IMPROVED ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM BASED ON MODIFIED SALP SWARM ALGORITHM AND GOLDEN EAGLE OPTIMIZER ALGORITHM FOR INTRUSION DETECTION IN NETWORKS

Alaa Majeed Shnain Al mrashde
Al-Furat Al-Awsat Technical University / Babil Technical Institute
alaa.shnen.iba@atu.edu.iq

Reception: 04/05/2023 **Acceptance**: 30/06/2023 **Publication**: 21/07/2023

Suggested citation:

Shnain Al mrashde, A.M. (2023). **Improved adaptive neuro-fuzzy inference system based on modified salp swarm algorithm and golden eagle optimizer algorithm for intrusion detection in networks**. *3C Tecnología. Glosas de innovación aplicada a la pyme, 12(2)*, 364-386. https://doi.org/10.17993/3ctecno.2023.v12n3e44.364-386

ABSTRACT

With the increase in the growth of computer networks throughout the past years, network security has become an essential issue. Among the numerous network security measures, intrusion detection systems play a dynamic function with integrity, confidentiality, and accessibility of resources. An Intrusion Detection System (IDS) is a software program or hardware device which monitors computer system and/or network activities for malicious activities and produces alerts to security experts. In IDS there are three major problems namely generating many alerts, a huge rate of false positive alerts, and unknown attack types per generated alerts. Alert management methods are used to manage these problems. One of the methods of alert management is alert reduction and alert classification. The proposed approach focuses on enhancing the efficiency of the adaptive neuro-fuzzy inference system (ANFIS) using a modified salp swarm algorithm (SSA) and Golden Eagle optimizer (GEOSSA). The present study uses the Golden Eagle optimization algorithm to improve SSA behaviors. The proposed model (GEO-SSA-ANFIS) intends to determine the appropriate parameters using the GEO-SSA algorithm because these parameters are considered the main component affecting the ANFIS forecasting process. The results of the intrusion detection based on the NSL-KDD dataset were better and more efficient compared with those models because the detection rate was 96.68% and the FAR result was 0.438%.

KEYWORDS

Intrusion Detection System (IDS), adaptive neuro-fuzzy inference system (ANFIS), modified salp swarm algorithm (SSA), Golden eagle optimizer (GEO), Intrusion Detection in Networks.

INDEX

https://doi.org/10.17993/3ctecno.2023.v12n3e44.364-386

https://doi.org/10.17993/3ctecno.2023.v12n3e44.364-386

1. INTRODUCTION

With the recent interest and progress in the development of Internet and communication technologies over the last decade, network security has emerged as a vital research domain. It employs tools like firewalls, antivirus software, and intrusion detection system (IDS) to ensure the security of the network and all its associated assets within cyberspace. Among these, a network-based intrusion detection system (NIDS) is the attack detection mechanism that provides the desired security by constantly monitoring the network traffic for malicious and suspicious behavior [1].

IDS solutions are one of the key security components that in combination with firewalls can effectively handle various types of security attacks. IDS schemes can be mainly classified as misuse detection schemes and anomaly detection schemes, which can be realized by using various machine learning techniques. Misuse detection or signature-based systems heavily depend on the signature of the security attacks and malicious behaviors and support multi-class classification. However, they cannot detect new attacks in which their signature is not available for the IDS. However, as an advantage, these schemes benefit from more accuracy in recognizing known malicious behaviors and their variants [2].

A possible protection mechanism such as intrusion detection is indispensable as it involves preventive action used to get rid of any malignant acts within the computer network. one merit is that they can locate previously unknown attacks, however, they retain to have a high false positive rate (FPR). Quite the contrary, the latter performs attack detection based on some known attack signatures. Utilizing a pattern-matching algorithm, an attack pattern candidate in the network is checked by comparing it with those predetermined signatures. This results in a lower FPR, but fails to detect novel attack patterns [3].

1.1. PROBLEM DEFINITION

Network security plays a vital role in avoiding financial loss, protecting customers from monetary damages, avoiding disabling or crippling services, and limiting severe information loss due to network intrusions. Attackers generally exploit the configurations and vulnerabilities of popular software to mount attacks against network computer systems. The damage caused by these attacks may vary from a little disruption in services to high financial losses. Existing conventional security techniques like firewalls are only used as the first line of defense [4].

Intrusion detection systems (IDS) perform the crucial task of detecting such attacks on computers and networks and alerting the system operators. An IDS may be placed in individual hosts in a network, in a dedicated central location, or distributed across a network. IDS's that are designed to detect attacks on a network of computers rather than a single host are called network intrusion detection systems (NIDS). These systems monitor network functionality in the form of network telemetry, which may include network traffic, metadata of network flows (eg: protocols such as NetFlow), and activity logs from hosts, and attempt to detect attack occurrences [5].

Because of the large volume of data, the network gets expanded with a false alarm rate of intrusion, and detection accuracy decreased. This is one of the significant issues when the network experiences unknown attacks. The principle objective was to expand the accuracy and reduce the false alarm rate (FAR) [6]. This study presents a novel forecasting model for detecting the abnormalities which have the largest effect on networks. The proposed method depends on improving the performance of the adaptive neuro-fuzzy inference system (ANFIS) using a modified salp swarm algorithm (SSA). The SSA simulates the behaviors of a salp swarm in nature during searching for food, and it has been developed as a global optimization method.

1.2. EXISTING CHALLENGES

With the large volume of data, the network gets expanded with false alarm rate of intrusion and detection accuracy decreased. This is one of the significant issues when the network experiences unknown attacks.

1.3. RESEARCH OBJECTIVES

- Improve the intrusion detection rate.
- Improve the security performance of the network.

1.4. THESIS STRUCTURE

The thesis organization is as follows: In the second chapter, we first explain the basic concepts. In the third chapter, it provides an overview of previous works in the field of intrusion detection in networks. In the fourth chapter, we describe the proposed solution. In the fifth chapter, we target the results and evaluation of the proposed algorithm, and at the end, conclusions and suggestions for future works are presented.

2. INTRUSION DETECTION SYSTEMS

Intrusion Detection System was initiated by Anderson in 1980. Various IDS products were created in 4 decades of IDS development. During its development, various kinds of problems arose, for instance, the high rate of false alarms which is sending alerts when there was no dangerous traffic. This will increase the network security analyst workload. If the network security analyst keeps getting false alarm alerts, then the actual attack may be infiltrated into one of these false alarms. Therefore, many studies on IDS focus on reducing the number of false alarms and increasing the ability to detect malicious traffic. On the other hand, traditional IDS is incapable of detecting unrecognized attacks. Changes in conditions and the network environment are very fast and the emergence of various new technologies on the network also raises various types of new attacks. Therefore, it is urgent to develop an IDS that can detect attacks that are not even recognized (unknown attacks) [7].

The components of IDS are as follows [8]:

(i) Monitoring Network: A network needs to be monitored to gather necessary packets containing network-related information.

(ii) Data Collection: It refers to gathering the details about the target system on which the attack is to be conducted. This can be achieved by performing queries using network commands or tools. For instance, packet-level details can be obtained by sniffing the packets flowing through the network using “Wireshark” or obtaining server and host-related details such as domain name using network commands like “nslookup”.

(iii) Analysis of Packet Details: This can be referred to as scanning the network packet for stealing confidential information.

(iv) Identifying and Storing the Signature/Attack Patterns: The next step after the analysis of packet details is to identify the attack patterns of already known attacks and novel attacks or signatures of some known exploits which can be used to launch insider attacks. These signatures and patterns are stored in the database for future reference; hence, the security administrator can easily report intrusive behavior, if found anomalous.

(v) Generating Alert: After recognizing the attack pattern, an alert/alarm is generated and reported to the security administrator. The alert is triggered based on the matching of the signature/pattern.

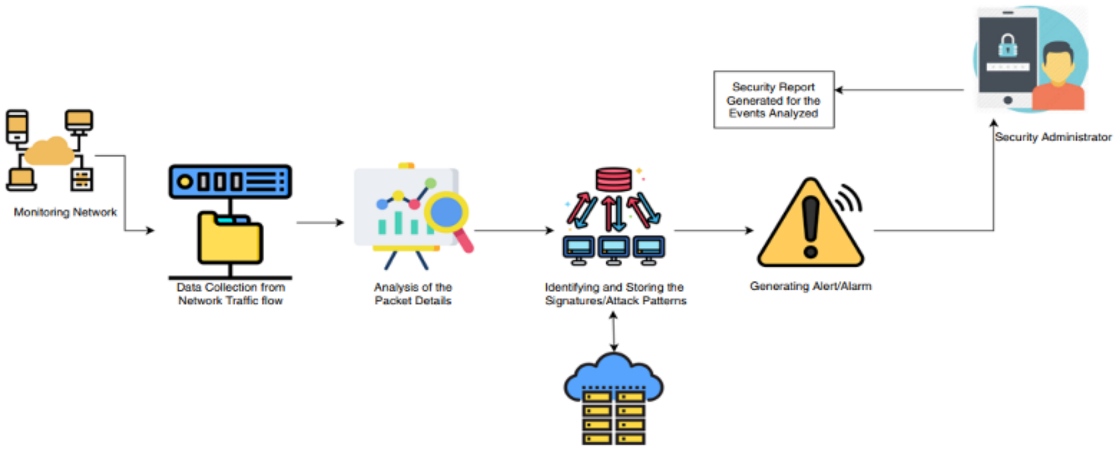

Security Report Generated for the Events Analyzed
Security Administrator
Monitoring Network
Data Collection from Network Traffic flow
Analysis of the Packet Details
Identifying and Storing the Signatures/Attack Patterns
Generating Alert/Alarm

Figure 1. Components of IDS [8]

2.1. NETWORK IDS (NIDS)

Technological development resulted from dependence on global networks when using several businesses, educational and social activities. As a result of the increasing use of computer networks, several issues occurred in Internet security. Hence, keeping the security of devices connected to the Internet is important to ensure system availability and integrity.

2.2. HOST IDS (HIDS)

The main aim of HIDS is to control the behavior and dynamic state of the computer system. The flow of packets and all the activities on a network has been scrutinized by HIDS. The system administrators receive some network alerts if any alternation or adjustment happens in the network. HIDS is gradually becoming crucial in securing a host computer framework and its network. HIDS is incorporated into the computer system to identify the intruder's abnormal behavior. It also protects the information from intruders, and the incidents are reported to the system administrator. If an attack happens on any other part of the network, then host-based IDS will not only detect an attack but will also monitor incoming and outgoing traffic. The file system located on the host performs audits of the users' login, currently, active processes, resource utilization, and much more can also be analyzed by a host-based IDS. Following are some of the advantages of HIDS [10]:

- All users' activities can be monitored in HIDS, whereas it is not conceivable in a network-based system.
- An attack that has originated from the host side can be identified by HIDS.
- The decrypted traffic to find a host-based system can analyze an attack signature. Thus, they also have the capability of monitoring encrypted traffic.
- No extra hardware is required as they can be easily installed on the existing host devices.
- For a small-scale network, Host-based IDS is cost-effective.

2.3. INTRUSION DETECTION SYSTEM TAXONOMY

IDSs are categorized into three groups, i.e., anomaly-based detection, signature-based detection, and specification-based detection [11]. To summarize the taxonomy, we show a conceptual diagram in Figure 2.

https://doi.org/10.17993/3ctecno.2023.v12n3e44.364-386

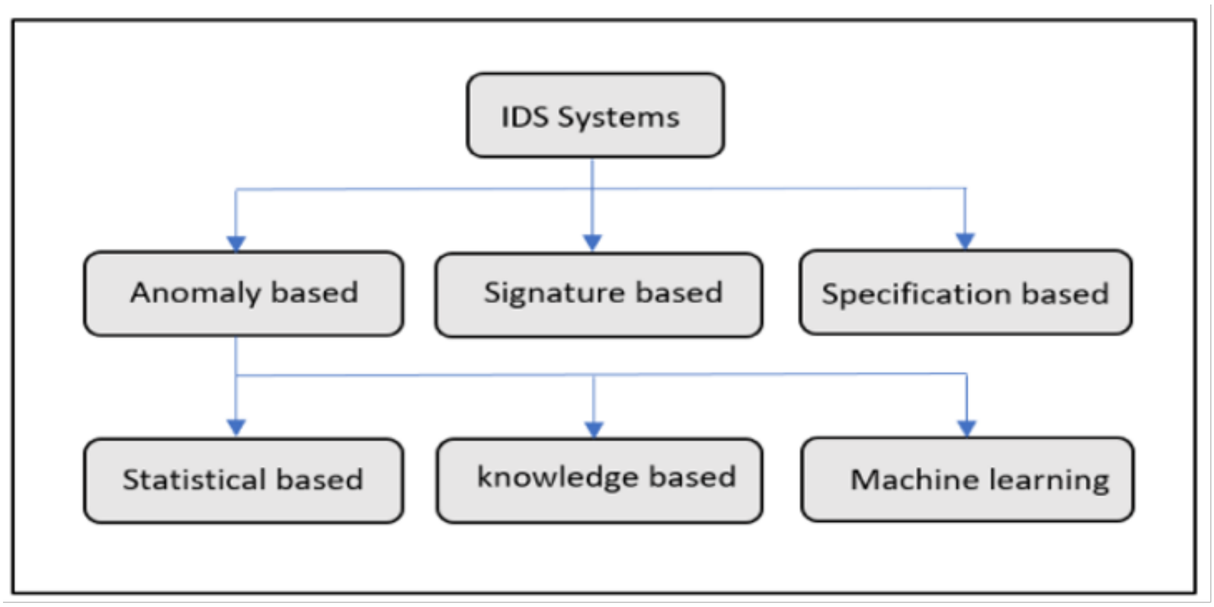


Figure 2. Taxonomy of intrusion detection system (IDS) systems [11].

2.3.1. ANOMALY-BASED TECHNIQUE

Anomaly detection refers to the deviation of network traffic from its normal profile. The normal profile is captured in the network's non-attack conditions and is represented mostly by statistical data. An example of such deviation is when a manager, who normally accesses the network in the daytime, uses his account to access it at night, which is regarded as a deviation of activity. Such a deviation is suspicious, and it can indicate an attack; however, such activity might not be associated with an attack. Thus, it is possible to have a false alarm based on this. Hence, continuous updates of the network's user activity patterns to avoid false alarms can be performed. In this study, we are interested in an anomaly-based IDS. Therefore, we create the following sub-taxonomy of such systems:

Statistical-based anomaly IDS: The statistical-based anomaly IDS matches the periodically captured statistical features from the traffic with a generated stochastic model of the normal operation or traffic. The attack is reported as the deviation between the two statistical patterns, i.e., the normal memorized one and the current captured one.

Knowledge-based anomaly IDS: In knowledge-based anomaly detection, numerous rules are provided by experts in the form of an expert system or fuzzy-based system to define the behavior of normal connections and attacks. In fuzzy-based anomaly detection, the rule-based is connected to inputs. A subset of the rules is enabled based on the input values, sometimes heuristics or a UML-based description of the attack's behavior is provided.

Machine learning-based anomaly IDS: An explicit or implicit model of the analyzed patterns is developed in a machine learning-based anomaly IDS. These

https://doi.org/10.17993/3ctecno.2023.v12n3e44.364-386

models are revised regularly to boost intrusion detection efficiency based on past results.

2.3.2. SIGNATURE-BASED TECHNIQUE

A signature-based technique is also referred to as knowledge-based or misuse detection. It uses the signature of the attack and performs matching between the current traffic and the signature, and then reports an attack on the existence of matching, otherwise, it does not report an attack. Such an approach does not suffer from a high rate of false alarms like other approaches. However, it requires a continuous update of the signature [11].

2.3.3. SPECIFICATION-BASED TECHNIQUE

A specification-based technique uses the specification or constraints to describe a certain program’s operation and reports any violation of such specification or constraints based on matching with the prior determined and memorized specification and constraints [11].

2.4. IDS DEPLOYMENT STRATEGIES

IDS can also be classified based on the deployment used to detect IoT attacks. In IDS Deployment strategies, IDS can be classified as distributed, centralized, or hybrid [12].

2.4.1. DISTRIBUTED IDS

In distributed placement, the IoT devices could be responsible for checking other IoT devices. Distributed IDS be made up of several IDS over a big IoT ecosystem, all of which communicate with each other, or with a central server that assists advanced intrusion detection systems, packet analysis, and incident response. Several IDS deploy distributed architectures. This includes a subset of the network checking the other nodes. Distributed IDS offers the incident analyst many advantages over centralized IDS. The main benefit is the capability to identify attack forms across a whole IoT ecosystem. This might increase prompt IoT attack prevention and detection. The additional supported benefit is to allow early detection of an IoT Botnet creating its way through corporate IoT devices. This data could then be used to detect and clean systems that have been infected by the IoT Botnet and stop further spread of the Botnet into the IoT ecosystem consequently taking down any IoT devices damaged that would otherwise have occurred. Furthermore, the advantage of distributed IDS rather than centralized IDS computing resources also implies reduced control over those resources.

2.4.2. CENTRALIZED IDS

In the centralized IDS location, the IDS is placed in central devices, for instance, in the boundary switch or a nominated device. All the information that the IoT devices collect and then send to the network boundary switch passes through the boundary switch. Consequently, the IDS positioned in a boundary switch can check the packets switched between the IoT devices and the network. Despite this, checking the network packets that pass through the boundary switch is not adequate to identify anomalies that affect the IoT devices. The network traffic is monitored in centralized IDS. This traffic is extracted from the network through different network data sources such as packet capture, NetFlow, etc. The computers connected in a network can be monitored by Network-based IDS. Moreover, NIDS is also capable of monitoring the external malicious activities that could have been commenced from an external threat at an earlier stage, before these threats expand to other computer systems. However, NIDS comes with some limitations such as its restricted ability to inspect the whole data in a high bandwidth network because of the volume of data passing through modern high-speed communication networks. NIDS deployed at several positions within a particular network topology, together with HIDS and firewalls, can provide concrete, resilient, and multi-tier protection against both external and insider attacks. Data source consists of system calls, application program interfaces, log files, and data packets that are extracted from well-known attacks. These data sources can be useful to classify intrusion behaviors from abnormal actions [12].

2.4.3. HIERARCHICAL IDS

In Hierarchical IDS, the network is separated into clusters. The sensor nodes that are adjacent to each other typically belong to the same cluster. Each cluster is assigned a leader, the so-called cluster head that screens the member nodes and plays a part in network-wide analyses.

2.5. RECENT IMPROVED SOLUTIONS TO INTRUSION DETECTION

With the advancement of recent technologies like Cloud services, and IOT devices, the Intrusion Detection System (IDS) has become a prominent technology to detect anomalies and attacks in the network. Many researchers have integrated IDS with data mining, fuzzy logic, neural networks, machine learning, and optimization techniques to improve the methods of detecting anomalies and attacks to improve the accuracy of detections. Information security is the main concern with advanced technologies like IoT and Cloud computing. With the increased usage of IoT networks and clouds in different domains, these have become more vulnerable targets for intruders and attackers. Many researchers have proposed different methods and approaches to detect malicious actions of intruders and found the need for security in the cloud and IOT to be implemented on the layers as well as protocol levels in the service models. Many IDS based on statistical methods, knowledge-based methods, and machine learning techniques have been studied and presented in Table 25.1 according to the algorithms used and the results produced with their detection performance. The existing AIDS techniques can be categorized as the following [13]:

1. Based on Statistical methods
2. Based on machine learning methods and Data mining techniques
3. Knowledge-based
4. Evolutionary methods
5. Based on Statistical methods

IDS based on the statistical methods uses a distribution model for normal behavior profiles to detect potential intrusions based on statistical metrics such as mean, median, standard deviations, and mode of packets. Statistical IDS generally uses one of the following models; Univariate, multivariate, and time series models. In univariate technique-based IDS statistical normal profile is created based on only one measure of behaviors in computer systems for identifying abnormalities in each metric. While the Multivariate technique considers more than one measure to specify the relationships between variables including multiple data variables which can be correlated. The multivariate statistical IDs face challenges to estimate correlations and distributions for high-dimensional data. Any time series data can be defined as a series of observations made over a certain time interval. A new observation can be considered abnormal if it is not occurring at that time interval. Researchers used any occurred abrupt variation in time series data for detecting network abnormalities.

2.5.1. BASED ON DATA MINING AND MACHINE LEARNING

Methods Machine learning process is used to infer knowledge from huge amounts of data by applying a set of rules, methods, or complex "transfer functions" to find out unusual data patterns and predict abnormal behavior of user profiles. Several researchers have applied machine learning techniques in the area of AIDS such as clustering, neural networks, association rules, decision trees, genetic algorithms, and nearest neighbor methods to improve accuracy and reduce the requirement of human interventions. Machine learning algorithms can be classified into supervised, unsupervised, and semi-supervised techniques which are extensively being used in building AIDS and finding patterns.

2.5.2. KNOWLEDGE-BASED

https://doi.org/10.17993/3ctecno.2023.v12n3e44.364-386

An expert system-based approach requires creating a knowledge base consisting of genuine traffic profiles with the help of human knowledge and detecting any action different from this defined profile as an intrusion. This technique helps in reducing false positive alarms but requires the regular update of the knowledge base regarding the normal expected behavior of traffic profiles. This group of techniques includes a Finite state machine, Description Language, Expert System, and Signature analysis-based IDS.

2.5.3. EVOLUTIONARY METHODS AND OPTIMIZATION TECHNIQUES

This group of techniques makes use of nature-inspired algorithms such as ACO, PSO, Genetic Algorithms, Evolutionary computing, etc. to improve accuracy. These evolutionary approaches based on the principle of evolution and the concept of fitness methods are used for classification and feature selection in intrusion detection systems.

3. THE REVIEW OF PREVIOUS WORKS

In this section, we study the works done in the field of intrusion detection systems based on Evolutionary Methods and Optimization Techniques.

4. THE PROPOSED METHOD

In this work, we proposed a new model for detecting the abnormalities present in the network or system using a modified salp swarm algorithm (SSA) to improve the performance of the adaptive neuro-fuzzy inference system (ANFIS). However, SSA still has some limitations such as it is easy to suck at a local point; therefore, we use the Golden Eagle optimizer (GEO) algorithm to improve the behavior of SSA. In addition, we argue that the proposed model can be successfully applied to detecting the abnormalities present in the network or system. Also, we can confirm that it can be used for future predictions. The description of the proposed method is presented in this section (see Fig. 4).

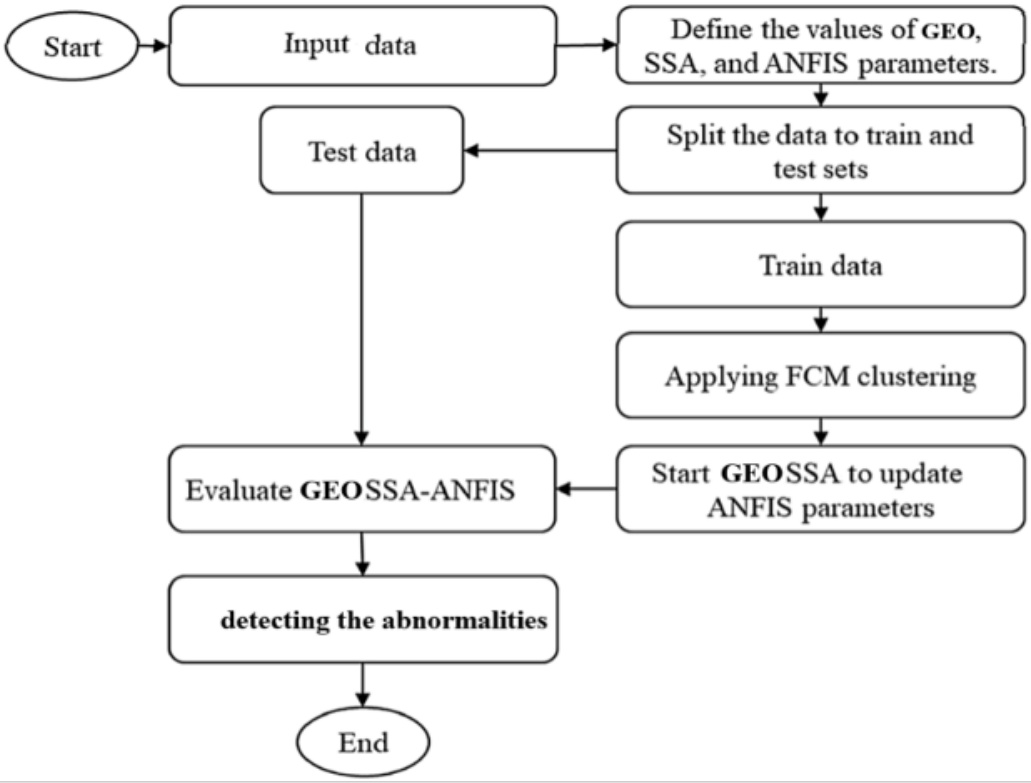

Start
Input data
Define the values of GEO, SSA, and ANFIS parameters.
Split the data to train and test sets
Test data
Train data
Applying FCM clustering
Start GEOSSA to update ANFIS parameters
Evaluate GEOSSA-ANFIS
detecting the abnormalities
End

Figure 3. Proposed GEO-SSA-ANFIS.

4.1. ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM-ANFIS

ANFIS was a fusion of a fuzzy inference system (FIS) and ANN which has the benefits of both ANN and FIS. In ANFIS architecture, ANN extricates fuzzy rules from input information and the fuzzy membership function's parameters were adaptably utilized during the process of hybrid learning. ANFIS could create a relation between input and output dependent on human knowledge by utilizing data pairs of input-output and applying an algorithm based on hybrid learning. ANFIS was a sort of multilayer feedforward network made from five layers. The layers in the ANFIS architecture include many nodes defined by the function of the node.

The proposed method improves the ANFIS model using GEO and SSA algorithms which are called GEO -SSA-ANFIS. It applies GEO and SSA to adjust the parameters of the ANFIS by feeding the best weights between Layer 4 and Layer 5.

The rule basis of the ANFIS is:

If $v(x_1)$ is A_j, $v(x_2)$ is B_j, hence C_j is

$$S_j = a_j v\left(x_1\right) + b_j v\left(x_2\right) + c_j v\left(x_n\right) + g_j \quad (1)$$

https://doi.org/10.17993/3ctecno.2023.v12n3e44.364-386

Where $v(x_1), v(x_2)\ldots v(x_n)$ are the inputs, A_j, B_j, and C_j are the fuzzy sets, S_j is the output inside the fuzzy zone defined by the fuzzy rule. a_j, b_j, c_j, and g_j are the parameters that are specified by the training process.

Layer 1- In this la1yer, every j node was a square node with a function of the node.

$$L_{1,j} = \mu_{A_1} v\left(x_1\right), L_{1,j} = \mu_{B_1} v\left(x_2\right), L_{1,j} = \mu_{C_1} v\left(x_n\right) \tag{2}$$

Generally $\mu_{A_1} v(x_1)$, $\mu_{B_1} v(x_2)$, $\mu_{C_1} v(x_n)$are selected to be bell-shaped with max=1 and min=0 and are specified as

$$\mu_{A_1} v\left(x_1\right) = \mu_{B_1} v\left(x_2\right) = \mu_{C_1} v\left(x_n\right) = \left(\frac{1}{1 + \left(\frac{x - u_j}{v_j}\right)^{2y_j}} \right) \tag{3}$$

Where u_j, v_j, y_j are the set of parameters. In this layer, those parameters were described as basis parameters. Layer 2- In this layer, every node was a circle node labeled that multiplies the product out and the input signals. Example,

$$L_{2,j} = wp_j = \mu_{A_j} v\left(x_1\right) \times \mu_{B_j} v\left(x_2\right) \times \mu_{C_j} v\left(x_n\right), j = 1{,}2 \tag{4}$$

Every output node demonstrates the rule's firing strength. Layer 3- Each node was a circle node called N. The jth node computes the proportion of the jth rule firing strength to the addition of all rule's firing strength as,

$$L_{3,j} = wp_j = \frac{wp_j}{\left(wp_1 + wp_2\right)}, j = 1{,}2 \tag{5}$$

Layer 4- In this layer each node j was a square node with a function of node.

$$L_{4,j} = wp_j S_j, j = 1{,}2 \tag{6}$$

Where wp_j were the layer-3 output and a_j, b_j, c_j, and g_j are the parameters set? These layer parameters were described as consequence parameters.

Layer 5- A single node was a circle node labeled in this layer which calculates the total output as the sum of all input signals.

$$L_{5,j} = \frac{\sum_j wp_j s_j}{\sum_j wp_j} \tag{7}$$

Hence the predefined threshold value φ and the output of the neural network Z are compared which is expressed in the below equation,

https://doi.org/10.17993/3ctecno.2023.v12n3e44.364-386

$$\text{output} = \begin{cases} \text{error,} & Z < \varphi \\ \text{no error ,} & Z \geq \varphi \end{cases} \tag{8}$$

To enhance the prescient accuracy of ANFIS and abstain from falling into the local optimum, parameter learning is performed by the modified salp swarm algorithm (SSA) and Golden Eagle optimizer (GEOSSA).

The GEO-SSA-ANFIS starts by preparing the inputs and dividing the problem into training and testing sets. Then the fuzzy c-mean (FCM) algorithm is used to determine a suitable number of the membership functions by clustering the dataset into different groups. Thereafter, the ANFIS uses these results to start the rest of the steps. The ANFISs parameters, namely the weights, are adapted using the GEO-SSA algorithm, where the GEO-SSA searches for the solution in the problem space by exploring various domains. In the first step in the proposed method, the GEO is used to generate the initial population of the SSA. Then the SSA uses this population to start searching for the best weights of the ANFIS. The fitness value of the population is calculated by the following fitness function.

$$Objective\ function = \|obs - pred\|^2 \tag{9}$$

Therefore, the selected weights are updated based on minimizing the error between the output and the real values in the training phase. These weights are passed to the ANFIS to prepare the output results of a given problem. The GEO-SSA works till meeting the stop condition in this paper which is the max number of iterations. After that, the test phase starts, and the best weights are passed to the ANFIS to produce the output.

4.2. SALP SWARM ALGORITHM (SSA)

Salp swarm algorithm (SSA) [24] is one of the recent optimization methods developed by Mirjalili et al. (2017). They tried to mathematically simulate the salp chains' behavior of the real salps. Salps are considered a kind of the Salpidaes family. They look like jellyfish in their moving behavior and their bodies contain a high water percentage. Salps use their swarm behavior (i.e., salp chain) in foraging and moving with fast harmonious changes. Food sources are the target of the swarm. The mathematical model of SSA begins by generating a population and then dividing it into leaders and followers based on their position. The leader is the front salp of the chain, and the followers are the rest of the salps. This study applies GEO and SSA to adjust the parameters of the ANFIS.

The position is formed in n-dimensions that denote the search space of a given problem, whereas the problem variables are represented by n. The position is frequently updated. The equation below is applied to update the position of the salp leader:

$$x_j^1 = \begin{cases} F_j + c_1\left(\left(ub_j - lb_j\right) \times c_2 + lb_j\right) & c_3 \leq 0 \\ F_j - c_1\left(\left(ub_j - lb_j\right) \times c_2 + lb_j\right) & c_3 > 0 \end{cases} \tag{10}$$

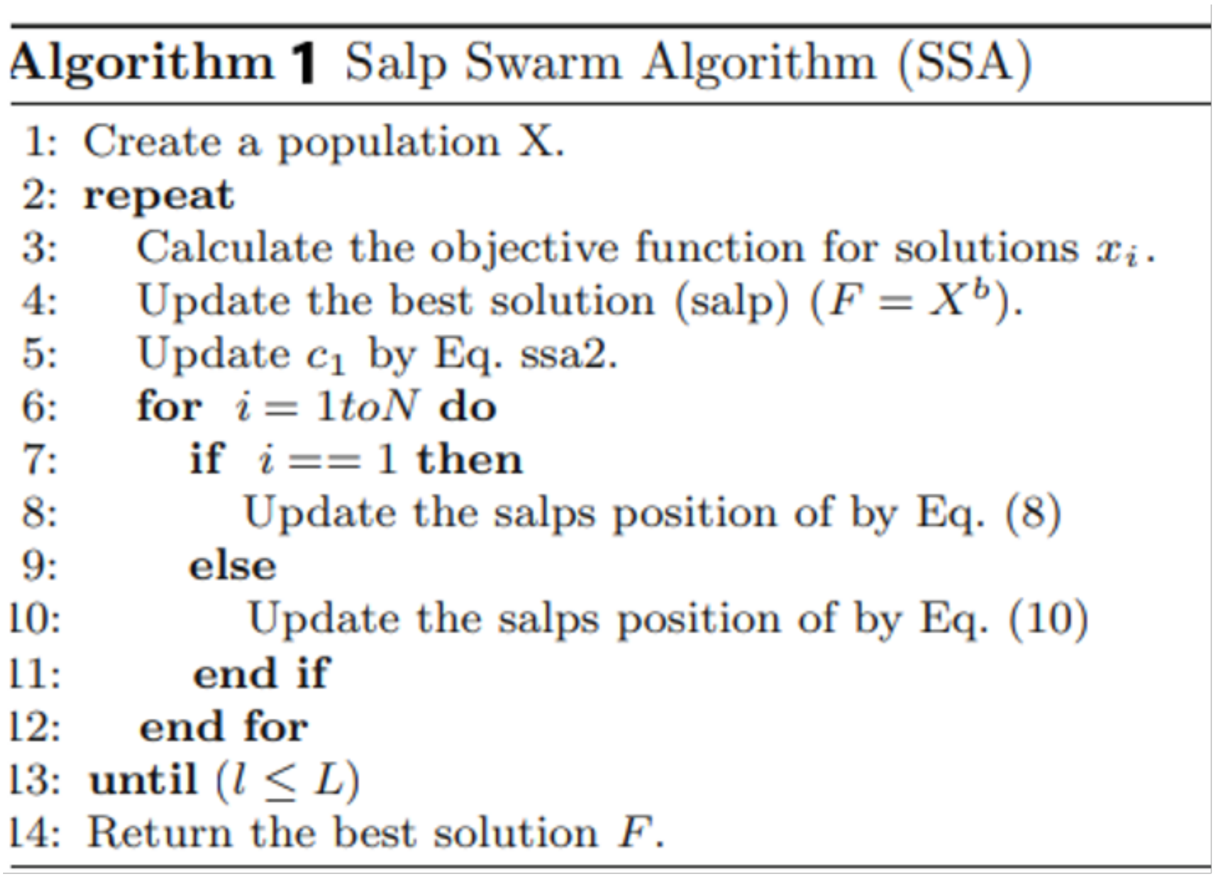
where x_j^1 denotes the leader's position in jth dimension. Fj denotes the food source. ub_j and lb_j are the upper and lower bounds, respectively. c_2 and c_3 are random variables in [0,1] that help in maintaining the search space. c_1 is a coefficient used to balance the exploitation and exploration phases. It is computed as follows:

$$c_1 = 2e^{-\left(\frac{4l}{L}\right)^2} \tag{11}$$

where l and L indicate the current loop and the max number of loops, respectively. Subsequently, the position of the followers is updated using the following equation:

$$x_j^i = \frac{1}{2}\left(x_j^i + x_j^{i-1}\right) \tag{12}$$

where x_j^i is the ith follower position, and $i > 1$. The entire sequence of the SSA is listed in Algorithm 1.



```
Algorithm 1 Salp Swarm Algorithm (SSA)
Create a population X.
repeat
    Calculate the objective function for solutions $x_i$.
    Update the best solution (salp) ($F = X^b$).
    Update $c_1$ by Eq. ssa2.
    for $i = 1toN$ do
        if $i == 1$ then
            Update the salps position of by Eq. (8)
        else
            Update the salps position of by Eq. (10)
        end if
    end for
until ($l \leq L$)
Return the best solution $F$.
```

Figure 4. Algorithm of Salp Swarm Algorithm (SSA)

https://doi.org/10.17993/3ctecno.2023.v12n3e44.364-386

4.3. GOLDEN EAGLE OPTIMIZER (GEO)

Golden Eagle Optimizer [25] is a swarm-intelligence metaheuristic algorithm and its multi-objective version is based on the golden eagles' hunting process. The authors are called Golden Eagle Optimizer (GEO) and Multi-Objective Golden Eagle Optimizer (MOGEO). GEO is founded on the intelligent adjustments of attack propensity and cruise propensity that golden eagles perform while searching for prey and hunting. MOGEO uses the same principles and is equipped with special tools to handle multi-objective problems. This study uses GEO as an optimization technique to improve the behavior of SSA.

This algorithm simulates the spiral foraging behavior of golden eagles. GEO completes the exploration and exploitation processes of the algorithm through attack vectors and cruise vectors.

4.3.1. ATTACK (EXPLOITATION)

The attack process of the golden eagle is represented by an attack vector A, which is shown in Eq. (13):

$$A_i = X_f^* - X_i \tag{13}$$

where A_i is the attack vector of the Ith golden eagle, X_f^* is the best position that golden eagle f has found so far, and X_i is the current position of golden eagle i.

where c_y is the yth element of C_i, b_j and by are the jth and yth elements of A_i, respectively. d corresponds to the value on the right side of Eq. (12). After determining the fixed variables of the cruise hyperplane, the cruise vector can be expressed by Eq. (4-17):

$$C_i = \left(c_1 = \text{ random }, c_2 = \text{ random }, \ldots, c_y = \frac{d - \sum_{j, j \neq y} b_j}{b_y}, \ldots, c_n = \text{ random} \right) \tag{14}$$

5. DATASET

NSL-KDD (National security lab–knowledge discovery and data mining) is the enhanced form of KDD99 to outperform its limitations. Initially, duplicated records in the training and test sets are eliminated. Second, there are different records chosen from the original KDD99 to accomplish dependable outcomes from classifier systems. Third, the issue of the unbalanced probability distribution was removed. The NSL-KDD data set has 125,973 training instances and 22,544 test instances, with 41 features, 38 consistent, and three categorical (discrete-valued) [26].

5.1. PARAMETERS' INITIALIZATION

To evaluate the quality of the proposed algorithm, we adjusted the parameters according to the parameters of the base paper [14]. In this way, 80% of the data were considered for training and 20% of the data were considered for testing. Table 1, shows the settings for the parameters of the proposed method.

Table 1. Initial values for the parameters in the proposed method

Parameters	Values
Train dataset	80 %
Test dataset	20 %
MaxIterations	20
nPop	30

5.2. EVALUATION CRITERIA

We will use more accurate measurement parameters to compare the proposed solution with algorithms. One of the criteria used to display the accuracy of data classification is the method of finding the accuracy.

The choice of a criterion for evaluating the effectiveness of the method depends on the problem we are trying to solve. Suppose several data samples are available. These data are given to the model individually and one class is received as output for each. The model predicted by the model and the actual data class can be displayed in a Table. Table 2 is called the confusion matrix.

Table 2. Confusion Table

	The label of predicted class		
	predicted **actual**	**Normal**	**Attack**
The actual class of label	**Normal**	True negative (TN)	False positive (FP)
	Attack	False negative (FN)	True positive (TP)

True positive: Samples that have been correctly detected as attacks by the test.

False positive: Samples that have been wrongly detected as attacks by the test.

True negative: Samples that have been correctly detected as Normal by the test.

False negative: Samples that have been wrongly detected as Normal by the test.

To evaluate the performance of the proposed method model, a comparative analysis with the base paper has been performed using several performance criteria.

MATLAB 2020b was used to implement the simulations. The used performance metrics are accuracy, Kappa.

5.2.1. ACCURACY CRITERION

The ability of a test to correctly distinguish fluctuations and normal events from others is called accuracy. To calculate the accuracy of a test, one must obtain the ratio of the sum of true positive and true negative samples to the total number of tested items. Mathematically, this ratio can be expressed as follows:

$$ACC = \frac{TN + TP}{FP + FN + TP + TN} \quad (15)$$

Detection rate refers to the test's ability to correctly detect attacks on the network (connections that are intrusion). The DR of the test method is the ratio of several connections correctly predicted to the total number of connections that are an intrusion. Mathematically, this can be expressed as follows:

$$\mathrm{DR} = \frac{TP}{TP + FN} \times 100 \quad (16)$$

False Positive Rate: This percentage falsely classifies normal cases as malware attacks compared to the total number of normal cases. This is given by the following formula.

$$\mathrm{FPR} = \frac{FP}{FP + \mathrm{TN}} \times 100 \quad (17)$$

Where FP and TN are true positive and negative numbers, respectively. A complete intrusion detection method must be 100% accurate while having a 0% false positive rate (FPR), which indicates that it can detect all possible attacks without error (incorrect classification), which is very difficult and probably impossible in real environments.

5.3. THE RESULTS' EVALUATION

In this thesis, the NSL-KDD dataset is reduced to a two-class problem, and the dataset is divided into normal and abnormal data. i.e. one class is considered normal and the rest of the classes are considered abnormal. Hence, the original NSL-KDD dataset is processed before testing. First, the character features in the NSL-KDD dataset are mapped to numeric features, three of which are as follows: "protocol type", which consists of three-character types. "Service", which includes 70 types of characters. The "flag" contains 11 characters. For these three parts, sorted substitutions are made using the decimal number starting with "1". After placement, the range is "Protocol Type" [1,3], the range is "Service" [1,70] and the range is "Flag" [1,5]. Since the range of each feature in the NSL-KDD dataset is approximately

https://doi.org/10.17993/3ctecno.2023.v12n3e44.364-386

different, it must be normalized before testing. The same as general datasets, each NSL-KDD dataset is divided into a training set and a testing set, which consists of the 80% training set and the 20% test set, and the same training set and test set are used for each algorithm. The number of populations and their repetition are 30 and 20, respectively.

GEOSSA has been used to find proper parameters for participating in the classification. The classification rate with ANFIS and the number of selected features act as the utility value for each solution. Table 3 shows the results related to the optimization of ANFIS parameters. By observing the results mentioned in this Table, it is determined that the use of the proposed method leads to better results in terms of accuracy, detection rate, and false positive rate. Using this mentioned method, the number of features is significantly reduced, which means reducing complexity and increased efficiency. Table 3. compares the proposed methods with other methods. The results of experiments performed in this study show that performing optimizing the ANFIS parameters leads to better results compared to other methods. In parameter optimization, the program runs in a shorter period and leads to higher accuracy, sensitivity, and specificity in implementation.

The results of the proposed method, and the base paper [14] in the NSL-KDD dataset are shown in Table 3, where "accuracy", "precision", "false positive rate", and "sensitivity" are provided for all two algorithms.

Table 3. Comparing the classification accuracy of the proposed method with the existing methods

Dataset	Evaluation criteria	Proposed method	Base article [14]
NSL-KDD	Accuracy	96.22	94.12
	False positive rate	438	3.45
	Detection Rate	96.68	95.80

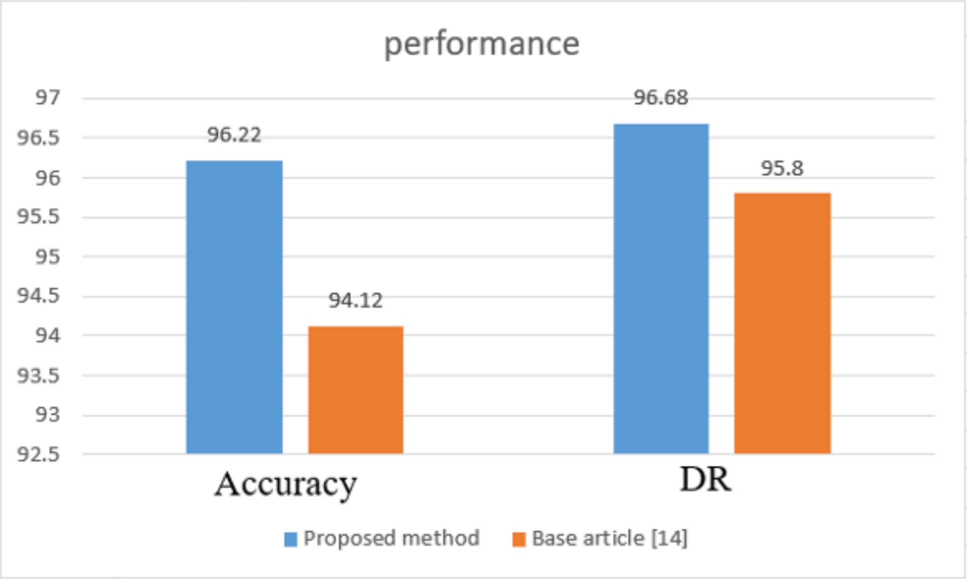


Figure 5. Comparison of the proposed method with existing methods

https://doi.org/10.17993/3ctecno.2023.v12n3e44.364-386

As shown in Figure 5, the proposed method is more efficient in the NSL-KDD dataset. Compared to the accuracy classification of the base paper method [14] the accuracy of the proposed method classification is higher, moreover, it has better sensitivity, accuracy, and false positives as shown in Table 3. For NSL-KDD random datasets, the classification accuracy of the proposed method can be more than 96.22%. The proposed method has a significant improvement in classification accuracy and better stability in network intrusion detection in comparison with existing methods.

The results of experiments performed in this study show that performing optimizing the ANFIS parameters leads to better results compared to other methods. The method of parameter optimization leads to a higher degree of accuracy, sensitivity, and specificity in the implementation.

The efficiency of intrusion detection is measured with the performance of detection rate and FAR. Because the detection rate and FAR are the essential parameters that are considered for the IDS to detect attacks. From the performance of the proposed model, the detection rate and false alarm rate are satisfactory compared with the other techniques as shown in Table 3. The GEOSSA-ANFIS model achieved a 96.68% detection rate and 0.438% false alarm rate, which is 3.012% FAR less than CSO-ANFIS [14] technique.

6. CONCLUSION

In this research, the intrusion detection system issues are presented and various techniques for solving the issues were discussed. ANFIS-based intrusion detection was a system proposed to detect attacks in networks. Because of the ANFIS, the combination of the fuzzy interference model and ANN has more advantages over other techniques. this thesis uses the Golden Eagle Optimizer (GEO) algorithm to improve the behavior of SSA. The proposed model (GEO-SSA-ANFIS) aims to determine the suitable parameters for the ANFIS by using the GEO-SSA algorithm since these parameters are considered the main factor influencing the ANFIS prediction process. Additionally, the GEO algorithm was used to optimize the ANFIS model to enhance its performance over intrusion detection which is an advantage for the IDS system. The proposed model has been used to solve the issues of intrusion detection and the model is validated using the familiar NSL-KDD dataset. The proposed model is compared with the other existing techniques like CSO-ANFIS. The results of the intrusion detection based on the NSL-KDD dataset were better and more efficient compared with those models because the detection rate was 96.68% and the FAR result was 0.438%.

6.1. FUTURE WORKS

The future work will be to enhance the detection and reduce the false alarm rate with a new machine learning-based classifier with another optimization technique for detecting attacks based on intrusion detection.

REFERENCES

(1) Ahmad, Z., Khan, A. S., Shiang, C. W., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1), e4150.

(2) Lansky, J., Ali, S., Mohammadi, M., Majeed, M. K., Karim, S. H. T., Rashidi, S., Hosseinzadeh, M., & Rahmani, A. M. (2021). Deep learning-based intrusion detection systems: A systematic review. IEEE Access, 9, 101574-101599.

(3) Tama, B. A., & Lim, S. (2021). Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation. Computer Science Review, 39, 100357.

(4) Kumar, G., Thakur, K., & Ayyagari, M. R. (2020). MLEsIDSs: Machine learning-based ensembles for intrusion detection systems—a review. The Journal of Supercomputing, 1-34.

(5) Gamage, S., & Samarabandu, J. (2020). Deep learning methods in network intrusion detection: A survey and an objective comparison. Journal of Network and Computer Applications, 169, 102767.

(6) Manimurugan, S., Majdi, A. Q., Mohmmed, M., Narmatha, C., & Varatharajan, R. (2020). Intrusion detection in networks using crow search optimization algorithm with adaptive neuro-fuzzy inference system. Microprocessors and Microsystems, 79, 103261.

(7) Alsyaibani, O. M. A., Utami, E., & Hartanto, A. D. (2021). Survey on Deep Learning Based Intrusion Detection System. Telematika, 14(2), 86-100.

(8) Thakkar, A., & Lohiya, R. (2021). A survey on intrusion detection system: Feature selection, model, performance measures, application perspective, challenges, and future research directions. Artificial Intelligence Review, 1-111.

(9) Jasim, A. D. (2022). A Survey of Intrusion Detection Using Deep Learning in Internet of Things. Iraqi Journal For Computer Science and Mathematics, 3(1), 83-93.

(10) Ayyagari, M. R., Kesswani, N., Kumar, M., & Kumar, K. (2021). Intrusion detection techniques in network environment: A systematic review. Wireless Networks, 27(2), 1269-1285.

(11) Adnan, A., Muhammed, A., Ghani, A. A. A., Abdullah, A., & Hakim, F. (2021). An Intrusion Detection System for the Internet of Things Based on Machine Learning: Review and Challenges. Symmetry, 13(6), 1011.

(12) Khraisat, A., & Alazab, A. (2021). A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecurity, 4(1), 1-27.

(13) Chauhan, M., & Agarwal, M. (2021). Study of Various Intrusion Detection Systems: A Survey. Smart and Sustainable Intelligent Systems, 355-372.

(14) Parfenov, D., Bolodurina, I., Zabrodina, L., & Zhigalov, A. (2021). Development of a solution for identifying network attacks based on adaptive neuro-fuzzy networks ANFIS. In 2021 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), (pp. 0491-0495). IEEE.

(15) Sinha, S., & Paul, A. (2020). Neuro-fuzzy based intrusion detection system for wireless sensor network. Wireless Personal Communications, 114(1), 835-851.

(16) Kondaiah, R., & Sathyanarayana, B. (2018). Trust based Genetic Neuro-Fuzzy System for Intrusion Detection and Self Adaptive Firefly integrated Particle Swarm Optimization Algorithm for Secure Routing in MANET. International Journal of Applied Engineering Research, 13(8), 5722-5735.

(17) Devi, R., Jha, R. K., Gupta, A., Jain, S., & Kumar, P. (2017). Implementation of intrusion detection system using adaptive neuro-fuzzy inference system for 5G wireless communication network. AEU-International Journal of Electronics and Communications, 74, 94-106.

(18) Brahma, A., & Panigrahi, S. (2015). A new approach to intrusion detection in databases by using artificial neuro fuzzy inference system. International Journal of Reasoning-based Intelligent Systems, 7(3-4), 254-260.

(19) Abd Elaziz, M., Ewees, A. A., & Alameer, Z. (2020). Improving adaptive neuro-fuzzy inference system based on a modified salp swarm algorithm using genetic algorithm to forecast crude oil price. Natural Resources Research, 29(4), 2671-2686.

(20) Mohammadi-Balani, A., Nayeri, M. D., Azar, A., & Taghizadeh-Yazdi, M. (2021). Golden eagle optimizer: A nature-inspired metaheuristic algorithm. Computers & Industrial Engineering, 152, 107050.

https://doi.org/10.17993/3ctecno.2023.v12n3e44.364-386