**An Enhanced Exploration and Exploitation of Modified Grey Wolf Optimizer for Fuzzy Rules Reduction in Cloud Intrusion Detection System (CIDS)**

Chidambaram Bagyalakshmi\(^1\), Erode Subramaniam Samundeeswari\(^2\), Varatharaj Arunkumar\(^3\)

\(^{1,2}\) Vellalar college for Women, Erode, 638012, India  
\(^3\) Kongu Engineering College, Erode, 638052, India

bagyachithra@gmail.com\(^1\), essamundeeswari@gmail.com\(^2\), arun15aug2002@gmail.com\(^3\)

doi: 10.17586/2226-1494-2021-21-6-912-918

**Abstract**

Optimization problems in the Cloud Intrusion Detection System (CIDS) contain numerous conflicting objectives, uni-modal and multi-modal functions and the difficulty level varying between linear to non-linear limits. The Grey Wolf Optimizer (GWO) is a meta-heuristic technique that is implemented based on social behavior of wolves and their hunting behavior. Significant improvement in exploration and exploitation of the search space in GWO can be obtained by modifying the control parameter \(a\). Works have already been carried out by many researchers by modifying the control parameter \(a\) in different manner to achieve better results. In a similar content, the authors of this study also modified the control parameter \(a\) and added a weight factor to the position of each wolf to attain the best possible results. Due to high demand for the cloud computing environment, intrusion detection in a cloud network plays a big role in maintaining the faith of the clients. Hence, CIDS is required to inspect the network packets to identify the abnormal behavior. For developing a system for cloud based IDS, the researchers created fuzzy rules to represent the relationship between the attributes and the nature of activity (normal or abnormal). The Modified Grey Wolf Optimizer (MGWO) algorithm is applied on eleven benchmark test functions and obtained good performance metrics. The results presented in this paper are promising; MGWO is used to reduce the fuzzy rules in developing fuzzy based CIDS. The performance of the proposed algorithm is compared with classical GWO, Particle Swarm Optimization (PSO), Cuckoo Search (CS) and variant of MGWO. The experimental results reveal that there is significant improvement in its performance.

**Keywords**

Grey Wolf Optimizer, CIDS, Modified Grey Wolf Optimizer, MGWO, Exploitation and Intrusion Detection System


**УДК 004.056.53**

Исследование модифицированного алгоритма оптимизации серых волков для редукции нечетких правил в облачной системе обнаружения вторжений

Чидамбарам Багьялакшми\(^1\), Эроде Субраманиам Самундесвари\(^2\), Варатхарадж Арункумар\(^3\)

\(^{1,2}\) Веллаларский женский колледж, Эроде, 638012, Индия  
\(^3\) Инженерный колледж Коргу, Эроде, 638052, Индия

bagyachithra@gmail.com\(^1\), essamundeeswari@gmail.com\(^2\), arun15aug2002@gmail.com\(^3\)

© Bagyalaksmi C., Samundeeswari E.S, Arunkumar V., 2021
до нелинейных. В данной работе авторы предложили, спроектировали и разработали CIDS на основе алгоритма оптимизации серых волков (Grey Wolf Optimizer, GWO) в качестве метаэвристического метода, который способен обнаруживать атаки в нескольких доступных средах. При разработке системы обнаружения вторжений на основе облака созданы нечеткие правила взаимосвязи между атрибутами и характером активности (нормальным и ненормальным). Предложенный алгоритм применен к описанию тестовых функций, на которых показаны хорошие показатели производительности. Модифицированный алгоритм оптимизации серых волков может быть использован для сокращения нечетких правил при разработке нечетких CIDS. Выполнено сравнение производительности предложенного модифицированного алгоритма с классическим оптимизатором серых волков, алгоритмом роя частиц (PSO), алгоритмом поиска кукушки (CS). Результаты экспериментов показывают высокую производительность предложенного алгоритма.

Ключевые слова:
оптимизатор серого волка, GWO, модифицированный оптимизатор серого волка, MGWO, исследование и обнаружения вторжений


Introduction

The implementation ways of the modern Cloud Intrusion Detection Systems (CIDS) discover a list of challenges for intrusion detection. A suitable approach is required to resolve them; therefore new optimization techniques are developed for this purpose. Some of the well-known optimization techniques such as Simulated Annealing, Genetic Algorithm, Ant Colony Optimization, Particle Swarm Optimization, Cuckoo Search, Firefly Algorithm, and Bee Algorithm are utilized in different fields of research. The advantages of these optimization approaches are simple, flexible and capability in keeping away from reaching nearby optimal value.

The Grey Wolf Optimizer (GWO) is one of the modern population-based optimization approaches [1]. The social behavior and pursuing behavior of grey wolves has been modeled mathematically with the aid of an algorithm. Wolves are classified as Alpha (α), Beta (β), Delta (δ) and Omega (ω) based on their roles. A wolf explores the prey and exploits it with the help of another wolf’s information. These exploration and exploitation are obtained through the control parameter a that improves the smooth transition between a wolf and its prey [2].

The emphasis is put on increasing the efficiency of the actual GWO by fine tuning of the control parameter a and on constructing the position-updated equation based on the weight factor to each wolf. This MGWO method is tested on benchmark functions and applied to optimize the fuzzy rules in CIDS. The fuzzy rules created in CIDS may be redundant; hence they should be reduced for designing an optimized fuzzy-based CIDS.

Brief Review of the Grey Wolf Optimizer Method

The GWO simulates the administrative behavior and searching manner of grey wolves, which are categorized into four important clusters such as α, β, δ, and ω wolves. Fig. 1 shows the hierarchy of the grey wolves.

In the top of hierarchy is α wolf, which leads the team as a head. The head of the wolf instructs other wolves to identify the sleeping place, hunting area, waking time and so on. The next hierarchy is occupied by β wolf, an ancillary wolf that helps with the decision-making of α wolves and other cluster activities. In the third level of hierarchy is δ wolf which offers the prey information to α and β wolves and also manages the ω wolves. The lower hierarchy is occupied by ω wolf which follows the commands of wolves in higher order α, β and δ that are considered as the best, second and third search agents and represent the best solutions. The ω wolves are the remaining candidate solutions.

Fig. 1. Hierarchy of grey wolves — from top to bottom

Fig. 2. Searching mechanism of grey wolves
Encircling Prey

Grey wolves (α, β and δ) hunt their prey by encircling them, which is considered to be sensible behavior. The mathematical model representing the principle of predation behavior is given below:

\[
D = |C \times X_p - X_i|, \quad (1)
\]

\[
X_{(t+1)} = X_p - \alpha \times D. \quad (2)
\]

Where \( t \) is current iteration, \( A \) and \( C \) are coefficient vectors, \( X_p \) is position of the prey and \( X \) is Position of grey wolf. The distance between grey wolves and the prey is represented by the notation \( D \) and is written mathematically in equation (1) and (2).

The GWO algorithm starts by means of producing a set of random selections such as the preliminary population. The preliminary population of each grey wolves’ positions are considered as random values, and their fitness values are calculated [3]. The fitness function also known as evaluation function, evaluates how close a given solution are considered as random values, and their fitness values are calculated [3]. The fitness function also known as evaluation function, evaluates how close a given solution are random vectors \( \alpha \), \( \beta \), \( \delta \), Coefficient Vectors — \( A \), \( C \), \( X \), \( D \), and is written as follows:

\[
A = 2 \times a \times r_1 - a \quad (3)
\]

\[
C = 2 \times r_2 \quad (4)
\]

\[
a = 2 \times (1 - \frac{t}{T}) \quad (5)
\]

Where \( a \) is linearly reduced from 2 to 0 and \( r_1 \) and \( r_2 \) are random vectors \( \in [0, 1] \). \( T \) is the maximum number of iterations and \( t \) is the current iteration.

Hunting

The searching mechanism is developed to simulate the grey wolves mathematically so that the wolves have a higher understanding of the knowledge about neighborhood prey. The best three wolves are sorted based on the fitness value. These three wolves are called best, second best, and third best. The selected wolves are called \( \alpha \) (best candidate solution), \( \beta \) and \( \delta \). Now \( \alpha \), \( \beta \) and \( \delta \) are turned into search agents in GWO and these different search agents are often get updated on their positions.

\[
A = 2 \times a \times r_1 - a \quad (3)
\]

\[
C = 2 \times r_2 \quad (4)
\]

\[
a = 2 \times (1 - \frac{t}{T}) \quad (5)
\]

Where \( a \) is linearly reduced from 2 to 0 and \( r_1 \) and \( r_2 \) are random vectors \( \in [0, 1] \). \( T \) is the maximum number of iterations and \( t \) is the current iteration.

The positions of the grey wolves are updated from equation (12). The role of each last wolf is calculated primarily based on the approximated role of the prey. The wolf \( \alpha \) acts as a final solution in GWO.

Background Studies

Exploration and exploitation are considered as key terms for modifying optimization algorithms. Algorithms are tuned by control parameters to achieve success in their performance. In classical GWO, exploration and exploitation are divided into two halves, but it could not balance between these two divisions. Detailed literature studies were conducted and significant remarks are tabulated in Table 1.

The motivation of the proposed MGWO is based on the literature which evaluates a significant modification over the control parameter \( a \). The control parameter \( a \) is modified to obtain smooth search space between exploration and exploitation.

Modified Grey Wolf Optimizer

This section elaborates the algorithm to explore the search process, namely the grey wolves search, based on the position of \( \alpha \), \( \beta \) and \( \delta \) wolves. The wolves move away from each other to search for the prey and come together to attack the prey. GWO algorithm allows search agents to update their position based on the location of \( \alpha \), \( \beta \), \( \delta \) after each iteration and attack towards the prey. Two basic parameters are required to be initialized, the first parameter is the maximum number of search agents or wolfs [1]. The second important parameter is the number of iterations fixed depending upon the type of application. The changeover between exploration and exploitation is made by the adaptive value of the control parameter \( a \). Too much exploration of search space may result in lower probability of getting trapped in local optima [4, 5]. At the same time higher exploration introduces more randomness and the optimal solution may not be obtained [6]. At the same time, excess exploitation is related to less randomness and the algorithm may not reach the global optima.

In the proposed MGWO algorithm, the greater number of iterations are allotted to the exploration phase and the smaller number of iterations are allotted to the exploitation stage.

\[
a = \sqrt{2 \times (1 - \frac{t}{T})}, \quad (13)
\]

The value of \( a \) decreased from 2 to 0 by the equation (13). There are a number of possibilities to improve the exploration rate. In this current work, various roots like square root (\( \sqrt{\cdot} \)), cube root (\( \sqrt[3]{\cdot} \)) and fourth root (\( \sqrt[4]{\cdot} \)) are applied on the decay function and found that square root (\( \sqrt{\cdot} \)) yields better results than other roots. Cubic root (\( \sqrt[3]{\cdot} \)) and fourth root (\( \sqrt[4]{\cdot} \)) provide an inadequate searching space between the prey and wolves.

In the original GWO algorithm, the position vector of a grey wolf is equally guided by the positions of \( \alpha \), \( \beta \) and \( \delta \) wolves as given by equation (12) [7]. As the most dominating member among the group is \( \alpha \) followed by \( \beta \).
and δ, in the proposed modified GWO, more weightage is given to α followed by β and δ to find the position vector of a grey wolf [8]. The weight factor is denoted by \( W \) that holds different factor for each wolf like \( W_1, W_2, \) and \( W_3 \).

\[
W = W_{n+1} + W_{n+1}.
\]

Two random numbers are set as initial values and each weight is the sum of the two preceding weights calculated from equation (14). These weights in each wolf positions could multiply from equation (15). Where \( n = 1, 2, 3 \) and \( t \) is the current iteration. Each wolf positions are updated in GWO search agents by adding weight factors, this helps to improve the next level of hunting.

**Algorithm 1: pseudocode of MGWO**

Initialize the search agent (grey wolf) population \( X_j \) (\( j = 1 \)
to \( n \) number of natural numbers)

Initialize Control Parameters \( a, A, \) and \( C \)

Calculate the fitness value — each search agent \( X_\alpha — \) best (or dominating) search agent \( X_\beta — \) second best search agent \( X_\delta — \) third best search agent

While \( (t < T) \)

For each search agent

Update the position of the current search agent by equation (15)

**Experimental Results for Benchmark Functions**

In this section, the proposed MGWO is implemented in R. MGWO is executed with following parameter settings as shown in Table 2. The input of data size may be either row wise or column wise data [9].

The performance of MGWO is applied on benchmark test functions, which is shown in Table 3 and statistical results are shown in Table 4. The proposed MGWO method is performed effectively on benchmark test functions [10], hence this method is implemented for fuzzy rules reduction in CIDS.

**Fuzzy Rules Reduction using MGWO**

An Intrusion Detection System (IDS) is a device or software application that monitors a network activity and reports to an administrator [11]. The utilization of cloud servers is increasing day by day. The increase in the workload of cloud servers leads to lack of network control,

<table>
<thead>
<tr>
<th>Author</th>
<th>Modified control parameter ( (a) )</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nitin Mittal et.al., (2016) [11]</td>
<td>( a = 2 \times (1 - \frac{T^2}{T^2}) )</td>
<td>Decay function for exponential terms is used iterations for ( a ). The iteration is divided into 70 % for exploration and 30 % for exploitation in the search space.</td>
</tr>
<tr>
<td>Sasmita Padhy et.al., (2017) [8]</td>
<td>( a = 2 \times (1 - \frac{T^3}{T^3}) )</td>
<td>The decay function is used to reduce ( a ) enhances the rate of exploration. The smooth course of the iteration process is divided into 75 % for exploration and 25 % for exploitation. Avoidance of unnecessary holding towards the local optima, which can cause randomness in exploitation, increases search space.</td>
</tr>
<tr>
<td>Fu Yan et.al., (2019) [6]</td>
<td>( a = 2 - 4 \times \exp\left(\frac{t}{T}\right) \times \cos\left(\frac{\pi}{2} \times \sqrt{\frac{T}{T}}\right) )</td>
<td>A continuous nonlinearity is observed in this trial. This nonlinearity gets increased along with the increasing iterations. An equal balance between exploration and exploitation is achieved in this algorithm.</td>
</tr>
<tr>
<td>Jia Cai et.al., (2020) [9]</td>
<td>( a_1 = 2 \times (1 - \frac{T}{t}) ) ( a_2 = a_1 \times (1 - \sin\left(\frac{\pi}{\mu} \times t\right)) ) ( \lambda = \frac{T-t}{T} ) ( a = \lambda \times a_1 + (1-\lambda) \times a_2 )</td>
<td>Global search space is used at the time of initializing the algorithm. In the final stage, a local search method is used. Mathematical modeling of the GWO method to reach the prey is obtained by reducing the value of ( a ). The rate of iteration used in GWO to explore and exploit is 74 % and 26 % respectively.</td>
</tr>
<tr>
<td>Ramin Ahmadi et.al., (2020) [4]</td>
<td>( a = 2 - 2 \times \frac{(\log T)^2}{T} )</td>
<td>The parameter ( a ) modified linearly which in turn makes certain changes over the exploration and exploitation phenomena. The ( k ) variable is maintained as a constant value between 0 and 1, change in the power of exploration and exploitation is done to reach the balance between them using ( k ) values. The iteration for exploration is 79 % and exploitation is 21 % in this experiment.</td>
</tr>
</tbody>
</table>
An Enhanced Exploration and Exploitation of Modified Grey Wolf Optimizer...

resulting in increased attacks on servers. Hence there is a need for cloud-based IDS to detect intrusions on servers.

Many benchmark datasets are available for cloud-based IDS problems and used by many researchers for their IDS based studies [12]. The NSL-KDD, one of the benchmark dataset, has 42 features and 2,26,283 records in four categories of attacks (DoS, R2L (Remote to user), U2R (user to root) and Probing). The records are characterized as normal and abnormal data, each abnormal record is classified into one of the four kinds of attacks.

In the previous paper [13], the CIDS model to Distributed Denial of Service (DDoS) attack using NSL-KDD was discussed. Feature selection methods (LVQ and PCA) and classification models (NB, DT and SVM) were used to classify the normal and abnormal records.

<table>
<thead>
<tr>
<th>Parameter settings</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>numVar</td>
<td>50</td>
</tr>
<tr>
<td>(the number of variables are determined with positive integer)</td>
<td>(it is called as data size)</td>
</tr>
<tr>
<td>numPopulation</td>
<td>20</td>
</tr>
<tr>
<td>maxIter</td>
<td>100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2. MGWO Parameter settings</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Table 3. Benchmark test functions</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Table 4. Statistical results for MGWO</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Table 5. The Performance of MGWO in fuzzy rules Reduction</th>
</tr>
</thead>
</table>

resulting in increased attacks on servers. Hence there is a need for cloud-based IDS to detect intrusions on servers.

Many benchmark datasets are available for cloud-based IDS problems and used by many researchers for their IDS based studies [12]. The NSL-KDD, one of the benchmark dataset, has 42 features and 2,26,283 records in four categories of attacks (DoS, R2L (Remote to user), U2R (user to root) and Probing). The records are characterized as normal and abnormal data, each abnormal record is classified into one of the four kinds of attacks. In the previous paper [13], the CIDS model to Distributed Denial of Service (DDoS) attack using NSL-KDD was discussed. Feature selection methods (LVQ and PCA) and classification models (NB, DT and SVM) were used to classify the normal and abnormal records.

resulting in increased attacks on servers. Hence there is a need for cloud-based IDS to detect intrusions on servers.

Many benchmark datasets are available for cloud-based IDS problems and used by many researchers for their IDS based studies [12]. The NSL-KDD, one of the benchmark dataset, has 42 features and 2,26,283 records in four categories of attacks (DoS, R2L (Remote to user), U2R (user to root) and Probing). The records are characterized as normal and abnormal data, each abnormal record is classified into one of the four kinds of attacks. In the previous paper [13], the CIDS model to Distributed Denial of Service (DDoS) attack using NSL-KDD was discussed. Feature selection methods (LVQ and PCA) and classification models (NB, DT and SVM) were used to classify the normal and abnormal records.

resulting in increased attacks on servers. Hence there is a need for cloud-based IDS to detect intrusions on servers.

Many benchmark datasets are available for cloud-based IDS problems and used by many researchers for their IDS based studies [12]. The NSL-KDD, one of the benchmark dataset, has 42 features and 2,26,283 records in four categories of attacks (DoS, R2L (Remote to user), U2R (user to root) and Probing). The records are characterized as normal and abnormal data, each abnormal record is classified into one of the four kinds of attacks. In the previous paper [13], the CIDS model to Distributed Denial of Service (DDoS) attack using NSL-KDD was discussed. Feature selection methods (LVQ and PCA) and classification models (NB, DT and SVM) were used to classify the normal and abnormal records.

resulting in increased attacks on servers. Hence there is a need for cloud-based IDS to detect intrusions on servers.

Many benchmark datasets are available for cloud-based IDS problems and used by many researchers for their IDS based studies [12]. The NSL-KDD, one of the benchmark dataset, has 42 features and 2,26,283 records in four categories of attacks (DoS, R2L (Remote to user), U2R (user to root) and Probing). The records are characterized as normal and abnormal data, each abnormal record is classified into one of the four kinds of attacks. In the previous paper [13], the CIDS model to Distributed Denial of Service (DDoS) attack using NSL-KDD was discussed. Feature selection methods (LVQ and PCA) and classification models (NB, DT and SVM) were used to classify the normal and abnormal records.
The fuzzy logic technique of Mamdani model is applied on the pre-processed NSL-KDD dataset to derive the relationship between effective fields involved in CIDS [14]. In this model, the authors defined 3 member variables (high, low and medium) for 19 fields and created 167 rules, however some of them may be redundant. The reduction of fuzzy rules using MGWO architecture is shown in Fig. 3. The MGWO algorithm is applied to optimize the fuzzy rules to generate an effective set of rules, and the smaller number of rules gives better understanding of relationship between features and attacks [15]. The comparative results of fuzzy rules reduction is shown in Table 5. The proposed algorithm reduces the number of rules effectively and the results are compared with GWO, PSO, CS and variants of MGWO algorithms.

**Conclusion**

The decay function used to modify the control parameter $a$ enhances exploration and modification of weights leads to better exploitation. This work effectively improves both exploration and exploitation of search spaces. The MGWO was tested using 11 benchmark test functions and compared with GWO, PSO and CS. The results of applying MGWO showed better performance compared to other algorithms. It is used for optimizing the fuzzy rules in CIDS.

---

**References**


---

**Литература**

An Enhanced Exploration and Exploitation of Modified Grey Wolf Optimizer...


Authors
Chidambaram Bagyalakshmi — M.Phil., Research Scholar, Vellalar college for Women, Erode, 638012, India, 57219419148, https://orcid.org/0000-0001-6630-3678, bagyachithra@gmail.com
Erode Subramaniam Samundeeswari — PhD, Associate Professor, Vellalar college for Women, Erode, 638012, India, 56153581400, https://orcid.org/0000-0002-1783-2634, essamundeeswari@gmail.com
Varatharaj Arunkumar — PhD, Assistant Professor, Kongu Engineering College, Erode, 638052, India, 15922361400, https://orcid.org/0000-0002-1399-9437, arun15aug2002@gmail.com

Авторы
Багьялакшми Чидамбарам — магистр, исследователь, Веллаларский женский колледж, Эроде, 638012, Индия, 57219419148, https://orcid.org/0000-0001-6630-3678, bagyachithra@gmail.com
Самундесвари Эроде Субраманиам — PhD, доцент, Веллаларский женский колледж, Эроде, 638012, Индия, 56153581400, https://orcid.org/0000-0002-1783-2634, essamundeeswari@gmail.com
Арункумар Варатхараатх — PhD, доцент, Инженерный колледж Коргу, Эроде, 638052, Индия, 15922361400, https://orcid.org/0000-0002-1399-9437, arun15aug2002@gmail.com

Статья поступила в редакцию 07.09.2021
Одобрена после рецензирования 25.10.2021
Принята к печати 30.11.2021

Работа доступна по лицензии «Attribution-NonCommercial»