Yugoslav Journal of Operations Research
# (20##), Number #, #-#
DOI: https://doi.org/10.2298/YJOR2402150057R

Research Article

AN EFFICIENT SPOTTED HYENA OPTIMIZATION BASED NETWORK LOG INTRUSIONS IN MASSIVE SERVER INFRASTRUCTURE

Rajalingam R[*]
Annamalai University, Chidambaram-608001, sairamsai936@gmail.com,
ORCID: 0000-0002-4067-7492

Dr. Kavitha K
Annamalai University, Chidambaram-608001, kavithacseau@gmail.com,
ORCID: 0000-0002-0024-3737

Received: February 2024 / Accepted: June 2024

Abstract: With advancement of information technology, intrusion is becoming more common in the internet era. The increased use of cloud services have also resulted in assaults on servers. In order to enhance network performance, this research offers a novel IDS (Intrusion Detection System) that can quickly identify large-scale server assaults in wireless networks. This work uses SHO (Spotted Hyena Optimisation) that mimics spotted hyena's hunting behaviours. SHO, a swarm based meta-heuristics a method uses masses in solving issues and identifies important elements in server assaults. This optimisation algorithm improves detection and accuracy and is also used to detect web log hacking attacks and fake web pages. It outperforms other existing methods like SSO (Slap Swarm Optimisation), GWO (Grey Wolf Optimisation), and PSO (Particle Swarm Optimisation) algorithms in some applications. The goal of the experiment was to examine the suggested strategy using a common dataset. The suggested study appears to have produced notable outcomes for of F1 score, detection accuracy, and FAR (false alarm rate).

Keywords: Intrusion Detection System, Spotted Hyena Optimizer, Slap swarm optimization, Grey Wolf Optimization, and particle swarm optimization, Security, Meta-heuristic algorithm, Computer system organization, Network security, Artificial intelligence, Problem solving and search.

MSC: 68M12, 68T20

[*] Corresponding author

1. INTRODUCTION

The creation of IDS improved network security and safeguarded company data. Additionally, it checks into the network administrator to identify harmful activity on the network and warns the administrator to secure the data by putting in place appropriate defences against such assaults to keep data credentials safe from hackers. [1] WAF, encryption, authorization, and VPN have been used to secure network configuration and internet communication in the process of data protection.[2] Intrusion detection is a newer addition to the security technology armament.

Figure 1 represents the Network IDS, a firewall security system that overviews network trafficking and works to analyse the hostile attacks that can probably originate from outside the organization and the exploitation within the organisation [3]. IDS come in two flavours: HBIDS (Host-based IDS) [5] and NIDS (Network-Based IDS) [4]. NIDSs evaluate activities of networks for identifying suspicious elements and monitor traffics from certain network segments or devices. Networks employ a variety of hashing algorithms, including MD5, to maintain file security. SLAS (Security Log Analysis Systems) are also known as LIDS (Log-based IDS) [6]. HIDSs are installed computers or servers called hosts, and keep track of their activities.

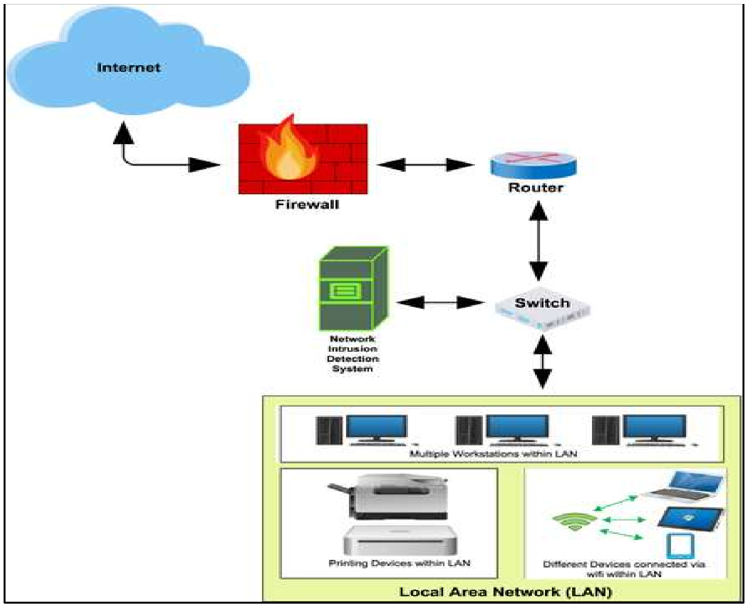
Internet
Firewall
Router
Switch
Network Intrusion Detection System
Multiple Workstations within LAN
Printing Devices within LAN
Different Devices connected via wifi within LAN
Local Area Network (LAN)

Figure 1: Network Intrusion Detection System

The technique used to find assaults on specific environments, computer abuse, policy breaches, and other types of improper behaviour is called LIDS. Using certain machine learning techniques (MLTs), logs can be leveraged as the primary source of knowledge [7].Better outcomes result from bio-inspired MLTs [8] that are being used in many applications. The meta-heuristic is a higher-level, problem-independent algorithm that provides a set of rules or tactics for designing heuristic optimisation. It is very successful in promoting the testing of the effectiveness of global solutions to challenges [9].

Through the logging, monitoring, and alerting method required for security, compliance, and service, IDS, which has been in use for years and is focused on network log intrusion and log management [10], provides an open source key to difficult and extremely common tests of log management and network monitoring caliber. However, leveraging these solutions to satisfy the functional, audit, and security requirements continues to be difficult for organisations of all sizes [31]–[44].

As a consequence of the problem, the research proposed a new SHO technique for detecting network log intrusion in huge networks. This is primarily made feasible by social bonds between spotted hyenas and co-operated conducts. Three SHO processes namely looking out for prey, encircling them and attacking identified preys can be expressed mathematically. SHO can also balance trade-offs between exploration and extraction phases. The recommended method is helpful for a variety of applications because to its straightforward and durable approach to solving complex and NP-hard problems. The comparison study reveals that it outperforms another meta-heuristic algorithm in terms of increased adaptability, flexibility, security, and QoS (Quality of Service) in several sectors.

2. LITERATURE REVIEW

Brandao and Georgieva [11] presented an efficient LIDS for predicting breaches, based on carefully chosen features. Several sources are combined into one dashboard from logs, with the main goal being to identify the distinguishing aspects. A few MLTs were moderately tried, and decision trees were used to forecast assaults. The biggest publicly accessible labelled log file dataset, KDD cup 1999, is used to illustrate the suggested technique.

Bul'ajoul et al [12] deployed in high-speed networks that used Cisco switch's parallel technologies with QoS, and Snort NIDPS dataset for actual networks. The findings of the experiment demonstrate the shortcomings of NIDPS, including their incapacity to analyse a large number of packets and discarding of packets during traffic loads, high-speed networks without inspecting them. By measuring the number of packets transmitted, examined, filtered, discarded, injected, and outstanding, Snorts analysis performance was measured. By employing QoS configuration options in a Cisco Catalyst 3560 series switch and parallel Snorts, it increases NIDPS performance and reduces the amount of lost packets.

For resource-constrained smart industries, a fog-assisted Deep Learning IDS WAS proposed BY Attique et al [13]. The suggested Cu-DNNGRU (cuda-deep neural network gated recurrent unit) framework was assessed using appropriate operating requirements including accuracy, precision, recall, and F1-score after being trained on the N-BaIoT dataset. Additionally, cutting-edge classifiers such as the Cu-LSTMDNN (cuda- long-short term Memory deep neural network), Cu-BLSTM (cuda- Bidirectional Long-Short Term Memory), and Cu-GRU (cuda- Gated Recurrent Units) were used to experimentally explore the Cu-DNNGRU. Their results of simulations demonstrated sample strengths in validations. The presented techniques achieved F1-score, accuracy, precision, and recall. Performance analysis demonstrated significantly higher values than other benchmarked schemes, and also strengthened competitive security.

Jemili and Bouras [14] proposed a new algorithm based on big data fuzzy analytics, a new intrusion detection system: The pre-processing training dataset is clustered and classified using the fuzzy C-mean (FCM) method. Both the UNSW-NB15 and CTU-13 are utilised as distributed huge datasets to show the effectiveness of the techniques. The recommended approach showed improved performance values for accuracy, FARs, precision, and alarm precision.

Morris et al. [15] proposed new IDS and BASED ON data logging FOR industrial control systems. First, the network traffic data record for a gas pipeline is shown. The gas pipeline's data log was kept, and it includes traces of both normal use and cyberattacks. Second, a programmable logic controller, a human machine interface, a gas pipeline model based on Somu links, and Modbus/TCP (Transmission Control Protocol) communications are included in a representation of an expandable virtual gas pipeline. The virtual gas pipeline makes it possible to simulate both typical behavior and cyberattacks. For training and testing purposes, the solution overlaps the virtual gas pipeline.

Vinayakumar et al [16] proposed DNNs (Deep neural networks), a subset of DNNs, created flexible and effective IDSs that could recognise and categorise unforeseen and unanticipated intrusions. Because of the rapid expansion of attacks and the continuing change in network behaviour, it is required to assess multiple datasets generated over time using both static and dynamic methodologies. This type of study allows for the identification of the best algorithm for successfully predicting incoming cyber threats. On various publicly available benchmark malware datasets, a detailed investigation of DNN and other classical MLTs were given [17]. Hyper-parameter selections of network topologies and DNN parameters included the usage of KDDCup 99 dataset. DNN model that performed well on KDDCup 99 was applied to other datasets, including Network Security Laboratory - Knowledge Discovery in Databases (NSL-KDD), Kyoto, UNSW-NB15, A Dataset for Intrusion Detection Systems in Wireless Sensor Networks (WSN-DS), and CICIDS 2017. Finally, the proposed method employed a scale-hybrid-IDS-AlertNet architecture, which was highly scalable and hybrid DNN, to effectively monitor network traffic and host-level events in real-time and proactively warn against possible intrusions.

Vigneswaran et al [18] used NIDSs to forecast attacks. KDDCup-99 dataset trained and assessed the network. DNNs with a learning rate of 0.1 were iterated 1000 times. To compare, the same dataset is trained using DNN with 1-5 layers and prominent MLTs. According to comparisons, DNNs with three layers surpassed all other standard MLTs.

Dionísio et al [19] proposed DNN to present the processing pipeline of a cutting-edge tool for safeguarding and processing cyber security data received from twitter. CNN (convolution neural network) can identify tweets that mention assets in an IT infrastructure and include security-related information. Bi_LSTM extracted identified entities from tweets for security warnings or indications of compromises. The suggested pipeline achieved average true positive/true negative rates, and F1-scores on three case studies.

Rassam et al [20] suggested the use of SIEM (Security Information and Event Management) is used to demonstrate the weaknesses of old technology/systems in addressing enormous data quantities and extremely sophisticated attacks. In order to address massive data quantities and complex threats, it addresses the requirements for Big Data analytics' effective implementation in the context of cyber threat intelligence and the cyber-security environment. To show the provenance of data based on the requirements and challenges of implementing Big Data analytics for cyber-security, the wine dataset is used as an input and effectively enhances the dependability of data obtained from diverse sources. Finally, the performance results highlight the difficulties that arise from such adoption and offer some approvals to address these difficulties.

3. PROPOSED METHODOLOGY

The background facts regarding the network log intrusion, the spotted hyena optimisation is explained in detail which clearly helps to understand the proposed method.

The Spotted Hyena Optimization (SHO) algorithm mimics the hunting behaviors of spotted hyenas through a series of iterative processes inspired by the animal's predatory tactics. Like hyenas hunting in groups, SHO employs multiple search agents to collaboratively locate and converge upon optimal solutions. This approach is applied to intrusion detection in network infrastructure by leveraging the collective intelligence of the search agents to identify and mitigate potential threats within the network logs.

SHO's specific features make it well-suited for detecting network log intrusions in massive server infrastructures compared to other meta-heuristic algorithms. Its ability to balance between exploration and exploitation, represented by the alternating phases of search and hunt, enables it to efficiently navigate the vast search space of network logs. Additionally, SHO's encircling approach and use of random vectors facilitate targeted searches for potential threats, enhancing its effectiveness in detecting intrusions.

3.1. Network Log Intrusion in Massive Server infrastructure

IDS detections of external and internal intrusions on computer and network resources, as well as on the data stored in these resources, is the main goal of an [21]. The majority of incursions are thought to originate from unreliable networks. Unauthorised access from the outside can be achieved via negotiating a permeable firewall, taking advantage of security flaws, tunneling using "harmless" protocols, or totally evading security precautions via unprotected links from external systems. Threats from malevolent "insiders" manifest due to lax physical protection, poor masquerade, or unauthorised data access attempts by users.

IDSs that protect distributed, real-time systems must take those systems' requirements into consideration. IDSs that address inherently (soft) real-time problems by analysing data as it arrives must be carried out quickly and upon completion, an alert must be sent in a timely manner to prevent further attacks from intrusion. They must not significantly tax available resources or clog up the network or computer infrastructure. They need to work deterministically and fail in a way that doesn't impair the system's operation.

3.2. General Architecture of a NIDS

Either a distributed or centralised architecture can be used to set up an IDS. The same device is utilised for data gathering and processing in a centralised design. The device could turn into a network bottleneck even if it is in queue [22]. In distributed designs, load-balances split traffics amongst sensors, which subsequently communicated acquired data to stations and management consoles for analyses. Figure 2 depicts a high-level perspective of the architecture of network IDSs.

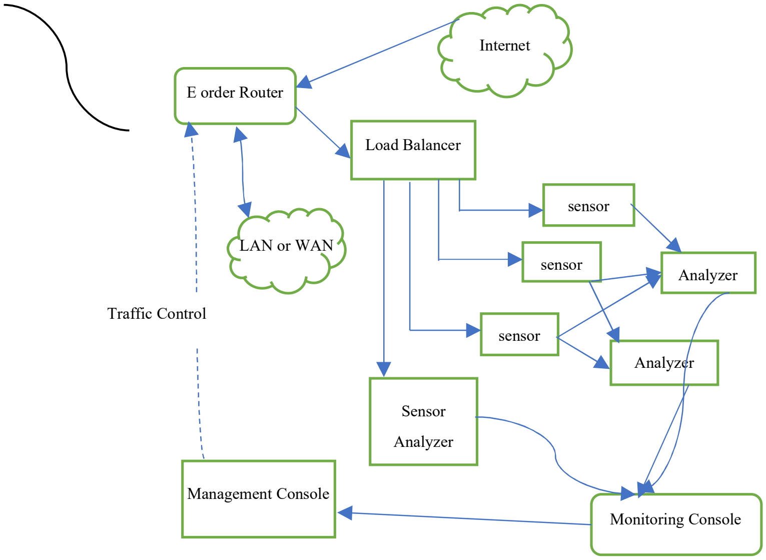

Figure 2: Generalized Network IDS Architecture

IDSs are five consecutive subprocesses that include the following:

1. Load balanced distribution of network traffics amongst sensors.
2. Sensing: Unusual traffic is separated from networks that are suspect.
3. Analysing: Identifying the traits and risks of suspicious traffic
4. Monitoring: generating reports, making operators aware of the hazard, and simplifying operator notification
5. Managing: IDS configuration is done as well as a reaction to the danger.

IDSs execute with five sequential sub-processes.

The overall sequential IDS process's cardinality of linkages between pairs of sub-processes is shown in Figure 3.

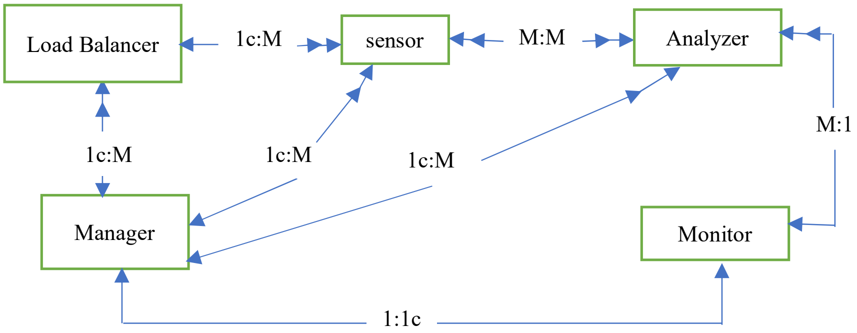

Figure 3: Relational cardinality of IDS subprocesses

In the process of monitoring, the operator is notified and given a graphical or textual view of the threat based on the security policy. Monitors must be adjusted in accordance with the protected network's traffic patterns.

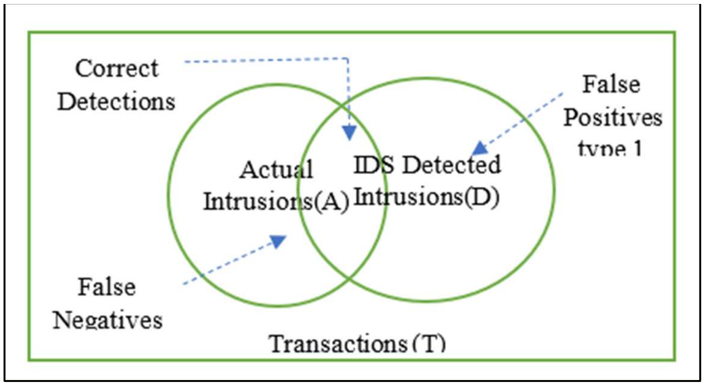

Figure 4: False Positive (Type 1) and False Negative (Type 2) Errors

False Positive Ratio =|D-A|/T

False Negative Ratio = |A - D|/T

Where:

- **D** = Detected instances
- **A** = Actual instances
- **T** = Total instances

Frequent alerts on unimportant or routine events have a high false-positive rate (Type 1 error), which causes operators to disregard the IDS. On the other hand, a sufficiently desensitized monitor will ignore real dangers. IDSs may give false senses of security and lead to false negatives (Type 2 errors). Figure 4 provides an example of the definitions of false positive and false negative.

3.3. Spotted Hyena Optimization for Network Log Intrusions in Massive Server Infrastructure

The spotted hyena optimisation approach is comparable to the spotted hyena, a large predatory canine that inhabits a variety of settings including arid and open areas. The spotted hyenas prey upon both small and large herbivores, such as zebras and wildebeests [23].

Due to its ability to trace, stalk, encircle, and ambush preys, spotted hyenas are one of the most socially skilled animals. This work’s spotted hyena optimisation network contains five sub-processes, making it the best at network intrusion hunting.

```
Input:
- P_i (i = 1, 2, ..., n): Initial population of search agents (spotted hyenas)
- MaxIterations: Maximum number of iterations
- FitnessFunction: Function to evaluate the fitness of search agents
Output:
- BestSearchAgents: Discovered best search agents
Procedure SHO:
1. Initialize variables h, B, E, N
2. Compute fitness of each search agent using FitnessFunction
3. Set P_h as the best search agents
4. Set C_h as groups or classes of unseemly optimal solutions
5. x = 0
6. While x < MaxIterations do:
7.    For each search agent in P_i do:
8.        Update position of the current search agent
9.    End for
10.   Update values of h, B, E, and N
11.   Check and adjust the range of each search agent if it exceeds the search range
12.   Recompute fitness of each search agent using FitnessFunction
13.   Update P_h if better solutions are found compared to prior optimal solutions
14.   Update groups C_h and P_h
15.   Increment x
16. End while
17. Return P_h
End Procedure
```

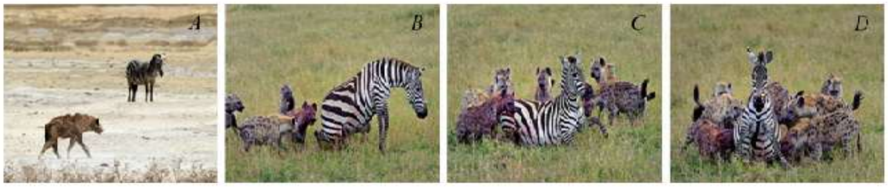

Figure 5: Various phases of spotted Hyena hunting behavior: (A) Searching and tracking (B) Chasing (C) troublesome and encircling (D) immobilize situation and attack prey

Spotted hyena's behaviours [24] can be represented mathematically using equations (1) and (2).

$$\underline{DIST}_{hyen} = \left|\underline{CVB}.\underline{PosH}_p(IT) - \underline{PosH}_{(x)}\right| \tag{1}$$

$$\underline{PosH}_{(IT+1)} = \underline{PosH}_{pos}(IT) - \underline{CVE}.\underline{DIST}_{hyena} \tag{2}$$

Where, $\underline{DIST}_{hye}$ implies prey's distances from spotted hyenas, IT denotes iterations, $\underline{CVB}$ and $\underline{CVE}$ are form vector coefficients. $\underline{PosH}_{pos}$ is the variable that represents the position of vector prey, while $\underline{PosH}$ is the variable that represents the position of hyenas. CVB and CVE stand for computer vectors obtained from equations (3) to (5)

$$\underline{CVB} = 2.RV\underline{d}_1 \tag{3}$$

$$\underline{CVE} = 2.\underline{hyena}.RV\ \underline{d}_2 - \underline{hyena}. \tag{4}$$

$$\underline{hyena} = 5 - \left(ITER * \frac{5}{MAX_{ITER}}\right) \tag{5}$$

Where $ITER = 1,2, \ldots\ldots MAX_{ITER}$. $\underline{hyena}$, reduce linearly in the range [5,0] and thus balance explorations/exploitations. They increase iterations as maximum (MAX_{ITER}). Random vectors in the range (0,1) are represented by $RV\underline{d}_1$ and $RV\underline{d}_2$. In general, spotted hyena placements are often modified to reflect the location of their prey. Various dimensions, including 2D, 3D, and multidimensional, are used for this update. Figure 6 shows the search for spotted hyenas in the vicinity of the best option.

Figure 6: Search for Spotted hyenas around the best- fit solution (exploration)

Hunting Processes: Hyenas hunt in groups and rely on trusted circles. To quantitatively define their conduct, it is assumed that optimum searches and optimal agents know prey locations. According to best and ideal search agents, location details are adjusted in other search agents. The mathematical explanation of hunting processes can be depicted as Equations (6-8).

$$\underline{DIST}_{hyena} = \left|\underline{CVB}.\underline{PosH}_{hyena} - \underline{PosH}_k\right| \tag{6}$$

$$\underline{PosH}_k\underline{PosH}_{hyen} \quad - \underline{CVE}.\underline{PosH}_{hyena} - \underline{PosH}_k \tag{7}$$

$$\underline{CLUST}_{hyena} = \underline{PosH_k} + \underline{PosH_{k+1}} + \cdots + \underline{PosH_{K+NSH^*}} \tag{8}$$

Figure 7: Spotted hyenas attack best-fit (exploitation)

Figure 7 represents best-fit spotted hyena attacks, where, first SH is represented by $\underline{PosH}_{hyena}$, while $\underline{PosH}_k$ represent positions of another SH. NSH indicate computed SHs obtained from equation (9).

$$N_{SH} = CNT_{Ns}\left(\underline{PosH_{hyen}}\ ,\underline{PosH_{hyena+1}},\underline{PosH_{hyena+2}},\dots,\left(\underline{PosH_{hyena}} + \underline{MRV}\right)\right) \quad (9)$$

Where $\underline{MRV}$ denotes a vector value chosen at random between 0.5 and 1, NS denotes the number of solutions, and NS computes the solutions of all SHs. The variable $\underline{MRV}$ represents the ideal search location, while $\underline{CLUST}_{hyena}$ represents the group of optimal solutions.

Exploitations: The vector value of $SH\left(\underline{hyena}\right)$ is reduced to capture the prey. The co-efficient variation for $\underline{CVE}$ is decreased for updating the SH's vector value that is decreased from 5 to 0 on course iteration processes. The group of SHs catch the preys successfully when satisfy the condition $|\underline{CVE}| < 1$. Mathematically can be shown below like equation (10).

To catch the prey, the vector value of $SH\left(\underline{hyena}\right)$ is reduced. For updating the SH's vector value, which is reduced from 5 to 0 on course iteration operations, the co-efficient variation for $\underline{CVE}$ is reduced. When the criterion $|\underline{CVE}| < 1$.. is met, the group of SHs successfully captures the preys. Equation (10) can be represented mathematically below.

$$\underline{PosH}_{hyena} \quad = \frac{\underline{CLUST}_{hyena}}{N\_SH} \quad (10)$$

Where, $\underline{PosH}_{hyena+1}$ updates other search agents' positions depending on best search agent positions, and best solutions are saved. SHO locates search agents in order to update locations of other search agents and catch preys. The general architectural design for the suggested Method is shown in Figure 8.

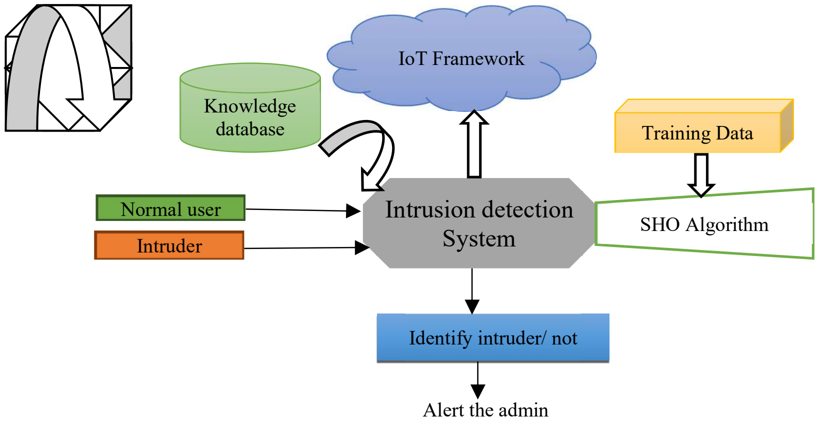

Figure 8: Architecture diagram for Proposed Method

The random search step determines a search factor's location based on a random selection of search factors rather than the best current search factor. As previously demonstrated, the |E|>1 technique emphasises random search and allows the SHO to do a worldwide search.

SHO steps are:

Step 1: Create the first spotted hyenas (P_i), where i=1, 2,..., n

Step 2: Initialize variables B, E, h, N, and max. iterations.

Step 3: Each search agent's fitness value is calculated.

Step 4: The most effective search agents are identified in the search environment.

Step 5: Utilizing Equations (8) and (9) to identify best solution groups. Using Eq. (10)

Step 6: Using Eq. (10) in Step 6, the location of search agents are updated.

Step 7: Modify search environments of search agents to prevent them from exceeding set limits.

Step 8: If there are better solutions than previous optimal solutions, determine adequacies of updated search agents and vector updates of Phs.

Step 9: Update the search agent Update the Spotted Hyenas (Ch) Group to fit.

Step 10: If satisfied, the algorithm finishes at step 10. If not, step 5 must be repeated.

Step 11: Present best optimum solutions on stopping.

Based on these estimates and idealisations, algorithms can condense the core SHO stages. Figure 9 depicts the flowchart of the SHO algorithm. Below is shown the SHO algorithm's pseudo code:

Algorithm 1: Pseudocode of the Spotted Hyenas Optimizer Algorithm

```
Input: $P_i\,(i = 1,2,\ldots\ldots,n)$ is the spotted hyenas' population.
Output: the best search agents are discovered
procedure SHO
The variables h, B, E, and N are initialled
Compute fitness of search agents
$P_h =$ best search agents
$C_h =$ groups or classes of all unseemly optimal solutions
    while (x<Max number of iterations) do
    for each search agent, do
          Update positions of current agents
end for
  update h, B, E, and N
The range of each search agent should be checked and adjusted if their value exceeds the search range.
   The fitness is computed to each search agent
  Update $P_h$ on finding better solutions than prior optimal solutions
  Update groups $C_h$ and $P_h$.
  $x = x + 1$
end while
return $P_h$
end
```

The SHO algorithm's most crucial characteristics are as follows:

- The SHO keeps the best solutions thus far throughout the iteration.
- The SHO encircling approach defines the locations of the solutions in a hyper-sphere that is generalised to huge dimensions.
- Random vectors *B* and *E*, which assist build hyperspheres with a range of random places, enable selective solutions.
- SHO offers focused solutions to identify potential prey positions.

- The vectors' corrected values, E and h, control extractions and explorations intents. It also switches between the two.
- half of iterations are searches (explorations) ($|E| \geq 1$) by vectors $\vec{E}$, and others are hunts (exploitations) ($|E| \leq 1$).

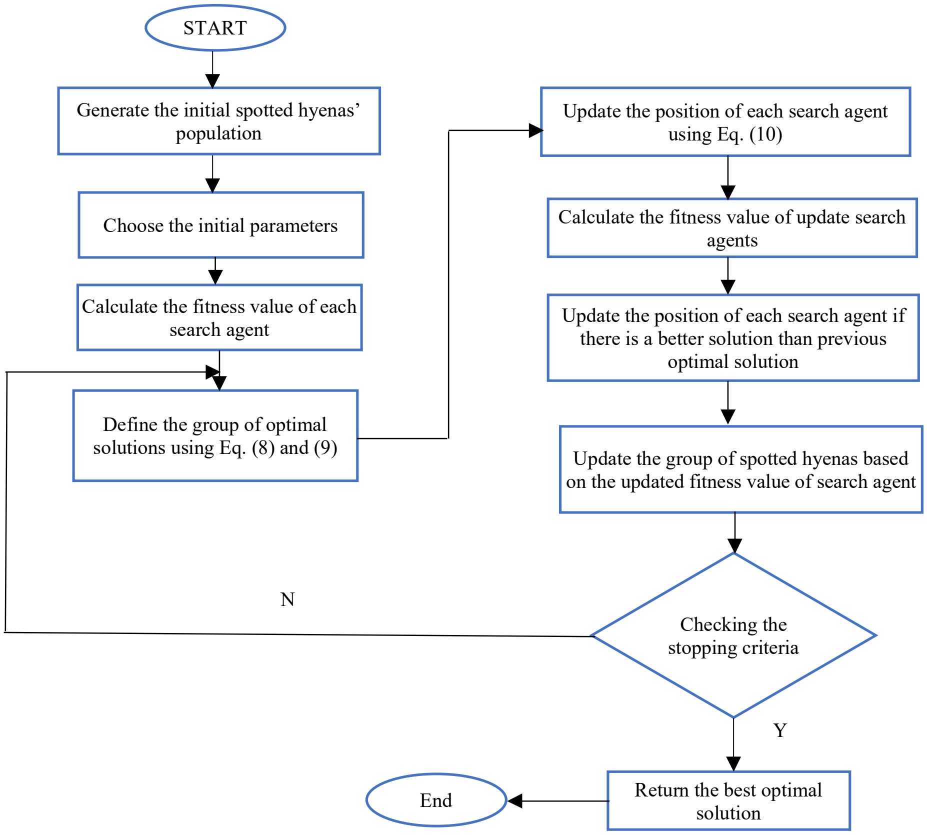
START
Generate the initial spotted hyenas' population
Choose the initial parameters
Calculate the fitness value of each search agent
Define the group of optimal solutions using Eq. (8) and (9)
Update the position of each search agent using Eq. (10)
Calculate the fitness value of update search agents
Update the position of each search agent if there is a better solution than previous optimal solution
Update the group of spotted hyenas based on the updated fitness value of search agent
Checking the stopping criteria
N
Y
Return the best optimal solution
End

Figure 9: SHO Algorithm flow chart

4. RESULTS AND DISCUSSION

The outcomes of several datasets used in this study for assessing the proposed SHO are described; the evaluation metrics showed that the suggested SHO was successful based on the results of experiments.

The proposed methodology, known as SHO (Spotted Hyena Optimization), appears to be tailored specifically for Intrusion Detection Systems (IDS) in the context of network log analysis. While the text primarily focuses on DDoS attacks, it suggests that the methodology can handle various intrusion tactics effectively.

1. DDoS Attacks: The DDoS (Distributed Denial of Service) attacks and discusses how the proposed IDS handles them. It assesses the detection efficiency of the SHO-based IDS for various types of DDoS attacks, such as LDAP, MSSQL, UDP, and SYN. The methodology aims to accurately detect these attacks while minimizing false positives, as indicated by the low False Alarm Rate (FAR) reported in the results.
2. Phishing Attempts: the SHO methodology's effectiveness in detecting phishing attempts could be inferred from its ability to analyze network packet data and identify anomalous behavior. Phishing attacks often involve deceptive emails or websites designed to trick users into divulging sensitive information. An IDS based on SHO could potentially detect patterns indicative of phishing activity by analyzing network traffic and identifying suspicious communication patterns or URLs.
3. Other Common Intrusion Tactics: The SHO-based IDS is likely designed to detect a wide range of intrusion tactics beyond just DDoS attacks. Since it utilizes a population-based metaheuristic algorithm inspired by spotted hyena hunting behaviors, it may be inherently adaptive and capable of identifying various types of anomalies in network traffic. This adaptability allows it to handle different types of attacks, including but not limited to port scanning, malware propagation, unauthorized access attempts, and data exfiltration.

4.1. Datasets

The assessment datasets allow us to assess how effectively the recommended method can detect intrusive behaviour, which is crucial for the validation of any detection methodology. Datasets used for commercial analyses of network packet are not easily accessible due to privacy issues. However, there are a few publicly accessible datasets that are frequently used as standards, such the Australian Defence Force Academy Linux Dataset (ADFA-LD), Network Security Laboratory - Knowledge Discovery in Databases (NSL-KDD), and the Defence Advanced Research Project Agency (DARPA) [25]. The characteristics and constraints of the current datasets used to create and assess IDS are covered in this section.

4.2. Performance metrics for Network Log Intrusion

In this paper, the appropriate performance assessment criteria were used to assess the effectiveness of the suggested IDS. These metrics are essential to comprehend the advantages and disadvantages of various classification algorithms and to select the optimal ones for usage in the detection process. IDS are evaluated using the following performance metrics: IDR (Intrusion Detection Rate), FAR, Precision, Recall, and F1-Score, True Negative Rate (TNR), ROC Curve, and Intrusion Detection Accuracy (IDAcc) [26]. The equations may be expressed using Equations (11-20).

- TPR: This statistic is calculated as the proportion of accurately anticipated attacks to all assaults. If every incursion is found, the TPR is 1, which is quite uncommon for an IDS. The Detection Rate (DR) or the Sensitivity are other names for TPR. The TPR can be expressed mathematically as

$$TPR = \frac{TP}{TP+FN} \tag{11}$$

- FPR: It is measured as the proportion of normal occurrences that are wrongly labelled as attacks to all of the normal occurrences.

$$FPR = \frac{FP}{FP+TN} \tag{12}$$

- FNR: when a detector misclassifies an anomaly as normal because it is unable to detect it. Math is used to express the FNR.

$$FNR = \frac{FN}{FN+} \tag{13}$$

- Classification rate or Accuracy: determines how well the IDS is able to identify typical or unusual traffic behaviours. It is calculated as the ratio of all cases that were successfully predicted to all instances:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+F} \tag{14}$$

$$PR = \frac{TRA}{TPA+FPR} \tag{15}$$

$$REC = \frac{TPA}{TPA+FNA} \tag{16}$$

$$IDRA = \frac{TPA}{TPA+FNA} \tag{17}$$

$$FS = 2 * \frac{(PR*REC)}{(PR+RE\ \ )} \tag{18}$$

$$IDA = \frac{TPA+TNR}{TPA+FNA+TNR+} \tag{19}$$

$$O\_IDR = \frac{\sum \quad TP\_Each-Attack\_Type}{\sum \quad TP\_Each-Attack\_Type \ \sum \quad FN\_Each-Attac\ \_Type} \tag{20}$$

The variables FS and IDA represent the F-Score, the Intrusion Detection Accuracy, and the Overall Intrusion Detection Rate, respectively. TNR stands for True Negative Rate, FPR stands for False Positive Rate, FA stands for FAR, PR stands for Precision, REC stands for Recall, IDRA stands for Intrusion Detection Accuracy, TPA and FNA stand for True Positive Attack and False Negative Attack, respectively.

4.3. Experimental Results:

The suggested model performed well when measured by the precision, accuracy, and F-score values, outperforming other existing IDSs like SSO [27], GWO [28], PSO [29], and SHO [30].

Figure 10 compares the proposed and current IDSs in terms of accuracy, recall, and F-score. The existing IDSs include SSO, GWO, PSO, and SHO. Here, experiments have been carried out in order to assess the proposed model. When compared to SSO, GWO, and PSO, the suggested SHO system archives greater values of 0.967%, 0.93%, 0.97, and 0.96.45.

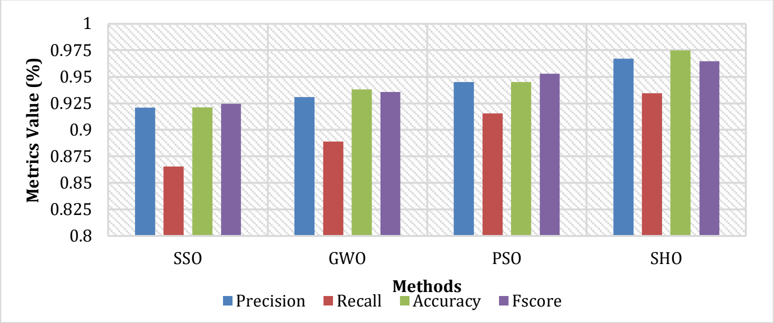

Figure 10: Comparative analysis of metrics and methods

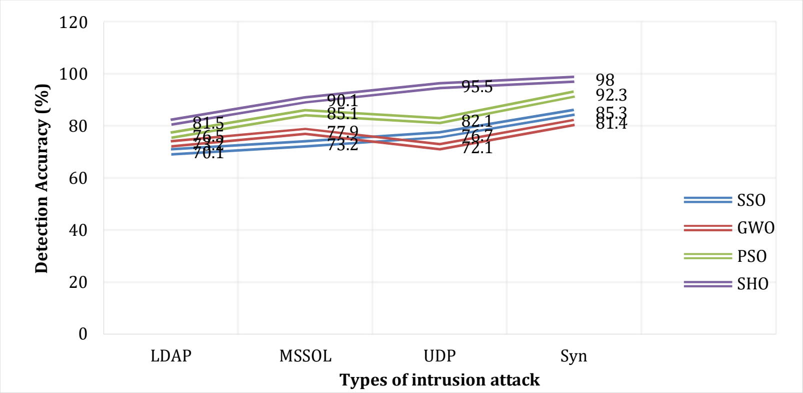

Figure 11: Results of detection accuracy analysis

Figure 11 displays the proposed IDS's detection efficiency for several DDoS attack types. It took into account four distinct kinds of intrusion attempts, including LDAP, MSSQL, UDP, and SYN. various assault-affected occurrences are taken into account while running tests to determine how well the proposed IDS handles various attack detection procedures. Additionally, the pertinent works are taken into account to demonstrate the effectiveness of the suggested IDS.

Table 1: Comparative analysis and performance of metrics with methods on different datasets

Methods / Dataset	**DARPA**		**NSL-KDD**		**ADFA-LD**	
	Accuracy (%)	**FAR (%)**	**Accuracy (%)**	**FAR (%)**	**Accuracy (%)**	**FAR (%)**
SSO	99.1	1.64	97.21	1.54	97.71	1.45
GWO	99.5	1.56	97.32	1.45	97.78	1.40
PSO	99.6	1.53	97.46	1.40	97.86	1.35
SHO	99.85	1.45	98.65	1.35	98.92	1.30

Figure 12 represents the outcome of the proposed methodology (SHO) when compared to the Accuracy results of other known methods are illustrated in Figure 5. The proposed SHO gives higher Accuracy results 98.92%, whereas other methods such as SSO, GWO, and PSO also gives higher precision for proposed SHO based different methods such as 99.85%, 98.35% and 98.92% respectively for DARPA, NSL-KDD and ADFA-LD datasets (Refer Table 1). The proposed algorithm gives highest results than the existing methods respectively.

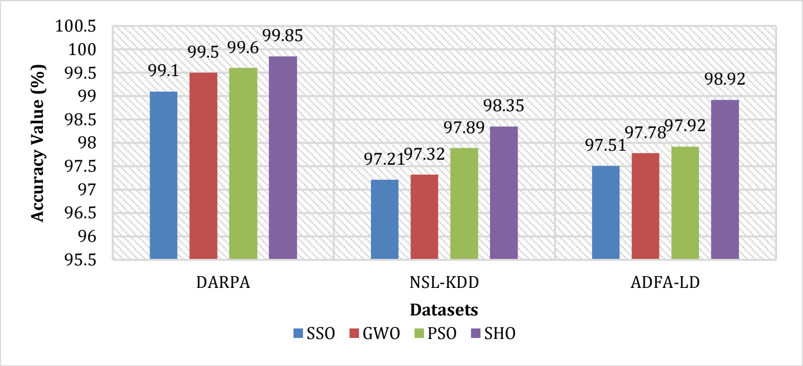

Figure 12: Accuracy comparison of metrics with different methods vs. Datasets

By contrasting the proposed model SHO with the current IDSs created using the SSO, GWO, and PSO existing algorithms, Figure 13 illustrates the lifetime of the suggested model SHO. The suggested approach effectively and accurately identifies DDoS with little to no FAR. The total detection accuracy of the suggested model is more than 1% higher than the accuracy of the existing methods, allowing it to obtain a final result of 97.95%.

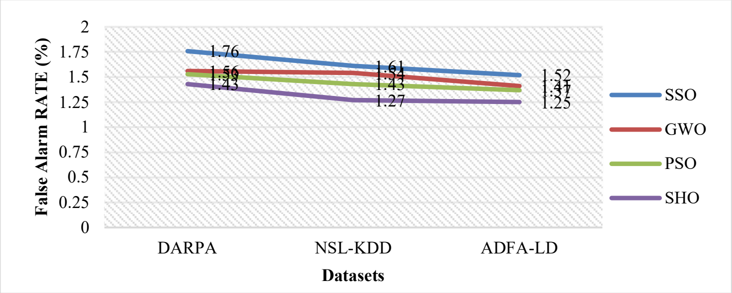

Figure 13: Far comparison of metrics with different methods vs. Datasets

5. CONCLUSIONS

SHO approach is a group-based population-based metaheuristic algorithm that uses spotted hyena hunting behaviours to tackle optimisation issues. It is intended for the creation and analysis of network log intrusion in enormous server architecture. The project's objective is to create an effective and efficient Intrusion Detection System using log analysis. Light-weight processor known as Agents will run in the background and be installed on the host machines. When communicating with the server, the Agents will send the required data from the host. A web panel that offers a user-friendly interface and a broad view of the hosts is also included. Remote server access and system monitoring are both possible with the web panel. The proposed method improves 95% intrusion detection attack in massive server infrastructure. Some Metrics like Precision, Recall, Accuracy, F-Measure Score, detection accuracy and FAR has been evaluated and compared with existing method. This work may be improved further by including light-weight feature optimisation to get greater detection accuracy in less time and with a lower FAR.

The results showcasing enhanced intrusion detection accuracy, especially in large server environments. As cyber threats continue to evolve in sophistication and frequency, such advancements in detection capabilities are crucial for maintaining robust cybersecurity defenses in dynamic digital landscape. Implementing the SHO-based Intrusion Detection System (IDS) can significantly enhance network security by accurately detecting various intrusion tactics like DDoS attacks and phishing attempts. This improves operational efficiency by minimizing false alarms and optimizing resource utilization through its population-based metaheuristic algorithm, ensuring effective threat mitigation.

Funding: This research received no external funding.

REFERENCES

[1] A. Javaid, Q. Niyaz, W. Sun, and M. Alam. “A deep learning approach for network intrusion detection system,” In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pp. 21-26, 2016.

[2] J. Pokala, and B. Lalitha, “A novel intrusion detection system for RPL based IoT networks with bio-inspired feature selection and ensemble classifier,” *Research square*, pp. 2-24, 2021.

[3] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad. “Network intrusion detection system: A systematic study of machine learning and deep learning approaches,” *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, pp. 1-29, 2021.

[4] S. Meftah, T. Rachidi, and N. Assem. “Network based intrusion detection using the UNSW-NB15 dataset,” *International Journal of Computing and Digital Systems*, vol. 8, no. 5, pp. 477-487, 2019.

[5] P. Deshpande, S.C. Sharma, S.K. Peddoju, and S. Junaid. “HIDS: A host based intrusion detection system for cloud computing environment,” *International Journal of System Assurance Engineering and Management*, vol. 9, pp. 567-576, 2018.

[6] U.K. Raut. “Log based intrusion detection system,” *IOSR Journal of Computer Engineering*, 2018, vol. 20, no. 5, pp. 15-22.

[7] M. Hasan, M.M. Islam, M.I.I. Zarif, and M.M.A. Hashem. “Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches,” *Internet of Things*, vol. 7, pp. 100059, 2019.

[8] Y. Zhang, and Y. Wang. “A novel energy-aware bio-inspired clustering scheme for IoT communication,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 10, pp. 4239-4248, 2020.

[9] A. Forestiero. “Metaheuristic algorithm for anomaly detection in Internet of Things leveraging on a neural-driven multiagent system,” *Knowledge-Based Systems*, vol. 228, pp. 107241, 2021.

[10] S. Smys, A. Basar, and H. Wang. “Hybrid intrusion detection system for internet of things (IoT),” *Journal of ISMAC*, vol.2, no. 04, pp. 190-199, 2020.

[11] A. Brandao, and P. Georgieva. “Log Files Analysis for Network Intrusion Detection,” In *2020 IEEE 10th International Conference on Intelligent Systems (IS)*, Varna, Bulgaria, 28-30 August 2020, pp. 328-333.

[12] W. Bul'ajoul, A. James, and M. Pannu. “Improving network intrusion detection system performance through quality of service configuration and parallel technology,” *Journal of Computer and System Sciences*, vol. 81, no. 6, pp. 981-999, 2015.

[13] D. Attique, H. Wang, and P. Wang. “Fog-assisted deep-learning-empowered intrusion detection system for RPL-based resource-constrained smart industries,” *Sensors*, vol. 22, no, 23, pp. 1-17, 2022.

[14] F. Jemili, and Bouras. “Intrusion detection based on big data fuzzy analytics,” In Open Data. *IntechOpen,* 2021.

[15] T.H. Morris, Z. Thornton, and I. Turnipseed. “Industrial control system simulation and data logging for intrusion detection system research,” *7th annual southeastern cyber security summit*, pp. 3-4, 2015.

[16] R. Vinayakumar, M. Alazab, K.P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman. “Deep learning approach for intelligent intrusion detection system,” *IEEE Access*, vol. 7, pp. 41525-41550, 2019.

[17] A. Brandao, and P. Georgieva. “Automatic log analysis to prevent cyber attacks,” *Advances in Intelligent Systems Research and Innovation*, pp. 1-25, 2022.

[18] R.K. Vigneswaran, R. Vinayakumar, K.P. Soman, and P. Poornachandran. “Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security," In *2018 9th International conference on computing, communication and networking technologies (ICCCNT)*, Bengaluru, India, 10-12 July 2018, pp. 1-6.

[19] N. Dionísio, F. Alves, P.M. Ferreira, and A. Bessani. “Cyberthreat detection from twitter using deep neural networks,” In *2019 international joint conference on neural networks (IJCNN)*, Budapest, Hungary, 14-19 July 2019, pp. 1-8. IEEE.

[20] M.A. Rassam, M. Maarof, and A. Zainal. “Big Data Analytics Adoption for Cybersecurity: A Review of Current Solutions, Requirements, Challenges and Trends,” *Journal of Information Assurance & Security*, vol. 12, no, 4, pp. 124-145, 2017.

[21] E. Kabir, J. Hu, H. Wang, and G. Zhuo. “A novel statistical technique for intrusion detection systems,” *Future Generation Computer Systems*, vol. 79, pp. 1-39, 2018.

[22] L. Ashiku, and C. Dagli. “Network intrusion detection system using deep learning,” *Procedia Computer Science*, vol. 185, pp. 239-247, 2021.

[23] G. Dhiman, and V. Kumar. “Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications,”*Advances in Engineering Software*, vol. 114, pp. 48-70, 2017.

[24] M. Sabahno, and F. Safara. “ISHO: improved spotted hyena optimization algorithm for phishing website detection,” *Multimedia Tools and Applications*, vol. 81, no. 24, pp. 34677-34696, 2022.

[25] R. Taj, A Machine Learning Framework for Host Based Intrusion Detection Using System Call Abstraction. Master’s Thesis, Dalhousie University, Halifax, NS, Canada, 2020.

[26] C.M. Nalayini, and J. Katiravan. “A new IDS for detecting DDoS attacks in wireless networks using spotted hyena optimization and fuzzy temporal CNN,” *Journal of Internet Technology*, vol. 24, no. 1, pp. 23-34, 2023.

[27] A. Alsaleh, and W. Binsaeedan. “The influence of salp swarm algorithm-based feature selection on network anomaly intrusion detection,” *IEEE Access*, vol. 9, pp. 112466-112477, 2021.

[28] A. Alzaqebah, I. Aljarah, O. Al-Kadi, and R. Damaševičius. “A modified grey wolf optimization algorithm for an intrusion detection system,” *Mathematics*, vol. 10, no. 6, pp. 1-16, 2022.

[29] R.O. Ogundokun, J.B. Awotunde, P. Sadiku, E.A. Adeniyi, M. Abiodun, and O.I. Dauda. An enhanced intrusion detection system using particle swarm optimization feature extraction technique. Procedia Computer Science, vol. 193, pp.504-512, 2021.

[30] G. Dhiman, and V. Kumar. “Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications,” *Advances in Engineering Software*, vol. 114, pp. 48-70, 2017.

[31] C.M. Nalayini, and J. Katiravan. “A new IDS for detecting DDoS attacks in wireless networks using spotted hyena optimization and fuzzy temporal CNN,” *Journal of Internet Technology*, vol. 24, no. 1, pp. 23-34, 2023.

[32] A.A. Almazroi, and N. Ayub. “Deep learning hybridization for improved malware detection in smart Internet of Things,” *Scientific Reports*, vol. 14, no. 1, pp. 1-18, 2024.

[33] J. Zhang, J.D. Peter, A. Shankar, and W. Viriyasitavat. “Public cloud networks oriented deep neural networks for effective intrusion detection in online music education,” *Computers and Electrical Engineering*, vol. 115, pp. 109095, 2024.

[34] S.A. Edalatpanah, F.S. Hassani, F. Smarandache, A. Sorourkhah, D. Pamucar, and B. Cui. “A hybrid time series forecasting method based on neutrosophic logic with applications in financial issues,” *Engineering applications of artificial intelligence*, vol. 129, pp. 107531, 2024.

[35] D.S. Vladislav. “Leakage Detection in Water Pipes: An Approach of Smart Water,” *Big Data and Computing Visions*, vol. 3, no. 1, pp. 8-14, 2023.

[36] C.C. Jorge, H.N. Jorge, M.R. Wendell, S.A. Edalatpanah, A.B. Shariq, S. Naz, J.C. Javier, and P.E. Gabriel. “Novel characterization and tuning methods for integrating processes,” *International Journal of Information Technology*, vol. 16, no. 3, 1387-1395. 2024.

[37] S. Mohammadi, N. Hemati, and N. Mohammadi. “Speech recognition system based on machine learning in persian language,” *Computational algorithms and numerical dimensions*, vol. 1, no. 2, pp. 72-83, 2022.

[38] M. Teimoori, H. Taghizadeh, J. Pourmahmoud, and Honarmand M. Azimi. “A multi-objective grey wolf optimization algorithm for aircraft landing problem,” *Journal of applied research on industrial engineering*, vol. 8, no. 4, pp. 386-398, 2021.

[39] S. Ali, Q. Li, and A. Yousafzai. “Blockchain and federated learning-based intrusion detection approaches for edge-enabled industrial IoT networks: A survey,” *Ad Hoc Networks*, vol. 152, pp. 1-31, 2024.

[40] M. Dirik. “Detection of counterfeit banknotes using genetic fuzzy system,” *Journal of fuzzy extension and applications*, vol. 3, no. 4, pp. 302-312, 2022.

[41] P. Ghasemi, H. Hemmaty, A. Pourghader Chobar, M.R. Heidari, and M. Keramati. “A multi-objective and multi-level model for location-routing problem in the supply chain based on the customer’s time window” *Journal of Applied Research on Industrial Engineering*, vol. 10, no. 3, pp.412-426, 2023.

[42] T. Wagner, A. Gepperth, and E. Engels. “A framework for the automated parameterization of a sensorless bearing fault detection pipeline,” *arXiv preprint arXiv:2303.08858,* pp. 1-8, 2023.

[43] M. Dirik. “Type-2 fuzzy logic controller design optimization using the PSO approach for ECG prediction,” *Journal of fuzzy extension and applications*, vol. 3, no. 2, pp. 158-168, 2022.

[44] F. Shahabi, F. Poorahangaryan, S.A. Edalatpanah, and H. Beheshti. “A multilevel image thresholding approach based on crow search algorithm and Otsu method,” *International Journal of Computational Intelligence and Applications*, vol. 19, no. 02, pp. 2050015, 2020.