An Enhanced Sine Cosine Algorithm for Feature Selection in Network Intrusion Detection

zahra asgari varzaneh soodeh Hosseini

Department of Computer Science, Faculty of Mathematics and Computer, Shahid Bahonar University of Kerman, Kerman, Iran,

Abstract

For computer networks to remain secure, intrusion detection is essential. Analyzing network traffic data is part of this activity to spot possible cyber threats. However, the curse of dimensionality presents a challenge because there are so many dimensions in the data. To overcome this challenge, feature selection is essential to creating a successful intrusion detection system. It involves removing irrelevant and redundant features, which enhances the classification model's accuracy and lowers the dimensionality of the feature space. Metaheuristic algorithms are optimization techniques inspired by nature and are well-suited to choose features for network intrusion detection. They are effective in exploring large search spaces and have been widely used for this purpose. In this study, we improve the Sine Cosine Algorithm named ISCA for feature selection by introducing a controlling parameter to balance exploration and exploitation. Based on the NSL-KDD dataset, the results show that compared to other competing algorithms, the ISCA performs better than other metaheuristic algorithms in terms of both the number of features selected and the accuracy of classification.

Keywords: Sine Cosine Algorithm, Metaheuristic algorithms, Feature selection, Network intrusion detection.

1. Introduction

With the development of computer technology and the increase in the use of the Internet, the issue of information security has become more important [1]. Any illegal action that is done to disrupt security goals such as integrity, confidentiality, and accessibility is called intrusion [2]. To create complete security in any computer system, in addition to intrusion prevention tools such as firewalls, authentication, and encryption, other systems called Intrusion Detection Systems (IDS) are needed [3]. If the intruder passes the aforementioned security mechanisms, an IDS detects it and finds a solution to deal with it. Intrusion detection is an essential component of network security that seeks to recognize and address attempts at unwanted entry and malicious activities within a network [4, 5]. IDS monitor the flow of network activity and distinguishes normal activities from suspicious activities and attacks. The two fundamental IDSs are host-based and network-based which all IDSs fall [6, 7]. The system's location distinguishes between these two types. The host-based is placed on the client's computer, while the network-based is disseminated throughout the network. In fact, an IDS works like a pattern recognition system, and by receiving a series of features, it detects the intrusion. Network traffic datasets may contain millions of samples containing a lot of features [8].

With the increasing complexity and sophistication of cyber threats, IDSs are becoming more critical for protecting network infrastructures [8, 9]. However, one of the main problems facing IDS is the high dimensionality of the data generated from network traffic, which could result in a decline in performance, increased false alarms, and high computational costs. Feature selection is one important phase in the process of building an effective IDS, as it aims to minimize the dimensionality of the data by choosing the most relevant features while retaining the essential information [10, 11]. The literature has presented a number of feature selection models, such as filter, wrapper, and embedding methods. However, due to the high dimensionality of IDS data, traditional feature selection methods may not be adequate, and alternative approaches are required [12].

Metaheuristic algorithms are a class of optimization techniques that have been utilized extensively in various fields, including feature selection [13-16]. The algorithms are modeled around natural phenomena, such as evolutionary processes and swarm intelligence, and is proficient at finding the best answers in intricate, high-dimensional search areas [17-19]. As such, metaheuristic algorithms have become increasingly popular in feature selection for IDS, as they can efficiently find the most pertinent features with the least amount of computational cost. In 2016, Mirjalili proposed a new search strategy that makes use of the sine and cosine trigonometric functions [20]. This search strategy called the SCA, is a stochastic algorithm based on population that follows basic optimization principles of exploration and exploitation. The SCA algorithm is simple yet effective, and it has been effectively used to solve a range of optimization issues. However, Similar to other metaheuristic algorithms, SCA has limitations such as falling into local optima and premature convergence [21]. To deal with these problems, we propose an enhanced version of the SCA algorithm called ISCA for feature selection in intrusion detection. The ISCA algorithm introduces a controlling parameter that balances exploration and exploitation. By tuning the controlling parameter, we aim to enhance the algorithm's efficiency by avoiding premature convergence and improving the standard of the chosen features.

In this paper, we assess how well the suggested algorithm performs. on the NSL-KDD dataset and compare it with the standard SCA algorithm. We prove that the ISCA algorithm outperforms the benchmark competing algorithms regarding the number of features selected and the classification accuracy. The outcomes of the experiments show how well our suggested method works as a feature selection tool for network intrusion detection.

The rest of this article is organized as follows: A summary of relevant works is provided in Section 2. Section 3 outlines the typical SCA algorithm and its basic principles. Details of our ISCA algorithm are provided in Section 4 and introduces a controlling parameter to balance exploration and exploitation. In Section 5, we describe the simulation setup and present the results of applying the proposed algorithm to intrusion detection datasets for feature selection. Finally, in Section 6, we provide an overview of our findings and consider potential avenues for further improving the proposed algorithm.

2. Related work

This section discusses the challenge of detecting network intrusions due to high-dimensional data and the significance of feature selection in building effective IDSs. Metaheuristic algorithms are well-suited for feature selection in network intrusion detection as they can effectively explore large search spaces. This section will review the literature on the use of metaheuristic algorithms for feature selection and their efficiency in raising intrusion detection systems' accuracy.

Alazzam et al. [22] They suggested an IDS wrapper feature selection approach that makes use of an optimizer inspired by pigeons. A new method for binarizing the continuous optimizer was introduced and compared to other methods used for binarizing intelligent swarm algorithms that are continuous. The proposed algorithm outperformed several state-of-the-art feature selection algorithms in terms of TPR, FPR, accuracy, and F-score on three popular datasets. The algorithm's proposed cosine similarity method for binarization converged more quickly than the sigmoid method. Beulah et al. [23] introduced a hybrid method that was used for feature reduction in any domain and integrates the best features from several feature selection techniques. Their method was applied to IDSs using the NSL-KDD data, and six predominant features were chosen. Performance investigation showed that their algorithm improves the detection rate and the accuracy of the IDS.

In [24], the ideal feature subset was found by the authors using the CFS-DE technique, which was followed by the weighted stacking approach. that improves classification performance by increasing the base classifiers' weights that produce strong training results and dropping those with bad results. Experiments were carried out on NSL-KDD and CSE-CIC-IDS2018 datasets, and the model had high accuracy, recall, precision, and F1-score on both datasets. When compared to other articles' models and conventional machine learning models, the suggested CFS-DE-weighted-Stacking IDS had the top-performing classification. Vijayanand and Devaraj [25] presented a wrapper-based model for intrusion detection using the modified WOA. They integrated WOA with genetic algorithm operators and picked useful features from the network data to accurately identify incursions. They used a support vector machine (SVM) to recognize intrusions based on the selected features.

In [26] authors proposed an opposition self-adaptive grasshopper optimization algorithm (GOA) based on perceptive ideas and mutation, to detect new malicious or anomalous attacks. Furthermore, an SVM employing gain actor-critic with SVM uses reinforcement learning to improve the capacity for detecting fresh cyberattacks. The suggested algorithm beats other evolutionary techniques and the fundamental GOA regarding accuracy, detection rate and false-positive rate for resolving intrusion detection system issues, according to comparative simulation results. Salo et al. [27] introduced a hybrid model for dimensionality reduction in IDS that integrates an ensemble classifier based on SVM, instance-based learning methods (IBK), and multilayer perceptron (MLP) with information gain (IG) and principal component analysis (PCA). Comparative analysis showed that the Compared to most current state-of-the-art methodologies, their method performs better regarding accuracy, false alarm rate and detection rate.

Ahmed et al. [28] introduced a modified model for enhancing the efficiency of IDS by proposing an alternative feature selection optimizer algorithm, called GTO. The method involved using a feature extraction model, such as a convolutional neural network (CNN), as a first step, to decrease the dataset's dimensionality. The FS model was then given the retrieved features to use in detection. The results of the devised approach were in contrast to prominent IDS methodologies, and their proposed algorithm based on performance criteria, showed superiority over all other techniques. Safaldin et al. [29] proposed an enhanced IDS called GWOSVM-IDS, which used the enhanced binary GWO with SVM. The algorithm was tested using 3, 5, and 7 wolves to determine the ideal number of wolves for the problem. The suggested approach sought to raise the detection rate and accuracy of intrusion detection while reducing processing time in a wireless sensor network environment by decreasing false alarm rates and the number of features resulting from the IDSs. The findings indicate that the suggested GWOSVM-IDS with seven wolves outperforms the other algorithms.

To improve network intrusion detection's efficacy, Jayalatchumy et al. [30] deployed a special intrusion detection system (IDS). Data-denoising techniques were initially used to deal with the imbalance in the data. The most important characteristics were then found using the improved Crow search method to improve intrusion attack classification. Using the chosen characteristics, an ensemble classifier classified the normal and invader labels in the last stage. The experimental findings show that the developed method provides a remarkable 99.2% accuracy for the NSL-KDD. In [31], a PCA was used by the suggested system to minimize the dimensionality of the dataset. Strong global search capacity was provided by the efficient feature selection process using an Improved HHO (IHHO). SVM is used in the first stage of a two-stage classifier, while KNN is used in the second. Combining the benefits of SVM and KNN is the main objective to minimize false alarm rate and enhance accuracy.

The key challenge in this study is selecting the most effective features in network intrusion detection using metaheuristic algorithms. In the studies conducted, most metaheuristic algorithms, despite their high computational complexity, still struggle with imbalances in the exploration and exploitation processes. The proposed ISCA addresses this by providing a simple yet computationally efficient method that achieves high accuracy, selects a smaller subset of features, and has faster execution times compared to other feature selection techniques.

3. Sine-cosine algorithm (SCA)

The sine-cosine algorithm, introduced by Mirjalili [20] is a swarm intelligence algorithm that utilizes the trigonometric functions of sine and cosine. This enables individuals to move freely to another region within the range of [0, 2π], thereby facilitating an exploration of a larger search space or exploitation of neighborhood space. Through multiple iterations of the population, the algorithm gradually obtains the global optimal position.

During each iteration, the SCA updates the position using Eq. 1, which is the update formula of the SCA algorithm as shown below:

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times \sin(r_2) \times |r_3 p_i^t - X_i^t|, & r_4 \leq 0.5 \\ X_i^t + r_1 \times \cos(r_2) \times |r_3 p_i^t - X_i^t|, & r_4 > 0.5 \end{cases} \tag{1}$$

where the number of iterations at this time is t, X_i^t is the particle i's location at iteration t, and the position of the global optimal particle at iteration t is determined by p_i^t and three random parameters that are uniformly distributed: $r_2 \in (0, 2\pi)$, $r_3 \in [0, 2]$, and $r_4 \in (0, 1)$.

In addition, the algorithm uses a linearly decreasing convergence parameter, r_1, which is expressed as shown in Eq. 2. This parameter regulates how the algorithm is explored and exploited.

$$r_1 = a - t\frac{a}{T} \tag{2}$$

Where, a = 2, the highest number of iterations is T, and the number of iterations that are currently in use is t.

4. Proposed Algorithm

In this paper, we propose a modification to the SCA algorithm called ISCA that enhances the balance between exploitation and exploration. The modification introduces a time-varying parameter, *Alpha*, that regulates the exploration rate of the algorithm in the beginning and the exploitation rate towards the end. The updated equation can be modified as follows:

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times \sin(r_2) \times |r_3 p_i^t - X_i^t| \times Alpha, & r_4 \leq 0.5 \\ X_i^t + r_1 \times \cos(r_2) \times |r_3 p_i^t - X_i^t| \times Alpha, & r_4 > 0.5 \end{cases} \tag{3}$$

where $Alpha$ is a parameter that changes throughout time has a high initial value and progressively falls over time, favoring exploration in the beginning and exploitation towards the end. Here is how to calculate $Alpha$'s value:

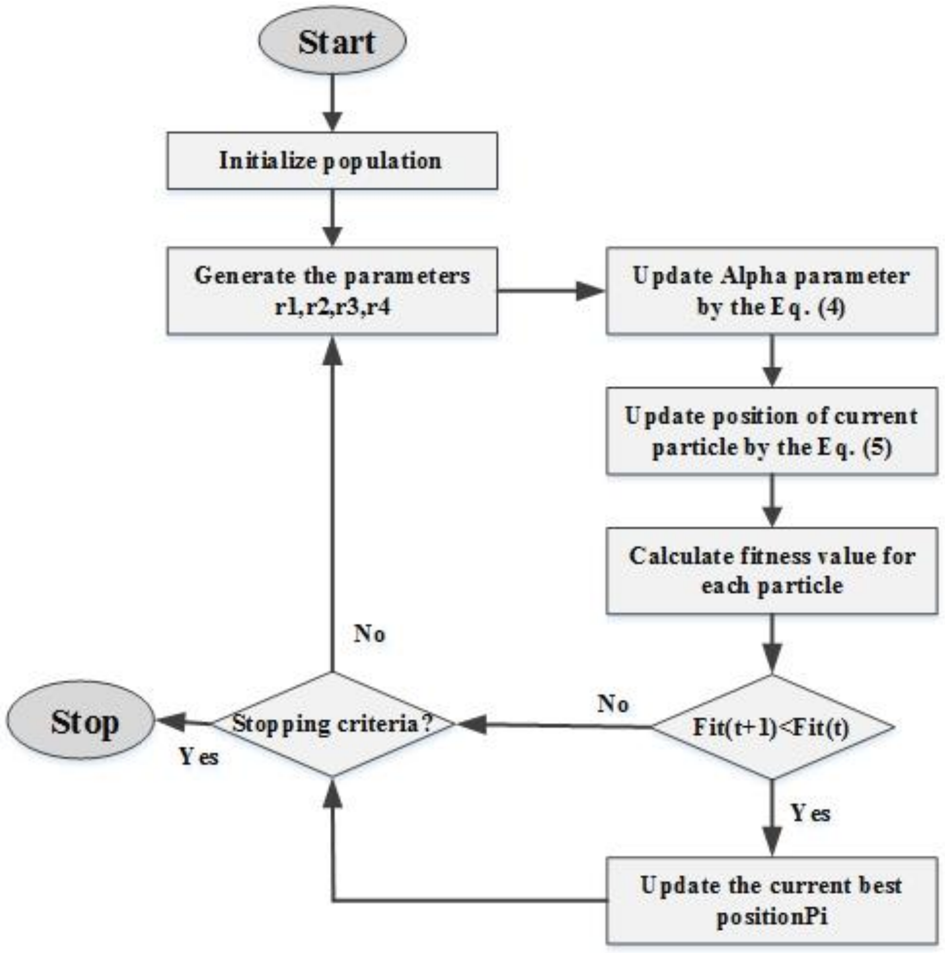
$$Alpha = alpha\_max - (alpha\_max - alpha\_min) \times (\frac{t}{T})\text{^k} \tag{4}$$

where $alpha\_max$ and $alpha\_min$ are the highest and lowest numbers of $Alpha$, t is the current iteration number, T is the highest number of iterations, and k is a parameter that controls the rate of change of $Alpha$. As the algorithm progresses through the iterations, the value of t increases from 1 to the maximum number of iterations, T. This means that the value of $Alpha$ will gradually decrease from $alpha\_max$ to $alpha\_min$ over the course of the optimization process. The role of t in this equation is to control the rate of change of $Alpha$. At the beginning of the algorithm (when t is small), the value of (t/T) will be small, and $Alpha$ will be closer to $alpha\_max$, favoring exploration. As the algorithm progresses (t increases), the value of (t/T) will get closer to 1, and $Alpha$ will decrease towards $alpha\_min$, favoring exploitation.

The value of k determines the rate at which $Alpha$ transitions from the exploration phase to the exploitation phase. A larger value of k will result in a faster decrease in $Alpha$, leading to a quicker shift from exploration to exploitation. Conversely, a smaller value of k will lead to a more gradual decrease in $Alpha$, allowing the algorithm to maintain a balance between exploration and exploitation for a longer period. The appropriate selection of $alpha\_max$ and $alpha\_min$, and k depends on the specific problem and the desired balance between exploration and exploitation. The ISCA algorithm allows for this balance to be dynamically adjusted, which can be particularly beneficial for complex optimization problems such as feature selection in network intrusion detection.

By incorporating the time-varying parameter $Alpha$, the ISCA algorithm aims to enhance the performance of the original SCA algorithm by ensuring a more effective exploration of the search space in the early stages and a more focused exploitation of the promising regions towards the end of the optimization process.

At the beginning of the algorithm, the value of $Alpha$ is close to $alpha\_max$, which favors exploration and allows the particles to investigate a more expansive search area. As the algorithm progresses, the value of $Alpha$ gradually decreases towards $alpha\_min$, which favors exploitation and allows the particles to focus on exploiting the local optima. Figure 1 shows the flowchart of the ISCA algorithm.



Start
Initialize population
Generate the parameters r1,r2,r3,r4
Update Alpha parameter by the Eq. (4)
Update position of current particle by the Eq. (5)
Calculate fitness value for each particle
Fit(t+1)<Fit(t)
No
Yes
Update the current best positionPi
Stopping criteria?
No
Yes
Stop

Fig 1. Flowchart of the ISCA Algorithm

4.1. Proposed ISCA Algorithm for feature selection

In this part, we introduce a model based on the ISCA for solving feature selection in IDSs. To solve this problem, we use the ISCA algorithm with a modification that creates a binary space by mapping the continuous search space, using the V-shaped transfer function. The transfer function with a V-shaped is a hyperbolic function that maps a real value in the range [0, 1] to a probability within the range [0, 1]. The probability values are calculated by Eq. 5 [32]:

$$V\left(x_i^d(t+1)\right) = \left|tanh(x_i^d(t))\right| \tag{5}$$

where $x_i^d(t)$ reveals the *i*th particle's position with dimension *d* and iteration *t*. Following the computation of the probability values, the position of each particle is calculated by Eq. 6.

$$b_i^d(t+1) = \begin{cases} x_i^d(t)^{-1} & if \quad r < V\left(x_i^d(t+1)\right) \\ x_i^d(t) & if \quad r \geq V(x_i^d(t+1)) \end{cases} \tag{6}$$

where, $b_i^d(t+1)$ is the i^{th} particle's binary position in dimension d at iteration t +1. Also, r is a random number in [0, 1].

The ISCA algorithm tackles the feature selection problem by using a population where each member represents a potential solution, i.e., a subset of available features. Each solution $x_i(t)$ in the population is a binary vector in which each element corresponds to a feature. A value of 1 represents the selection of a feature, while a value of 0 indicates non-selection. We adapt the ISCA algorithm for the feature selection problem by specifying the objective function that reflects the balance between feature count and accuracy chosen.

Let X represent the $N * D$ feature matrix, where N denotes the number of samples and D the number of features. Let Y be the corresponding label vector with dimensions $N * 1$. Our objective is to find a subset of features F that reduces the number of selected features while increasing classification accuracy. Moreover, to assess the quality of solutions, we employ a K-Nearest Neighbors (KNN) classifier [33]. Every time the algorithm runs, a subset of the features is selected by the solution, and the KNN is trained using the samples of data that correspond to the selected feature subset. The KNN classifier's accuracy is then calculated using the remaining data samples that were not used for training. This procedure helps to evaluate the performance of the solution regarding its capacity to classify new data samples accurately based on the selected features. The objective function is defined as follows:

$$Fit_i = \alpha E + \beta \frac{|F|}{D} \tag{7}$$

where E is the KNN classifier's error rate trained using the selected features, D is the dataset's total features, and |F| denotes the number of features that have been chosen. The weighting factors β and α check the proportionate significance of feature subset size and accuracy, respectively. In this paper, we set α to 0.99 and β to 0.01 based on previous studies [34].

5. Experimental results

In this section, the efficiency of the ISCA algorithm proposed in this paper is evaluated in selecting the optimal features for network intrusion detection. For this purpose, experiments were conducted using the dataset NSL-KDD by the ISCA algorithm and other competing metaheuristic algorithms including HHO [35], SSA [36], GWO [37], and DE [38] algorithms and two state - of – the art models. HHO is a population-based algorithm inspired by the cooperative hunting behavior of Harris hawks. It utilizes exploration and exploitation phases to efficiently search the solution space. SSA is a swarm-based optimization algorithm that mimics the foraging behavior of salps. It uses a leader-follower structure to guide the swarm towards the global optimum. GWO is a nature-inspired algorithm that simulates the social hierarchy and hunting mechanism of grey wolves. It employs an adaptive mechanism to balance exploration and exploitation. DE is an efficient optimization algorithm that utilizes mutation, crossover, and selection operations to evolve a population of candidate solutions towards the global optimum.

The KDD Cup99 dataset has been changed to create a preprocessed NSL-KDD dataset to remove duplicate records and reduce its size. The data includes simulated network traffic data, including various types of attacks, including probe attacks, remote-to-local (R2L), user-to-root (U2R), and denial of service (DoS) attacks [39]. By using the NSL-KDD data, we can assess how well our suggested algorithm works in detecting and classifying several attack kinds commonly found in network traffic data. The preprocessing steps utilized for the dataset ensure that the data is representative of real-world scenarios and can provide reliable evaluation results.

To ensure a fair comparison, all experiments are conducted in an identical experimental setup. We implement the ISCA and run all comparative algorithms in MATLAB R2019b programming language. We conduct 20 independent runs of all algorithms on a desktop PC equipped with a 2.40 GHz Intel® Core^{TM} i5 processor and 6.0 GB of RAM. The experiments are conducted using the MATLAB 2019b platform and run on Windows 7.

5.1. Data preprocessing

To perform the procedure for data mining and apply classification algorithms on the intrusion detection datasets, it is necessary to perform the preprocessing stage [40]. Some of the most significant pre-processing procedures in this study including data transfer, data normalization, and feature selection are performed. At first, all symbolic data are transferred into numerical values. Also, the data related to the data class are identified by the numbers 0 and 1, where 0 identifies the normal class and 1 indicates the attack class. Also, duplicate records are removed and missing values are managed. In the next step, data normalization is done. In the scaling process, each feature's data values are placed in a proportionate range. In this study, data normalization is done in the range [0, 1], based on the Eq. 8 [41].

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (8)$$

Where, X_{max} indicates the highest possible value of the X feature, and X_{min} determined the lowest possible number of the X feature. The $X_{normalized}$ value is the normalized equivalent of the X feature. Finally, the objective of the feature selection is to minimize the number of features in the dataset to boost the effectiveness of classification and reduce the complexity of calculations. In this study, a wrapper-based technique is proposed to minimize the number of data set features using an improved algorithm.

5.2. Parameter settings of algorithms and Evaluation metrics

For all experiments, we employ the KNN as a classifier with k = 5 to classify features obtained from the algorithms. To train the KNN, we perform K-fold cross-validation on each dataset, where K is set to 10. The dataset is split up into K identical pieces at random, with the remaining K-1 folds being employed for training and one-fold serving as the testing set. To ensure a fair comparison, we conduct 20 independent runs of all algorithms, using a uniform random distribution used to create the starting population. The highest number of iterations and population size are set at 100 and 20, respectively, across all algorithms. The algorithms' parameter configurations are consistent with their initial papers and are summarized in Table 1.

Table 1. Parameters setting of algorithms

Algorithms	**Parameters**
GWO	A= [2,0] (Linearly decreasing)
DE	CR= 0.7
SCA	r_1=[2,0] (Linearly decreasing), r_2=[0, 2] r_3=[0, 2] , r_4=[0, 1]
SSA	C1= [2,0] (Linearly decreasing)
HHO	E= [2, −2] →0 (Linearly decreasing)

The proposed ISCA and comparative algorithms are evaluated according to a range of performance measures, such as the mean fitness value, size of selected features, and accuracy. With D being the total count of features in the original data and $Avg.size^m$ being the mean number of features picked from the dataset, Eq. (9) calculates the mean of the proportion of chosen features to complete features across 20 runs.

$$Average\ selection = \frac{1}{20}\sum_{m=1}^{20}\frac{Avg.size^m}{D} \tag{9}$$

The fitness values' average is determined by the mean fitness value by running each of the algorithms 20 times independently as follows:

$$Mean\ fitness = \frac{1}{20}\sum_{m=1}^{20} f \tag{10}$$

Eq. (11) formulates the average accuracy value, which is the mean of the classification accuracy values acquired by executing the method 20 times. $Accuracy^{m}$ is the accuracy obtained from *the m* runs.

$$Average\ Accuracy = \frac{1}{20}\sum_{m=1}^{20} Accuracy^{m} \tag{11}$$

5.3. Numerical results and discussion

In this subsection, the outcomes of the ISCA algorithm are verified in contrast to a few cutting-edge feature selection techniques introduced in previous subsections. All similar methods have the same parameter values, and each algorithm yields results after 20 runs. Table 2 compares the ISCA algorithm with other competing algorithms regarding fitness value, accuracy, and the number of selected features on the NSL-KDD dataset. The obtained numerical outcomes in the table prove that the ISCA algorithm with a classification accuracy of 0.998 has a higher efficiency than other algorithms, and its fitness value is also the lowest among competitors with a value of 0.002. Regarding the quantity of features selected by the ISCA algorithm is less than other competing algorithms.

Table 2. Comparison of ISCA with competing algorithms of 20 runs

Algorithms	Accuracy	Fitness value	No. of features
BHHO	0.927	0.073	18
BSSA	0.925	0.075	21
BGWO	0.923	0.077	24
BDE	0.923	0.077	24
BHOA**[42]**	0.941	0.059	16.7
Alazzam et al.**[22]**	0.869	0.131	18
ISCA	**0.998**	**0.002**	**14**

In addition to the basic meta-heuristic algorithms, two other improved meta-heuristic algorithms were also considered for comparison. Both the BHOA[42] algorithm and the method proposed by Alazzam et al.[22]. In terms of accuracy, fitness, and the number of selected features, they are weaker than the proposed ISCA algorithm. But BHOA is considered the best algorithm after ISCA and is superior to other competitors.

Also, the convergence curve of the compared metaheuristic algorithms is presented in Figure 2. According to this figure, ISCA outperformed all competitive algorithms in convergence over the NSL-KDD dataset. Thus, the comparison of ISCA convergence to other algorithms proved its superior performance regarding fitness value. The convergence curve of the ISCA algorithm shows that the presence of the controller parameter introduced in the algorithm causes it to depart from the local optimum and converge to the general optimal solution.

The dynamic exploration-exploitation balance and the sine-cosine-based updates enable ISCA to identify the optimal or near-optimal feature subset, leading to improved classification accuracy compared to the original SCA. Moreover, the adaptive exploration-exploitation mechanism in ISCA accelerates the convergence of the algorithm, allowing it to reach the optimal solution in fewer iterations than the original SCA.

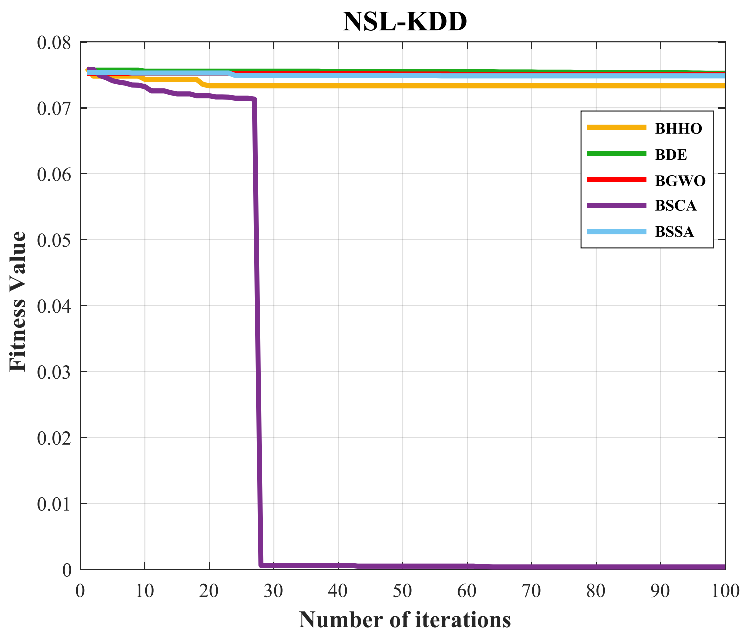
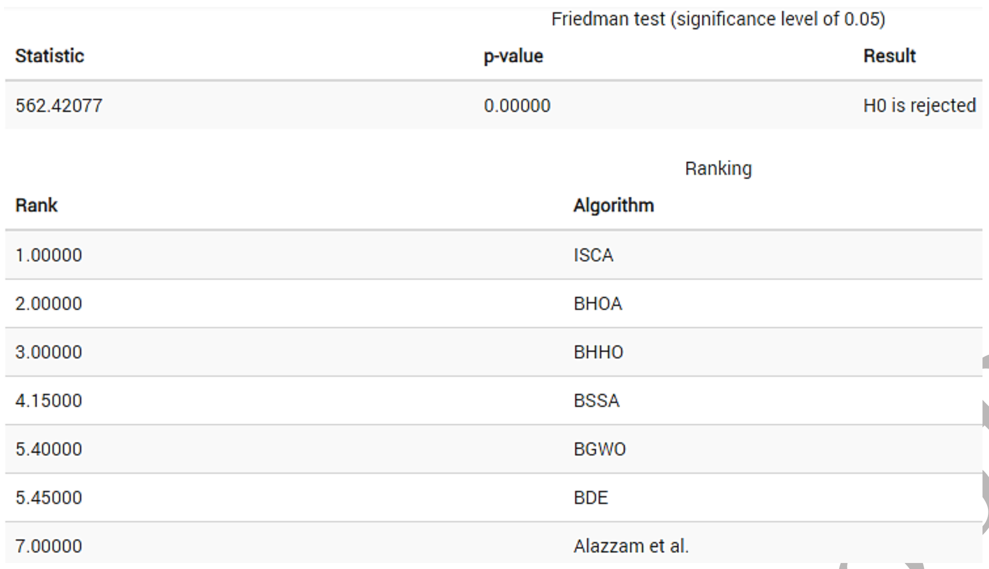


Fig 2. Convergence comparison of the ISCA and comparative algorithms

The implementation complexity of the proposed ISCA is O(N×M), where N represents the number of algorithm iterations and M is the population size. This is because the algorithm performs a fixed number of computations for each feature, including sine and cosine calculations, as well as the update step, and this is repeated for each iteration and over the entire population. However, in the proposed method, the computational complexity has not increased compared to the original SCA, and it remains at O(N×M). This complexity is lower than that of other comparative algorithms used for comparison, which means that the computation time of the improved SCA is also lower than its competitors.



Friedman test (significance level of 0.05)

Statistic	p-value	Result
562.42077	0.00000	H0 is rejected

Ranking

Rank	Algorithm
1.00000	ISCA
2.00000	BHOA
3.00000	BHHO
4.15000	BSSA
5.40000	BGWO
5.45000	BDE
7.00000	Alazzam et al.

Fig 3. Friedman test results

Friedman's statistical test can be used to compare the performance of the ISCA algorithm with other algorithms for feature selection in intrusion detection. The results of Friedman's test are displayed in Fig. 3, comparing the efficiency of the ISCA algorithm to that of its competitors based on fitness values. The figure illustrates that this algorithm outperformed other competing algorithms in terms of rank. Therefore, the ISCA algorithm has shown its effectiveness in addressing optimization problems, especially binary problems like feature selection.

The reasons for the high efficiency of the ISCA algorithm are: The ISCA algorithm employs a dynamic adjustment of the exploration-exploitation balance through the *Alpha* parameter. This parameter is controlled by the iteration count (t) and the maximum number of iterations (T). The dynamic balance allows the algorithm to effectively explore the search space in the early stages and gradually shift towards exploitation as the optimization progresses. This leads to more accurate identification of the optimal feature subset. The feature selection mechanism of the ISCA algorithm combines the sine and cosine functions in a novel way, allowing for a more comprehensive exploration of the feature space. This enhanced feature selection approach enables the algorithm to more effectively identify the most relevant features, resulting in a compact set of selected features while maintaining high classification accuracy.

6. Conclusion

Detecting network intrusions is a crucial assignment in preserving computer networks' integrity and security. However, network traffic data's high dimensionality presents a problem for intrusion detection systems, as it can result in a lot of features, many of which can be irrelevant or redundant. To overcome this challenge, feature selection is an essential stage in creating efficient intrusion detection systems, as it helps to minimize the dimensionality of the feature space and increase the classification model's accuracy. In this study, we proposed an improved model of the SCA to select effective features in network intrusion detection. Our proposed algorithm introduced a controlling parameter to balance exploration and exploitation, which enhanced the accuracy and addressed the dimensionality curse of the classification model. We evaluated the efficiency of our introduced ISCA algorithm on the NSL-KDD data and showed that it outperformed other metaheuristic algorithms in the number of features chosen and the accuracy of classifier. The proposed ISCA algorithm can be used as an effective tool for building intrusion detection systems that are both accurate and efficient. By lowering the feature space's dimensionality, our proposed algorithm helps to focus on the most essential characteristics and raise the classification model's accuracy.

Although our proposed algorithm showed promising results, there are still several avenues for future research. One direction is to investigate the use of hybrid algorithms that combine metaheuristic algorithms with other optimization techniques. Another direction is to evaluate the proposed algorithm's performance on other datasets and in real-world intrusion detection systems. Finally, more research is needed to investigate the interpretability of the selected features and their relevance to different types of cyber threats.

Declarations:

Authors' contributions: Conceptualization, Z.Asghari; methodology, Z.Asghari.; software, Z. Asghari; validation, Z. Asghari and S.Hosseini; formal analysis, Z. Asghari and S.Hosseini; investigation, S.Hosseini; resources, Z. Asghari; data curation, S.Hosseini.; writing—original draft preparation, Z. Asghari; writing—review and editing, S.Hosseini.; visualization, Z. Asghari.; supervision, S.Hosseini.; project administration, S.Hosseini.;

Funding: This research received no external funding.

Data Availability declaration: The datasets analyzed during the current study are available at the, https://ieee-dataport.org/documents/nsl-kdd-0. For the academic/public use of this dataset, the authors have to cite original papers.

Conflict of Interest: The authors declare that they have no conflict of interest.

Human Participants and/or Animals: This article does not contain any studies with human participants and/or animals performed by any of the authors.

Ethics approval and consent to participate: Not applicable.

Acknowledgments: Not applicable.

References

[1] G. A. Marin, "Network security basics," *IEEE security & privacy*, vol. 3, no. 6, pp. 68-72, 2005.

[2] J. R. Vacca, *Computer and information security handbook*. Newnes, 2012.

[3] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19-31, 2016.

[4] Z. A. Varzaneh and M. Kuchaki Rafsanjani, "Intrusion detection system using a new fuzzy rule-based classification system based on genetic algorithm," *Intelligent Decision Technologies*, vol. 15, no. 2, pp. 231-237, 2021.

[5] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16-24, 2013.

[6] B. Gupta, D. P. Agrawal, and S. Yamaguchi, *Handbook of research on modern cryptographic solutions for computer and cyber security*. IGI global, 2016.

[7] J. M. Vidal, M. A. S. Monge, and S. M. M. Monterrubio, "Anomaly-based intrusion detection: adapting to present and forthcoming communication environments," in *Handbook of Research on Machine and Deep Learning Applications for Cyber Security*: IGI Global, 2020, pp. 195-218.

[8] M. K. Rafsanjani and Z. A. Varzaneha, "Intrusion detection by data mining algorithms: a review," *Journal of New Results in Science*, vol. 2, no. 2, pp. 76-91, 2013.

[9] S. Axelsson, "Intrusion detection systems: A survey and taxonomy," 2000.

[10] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, "Building an intrusion detection system using a filter-based feature selection algorithm," *IEEE transactions on computers*, vol. 65, no. 10, pp. 2986-2998, 2016.

[11] A. Alazab, M. Hobbs, J. Abawajy, and M. Alazab, "Using feature selection for intrusion detection system," in *2012 international symposium on communications and information technologies (ISCIT)*, 2012, pp. 296-301: IEEE.

[12] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, pp. 152-160, 2018.

[13] Z. A. Varzaneh, S. Hossein, S. E. Mood, and M. M. Javidi, "A new hybrid feature selection based on Improved Equilibrium Optimization," *Chemometrics and Intelligent Laboratory Systems*, vol. 228, p. 104618, 2022.

[14] J. Huang, Y. Chen, A. A. Heidari, L. Liu, H. Chen, and G. Liang, "Improved Runge Kutta Optimization Using Compound Mutation Strategy in Reinforcement Learning Decision Making for Feature Selection," *Journal of Bionic Engineering,* pp. 1-37, 2024.

[15] A. I. Hammouri, M. A. Awadallah, M. S. Braik, M. A. Al-Betar, and M. Beseiso, "Improved Dwarf Mongoose Optimization Algorithm for Feature Selection: Application in Software Fault Prediction Datasets," *Journal of Bionic Engineering,* pp. 1-34, 2024.

[16] M. H. Nadimi-Shahraki, H. Zamani, Z. A. Varzaneh, A. S. Sadiq, and S. Mirjalili, "A Systematic Review of Applying Grey Wolf Optimizer, its Variants, and its Developments in Different Internet of Things Applications," *Internet of Things,* p. 101135, 2024.

[17] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Metaheuristic algorithms: A comprehensive review," *Computational intelligence for multimedia big data on the cloud with engineering applications,* pp. 185-231, 2018.

[18] X.-S. Yang, *Nature-inspired metaheuristic algorithms*. Luniver press, 2010.

[19] G. Hu, Y. Guo, and G. Sheng, "Salp Swarm Incorporated Adaptive Dwarf Mongoose Optimizer with Lévy Flight and Gbest-Guided Strategy," *Journal of Bionic Engineering,* pp. 1-35, 2024.

[20] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-based systems,* vol. 96, pp. 120-133, 2016.

[21] M. H. Nadimi-Shahraki, H. Zamani, Z. Asghari Varzaneh, and S. Mirjalili, "A systematic review of the whale optimization algorithm: theoretical foundation, improvements, and hybridizations," *Archives of Computational Methods in Engineering,* vol. 30, no. 7, pp. 4113-4159, 2023.

[22] H. Alazzam, A. Sharieh, and K. E. Sabri, "A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer," *Expert systems with applications,* vol. 148, p. 113249, 2020.

[23] J. Rene Beulah and D. Shalini Punithavathani, "A hybrid feature selection method for improved detection of wired/wireless network intrusions," *Wireless Personal Communications,* vol. 98, no. 2, pp. 1853-1869, 2018.

[24] R. Zhao, Y. Mu, L. Zou, and X. Wen, "A hybrid intrusion detection system based on feature selection and weighted stacking classifier," *IEEE Access,* vol. 10, pp. 71414-71426, 2022.

[25] R. Vijayanand and D. Devaraj, "A novel feature selection method using whale optimization algorithm and genetic operators for intrusion detection system in wireless mesh network," *IEEE Access,* vol. 8, pp. 56847-56854, 2020.

[26] A. K. Shukla, "Detection of anomaly intrusion utilizing self-adaptive grasshopper optimization algorithm," *Neural Computing and Applications,* vol. 33, no. 13, pp. 7541-7561, 2021.

[27] F. Salo, A. B. Nassif, and A. Essex, "Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection," *Computer networks,* vol. 148, pp. 164-175, 2019.

[28] I. Ahmed, A. Dahou, S. A. Chelloug, M. A. Al-qaness, and M. A. Elaziz, "Feature selection model based on gorilla troops optimizer for intrusion detection systems," *Journal of Sensors,* vol. 2022, pp. 1-12, 2022.

[29] M. Safaldin, M. Otair, and L. Abualigah, "Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks," *Journal of ambient intelligence and humanized computing,* vol. 12, pp. 1559-1576, 2021.

[30] D. Jayalatchumy, R. Ramalingam, A. Balakrishnan, M. Safran, and S. Alfarhood, "Improved Crow Search-Based Feature Selection and Ensemble Learning for IoT Intrusion Detection," *IEEE Access,* 2024.

[31] U. Nandhini and S. K. SVN, "An improved Harris Hawks optimizer based feature selection technique with effective two-staged classifier for network intrusion detection system," *Peer-to-Peer Networking and Applications,* pp. 1-35, 2024.

[32] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "BGSA: binary gravitational search algorithm," *Natural computing,* vol. 9, pp. 727-745, 2010.

[33] P. Cunningham and S. J. Delany, "k-Nearest neighbour classifiers-A Tutorial," *ACM computing surveys (CSUR),* vol. 54, no. 6, pp. 1-25, 2021.

[34] N. Al-Madi, H. Faris, and S. Mirjalili, "Binary multi-verse optimization algorithm for global optimization and discrete problems," *International Journal of Machine Learning and Cybernetics*, vol. 10, pp. 3445-3465, 2019.

[35] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future generation computer systems,* vol. 97, pp. 849-872, 2019.

[36] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems," *Advances in engineering software,* vol. 114, pp. 163-191, 2017.

[37] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software,* vol. 69, pp. 46-61, 2014.

[38] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization,* vol. 11, pp. 341-359, 1997.

[39] S. Revathi and A. Malathi, "A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection," *International Journal of Engineering Research & Technology (IJERT),* vol. 2, no. 12, pp. 1848-1853, 2013.

[40] S. García, J. Luengo, and F. Herrera, *Data preprocessing in data mining*. Springer, 2015.

[41] S. K. Sahu, S. Sarangi, and S. K. Jena, "A detail analysis on intrusion detection datasets," in *2014 IEEE international advance computing conference (IACC),* 2014, pp. 1348-1353: IEEE.

[42] Z. Asghari Varzaneh, S. Hosseini, and M. M. Javidi, "A novel binary horse herd optimization algorithm for feature selection problem," *Multimedia Tools and Applications,* pp. 1-35, 2023.