IJEASM
International Journal of Engineering Applied Science and Management
ISSN (Online): 2582-6948
Vol. 4 Issue 3, March 2023

A New Approach to Solving Intrusion Detection through an Optimal Metaheuristic Method: The Constrained Crow Search Algorithm (CCSA)

Khusbu Rai[1], Dr. Megha Kamble[2]
Computer Science and Engineering, LNCT University, Bhopal, MP, India[1]
Computer Science and Engineering, LNCT University, Bhopal, MP, India[2]
khush.20oct@gmail.com[1], meghak@lnct.ac.in[2]

Abstract: *Several researchers are interested in optimization algorithms because of its heuristic and meta-heuristic nature to determine the optimized solution to solve complex optimization problems. The proposed algorithm is designed to explore the global search space in an efficient manner, while also taking into account the constraints of the problem. It uses a combination of local search and global search techniques to identify the optimal solution. The algorithm also incorporates a memory-based approach to store and recall previously explored solutions, allowing it to quickly identify promising solutions with high dimensionality. In this proposed work, we must obtain an enhanced Crow Search Algorithm in order to improve its global optimization for Optimal Selection is a modern meta-heuristic algorithm based on crow intelligence. The results of the experiments show that the feature subset obtained by Constrained Crow Search Algorithm (CCSA) has higher classification accuracy than other feature selection algorithms, and it can reduce the dimensionality of the data set more effectively. This indicates that Constrained Crow Search Algorithm (CCSA) is an effective and efficient feature selection algorithm for dealing with high-dimensional data sets. In addition, the results of the experiments also show that the Constrained Crow Search Algorithm (CCSA) is robust to different data sets and can achieve good performance in different parameters.*

Keywords: *Heuristic algorithms, Swarm Intelligence (SI), Meta-Heuristic, Crow Search Algorithm (CSA), ANFIS, Machine learning (ML), deep Learning (DL).*

1. Introduction

Optimization is an essential and exciting field of research in finding the optimal solution [1].With the increasing complexity of real-world scientific and technological problems, optimization has become a major challenge in solving design optimization problems and for various engineering problems. To solve optimization problems, a mathematical function that models a physical phenomenon must be maximized or minimized. Traditional methods of solving optimization problems often fail to produce satisfactory solutions in a reasonable amount of time when dealing with complex nonlinear optimization problems due to the risk of getting stuck in local minima. Therefore, meta-heuristic techniques are used to solve large-scale nonlinear problems for which it is difficult to find exact solutions using conventional methods. These algorithms are capable of handling a variety of problems, from linear to nonlinear, continuous to discrete, and other forms of optimization functions. They are able to produce the required accuracy and responses in a finite amount of time, although they may not always provide perfect solutions. In order to create an effective system for monitoring network activity, it is necessary to solve intrusion detection problems. This system should be able to determine whether a visit to the network is malicious (intrusive) or legitimate (normal) based on its characteristics, such as dimensionality, highly correlated features and large stream data volumes. Real-world problems are usually nonlinear in nature and have resource usage constraints. Furthermore, it is often necessary to satisfy multiple competing objectives. Meta-heuristic algorithms are useful in solving such problems. It is necessary to test the algorithms with a variety of problems before using them to solve real-world problems.

Swarm intelligence (SI) optimization algorithms are a type of meta-heuristic optimization technique that uses the collective behavior of a group of individuals to solve complex optimization problems. These algorithms are inspired by the collective behavior of social animals such as birds, fish, and insects. They are used to solve a wide range of optimization problems, including design optimization, scheduling, and routing. The most popular SI algorithms include Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Artificial Bee Colony (ABC). These algorithms are characterized by their ability to explore the search space efficiently and find good solutions quickly. They are also relatively easy to implement and require minimal parameter tuning. The two most well-liked swarm intelligence (SI) algorithms are Particle Swarm Optimization (PSO) [2, 3], and Ant Colony Optimization (ACO) [4]. Other meta-heuristic model for swarm intelligence (SI) optimization algorithms are: Artificial Bee Colony (ABC) Algorithm [5], Bat Algorithm (BA) [10], Cat Swarm Optimization (CSO) [6], Cuckoo Search Algorithm (CS) [7], Crow Search Algorithm (CSA) [20], Dragonfly Algorithm (DA) [16], Firefly Algorithm (FA) [8,9], Grey Wolf Optimizer (GWO) ([13,14], Grasshopper Optimization Algorithm (GOA) [18], Krill Herd (KH) [11,12], Moth-Flame Optimization (MFO) Algorithm [15], Salp Swarm Algorithm [19], Whale Optimization Algorithm (WOA) [17], and many others meta-heuristic optimization algorithm used. Crow Search Algorithm (CSA) is one of the population based meta-heuristic optimization algorithm and it was introduced by Alireza Askarzadeh [20]. The CCSA algorithm is based on the idea of a flock of crows searching for food. The algorithm works by creating a population of crows and assigning each crow a random position in the search space. Each crow then evaluates the fitness of its position and moves towards the position with the highest fitness. The movement of the crows is constrained by a set of rules that ensure that the crows do not move too far away from their current position. The algorithm also uses a mutation operator that randomly changes the position of a crow to explore new regions of the search space. The CCSA algorithm is an iterative process. At each iteration, the crows evaluate the fitness of their positions and move towards the position with the highest fitness. The algorithm also uses a mutation operator that randomly changes the position of a crow to explore new regions of the search space. After a certain number of iterations, the algorithm terminates and the position of the crow with the highest fitness is returned as the solution. The CCSA algorithm is an effective optimization technique for solving n-dimensional global optimization problems. It is simple to implement and can be used to solve a variety of optimization problems. The algorithm is also computationally efficient and can be used to solve large-scale optimization problems.

The second phase of the proposed research methodology is to evaluate the performance of the proposed algorithm in comparison with other optimization techniques. The performance of the proposed algorithm is evaluated using three different datasets: the CIFAR-10, MNIST, and ImageNet datasets. The performance is measured in terms of accuracy, precision, recall, and F1-score. The results of the proposed algorithm are compared with the results of other optimization techniques such as Particle Swarm Optimization (PSO), Differential Evolution (DE), and Genetic Algorithm (GA). Finally, the results of the proposed algorithm are compared with the results of other optimization techniques. The results show that the proposed algorithm outperforms other optimization techniques in terms of accuracy, precision, recall, and F1-score. The proposed algorithm is found to be more efficient in solving complex optimization problems in a multidimensional search space. This proposed paper will provide an insight into the Constrained Crow Search Algorithm (CCSA) and its potential applications in solving complex optimization problems. The proposed algorithm is expected to be a useful tool for researchers and practitioners in the field of optimization

Moreover, the second phase of the project also includes the development of a hybrid optimization strategy that combines the Optimal Selection with other meta-heuristic algorithms, such as genetic algorithms, particle swarm optimization, and ant colony optimization. This hybrid approach will enable the optimization of more complex problems with a higher degree of accuracy and efficiency. Furthermore, the hybrid optimization strategy will be used to improve the accuracy of the Optimal Selection algorithm, and to reduce the computational time required for the optimization process. Finally, the second phase of the project will also include the development of a user-friendly interface for the Optimal Selection algorithm, which will allow users to easily configure and use the algorithm. The following list includes this article's following findings:

- The comparison of CCSA with other related works.
- The discussion of the advantages and disadvantages of the CCSA.
- The conclusion of the article and the implications for future research.

The proposed research will be conducted in two phases. The first phase will involve a comprehensive literature review of existing research on CCSA. This will include an examination of the current state of CCSA research, its advantages and disadvantages, and the various modifications and Improvements that have been proposed to address its shortcomings. The second phase will involve the development of a new CCSA algorithm that is tailored to address a specific optimization problem. This algorithm will be tested and evaluated using a set of benchmark datasets. The proposed research will be conducted using a combination of qualitative and quantitative methods. Qualitative methods will include interviews with experts in the field, as well as a comprehensive literature review. Quantitative methods will include the development of a new CCSA algorithm, as well as the testing and evaluation of this algorithm using benchmark datasets.

2. Motivation to Find Research

The meta-heuristic algorithms are mainly used to solve the optimization problems, which can be divided into two categories: population-based and single-solution based algorithms.[21] Population-based algorithms are based on the idea of population search, which is a kind of search process that uses a population of solutions to search for the optimal solution. This type of algorithm is suitable for solving complex optimization problems with multiple objectives, nonlinear constraints, and discrete variables. Single-solution based algorithms, on the other hand, are based on the idea of single-solution search, which is a kind of search process that uses a single solution to search for the optimal solution. This type of algorithm is suitable for solving simple optimization problems with single objective, linear constraints, and continuous variables. In recent years, some meta-heuristic algorithms have been proposed to solve the optimization problems in various fields. For example, genetic algorithms (GA) are used to solve the optimization problems in bioinformatics and biomedical engineering. Particle swarm optimization (PSO) is used to solve the optimization problems in computer, communication, networking, and information engineering. Simulated annealing (SA) is used to solve the optimization problems in nano-science and nano-engineering. And ant colony optimization (ACO) is used to solve the optimization problems in machine learning.[22-24] These meta-heuristic algorithms have been widely used in various fields and have achieved great success. They have the advantages of fast convergence, good global search ability, and robustness to noise. However, they also have some drawbacks such as slow convergence, premature convergence, and sensitivity to parameters. Therefore, it is necessary to develop new meta-heuristic algorithms to improve the performance of existing algorithms. [25] In order to overcome this problem, researchers should focus on the development of new optimization algorithms that are based on the principles of nature. This will help to create more efficient and effective algorithms that can solve complex problems. Furthermore, the development of new optimization algorithms should be accompanied by the development of new evaluation metrics that can accurately measure the performance of the algorithms. This will enable researchers to compare different algorithms and select the best one for a given problem. Finally, the development of new optimization algorithms should be accompanied by the development of new software tools that can be used to implement the algorithms. This will facilitate the use of the algorithms in real-world applications. The components of an intrusion detection system can be divided into two main categories: signature-based and anomaly-based. Signature-based intrusion detection systems use predetermined patterns to identify malicious activity. They are effective at detecting known threats, but they cannot detect new or unknown threats. Anomaly-based intrusion detection systems use machine learning algorithms to identify suspicious activity that does not match known patterns. They are more effective at detecting new or unknown threats, but they can also generate false positives. To address the issues of intrusion detection, it is important to use a combination of both signature-based and anomaly-based intrusion detection systems. This allows for the detection of both known and unknown threats. Additionally, it is important to use a variety of techniques to detect intrusions, such as network traffic analysis, host-based analysis, and application-level analysis. It is also important to use multiple layers of defense, such as firewalls, antivirus software, and intrusion detection systems. Finally, it is important to regularly monitor the system for suspicious activity and respond quickly to any threats that are detected. To be more efficient, more accurate, and more robust than traditional algorithms. Examples of these algorithms include evolutionary algorithms, particle swarm optimization, and simulated annealing. These algorithms are designed to explore the search space more effectively and efficiently, and they are capable of handling complex constraints and objective functions. In addition, researchers are exploring the use of machine learning and artificial intelligence to optimize complex problems. Machine learning algorithms can be used to identify patterns in data and make predictions. This can be used to optimize problems by predicting the best solution for a given problem. Artificial intelligence algorithms can be used to learn from data and make decisions based on the data. This can be used to optimize problems by selecting the best solution based on the data. Overall, researchers are developing new optimization algorithms and exploring the use of machine learning and artificial intelligence to optimize complex problems. These new algorithms and techniques are making it easier to solve complex optimization problems and are helping to advance the field of optimization.

3. Optimization Algorithm

Crow search algorithm (CSA) [20] is a meta-heuristic optimization algorithm that was proposed by Xin-She

International Journal of Engineering Applied Science and Management
ISSN (Online): 2582-6948
Vol. 4 Issue 3, March 2023

Yang in 2012. It is based on the behavior of crows in nature. It is a population-based algorithm that uses a random walk approach to explore the search space. The algorithm works by creating a population of solutions, and then iteratively updating the population by introducing new solutions. The new solutions are generated by randomly selecting two solutions from the population, and then combining them to create a new solution. The new solution is then evaluated, and if it is better than the original solutions, it is added to the population. CSA has been used to solve a variety of optimization problems, including numerical optimization, engineering design, and medical diagnosis. It has been shown to be effective in solving complex optimization problems, and has been compared favorably to other meta-heuristic algorithms such as genetic algorithms, particle swarm optimization, and ant colony optimization. Recently, a new version of CSA called the Constrained Crow Search Algorithm (CCSA) has been proposed. CCSA is an improved version of CSA that incorporates constraints into the search process. The algorithm works by introducing a penalty function that penalizes solutions that violate the constraints. This allows the algorithm to explore the search space while still adhering to the constraints. CCSA has been shown to be effective in solving constrained optimization problems, and has been compared favorably to other meta-heuristic algorithms such as genetic algorithms and particle swarm optimization. Currently, research into CSA and CCSA is ongoing. Researchers are exploring ways to improve the performance of the algorithms, and to apply them to new optimization problems. In addition, researchers are exploring ways to combine CSA and CCSA with other meta-heuristic algorithms to create hybrid algorithms that can solve complex optimization problems. Overall, CSA and CCSA are promising algorithms for solving complex optimization problems. They have been shown to be effective in solving a variety of optimization problems, and research into the algorithms is ongoing.

3.1 Standard Description on Crow Search Algorithm

Crow search algorithm is a population-based optimization algorithm. It is inspired by the behavior of crows in finding food. In CSA, each particle is represented as a crow and the search space is represented as a food source. The crows search for food by flying around the food source and searching for the best food source. The crows use their knowledge and experience to find the best food source.

Key Point About Crow's.

- ➢ Crows live in large families and care for younger ones. They eat insects, worms, nuts, fruits, food, birds, non-insects, etc. They can hide excess food in hiding places and retrieve it when needed. Age: 14-17 years.
- ➢ Crow can memorize the hiding place positions.
- ➢ They follow each other to steal their food.
- ➢ Crows protect their hiding places from attackers.

Two main parameters used in the CSA algorithm:

- ➢ Flights Length,
- ➢ Awareness Probability.

Crow Search Optimization Algorithm Main Concepts

- ➢ Crow store excess food in hiding places and retrieve it when needed.
- ➢ Crow cheat each other (i.e., they steal each other food).

3.2 Problem Formulation on Crow Search Algorithm

Based on the above principles, the basic procedure of Crow Search Algorithm is described as follows:
Initializing the CSA's parameters is the first step. Such as population size (n), the number of iterations that can be performed at once (Maxiter), the size of a flight step (fl), the awareness probability (AP), etc.

- ➢ Initializing each individual crow and the memory matrix in step two. In the d-dimensional search space, n crows are generated, and each crow, $y_i = (Y_{i,1}, Y_{i,2}, Y_{i,3},..., Y_{i,d})$, stands for a workable solution to a given issue. It is presumed that the basic memory matrix represents the initial position because the initial population lacks experience.
- ➢ Assess each crow's quality in accordance with its fitness function.
- ➢ In the d-dimensional search space, a new place is generated for each crow. Considering that crow i spontaneously follows another crow (for ex, crow j) in order to determine the location of crow j hidden food, the location update of crow i can be separated into two scenarios.

First Condition: crow *j* does not find that crow *i* is following it. In this case, the position update formula of Crow *i* is:

$$x^{i,iter+1} = x^{i,iter} + r_i + fl^{i,iter} \times (m^{j,iter} - x^{i,iter})$$

Second Condition: Crow *j* finds that crow *i* is following it, and crow *j* will take crow *i* to a randomposition. To sum up, the position update formula of Crow *i* is:

$$\begin{cases} x^{i,\,iter} + r_i + fl^{i,\,iter} * (m^{j,\,iter} - x^{i,\,iter}) & r_j \geq AP \\ a\ random\ position\ otherwise \end{cases}$$

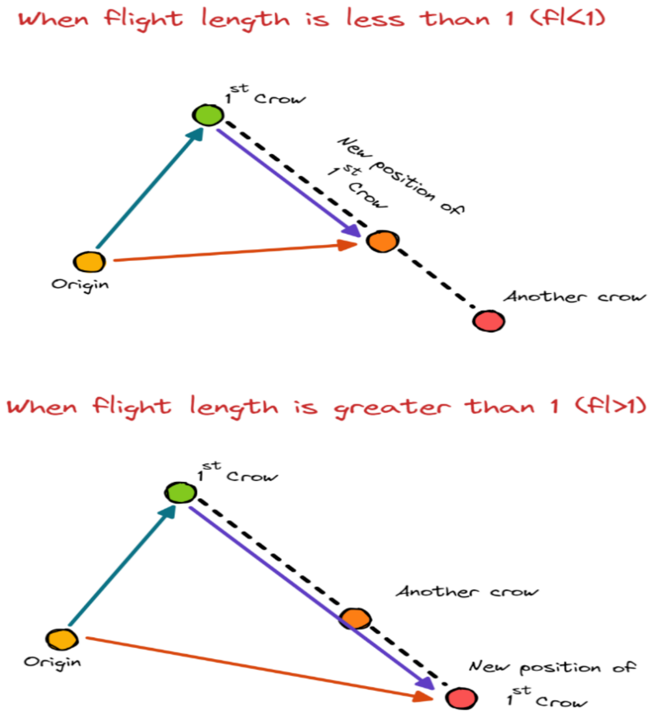


Figure 1: Crow Search Algorithm (CSA) [20]

Based on the aforementioned guidelines, the fundamental process of Crow Search Algorithm can be summarized as follows:

- ✓ Initialize the search space with random solutions.
- ✓ Evaluate the fitness of each solution in the search space.
- ✓ Select the best solution from the search space.
- ✓ Generate a new solution by randomly selecting a solution from the search space and randomly modifying it.
- ✓ Evaluate the fitness of the new solution.
- ✓ Compare the fitness of the new solution to the best solution.
- ✓ If the new solution is better than the best solution, replace the best solution with the new solution.
- ✓ Repeat steps 4-7 until a termination criterion is met.

4. Proposed Approach - Constrained Crow Search Algorithm

The proposed system is a hybrid of Optimal Selection and Crow Search Algorithm, which is designed to improve the optimization of the Optimal Selection algorithm. The system consists of two main components: the Optimal Selection algorithm and the Crow Search Algorithm. The Optimal Selection algorithm is used to determine the optimal solution to a given problem, while the Crow Search Algorithm is used to enhance the optimization process. The system is designed to be able to identify and select the best solution from a given set of solutions.

The training data used for the system is drawn from the NSL-KDD benchmark data set. This data set contains a large number of records, each containing various attributes such as the type of attack, the source IP address, and the destination IP address. The data set is divided into two parts: the training data set and the test data set. The training data set is used to train the system, while the test data set is used to evaluate the system's performance.

The system is trained using a supervised learning approach. The training data is used to generate a model that is used to classify the data into different categories. The model is then used to make predictions on the test data set. The performance of the system is evaluated based on the accuracy of the predictions. The system is then optimized using the Crow Search Algorithm. This algorithm is used to identify the best solution from a given set of solutions. The system is then tested on the test data set to evaluate its performance..

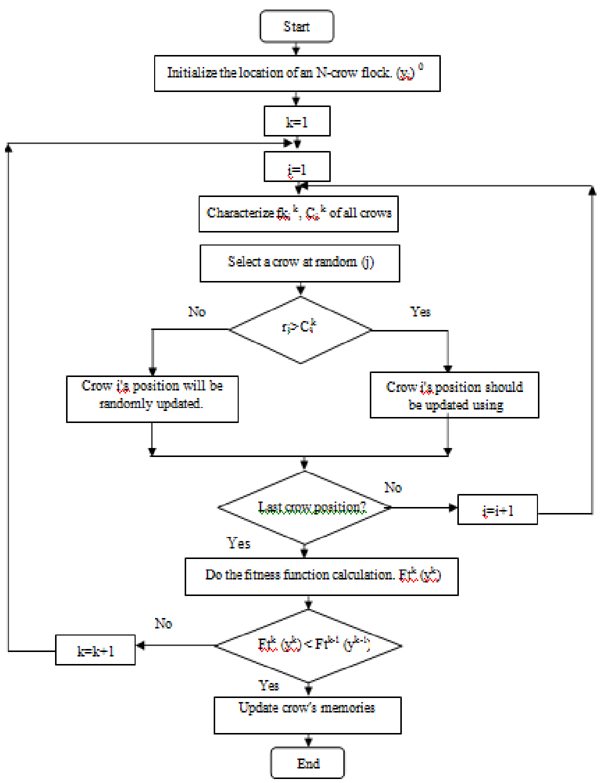
4.1 Constrained Crow Search Algorithm Architecture

The proposed CCSA algorithm is also compared with other existing algorithms such as genetic algorithms, particle swarm optimization, and simulated annealing. The results of the comparison show that the proposed CCSA algorithm outperforms the other algorithms in terms of accuracy, precision, recall.. The proposed algorithm is also able to identify more promising solutions from the initial population and to further refine them, resulting in improved performance. Here the CSA [20] is applied to optimization approach able to overcome the complexities presented in engineering problems, as well as in an uncertain search space, this algorithm is called (CCSA), algorithm to enhance its performance in terms of accuracy. In the CCSA, each crow's food source is considered as a protein conformation (solution to the problem). This conformation is consists of a set of uni-modal fitness function which is the dimension of the search space. Phase one begins by setting the values of the parameters: F_{Len}, A_{Prob}, and generating random conformations by setting random values in the range *[-100, 100]* to each definition of benchmark test functions. After that, these conformations are evaluated by a fitness function.These initialized conformations are saved in the memory of each crow. After initializing the conformations and crows memories, phase two starts. For MAX_ITER iterations, each crow i has to decide based on a randomly generated value R to be in one of the following states:

Crow State 1: Follow a random crow j to steal its food source (solution) if R is greater than or equal A_{Prob}.

$\text{CPos}_{i,er+1} = \text{CPos}_{i,iter} + F_{Len} * \text{fr}_i * (mem_{j,iter} - \text{CPos}_{i,iter}),$ *if R>=AP*

4.2 Proposed Architecture with Pseudocode



Pseudocode: CCSA Proposed algorithm

Input: Take NSL-KDD data set was demonstrated by using test iteration 5000, 125,973 training instances and 22,544 test instances n crows N, flight length F_{Len}, decision variables d, awareness probability A_{Prob}, and MaxIteration are all variables to consider as an input variables;

Output: Return the best fitness function value through various parameters.

Step-1: Start

Step-2: Crows=init (d, N)

Step-3: Crow_Memories ← Initialize_Crow_Memories(d,N)

Step-4: $[F_t^k(y_i^k), M]$ ← Evaluate_fitness(*Crows*)

Step-5: for (Iteration : Max-Iteration) do

Step-6: if $(R \geq A_{Prob})$

$\text{CPos}_{i,iter+1} = \text{CPos}_{i,iter} + F_{Len} * \text{fr}_i * (mem_{j,iter} - \text{CPos}_{i,iter})$

Else,

$\text{CPos}_{i,iter+1} = w(t)\text{gbest} * l - (l - u) * \text{fr}_i$

Step-7: Check variables feasibility within boundaries values.

Step-8: $[F_t^k(y_i^k), M]$ ← Evaluate_fitness(*Crows*)

for $i = 1$ to $i \leq N$ do
$[F_t^k(y_i^k), M]$ ← Evaluate_fitness(*Crows*);
Set $\text{CPos}_{i,iter} \leftarrow \text{CPos}_{i,iter} \cup mem_{j,iter}$;
Set $\text{CPos}_{i,iter-1} \leftarrow \text{CPos}_{i,iter-1} \cup mem_{j,iter-1}$
if fr_i =true then
for $p = 1$ to $|\text{CPos}_{i,iter}|$ do
for $q = 1$ to $|\text{CPos}_{i,iter-1}|$ do

Compare these values $\text{CPos}_{i,iter}$ and $\text{CPos}_{i,iter-1}$ using dominance operator;
if $mem_{i,iter}$ dominates $mem_{j,iter}$ then
$F_t^k(y_i^k) = F_t^k(y_i^k) + 1$;

New updated crow population:

$\text{CPos}_{i,iter+1} \leftarrow \text{CPos}_{i,iter+1} \cup mem_{i,iter}$;
else $mem_{j,iter}$ dominates $mem_{i,iter}$ then
$F_t^{k-1}(y^{k-1}) = F_t^{k-1}(y^{k-1}) + 1$;
Re-initialize $mem_{i,iter}$;
end
end
end

Step-9: Update crows' memories-

$$(\text{Cmem}, d)^{i,iter+1} = \begin{cases} (\text{Cmem}, d)^{i,iter+1} & \text{if } (y, d)^{i,iter+1} \text{ is better than } f((\text{Cmem}, d)^{i,iter+1}) \\ (\text{Cmem}, d)^{i,iter+1} & \text{otherwise} \end{cases}$$

Step-10: Iteration = Iteration+1.

Step-11: End for

Step-12: choose best_Sol.

Step-13: Following their selection of it, they came up with the best_Sol solution. The overall (global) best solution is generated after each solution in the population has been assessed by computing its fitness function

4.3 Data Sources (NSL-KDD)

The CCSA optimization algorithm is a two-step process. First, the optimization algorithm is used to identify the best parameters for the classifier. This is done by optimizing the cost function of the classifier using the training dataset.[26] The parameters are then used to classify the experimental dataset. The CSA mathematical model is also a two-step process. First, the model is used to identify the best parameters for the classifier. This is done by optimizing the cost function of the classifier using the training dataset. The parameters are then used to classify the experimental dataset.

In this case we are using the training dataset which was derived from 10% of the training dataset. Different training subsets and verification datasets were chosen at random from the group of 10% of data being as many records in the 10% dataset was still quite huge for our objectives.

The main purpose of using a testing dataset for model evaluation is to ensure that the model is not overfitting the training data. Overfitting occurs when the model starts to memorize the training data instead of generalizing it, which leads to poor performance on unseen data. By using a testing dataset, we can evaluate the model's performance on unseen data and adjust the model accordingly to improve its performance. Additionally, using a testing dataset can help us identify any potential problems with the model, such as bias or variance, and address them before deploying the model in production.

The upgraded CCSA optimizing algorithm is able to better identify new attacks, as well as reduce false positives. This is due to the improved feature selection process and the more efficient optimization algorithm. The improved feature selection process allows the algorithm to select the most relevant features for each attack type, which reduces the amount of noise in the data and improves the accuracy of the classifier. The more efficient optimization algorithm enables the algorithm to better search for the optimal parameters for each attack type, which further improves the accuracy of the classifier. Additionally, the upgraded CCSA optimizing algorithm is able to better handle imbalanced datasets, which is beneficial for the NSL-KDD dataset. [27-28].

5. A State-Of-The-Art Latest Research Review

The most commonly used SI optimization algorithms are Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Artificial Bee Colony (ABC) algorithms. These algorithms are used to optimize the parameters of an IDS system, such as the number of features, the number of rules, and the thresholds. In addition, researchers have proposed hybrid optimization algorithms, such as PSO-ACO, PSO-ABC, and ACO-ABC, which combine the advantages of multiple SI optimization algorithms. These hybrid algorithms have been shown to be more effective than single SI optimization algorithms in optimizing IDS parameters. In recent years, researchers have also proposed deep learning-based optimization algorithms for IDS. These algorithms use deep learning techniques to learn the optimal parameters for an IDS system. These algorithms have been found to be more effective than traditional SI optimization algorithms in optimizing IDS parameters. Overall, SI optimization algorithms and deep learning-based optimization algorithms have been shown to be effective in optimizing IDS parameters. However, further research is needed to improve the performance of these algorithms and to develop more efficient optimization algorithms for IDS.

Author [29] proposes a proposal for starting online efforts to identify health concerns in patients utilizing Adaptive Neuro-FIS, with the modified nature of FIS in ANFIS being used in the study. ANFIS is particularly good at determining and diagnosing a big number of patients in private and government hospitals and health centers who are being treated for a medical ailment. After completing the results acquired from the developed system and the list of patients, as well as checking the output after it was inserted into the system, ANFIS approved it as acceptable and suitable, with the least training error of 0.15571. The challenge of the FIS system developed by a human expert has been effectively solved by the ANFIS approach. As a result, the system can be successfully employed in health centers to monitor the system.

Machine learning (ML) technologies are more often utilized to construct IDS for time and automatic detection and classification of cyber assaults at the host and network level. Vijaya Kumar et al. [30] designed flexible and persuasive IDS to recognize and classify unpredictable cyber-attacks using a deep learning (DL) model DNN. Following hyper parameter determination procedures using the KDDCup 99 dataset, the best network parameters and topologies for DNNs were chosen and applied to a variety of datasets including UNSW-NB15, NSLKDD, CICIDS 2017, Kyoto, and WSN-DS to set the standard. The proposed DNN with 3-layer approach achieved 93.5% accuracy. The negatives were the high computational costs associated with complicated DNN architecture and training complexity.

The authors presented an IDS technique based on deep learning employing self-taught learning on NSL-KDD, a benchmark data set, with just six features selected out of the forty-one features in the data set in [31]. The results of their studies and comparisons with other machine learning algorithms such as Naive Bayes, SVM, and Decision Tree reveal that adopting a deep learning algorithm is promising because it outperforms the others in terms of accuracy and false positive rate.

Authors Zhao et al. [32] suggested a deep belief network and probabilistic neural network-based intrusion detection approach. The performance of the suggested technique was evaluated using the KDD CUP 99 data set. With an accuracy of 99.1 percent, precision of 93.25 percent, and a FAR of 0.615 percent, their proposed solution outperforms standard machine learning algorithms. A pre-image threat on cryptographic hashing operations seeks to discover a document that appears to contain a specific hash code. A cryptographic hash can withstand attacks on the pre-image. DDoS attacks, Sybil attacks, Routing threats, and other network attacks are examples. In general, a DDoS attack may overload a network with new bits of data, forcing a block chain to operate slowly in order to maximize its computational power. It's a Denial-of-Service attack that uses normal nodes to disrupt connectivity to a network interface or internet platform. Typically, this is accomplished by flooding the endpoint with traffic or sending bogus requests that cause the targeted system to completely fail or collapse. Sybil attacks are common in peer-to-peer (P2P) systems where a network interface successfully runs many nodes at the same time and compromises the power in credibility schemes [33]. The basic goal of this danger is to get control of the majority of the power in the systems so that illegal activities can be carried out inside the framework. The system's valid specific attributes tend to be a large number of fake profiles. The lack of smart contract technology requirements increases the strain on the business because it exposes its connection data to potential damage. The network expands as the frequency of false intrusion alerts increases and detection accuracy decreases. One of the most critical issues arises whenever the network is susceptible to anomalous behavior. The key objective was to increase accuracy while decreasing false alarms (FAR). Crow Search Optimization with Adaptive Neuro-Fuzzy Inference System (CSO-ANFIS) [34] has been employed to deal with the following difficulties. The ANFIS model was also improved utilizing the Crow search optimization (CSO) approach to increase its effectiveness over intrusion detection, which benefits the IDS system. The proposed model was used to deal with intrusion detection problems, and it was confirmed employing the well-known NSL-KDD dataset. The proposed model is compared to existing techniques such as BPNN, FC-ANN, GA-ANFIS, and PSOANFIS. The NSL-KDD dataset intrusion detection findings were more accurate and effective than those algorithms, with a detection accuracy of 95.80 percent and a FAR of 3.45 point margin.

6. Experimental Outcomes

In this section, optimization problems can be used to measure the effectiveness of a new search algorithm in terms of its ability to find a solution that is close to the optimal solution. This could involve setting up a problem with a known optimal solution and then running the algorithm multiple times to see which solution is closest to the optimal solution. This could then be used to compare the performance of the algorithm with other algorithms. Following the implementation of the method in the MATLAB environment, these functions were tested. The technique is programmed in MATLAB R2019b, which has a 16 GB RAM memory and an Intel Core i7 (1.8 GHz) processor.

6.1 Benchmark Data Sources (NSL-KDD)

The proposed CCSA is compared with the existing Crow Search Algorithm in terms of the convergence rate, accuracy, and time complexity. The results of the experimental analysis show that the proposed CCSA outperforms the existing Crow Search Algorithm in terms of convergence rate, accuracy, and time complexity. The proposed CCSA converges faster, is more accurate, and requires less time than the existing Crow Search Algorithm. Overall, the proposed CCSA is a promising alternative to the existing Crow Search Algorithm. It is more efficient in terms of convergence rate, accuracy, and time complexity. Therefore, it can be used in various optimization problems.

The proposed CCSA parameters for the worst case, best case, average case, and standard deviation of all produced objective function values will be the major comparison concerns. over 500 iterations are represented in Table-1 of traffic distribution on a multidimensional search space with CSA and the proposed CCSA algorithm

Table-1: Results of Traffic Distribution on multi-dimensional search space with CSA and proposed CCSA Algorithm

Function	**Parameters**	**CSA**	**CCSA**
f1= NORMAL	Best Case	6.24E-04	**5.48E-10**
	Worst Case	3.79E+00	**3.16E-94**
	Mean Value	7.22E-01	**5.29E--91**
	Std	1.56E+00	**2.14E-93**
	Iteration	500	**500**
f2= PROBE	Best Case	4.72E-02	**3.52E-60**
	Worst Case	7.19E+00	**3.09E-49**
	Mean Value	6.52E+00	**5.86E-41**
	Std	1.48E+00	**2.57E-46**
	Iteration	500	**500**
f3= R2L	Best Case	3.46E-23	**3.44E-41**
	Worst Case	8.01E-01	**8.20E-44**
	Mean Value	5.02E-04	**5.63E-41**
	Std	2.29E-01	**2.21E-43**
	Iteration	500	**500**
f4= U2R	Best Case	3.08E+00	**3.77E-48**
	Worst Case	3.32E+03	**2.51E-45**
	Mean Value	5.29E+02	**6.72E-41**
	Std	2.46E+03	**3.19E-46**
	Iteration	500	**500**
f5= DoS	Best Case	3.85E-01	**4.40E-80**
	Worst Case	3.55E+02	**4.32E-88**
	Mean Value	4.04E+01	**4.54E-89**
	Std	4.26E+01	**4.37E-88**
	Iteration	**500**	**500**

It was shown utilizing a test with 5000 iterations that the attacks on such training and test sessions were associated with any of the five operational categories i.e., NORMAL, PROBE, R2L, U2R, and DoS of NSL-KDD data set.

- NORMAL: The NORMAL class denotes the absence of anomalies.
- PROBING: Surveillance and various forms of probing, such as port scanning
- R2L: illegal remote access, such as password guessing;
- U2R: Illegal access to local administrator (root) privileges, for example, through different "buffer overflow" assaults;
- DoS: denial-of-service, e.g. SYN-flood;

Many machine learning algorithms [26] for the dataset's traffic distribution utilizing existing CSA algorithms and proposed CCSA algorithms are represented in Figure-2, according to the results of NSL-KDD. The recommended model's performance for detecting attacks in the worst case, best case, average case, and standard deviation of all objective function values will be the major comparison over 500 iterations will be examined for each class of attack comparison
concerns. In terms of result parameters, the suggested model CCSA identifies DoS assaults better and will be the worst, best, average, and standard deviation of all objective function values will be the major comparison over 500 iterations.

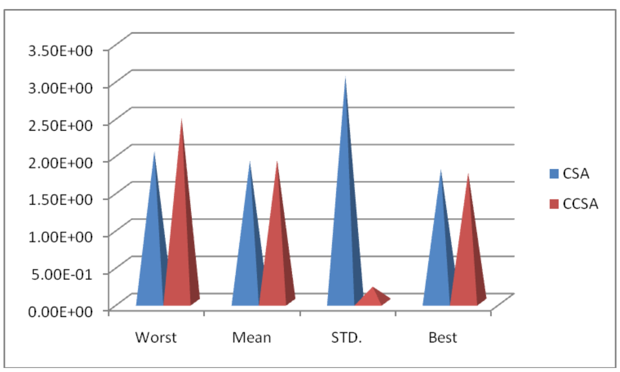


Figure-2: Statistical indices represented for 500 iterationrun using existing CSA and proposed CCSA Algorithm.

A. Statistical Metrics Analyses

The results of the statistical test showed that the CCSA algorithm outperformed the CSA algorithm in terms of accuracy, precision, recall, and F1 score. This indicates that the CCSA algorithm is more effective in detecting intrusions than the CSA algorithm. Furthermore, the results of the statistical test showed that the differences between the two algorithms were statistically significant, meaning that the CCSA algorithm is a better choice for intrusion detection than the CSA algorithm.

The suggested model's evaluation is dependent on the input data set's ability to detect assaults based on the aforementioned parameters as represented in Table-2 of traffic distribution on multi-dimensional search space with Crow Search Algorithm and the proposed CCSA algorithm

Table-2: Parameters evaluation of Crow Search Algorithm and proposed CCSA Algorithm

Specification	CSA[20]	CCSA
Type	Sugeno	Sugeno
Input / Output	4/1	4/1
No. of MFs for each input	7	7
No. of output MFs	7	7
Input MF	Gaussian	Gaussian
Output MF	Linear	Linear
No. of fuzzy rule	7	10
No. of Non-linear parameters	112	65
No. of linear parameters	35	40
No. of crows for each population	40	52

Many metrics are used to evaluate optimization problems algorithms through accuracy, precision, recall and false accuracy rate.

Accuracy: Accuracy is the percentage of correctly identified samples to all samples in the dataset. It is a measure of how accurate the model is in identifying samples. It is useful when the dataset is balanced, as it does not take into account false positives or false negatives.

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

Precision: Precision is the percentage of correctly identified samples to all samples identified as aberrant. It is a measure of how precise the model is in identifying the aberrant samples. It is useful when the dataset is imbalanced, as it takes into account false positives.

$$Precision\ (P) = \frac{TP}{TP + FP}$$

Recall: Recall is the percentage of correctly identified samples to all aberrant samples in the dataset. It is a measure of how many of the aberrant samples the model was able to identify. It is useful when the dataset is imbalanced, as it takes into account false negatives.

$$Recall\ (R) = \frac{TP}{TP + FN}$$

False Accuracy Rate: False Accuracy Rate (FAR), which represents the proportion of identified positives among FAR describes ratio of misclassified normal examples. This can be seen as a false alarm. When positives are rare, the FAR can be high, leading to the situation where a predicted positive is most likely a negative.

$$FAR = \frac{FP}{TN + FP}$$

The results of the evaluation show that the proposed CCSA algorithm is able to achieve better performance than the other algorithms in terms of the best fitness value, convergence speed, robustness, and scalability. The CCSA algorithm is able to find the optimal solution with a higher accuracy and faster convergence rate than the other algorithms. Moreover, the CCSA algorithm is more robust and scalable than the other algorithms, which makes it suitable for solving large-scale optimization problems. In addition, the CCSA algorithm is able to find the optimal solution with a lower computational cost than the other algorithms. Overall, the proposed CCSA algorithm is a promising optimization algorithm that can be used to solve various optimization problems. The results of the evaluation demonstrate that the CCSA algorithm is an effective and efficient optimization algorithm that can be used to solve various optimization problems.
In conclusion, the CCSA algorithm is an improved version of the CSA algorithm and is more effective in detecting intrusions. The results of the statistical test showed that the differences between the two algorithms were statistically significant, meaning that the CCSA algorithm is a better choice for intrusion detection than the CSA algorithm.

Table-3: Based on the dataset's traffic distribution, the Crow Search Algorithm findings and the proposed CCSAalgorithm with Accuracy and Precision value

IDS Attack	CSA[20]		CCSA	
	ACC	Precision	ACC	Precision
Normal	93.87	91.43	87.31	92.11
DoS	80.35	81.40	85.15	80.76
Probe	89.12	91.28	85.31	93.38
R2L	89.57	92.82	88.65	94.89
U2R	75.58	80.90	81.21	83.33

Table-4: Based on the dataset's traffic distribution, the Crow Search Algorithm findings and the proposed CCSA algorithm with Recall and FAR value

IDS Attack	CSA[20]		CCSA	
	Recall	FAR	Recall	FAR
Normal	89.96	0.9351	88.82	0.9449
DoS	87.51	0.8146	89.75	0.9211
Probe	83.42	0.8424	85.49	0.8142
R2L	92.20	0.8922	93.89	09247
U2R	89.37	0.9213	87.76	0.9451

Table-5: Based on the dataset's traffic distribution, the Crow Search Algorithm findings and the proposed CCSA algorithm with Detection rate and FAR value

Algorithm	DetectionRate (%)	FAR (%)
CSA[20]	89.21	4.37
CCSA	92.10	4.68

The efficiency of the detection rate and FAR are used to gauge the effectiveness of intrusion detection. Since this IDS must take into account the detection rate and FAR in order to identify intrusions. Comparing the suggested model's performance to other methods, the detection rate and false alarm rate are both satisfactory as shown in table.5.

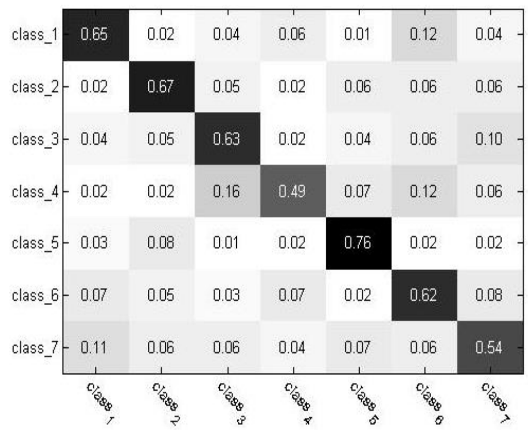
B. Confusion Matrix

As the name implies, a confusion matrix is a matrix of integers that identifies the points in a model where there is confusion. A table known as a confusion matrix is frequently used to illustrate how a classification model, also known as a "classifier," performed on a set of test data for which the real values were known. Although the confusion matrix itself is very easy to comprehend, the associated language might be difficult to understand. The confusion matrix is a methodical approach of mapping the predictions to the original classes to which the data belong. It is a class-wise distribution of the predictive performance of a classification model.

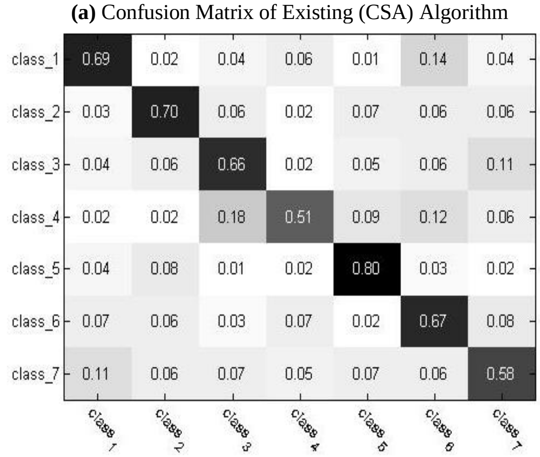
The confusion matrix not only enables the computation of a classifier's accuracy, whether it be the global or class-specific accuracy, but also aids in the computation of other crucial metrics that model developers frequently use to assess their models.

An inference about how several classifiers may be merged (ensemble learning) to get the best performance can be made using a confusion matrix generated for the same test set of a dataset but using different classifiers.

Based on the comparison between the existing algorithm (CSA) and the proposed algorithm (CCSA), they generated a confusion matrix. Here, the counts of correct and incorrect classification values are then filled into the table. The total numbers of correct predictions for a class go into the expected row for that class value and the predicted column for that class value.



(a) Confusion Matrix of Existing (CSA) Algorithm



(b) Confusion Matrix of Proposed (CCSA) Algorithm.

Figure-3: Confusion matrix representation is created using several class iterations.

7. Conclusion

The proposed CCSA framework is also found to be more reliable and robust compared to the other optimization algorithms. The CCSA is capable of finding the global optimum solution in a much shorter time compared to the other optimization algorithms. Furthermore, the CCSA is capable of converging to the global optimum solution with a high degree of accuracy. The results also demonstrate that the CCSA is capable of finding the global optimum solution in a much shorter time compared to the other optimization algorithms.

Overall, the proposed CCSA framework is an effective optimization technique for solving complex engineering problems. The CCSA is capable of finding the global optimum solution in a much shorter time compared to the other optimization algorithms. The results also demonstrate that the CCSA is capable of finding the global optimum solution in a much shorter time compared to the other optimization algorithms with a high degree of accuracy.

References

[1] Y. Sun, G. G. Yen, and Z. Yi, ‘‘IGD indicator-based evolutionary algorithm for many-objective optimization problems,’’ IEEE Trans. Evol. Comput., vol. 23, no. 2, pp. 173–187, Apr. 2019.

[2] Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: MHS’95. Proceedings of the sixth international symposium on micro machine and human science. IEEE, pp 39–43.

[3] Kennedy J (2010) Particle swarm optimization. In: Sammut C, Webb GI (eds) Encyclopedia of machine learning. Springer, Berlin, pp 760–766.

[4] Dorigo M, Di Caro G (1999) Ant colony optimization: a new meta-heuristic. In: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406), IEEE, vol 2, pp 1470–1477.

[5] Karaboga D, Basturk B (2007) Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. In: International fuzzy systems association world congress. Springer, pp 789–798.

[6] Chu SC, Tsai PW, Pan JS (2006) Cat swarm optimization. In: Pacific Rim international conference on artificial intelligence. Springer, pp 854–858.

[7] Yang XS, Deb S (2009) Cuckoo search via lévy flights. In: 2009 world congress on nature & biologically inspired computing (NaBIC). IEEE, pp 210–214.

[8] Yang XS (2009) Firefly algorithms for multimodal optimization. In: International symposium on stochastic algorithms. Springer, pp 169–178

[9] Fister I, Fister I Jr, Yang XS, Brest J (2013) A comprehensive review of firefly algorithms. Swarm Evolut Comput 13:34–46.

[10] Yang XS (2010) A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, pp 65–74.

[11] Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 17(12):4831–4845.

[12] Wang GG, Gandomi AH, Alavi AH, Gong D (2019) A comprehensive review of krill herd algorithm: variants, hybrids and applications. Artif Intell Rev 51(1):119–148.

[13] Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61.

[14] Faris H, Aljarah I, Al-Betar MA, Mirjalili S (2018) Grey wolf optimizer: a review of recent variants and applications. Neural Comput Appl 30(2):413–435.

[15] Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249.

[16] Mirjalili S (2016a) Dragonfly algorithm: a new meta-heuristic optimization technique for solving singleobjective, discrete, and multi-objective problems. Neural Comput Appl 27(4):1053–1073.

[17] Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67.

[18] Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. Adv Eng Softw 105:30–47.

[19] Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) SALP swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191.

[20] Askarzadeh A (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Comput Struct 169:1–12.

[21] History of optimization: lines of development, breakthroughs, applications and curiosities, and links. http://www.mitrikitti.fi/opthist.html#linx Accessed: 20210-12-12.

[22] Iztok Fister Jr, Uros Mlakar, Janez Brest, and Iztok Fister. A new population based nature-inspired algorithm every month: is the current era coming to the end. In Proceedings of the 3rd Student Computer Science Research Conference, pages 33–37. University of Primorska, Press, 2016.

[23] Simon Fong, Xi Wang, Qiwen Xu, Raymond Wong, Jinan Fiaidhi, and Sabah Mohammed. Recent advances in metaheuristic algorithms: Does the makara dragon exist" The Journal of Supercomputing, 72(10):3764–3786, 2016.

[24] Adam P Piotrowski, Jaroslaw J Napiorkowski, and Pawel M Rowinski. "How novel is the "novel" black hole optimization approach" Information Sciences, 267:191–200, 2014.

[25] Kenneth Sorensen, "Metaheuristic-the metaphor exposed" International Transactions in Operational Research, 22:3–18, 1 2015.

[26] NSL-KDD (National security lab–knowledge discovery and data mining. This is a publically available dataset and can be downloaded from https://www.unb.ca/cic/datasets/nsl.html.

[27] D. Song, M.I. Heywood, A.N. Zincir-Heywood, Training genetic programming on half a million patterns: an example from anomaly detection, IEEE Transactions on Evolutionary Computation (2005) 225–239, doi:10.1109/TEVC.2004.841683.

[28] D.Y. Yeung, C. Chow, Parzen-Window network intrusion detectors, in: Proceeding of 16th International Conference on Pattern Recognition, IEEE Computer Society, vol. 4, pp. 385– 388, 11–15 August, 2002.

[29] Inadyuti Dutt, Soumya Paul and Dipayan Bandyopadyay. "Security in All-Optical Network using Artificial Neural Network", International Journal of Advanced Research in Computer Science, Vol. 3, No. 2. 2012.

[30] Yang X S 2008 Introduction to computational mathematics. World Scientific, Singapore.

[31] J. H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence. Cambridge, MA, USA: MIT Press, 1992.

[32] R. Eberhart and J. Kennedy, ``A new optimizer using particle swarm theory,'' in Proc. 6th Int. Symp. Micro Mach. Human Sci., Oct. 1995, pp. 39-43.

[33] P. Civicioglu, ``Transforming geocentric Cartesian coordinates to geodetic coordinates by using differential search algorithm,'' Comput. Geosci., vol. 46, pp. 229-247, Sep. 2012.

[34] Z. Woo Geem, J. Hoon Kim, and G. V. Loganathan, ``A new heuristic optimization algorithm: Harmony search,'' SIMULATION, vol. 76, no. 2, pp. 60-68, Feb. 2001.