Optimizing SQL injection detection using BERT encoding and AdaBoost Classification

Miodrag Zivkovic[1]*, Luka Jovanovic[1], Milos Bukumira[1], Milos Antonijevic[1], Djordje Mladenovic[2], Maryam Al Washahi[3], and Nebojsa Bacanin[1]

[1]Singidunum University, Belgrade, Serbia
mzivkovic@singidunum.ac.rs*, ljovanovic@singidunum.ac.rs, milos.bukumira.20@singimail.rs, mantonijevic@singidunum.ac.rs, nbacanin@singidunum.ac.rs
[2]College of Academic Studies "Dositej", Belgrade, Serbia
djordjemladenovic@akademijadositej.edu.rs
[3]Modern College of Business and Science, Muscat, Oman
maryam.washahi@mcbs.edu.om

Abstract. SQL injection attacks are still considerable threat to the web applications and organizations security in general, giving the attackers the opportunity to cause execution of arbitrary SQL queries sent through user input fields. Traditional defensive mechanisms to mitigate these threats often rely on static rules that may not adapt efficiently to the ever-evolving attack patterns. Recently, machine learning models are regarded as very promising to detect and prevent these attacks by enhancing the strenght of data-driven methods. This research proposes AdaBoost classifier to mitigate SQL threats. An altered variant of whale optimization algorithm has been introduced and employed to optimize the hyperparameters of the AdaBoost for this challenging problem. The outcomes were compared to the scores attained by other powerful optimizers. The suggested method achieved supreme results, with the highest obtained accuracy of slightly over 98.9%, exhibiting exciting potential in this field.

Keywords: SQL injection · BERT · AdaBoost · Metaheuristics optimization · Swarm intelligence · WOA.

1 Introduction

Structured Query Language (SQL) is used to work with data in a relational database management system (RDBMS). Its application is in the handling of structured data, where there are relationships between variables and entities. The language was designed by Donald D. Chamberlin and Raymond F. Boyce of IBM back in the 1970s [19]. It is designed to manipulate the data stored in the database with an intuitive approach. Its popularity and application to this day lie in its simplicity and efficiency, which are possible with a single command. It

N. Bacanin and H. Shaker (eds.), *Proceedings of the 2nd International Conference on Innovation in Information Technology and Business (ICIITB 2024)*,
Advances in Computer Science Research 113,
https://doi.org/10.2991/978-94-6463-482-2_10

has been developed over the years and has been followed by numerous standards from ANSI and ISO since 1986 and 1987 [23]. It has experienced popularity and its ubiquity in the management of relational databases, but due to its status it is prone to certain vulnerabilities, primarily in SQL queries due to backward compatibility, which is often a problem in the security of the databases themselves.

Artificial intelligence (AI), with an emphasis on natural language processing (NLP), can contribute to improving SQL security, primarily on databases, complex query automation, identifying potential security risks in queries, but also providing an interface for interacting with databases. The integration of AI and SQL is certainly a challenge that will become more and more common, primarily due to the possibility of automation but also the improvement of security standards, but there are also numerous challenges [41]. First of all, specific settings of hyperparameters, which are present in deep learning, some of the basic ones such as learning speed, number of layers of neural networks, batch size, but also others that have an impact on improving the efficiency of the model itself. This should be accompanied by extensive testing and the use of advanced optimization algorithms.

The optimization of hyperparameters can be improved and facilitated by the application of metaheuristics based on algorithms that draw their inspiration from processes in natural environments, first and foremost genetic algorithm [44], red fox algorithm [45], particle swarm [60] and many others should be highlighted. The optimization of hyperparameters can be improved and facilitated by the application of metaheuristics based on algorithms that draw their inspiration from processes in natural environments.

Their ability is to efficiently search large solution spaces, which results in optimal hyperparameter values. Metaheuristic parameters are classified as NP-hard, finding the optimal solution grows exponentially with its complexity. This further complicates the integration of AI, but also the existing solutions must not disrupt the existing SQL communication with the bases, adhering to standards and protocols. Efficient and effective use of AI in the SQL environments itself is essential for only safe integration.

2 Related Works

The connection of artificial intelligence (AI), through the fields of natural language processing (NLP) or machine learning, as already mentioned, can improve the security of the system itself, but also automate queries, however, it can also be used in the identification of potential risks. The biggest challenges are certainly with the optimization of hyperparameters of deep learning, but also with the application of the metaheuristic algorithms themselves.

In this scientific paper [1], the SQL vulnerability is described, it is not a new form of attack, and machine learning algorithms were used to detect sk injection attacks on websites. A total of seven machine learning algorithms (Naive Bayes, Neural-Network, SVM, Random-Forest, KNN, and Logistic Regression) were used, and their learning was based on SQL queries. Naive-Bayes proved to be the best solution where the results were 0.99 accuracies, 0.98 precision, 1.00 recall, and a 0.99 f1-score.

The authors of this scientific paper [2], focused on the detection of SQL injections with the help of a probabilistic neural network (PNN), where a meta-heuristic algorithm was used to optimize BAT. This study is based on a dataset of 600 SQL injections and 3500 normal queries. The results of this study are accuracy 99.19 percent, precision of 0.995 percent, a recall of 0.981 percent, and an F-Measure of 0.928 percent.

2.1 Bert

Bidirectional Encoder Representations from Transformers, presented in 2018(BERT) [24], and based on the paper "attention is all you need" [59], relies on the attention mechanism to interpret text and symbolism of sentences. It was patented by a team of scientists and researchers from the company Google, and today a large number of modern NLP applications, which include search, speech recognition and translation, are based on it.

The architecture and mechanical basis of BERT is based on transformers, which work according to the system of attention, which gives it the ability to change the focus on different segments of input data in the processing process itself. It contributes to the understanding of the meaning of the words in the sentence. Transformers are based on multiple heads of attention, so they process parts of the sentence in parallel, which increases the speed and efficiency of processing.

It has the ability to successfully process natural language in both directions. It is crucial for a deeper understanding of the language itself. Which is revolutionary because previous models were based on processing language in one direction. Using the "masked language model" (MLM) concept, training takes place by randomly hiding words in a sentence and then predicting the hidden words based on the content. All this helps him learn to understand language patterns and structures.

Bert has the ability to use the concept of transfer learning, it can be trained on a large dataset and then adjusted for special tasks. Its application in the field of NLP is vast, from text classification, text summarization, question answering, and many others. Because of all that, today BERT is offered as an ideal solution for many tasks of this type. With its ability to understand and interpret, it opened a new chapter in the field of intelligent systems that can communicate and understand human language.

2.2 Ada Boost

In the last decade, we have witnessed a constant growth in the use of machine learning. There are numerous alogrithms that have disproportionate contributions depending on the area in which they are used and on which examples. Where AdaBoost (Adaptive Boosting) stands out as a bridge between numerous optimization algorithms, its role is to connect weak algorithms into a group, resulting in one strong algorithm. It was developed by Freund and Schapire in 1995, and its application today is constantly increasing. Weak classifiers are considered to be those whose performance ranks slightly better than random guessing. Through each iteration, AdaBoost adds weak classifiers to the final model, where it balances the weight a classifier receives based on its accuracy.

When there is a classification error, the classification weights are decreased, while good classifications are rewarded with increased weights. This is the equation for the error of a weak classifier: 1:

$$\epsilon_t = \frac{\sum_{i=1}^{N} w_{i,t} \cdot \mathrm{I}(h_t(x_i) \neq y_i)}{\sum_{i=1}^{N} w_{i,t}} \tag{1}$$

where ϵ_t is the weighted error of the weak learner t-th is the iteration, N how many training samples, the weight value i-th instance in the t-th. Assumption of weak learner i-th case in the t-th round shown by the expression $h_t(x_i)$. The actual label of the i-th instance that points to the variable y_i. Here, the function $\mathrm{I}(\cdot)$ returns 0 if the case is false, and 1 if the case in the parentheses is true.

When the weights are achieved, we get new classifiers and then the process of modifying the weights themselves takes place again. For the result of accurate classifiers, a large group of classifiers is required. A linear model is a combination of sub-models with their results. The equation for calculating the weight of the classifier in the ensemble:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right) \tag{2}$$

Where the weight (α_t) changes for each weak student in the total set. It denotes the contribution of the weak learner, in the mixed exposure model, and rests on the weighted error(ϵ_t). The following equation is used to update the weights:

$$w_{i,t+1} = w_{i,t} \cdot \exp\left(-\alpha_t \cdot y_i \cdot h_t(x_i)\right) \tag{3}$$

where y_i marks the true mark of the i-th instance, $h_t(x_i)$ represents the prediction result of the weak student i-th instance in the t-th round, and $w_{i,t}$ denotes the weight, i-th instance in the t-th round.

AdaBoost plays a significant role in the field of machine learning. The advantages of adaboost are that bias and variance are reduced, which again leads to robust models. Its disadvantage is sensitivity to noisy data and exceptions, which can result in its adaptation in certain scenarios.

2.3 Metaheuristics algorithms

Taking inspiration from the organisms that thrive in vast groups and gaining advantage from collective behavioral patterns, swarm intelligence methods exhibit high efficacy in case a sole individual can not complete the assignment on its own. The family of swarm techniques achieved significant success for tackling NP-hard challenges.

These algorithms have demonstrated excellent capabilities in solving a broad range of real-life problems. Prominent application examples include medicine [38, 62, 11, 33, 40, 32, 66], and credit card frauds identification [29, 46]. Moreover, swarm algorithms have proven themselves in cloud computing problems [49, 9], plant classification[15], predicting green energy production [57, 6], economic problems [34, 54, 56, 13, 52, 47], enhancing the audit opinion [58], defect identification in software testing [68], feature selection [7, 18, 30, 65], computer security, phishing and intrusion detection [64, 3, 53, 36, 51, 16, 35, 12], environmental monitoring and pollutants tracking [37, 31, 8, 42], as well as improving wireless sensor networks optimization [63, 5] and general optimization of machine learning models [10, 55, 14, 25, 50, 67, 4, 26, 20].

3 Methods

3.1 Whale Optimization Algorithm

The whale optimization algorithm (WOA) is a metaheuristic optimization method that was proposed in 2016[43]. It is based on the social conduct of hump-back whales. WOA leverages a hunting method described as the bubble-net strategy, which is modeled after how whales hunt, to solve optimization problems. Since its launch, several challenges have been used to evaluate its optimizing skills. When hunting, humpback whales collaborate by employing the bubble-net technique. The whales swim underneath their pray fish, releasing bubbles as they go and swimming in circles that approach the surface of the water. Consequently, the prey gets trapped in the bubble ring and is forced to rise to the surface, which restricts its ability to escape [43].

Whales use the net of bubbles assault technique and imitate circling movements to surround their prey. On the other hand, exploration simulates a pseudo-random hunt for possible targets. WOA operates as a population-based algorithm, where the most promising solution (i.e., the option with the highest fitness) represents the prey and the rest possibilities in the population represent individual whales. Every contender looks around throughout exploitation, progressing to the best available option. Subsequently, upon obtaining the fitness values of the solutions, the solutions adjust their locations based on the prevailing optimal choice. Eqs. (4) and (5) illustrate this:

$$\vec{D} = |\vec{C} \cdot \vec{X}^*(t) - \vec{X}(t)| \tag{4}$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D}, \tag{5}$$

Here, $\vec{X}(t)$ and $\vec{X}^*(t)$ denote conceivable results $\vec{A}$ and $\vec{C}$ imply the coefficient vectors, and the currently leading solution for the moment t. The element-wise multiplicator is expressed as the operator $\cdot$.

The following formulas are used to compute the vectors $\vec{A}$ and $\vec{C}$

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \tag{6}$$

$$\vec{C} = 2 \cdot \vec{r}. \tag{7}$$

In the range $[0, 1]$, $\vec{r}$ refers to a pseudo-random vector drawn from a normal distribution, and $\vec{a}$ denotes a vector that diminishes linearly from 2 to 0 throughout the course of the iterations.

All prospective contenders $\vec{X}$ are continuously shifted toward a neighborhood of current best alternatives, $\vec{X}^*$, after being corrected by vector $\vec{r}$ for the coefficients $\vec{A}$ and $\vec{C}$.

As can be observed in Eq.(8), the air bubble rings are linearly reduced α beginning at 2 and approaching 0 in each repetition.

$$\vec{a} = 2 - t\frac{2}{maxIter}, \tag{8}$$

the present iteration and the top number of iterations are marked as t and $maxIter$.

The gap between the globally favored option $(\vec{X}^*(t))$ and the current choice for solution $(\vec{X}(t))$ being calculated signals the start of the spiraling movement. Next, assuming Eq. (9), the altered location of option $(\vec{X}(t+1))$ is obtained:

$$\vec{X}(t+1) = \vec{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t), \tag{9}$$

Here, $\vec{D'}$ reflects the space that remains after dividing the i-th feasible choice from the globally optimal option, which is determined by applying the formula $\vec{D'} = |\vec{X}^*(t) - \vec{X}(t)|$. The fixed value b is used to benchmark the logarithmic spiral's form. Finally, the pseudo-random number l from the range $(-1, 1)$ is presented.

According to Eq. (10), the circular action is able to be mathematically described as alternating the two mechanisms with equal probabilities p for each iteration.

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D} \text{ , if } p < 0.5 \\ \vec{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \text{ , if } p \geq 0.5 \end{cases}. \tag{10}$$

In the exploration step, instead of updating the global best, every participant updates its position based on where a random solution is located. In the event that the generated random numbers are more than or equal to $(|A| \geq 1)$, the $\vec{A}$ vector allows the people to be directed to a random solution. This is shown by the equation that follows: Eq. (11):

$$\vec{X}(t+1) = \vec{X}_{rnd}(t) - \vec{A} \cdot \vec{D}, \tag{11}$$

$\vec{D}$ marks how much space there is from the i-th possibility to the random solution rnd in iteration number t, computed as $\vec{D} = |\vec{C} \cdot \vec{X}_{rnd}(t) - \vec{X}(t)|$.

3.2 Adaptive genetic WOA

Despite generally good performance, WOA still holds untapped potential for optimization. While extensive analysis using standard CEC functions [21] has yielded promising results, the algorithm might be improved by prioritizing exploration action during beginning stages, while prioritizing exploitation in later phases. To address this, this study integrates mechanisms based on the genetic algorithms (GA), firefly algorithms (FA), and quasi-reflexive learning (QRL) [17] to enhance performance.

The initial enhancement focuses on boosting exploration even before completing the first iteration of the algorithm. This is accomplished by integrating QRL into the initialization process. Specifically, the first half of the population is generated making use of the conventional mechanisms of the original WHO algorithm. However, the other portion of agents is created as quasi-reflexive opposites of the current agents, as outlined in Eq. 12.

$$A_z^{qr} = rand\left(\frac{lb_z + ub_z}{2}, a_z\right) \tag{12}$$

here the lb and ub mark minimum and maximum constraints of the search scope while rad indicates a random value in the range.

Additional upgrades in convergence are gained when the algorithm is mixed with the FA [61]. The firefly search mechanism is depicted in Eq. (13)

$$X_i(t+1) \ = \ X_i(t) \ + \ \beta \ e^{-\gamma \ r_{ij}^2} \ (X_j(t) \ - \ X_i(t)) \ + \ \alpha \ \epsilon_i(t), \tag{13}$$

in which β_0 stands for the allure at $r = 0$. Eq. (13) is often replaced by Eq. (14) for better practical use.

$$\beta(r) = \frac{\beta_0}{(1 + \gamma \times r^2)} \tag{14}$$

here, $X_i(t)$ marks position currently occupied by agent i during iteration t, while r_{ij} stands for the position of agent j during the t iteration. The variable β denotes the separation of agent i from agent j, and shows the amount of shared attraction. Marking the agent attraction coefficient, β regulates the interaction. Next, γ is the coefficient of light absorption, α controls the randomness level, and lastly $\epsilon_i(t)$ stands for a stochastic vector.

After starting, the algorithm progressively explores potentially good options within the search spaces through two stages. The first stage occurs during the initial half of iterations, focusing on aggressively exploring the search space with larger agent jumps. To enable contributions from both algorithms to optimization, a arbitrary number in the extent of $[0, 1]$ is generated for ψ, while a predetermined empirical value is assigned to the introduced parameter $\theta = 0.8$. In case of $\psi > \theta$, the FA search is activated; otherwise, the default search employed by the WHO is utilized. Following each iteration, θ is decremented by 0.4, increasing the likelihood of utilizing the FA search in subsequent iterations. This way, the hybrid algorthm can keep convergence in later stages a priority while gaining benefits form the initial exploration boost added by QRL.

Thus, the algorithm resulting from this modification should be called Hybrid Adaptive WOA (HAWOA). For the pseudocode please see Algorithm 1.

Algorithm 1 Introduced modified algorithm (AGWOA) pseudocode.

```
Produce 1/2 of potential solutions in S
Apply QRL to S to generate remaining solutions
Set θ = 0.8
while T > t do
    Apply objective evaluation to S
    Select an arbitrary number ψ
    if ψ > θ then
        Apply FA search to S
    else
        Apply WOA search to S
    end if
    Update adaptive parameter as θ = θ − 0.4
end while
return Best attained solution
```

4 Simulation Setup

Dataset for SQL injection attacks identification, used for the simulations in this manuscript, is openly available at https://www.kaggle.com/datasets/sajid576/sql-injection-dataset. It is comprised of the raw SQL queries and label with values 0 (non-malicious) and 1 (malicious). 70% of dataset has been used for training, while remaining 30% was dedicated to testing. Adaboost hyperparameters chosen for optimization, along with corresponding search intervals included number of estimators $[10, 50]$, depth $[1, 10]$ and learning rate $[0.1, 2]$.

The AdaBoost model was tuned by introduced AGWOA metaheuristics. The experimental evironment was developed in Python, along with common set of libraries dedicated to machine learning, like scikit-learn, scipy, numpy, pandas and seaborn. The performance of the proposed AdaBoost tuned by AGWOA algorithm (short AB-AGWOA) has been compared to cutting-edge algorithms that included baseline WOA, GA [44], ACB [39], HHO [27] , RFO [48] and COA [28]. All algorithms were configured with six solutions in the population, 8 iterations per run, and 30 separate executions.

Since SQL injection dataset is not balanced, the authors opted for the Cohen's kappa metric κ as the fitness function required to be maximized [22], which may be obtained by:

$$\kappa = \frac{c_o - c_e}{1 - c_e} = 1 - \frac{1 - c_o}{1 - c_e} \tag{15}$$

where observed and expected results vectors are labeled as c_o and c_e. This measurement takes into account the class imbalance, consequently being capable of providing more robust predictions than just simply observing accuracy.

5 Experimental outcomes

The simulation results regarding the Cohen's kappa (objective) and classification error (used as the indicator function) are summarized in Tables 1 and 2. The supreme outcomes in every category are shown in bold font. The most superior model was produced by the suggested AGWOA method, and achieved supreme outcomes by finishing first in all categories observed for both Cohen's kappa and classification error rate. Other metaheuristics algorithms have also attained solid outcomes, finishing closely behind the AGWOA algorithm.

Figure 1 depicts the violin and box plots of the objective function (Cohen's coefficient) across thirty separate runs. From the graphs, it is obvious that the introduced AGWOA attains remarkable stability, in other words in all independent runs the results were very close to the result obtained in the best run. It may also be highlighted that the HHO finished second for the best metrics, however, it had the lowest stability, affecting the mean and median scores.

Next, Fig. 2 provides insight into the swarm plots of fitness and indicator functions, presenting the positions of the individuals in the populace of each algorithm within the last round of execution of the best run. Again, AGWOA exhibited remarkable stability, where nearly all solutions were concentrated in the near proximity of the best individual. HHO also exhibited admirable swarm plot in its best run. Fig. 3 puts forward the convergence diagrams of the objective and indicator functions during the best runs of each regarded metaheuristics. Suggested AGWOA once again yielded superior performance, exhibiting excellent convergence capabilities that indicate the ability of the algorithm to avoid local minimum traps that could hinder the performance in case of premature convergence, as exhibited by other algorithms (most notably GA and HHO).

Method	Best	Worst	Mean	Median	Std	Var
AB-AGWOA	**0.976358**	**0.972915**	**0.974419**	**0.974144**	**0.001330**	**1.77E-06**
AB-WOA	0.975357	0.964537	0.970879	0.971374	0.003455	1.19E-05
AB-GA	0.973936	0.969393	0.972078	0.972189	0.001658	2.75E-06
AB-ABC	0.972925	0.967015	0.970932	0.971843	0.002121	4.50E-06
AB-HHO	0.976060	0.954213	0.967598	0.970326	0.007414	5.50E-05
AB-RFO	0.972575	0.965884	0.971090	0.972196	0.002382	5.67E-06
AB-COA	0.974626	0.962755	0.970565	0.971832	0.003899	1.52E-05

Table 1. SQL injection detection optimized model overall Cohen's kappa (objective function) outcomes.

Detailed evaluations of the scores attained by the top-produced models created by every metaheuristics are shown in Table 3. The suggested AB-AGWOA method provided the supreme results, leading the charts in almost every score tracked in the simulations. HHO was the only method able to attain some better scores, namely for the precision on the normal class, recall for the injection class, and macro average recall. Aiming to support the replication of the simu-

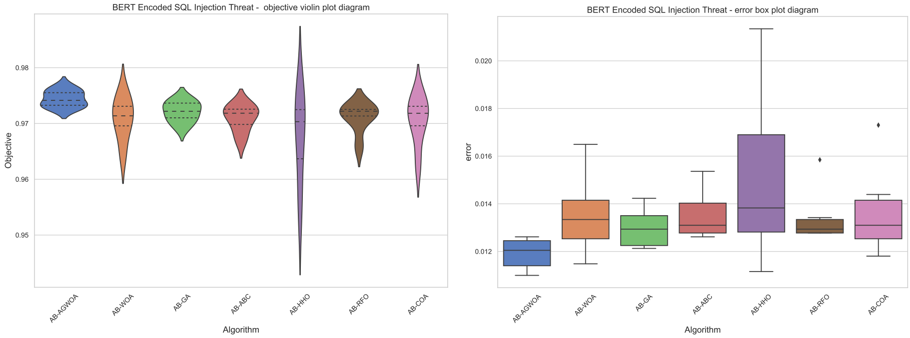

Fig. 1. Cohen's kappa coefficient violin and box plots for all regarded metaheuristics

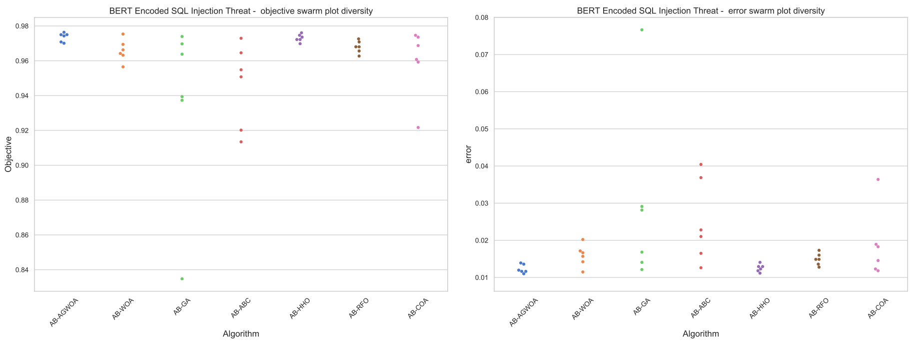

Fig. 2. Swarm plots of both Cohen's kappa and classification error

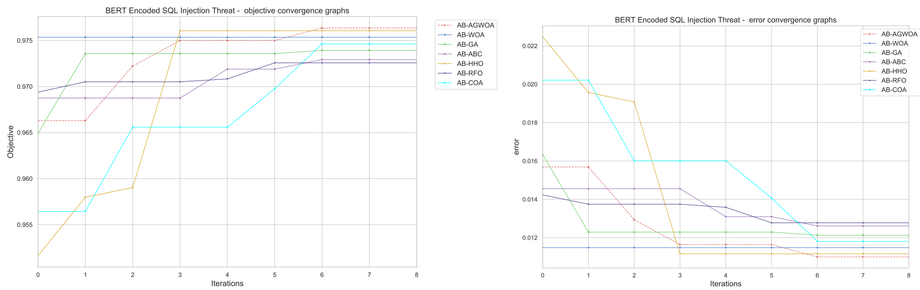

Fig. 3. Convergence diagrams of both Cohen's kappa and classification error

Method	Best	Worst	Mean	Median	Std	Var
AB-AGWOA	**0.010996**	**0.012613**	**0.011912**	**0.012047**	**0.000624**	**3.89E-07**
AB-WOA	0.011481	0.016494	0.013556	0.013341	0.001599	2.56E-06
AB-GA	0.012128	0.014230	0.012991	0.012937	0.000774	5.99E-07
AB-ABC	0.012613	0.015362	0.013530	0.013098	0.000987	9.73E-07
AB-HHO	0.011158	0.021345	0.015093	0.013826	0.003453	1.19E-05
AB-RFO	0.012775	0.015847	0.013449	0.012937	0.001098	1.21E-06
AB-COA	0.011805	0.017303	0.013691	0.013098	0.001805	3.26E-06

Table 2. SQL injection detection optimized model overall classification error (indicator) outcomes.

lation outcomes, therefore facilitating the reproducibility, the chosen collections of AdaBoost hyperpamareters are listed in Table 4.

Table 3. Detailed metric comparisons among the top performing models.

Approach	Score	Normal	Injection	Accuracy	Macro avg	Weighted avg
AB-AGWOA	precision	0.987261	**0.992032**	**0.989004**	**0.989647**	**0.989029**
	recall	**0.995376**	0.978175	**0.989004**	0.986776	**0.989004**
	f1-score	**0.991302**	**0.985055**	**0.989004**	**0.988179**	**0.988988**
AB-WOA	precision	0.989247	0.987270	0.988519	0.988258	0.988515
	recall	0.992551	0.981667	0.988519	0.987109	0.988519
	f1-score	0.990896	0.984460	0.988519	0.987678	0.988512
AB-GA	precision	0.986990	0.989399	0.987872	0.988195	0.987882
	recall	0.993835	0.977739	0.987872	0.985787	0.987872
	f1-score	0.990401	0.983535	0.987872	0.986968	0.987857
AB-ABC	precision	0.988226	0.985946	0.987387	0.987086	0.987382
	recall	0.991780	0.979921	0.987387	0.985851	0.987387
	f1-score	0.990000	0.982925	0.987387	0.986462	0.987379
AB-HHO	precision	**0.990005**	0.986854	0.988842	0.988429	0.988838
	recall	0.992294	**0.982977**	0.988842	**0.987635**	0.988842
	f1-score	0.991148	0.984911	0.988842	0.988030	0.988838
AB-RFO	precision	0.987973	0.985940	0.987225	0.986957	0.987220
	recall	0.991780	0.979485	0.987225	0.985633	0.987225
	f1-score	0.989873	0.982702	0.987225	0.986288	0.987216
AB-COA	precision	0.986996	0.990274	0.988195	0.988635	0.988211
	recall	0.994349	0.977739	0.988195	0.986044	0.988195
	f1-score	0.990659	0.983967	0.988195	0.987313	0.988180
	records	3893	2291			

Lastly the PR (precision-recall) curve and confusion matrix of the AdaBoost model optimized by suggested AGWOA are depicted in Fig. 4. Summarizing the overall simulation outcomes, it is possible to conclude that the AB-AGWOA model is very capable to handle the SQL injection task. It can also be concluded that other contending optimizers also performed very well in this challenge.

Table 4. Best constructed model parameters selected by each optimizer.

Methods	Number of estimators	Depth	Learning rate
AB-AGWOA	5	5	0.585448
AB-WOA	5	5	0.980577
AB-GA	5	5	0.597019
AB-ABC	5	5	1.129098
AB-HHO	5	5	0.630079
AB-RFO	5	5	1.246535
AB-COA	5	5	0.705048

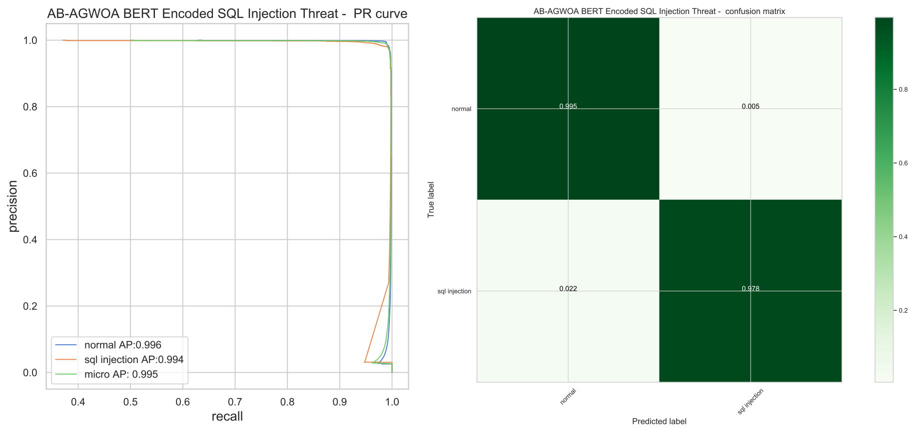

Fig. 4. PR curve and confusion matrix of the AdaBoost tuned by AGWOA

6 Conclusion

This research examined the capabilities of AdaBoost model tuned by metaheuristics algorithms for the SQL injection attacks identifications. SQL injection is a considerable threat, that may compromise the integrity of any institution, leading to financial losses, decrease of customer satisfaction, and ultimately even to the collapse of the entire organisation. AdaBoost has been optimized by an enhanced variant of famous WOA metaheuristics, and given the task to identify SQL injection threats. The simulation outcomes were compared to few other contending cutting-edge metaheuristics, and the suggested method achieved supreme results, as the top-constructed model obtained the accuracy of approximately 98.9%.

The drawbacks of the presented research are related to the high computational requirements of the simulations, resulting in lower number of solutions and iterations per run, and also limiting the count of metaheuristics algorithms evaluated in the comparative analysis. Moreover, the hyperparameters' search space has also been narrowed. In the future, we will aim to overcome these limitations if additional computing resources became available. The introduced AGWOA shall also be validated on other optimization challenges, however, it is outside of this paper's scope.

Acknowledgment

This research was supported by the Science Fund of the Republic of Serbia, Grant No. 7373, Characterizing crises-caused air pollution alternations using an artificial intelligence-based framework - crAIRsis and Grant No. 7502, Intelligent Multi-Agent Control and Optimization applied to Green Buildings and Environmental Monitoring Drone Swarms - ECOSwarm.

References

1. Manar Hasan Ali AL-Maliki and Mahdi Nsaif Jasim. Comparison study for nlp using machine learning techniques to detecting sql injection vulnerabilities. *International Journal of Nonlinear Analysis and Applications*, 14(8):283–290, 2023.
2. Fawaz Khaled Alarfaj and Nayeem Ahmad Khan. Enhancing the performance of sql injection attack detection through probabilistic neural networks. *Applied Sciences*, 13(7), 2023.
3. Nadheera AlHosni, Luka Jovanovic, Milos Antonijevic, Milos Bukumira, Miodrag Zivkovic, Ivana Strumberger, Joseph P Mani, and Nebojsa Bacanin. The xgboost model for network intrusion detection boosted by enhanced sine cosine algorithm. In *International Conference on Image Processing and Capsule Networks*, pages 213–228. Springer, 2022.
4. Nebojsa Bacanin, Milos Antonijevic, Nikola Vukobrat, Timea Bezdan, and Miodrag Zivkovic. Enhanced seagull optimization algorithm adapted for artificial neural network training. In *ICT with Intelligent Applications: Proceedings of ICTIS 2021, Volume 1*, pages 753–761. Springer, 2022.
5. Nebojsa Bacanin, Uros Arnaut, Miodrag Zivkovic, Timea Bezdan, and Tarik A Rashid. Energy efficient clustering in wireless sensor networks by opposition-based initialization bat algorithm. In *Computer Networks and Inventive Communication Technologies: Proceedings of Fourth ICCNCT 2021*, pages 1–16. Springer, 2022.
6. Nebojsa Bacanin, Luka Jovanovic, Miodrag Zivkovic, Venkatachalam Kandasamy, Milos Antonijevic, Muhammet Deveci, and Ivana Strumberger. Multivariate energy forecasting via metaheuristic tuned long-short term memory and gated recurrent unit neural networks. *Information Sciences*, 642:119122, 2023.
7. Nebojsa Bacanin, Aleksandar Petrovic, Miodrag Zivkovic, Timea Bezdan, and Milos Antonijevic. Feature selection in machine learning by hybrid sine cosine metaheuristics. In *International Conference on Advances in Computing and Data Sciences*, pages 604–616. Springer, 2021.
8. Nebojsa Bacanin, Marko Sarac, Nebojsa Budimirovic, Miodrag Zivkovic, Ahmad Ali AlZubi, and Ali Kashif Bashir. Smart wireless health care system using graph lstm pollution prediction and dragonfly node localization. *Sustainable Computing: Informatics and Systems*, 35:100711, 2022.
9. Nebojsa Bacanin, Vladimir Simic, Miodrag Zivkovic, Melfi Alrasheedi, and Aleksandar Petrovic. Cloud computing load prediction by decomposition reinforced attention long short-term memory network optimized by modified particle swarm optimization algorithm. *Annals of Operations Research*, pages 1–34, 2023.

10. Nebojsa Bacanin, Catalin Stoean, Miodrag Zivkovic, Dijana Jovanovic, Milos Antonijevic, and Djordje Mladenovic. Multi-swarm algorithm for extreme learning machine optimization. *Sensors*, 22(11):4204, 2022.
11. Nebojsa Bacanin, Miodrag Zivkovic, Fadi Al-Turjman, K Venkatachalam, Pavel Trojovskỳ, Ivana Strumberger, and Timea Bezdan. Hybridized sine cosine algorithm with convolutional neural networks dropout regularization application. *Scientific Reports*, 12(1):6302, 2022.
12. Nebojsa Bacanin, Miodrag Zivkovic, Milos Antonijevic, K Venkatachalam, Jinseok Lee, Yunyoung Nam, Marina Marjanovic, Ivana Strumberger, and Mohamed Abouhawwash. Addressing feature selection and extreme learning machine tuning by diversity-oriented social network search: an application for phishing websites detection. *Complex & Intelligent Systems*, 9(6):7269–7304, 2023.
13. Nebojsa Bacanin, Miodrag Zivkovic, Luka Jovanovic, Milica Ivanovic, and Tarik A Rashid. Training a multilayer perception for modeling stock price index predictions using modified whale optimization algorithm. In *Computational Vision and Bio-Inspired Computing: Proceedings of ICCVBIC 2021*, pages 415–430. Springer, 2022.
14. Nebojsa Bacanin, Miodrag Zivkovic, Mohamed Salb, Ivana Strumberger, and Amit Chhabra. Convolutional neural networks hyperparameters optimization using sine cosine algorithm. In *Sentimental Analysis and Deep Learning: Proceedings of ICSADL 2021*, pages 863–878. Springer, 2022.
15. Nebojsa Bacanin, Miodrag Zivkovic, Marko Sarac, Aleksandar Petrovic, Ivana Strumberger, Milos Antonijevic, Andrija Petrovic, and K Venkatachalam. A novel multiswarm firefly algorithm: An application for plant classification. In *International Conference on Intelligent and Fuzzy Systems*, pages 1007–1016. Springer, 2022.
16. Nebojsa Bacanin, Miodrag Zivkovic, Catalin Stoean, Milos Antonijevic, Stefana Janicijevic, Marko Sarac, and Ivana Strumberger. Application of natural language processing and machine learning boosted with swarm intelligence for spam email filtering. *Mathematics*, 10(22):4173, 2022.
17. Jameer Basha, Nebojsa Bacanin, Nikola Vukobrat, Miodrag Zivkovic, K Venkatachalam, Stepan Hubálovskỳ, and Pavel Trojovskỳ. Chaotic harris hawks optimization with quasi-reflection-based learning: An application to enhance cnn design. *Sensors*, 21(19):6654, 2021.
18. Timea Bezdan, Dusan Cvetnic, Luka Gajic, Miodrag Zivkovic, Ivana Strumberger, and Nebojsa Bacanin. Feature selection by firefly algorithm with improved initialization strategy. In *7th conference on the engineering of computer based systems*, pages 1–8, 2021.
19. Raymond F Boyce and Donald D Chamberlin. *Using a structured English query language as a data definition facility.* IBM Thomas J. Watson Research Division, 1973.
20. Milos Bukumira, Miodrag Zivkovic, Milos Antonijevic, Luka Jovanovic, Nebojsa Bacanin, and Tamara Zivkovic. The extreme gradient boosting method optimized by hybridized sine cosine metaheuristics for ship vessel classification. In *International Conference on Advances in Data-driven Computing and Intelligent Systems*, pages 255–270. Springer, 2023.
21. Ran Cheng, Miqing Li, Ye Tian, Xiaoshu Xiang, Xingyi Zhang, Shengxiang Yang, Yaochu Jin, and Xin Yao. Benchmark functions for the cec'2018 competition on many-objective optimization. Technical report, 2018.
22. Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

23. Chris J Date. *A Guide to the SQL Standard.* Addison-Wesley Longman Publishing Co., Inc., 1989.
24. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
25. Masa Gajevic, Nemanja Milutinovic, Jelena Krstovic, Luka Jovanovic, Marina Marjanovic, and Catalin Stoean. Artificial neural network tuning by improved sine cosine algorithm for healthcare 4.0. In *Proceedings of the 1st international conference on innovation in information technology and business (ICIITB 2022)*, volume 104, page 289. Springer Nature, 2023.
26. Luka Gajic, Dusan Cvetnic, Miodrag Zivkovic, Timea Bezdan, Nebojsa Bacanin, and Stefan Milosevic. Multi-layer perceptron training using hybridized bat algorithm. In *Computational Vision and Bio-Inspired Computing: ICCVBIC 2020*, pages 689–705. Springer, 2021.
27. Ali Asghar Heidari, Seyedali Mirjalili, Hossam Faris, Ibrahim Aljarah, Majdi Mafarja, and Huiling Chen. Harris hawks optimization: Algorithm and applications. *Future generation computer systems*, 97:849–872, 2019.
28. Heming Jia, Honghua Rao, Changsheng Wen, and Seyedali Mirjalili. Crayfish optimization algorithm. *Artificial Intelligence Review*, 56(Suppl 2):1919–1979, 2023.
29. Dijana Jovanovic, Milos Antonijevic, Milos Stankovic, Miodrag Zivkovic, Marko Tanaskovic, and Nebojsa Bacanin. Tuning machine learning models using a group search firefly algorithm for credit card fraud detection. *Mathematics*, 10(13):2272, 2022.
30. Dijana Jovanovic, Marina Marjanovic, Milos Antonijevic, Miodrag Zivkovic, Nebojsa Budimirovic, and Nebojsa Bacanin. Feature selection by improved sand cat swarm optimizer for intrusion detection. In *2022 International Conference on Artificial Intelligence in Everything (AIE)*, pages 685–690. IEEE, 2022.
31. Gordana Jovanovic, Mirjana Perisic, Nebojsa Bacanin, Miodrag Zivkovic, Svetlana Stanisic, Ivana Strumberger, Filip Alimpic, and Andreja Stojic. Potential of coupling metaheuristics-optimized-xgboost and shap in revealing pahs environmental fate. *Toxics*, 11(4):394, 2023.
32. Luka Jovanovic, Nebojsa Bacanin, Miodrag Zivkovic, Milos Antonijevic, Aleksandar Petrovic, and Tamara Zivkovic. Anomaly detection in ecg using recurrent networks optimized by modified metaheuristic algorithm. In *2023 31st Telecommunications Forum (TELFOR)*, pages 1–4. IEEE, 2023.
33. Luka Jovanovic, Marko Djuric, Miodrag Zivkovic, Dijana Jovanovic, Ivana Strumberger, Milos Antonijevic, Nebojsa Budimirovic, and Nebojsa Bacanin. Tuning xgboost by planet optimization algorithm: An application for diabetes classification. In *Proceedings of fourth international conference on communication, computing and electronics systems: ICCCES 2022*, pages 787–803. Springer, 2023.
34. Luka Jovanovic, Dejan Jovanovic, Nebojsa Bacanin, Ana Jovancai Stakic, Milos Antonijevic, Hesham Magd, Ravi Thirumalaisamy, and Miodrag Zivkovic. Multistep crude oil price prediction based on lstm approach tuned by salp swarm algorithm with disputation operator. *Sustainability*, 14(21):14616, 2022.
35. Luka Jovanovic, Dijana Jovanovic, Milos Antonijevic, Bosko Nikolic, Nebojsa Bacanin, Miodrag Zivkovic, and Ivana Strumberger. Improving phishing website detection using a hybrid two-level framework for feature selection and xgboost tuning. *Journal of Web Engineering*, 22(3):543–574, 2023.
36. Luka Jovanovic, Dijana Jovanovic, Milos Antonijevic, Miodrag Zivkovic, Nebojsa Budimirovic, Ivana Strumberger, and Nebojsa Bacanin. The xgboost tuning by

improved firefly algorithm for network intrusion detection. In *2022 24th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pages 268–275. IEEE, 2022.

37. Luka Jovanovic, Gordana Jovanovic, Mirjana Perisic, Filip Alimpic, Svetlana Stanisic, Nebojsa Bacanin, Miodrag Zivkovic, and Andreja Stojic. The explainable potential of coupling metaheuristics-optimized-xgboost and shap in revealing vocs' environmental fate. *Atmosphere*, 14(1):109, 2023.
38. Luka Jovanovic, Miodrag Zivkovic, Milos Antonijevic, Dijana Jovanovic, Milica Ivanovic, and Hothefa Shaker Jassim. An emperor penguin optimizer application for medical diagnostics. In *2022 IEEE Zooming Innovation in Consumer Technologies Conference (ZINC)*, pages 191–196. IEEE, 2022.
39. Dervis Karaboga and Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of global optimization*, 39:459–471, 2007.
40. Stanislava Kozakijevic, Mohamed Salb, Ali Elsadai, Joseph Mani, Kanchana Devi, Anni Dasho Sharko, and Suresh Muthusamy. Seizure detection via time series classification using modified metaheuristic optimized recurrent networks. *Theoretical and Applied Computational Intelligence*, 1(1):82–94, 2023.
41. Guoliang Li, Xuanhe Zhou, and Lei Cao. Ai meets database: Ai4db and db4ai. In *Proceedings of the 2021 International Conference on Management of Data*, pages 2859–2866, 2021.
42. Joseph Mani, Hotefa Shaker, Luka Jovanovic, et al. Sunspot occurrence forecasting with metaheuristic optimized recurrent neural networks. *Theoretical and Applied Computational Intelligence*, 1(1):15–26, 2023.
43. Seyedali Mirjalili and Andrew Lewis. The whale optimization algorithm. *Advances in engineering software*, 95:51–67, 2016.
44. Seyedali Mirjalili and Seyedali Mirjalili. Genetic algorithm. *Evolutionary algorithms and neural networks: Theory and applications*, pages 43–55, 2019.
45. Hardi Mohammed and Tarik Rashid. Fox: a fox-inspired optimization algorithm. *Applied Intelligence*, 53(1):1030–1050, 2023.
46. Aleksandar Petrovic, Nebojsa Bacanin, Miodrag Zivkovic, Marina Marjanovic, Milos Antonijevic, and Ivana Strumberger. The adaboost approach tuned by firefly metaheuristics for fraud detection. In *2022 IEEE World Conference on Applied Intelligence and Computing (AIC)*, pages 834–839. IEEE, 2022.
47. Aleksandar Petrovic, Luka Jovanovic, Miodrag Zivkovic, Nebojsa Bacanin, Nebojsa Budimirovic, and Marina Marjanovic. Forecasting bitcoin price by tuned long short term memory model. In *1st International conference on innovation in information technology and business (ICIITB 2022)*, pages 187–202. Atlantis Press, 2023.
48. Dawid Połap and Marcin Woźniak. Red fox optimization algorithm. *Expert Systems with Applications*, 166:114107, 2021.
49. Bratislav Predić, Luka Jovanovic, Vladimir Simic, Nebojsa Bacanin, Miodrag Zivkovic, Petar Spalevic, Nebojsa Budimirovic, and Milos Dobrojevic. Cloud-load forecasting via decomposition-aided attention recurrent neural network tuned by modified particle swarm optimization. *Complex & Intelligent Systems*, pages 1–21, 2023.
50. Mohamed Salb, Nebojsa Bacanin, Miodrag Zivkovic, Milos Antonijevic, Marina Marjanovic, and Ivana Strumberger. Extreme learning machine tuning by original sine cosine algorithm. In *2022 IEEE World Conference on Applied Intelligence and Computing (AIC)*, pages 143–148. IEEE, 2022.

51. Mohamed Salb, Luka Jovanovic, Nebojsa Bacanin, Milos Antonijevic, Miodrag Zivkovic, Nebojsa Budimirovic, and Laith Abualigah. Enhancing internet of things network security using hybrid cnn and xgboost model tuned via modified reptile search algorithm. *Applied Sciences*, 13(23):12687, 2023.
52. Mohamed Salb, Miodrag Zivkovic, Nebojsa Bacanin, Amit Chhabra, and M Suresh. Support vector machine performance improvements for cryptocurrency value forecasting by enhanced sine cosine algorithm. In *Computer Vision and Robotics: Proceedings of CVR 2021*, pages 527–536. Springer, 2022.
53. Nikola Savanović, Ana Toskovic, Aleksandar Petrovic, Miodrag Zivkovic, Robertas Damaševičius, Luka Jovanovic, Nebojsa Bacanin, and Bosko Nikolic. Intrusion detection in healthcare 4.0 internet of things systems via metaheuristics optimized machine learning. *Sustainability*, 15(16):12563, 2023.
54. Marko Stankovic, Nebojsa Bacanin, Miodrag Zivkovic, Luka Jovanovic, Joseph Mani, and Milos Antonijevic. Forecasting ethereum price by tuned long short-term memory model. In *2022 30th Telecommunications Forum (TELFOR)*, pages 1–4. IEEE, 2022.
55. Marko Stankovic, Jelena Gavrilovic, Dijana Jovanovic, Miodrag Zivkovic, Milos Antonijevic, Nebojsa Bacanin, and Milos Stankovic. Tuning multi-layer perceptron by hybridized arithmetic optimization algorithm for healthcare 4.0. *Procedia Computer Science*, 215:51–60, 2022.
56. Marko Stankovic, Luka Jovanovic, Nebojsa Bacanin, Miodrag Zivkovic, Milos Antonijevic, and Petar Bisevac. Tuned long short-term memory model for ethereum price forecasting through an arithmetic optimization algorithm. In *International Conference on Innovations in Bio-Inspired Computing and Applications*, pages 327–337. Springer, 2022.
57. Catalin Stoean, Miodrag Zivkovic, Aleksandra Bozovic, Nebojsa Bacanin, Roma Strulak-Wójcikiewicz, Milos Antonijevic, and Ruxandra Stoean. Metaheuristic-based hyperparameter tuning for recurrent deep learning: application to the prediction of solar energy generation. *Axioms*, 12(3):266, 2023.
58. Mihailo Todorovic, Nemanja Stanisic, Miodrag Zivkovic, Nebojsa Bacanin, Vladimir Simic, and Erfan Babaee Tirkolaee. Improving audit opinion prediction accuracy using metaheuristics-tuned xgboost algorithm with interpretable results through shap value analysis. *Applied Soft Computing*, 149:110955, 2023.
59. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
60. Dongshu Wang, Dapei Tan, and Lei Liu. Particle swarm optimization algorithm: an overview. *Soft computing*, 22:387–408, 2018.
61. Xin-She Yang. Firefly algorithm, stochastic test functions and design optimisation. *International journal of bio-inspired computation*, 2(2):78–84, 2010.
62. Miodrag Zivkovic, Nebojsa Bacanin, Milos Antonijevic, Bosko Nikolic, Goran Kvascev, Marina Marjanovic, and Nikola Savanovic. Hybrid cnn and xgboost model tuned by modified arithmetic optimization algorithm for covid-19 early diagnostics from x-ray images. *Electronics*, 11(22):3798, 2022.
63. Miodrag Zivkovic, Nebojsa Bacanin, Tamara Zivkovic, Ivana Strumberger, Eva Tuba, and Milan Tuba. Enhanced grey wolf algorithm for energy efficient wireless sensor networks. In *2020 zooming innovation in consumer technologies conference (ZINC)*, pages 87–92. IEEE, 2020.
64. Miodrag Zivkovic, Luka Jovanovic, Milica Ivanovic, Nebojsa Bacanin, Ivana Strumberger, and P Mani Joseph. Xgboost hyperparameters tuning by fitness-dependent

optimizer for network intrusion detection. In *Communication and intelligent systems: Proceedings of ICCIS 2021*, pages 947–962. Springer, 2022.
65. Miodrag Zivkovic, Luka Jovanovic, Milica Ivanovic, Aleksa Krdzic, Nebojsa Bacanin, and Ivana Strumberger. Feature selection using modified sine cosine algorithm with covid-19 dataset. In *Evolutionary computing and mobile sustainable networks: Proceedings of ICECMSN 2021*, pages 15–31. Springer, 2022.
66. Miodrag Zivkovic, Catalin Stoean, Aleksandar Petrovic, Nebojsa Bacanin, Ivana Strumberger, and Tamara Zivkovic. A novel method for covid-19 pandemic information fake news detection based on the arithmetic optimization algorithm. In *2021 23rd international symposium on symbolic and numeric algorithms for scientific computing (SYNASC)*, pages 259–266. IEEE, 2021.
67. Miodrag Zivkovic, Ana Vesic, Nebojsa Bacanin, Ivana Strumberger, Milos Antonijevic, Luka Jovanovic, and Marina Marjanovic. An improved animal migration optimization approach for extreme learning machine tuning. In *International Conference on Intelligent and Fuzzy Systems*, pages 3–13. Springer, 2022.
68. Tamara Zivkovic, Bosko Nikolic, Vladimir Simic, Dragan Pamucar, and Nebojsa Bacanin. Software defects prediction by metaheuristics tuned extreme gradient boosting and analysis based on shapley additive explanations. *Applied Soft Computing*, 146:110659, 2023.