Enhancing Cybersecurity by relying on a Botnet Attack Tracking Model using Harris Hawks Optimization

Ali Ibrahim Ahmed *
Al-Noor University,Mosul, Iraq.
AbdulSattar M. Khidhir †
Northern Technical University, Mosul, Iraq.
Shatha A. Baker ‡
Northern Technical University, Mosul, Iraq.
Omar I. Alsaif §
Northern Technical University, Mosul, Iraq.
Ibrahim Ahmed Saleh ¶
Mosul University, Mosul, Iraq

June 22, 2024

Abstract

A botnet attack is a major cybersecurity threat that involves coordinated control of a network of infected computers, enabling large-scale distributed denial of service (DDoS) attacks, malware spreading, and other cybercrime activities. Proactive security measures and advanced threat intelligence systems are essential to detect and mitigate these assaults. This paper proposes the Harris Hawks Optimization (HHO) algorithm, which employs exploration and exploitation techniques to find optimal solutions for analyzing botnet attack pathways. The proposed approach involves HHO as a feature selector for extracting features from anomalous network traffic. The algorithm's impact on botnet IP positioning performance is analyzed, considering different optimization modes and control center accuracy. The paper is organized into sections covering attack path establishment and analysis, system testing and verification, and a central leadership entity controls it [1]. Botnets are created based on the use of malicious software packages to infect important and sensitive devices in the network, thus making servers, computers, and Internet of Things devices vulnerable [2]. To detect these attacks and limit their impact requires many proactive security measures such as strong network security settings, regular software upgrades, etc. [3]. HHO is a powerful method that has the potential to solve many functional optimization problems and provides a suitable environment for engineering applications, as it mimics the exploration and exploitation phases during the foraging process of Harris Hawks [4]. A model based on HHO algorithm is proposed in this paper that has the ability to track and analyze bot attack paths by extracting a set of features during abnormal network traffic. The results were analyzed and their impact on the performance of robot networks was discussed, based on the use of different

experimental results. After configuring the network topology and determining the attack path based on HHO, the performance of the algorithm and its effectiveness in preventing IP addresses from being spoofed are verified. The results showed convergence in being able to correct attack paths and effective performance in repelling the interference of fake IP addresses.

Keywords: Botnet attack, Harris Hawks Optimization, zombie virus

1 Introduction

Botnet attacks pose a major threat to cybersecurity because they have the potential to allow criminals to launch large-scale DDoS attacks, launch spam campaigns, and engage in various forms of cybercrime. It is coordinated based on a network of infected devices called "zombies" or "bots", and

optimization modes, determining the IP position, and the extent of the influence of the control center's accuracy. The rest of the paper is organized as follows: Section 2 provides an overview of the HHO algorithm's concept and formulation, while Section 3 provides a discussion of the proposed materials and methods. The experiments and results analysis are presented in Section 4. Section 5 finally includes the conclusion.

2 Harris Hawks Optimization

This is an algorithm proposed in 2019 by Heidari et al and is a descriptive heuristic [5]. It draws inspiration from the

*Reasearch Assistant
†Department of Electronic Technologies Northern Technical University.
‡Department of Electronic Technologies Northern Technical University.
§Department of Electronic Technologies Northern Technical University.
¶3Department of Software, College of Computer and Mathematics.

predation behavior of Harris hawks, specifically their hunting technique of capturing prey, such as hares. In a variety of optimization tasks and problem domains, HHO outperforms other well-known methods [6].In this optimization algorithm, the preysymbolizes the search for the optimal solution, and the candidate solutions are represented by the Harris hawks. Figure 1 illustrates the two primary stages of the algorithm, the exploration phase, the algorithm focuses on discovering new potential solutions by exploring unknown regions of the search space. While, the exploitation phase aims to refine and improve the solutions that have shown promising results during the exploration phase. In order to arrive at an optimal solution, convergence requires striking an equilibrium between exploration and exploitation. Through this iterative process, HHO gradually guides the search towards the global minimum by emulating the predatory behavior of Harris hawks [7].

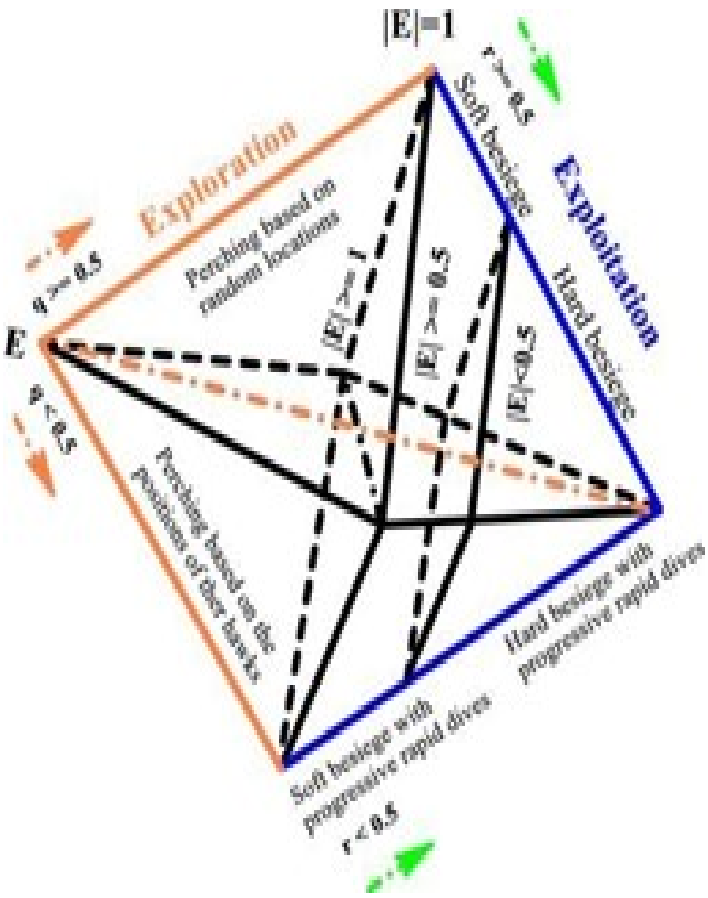

Figure 1: HHO Phases

2.1 Exploration Phase

Every Harris hawk in the population is considered a potential fix. During each iteration, every possible solution's fitness value is assessed in relation to the intended prey. In this case, exploitation refers to local research conducted within the area identified through exploration stage. Harris hawks initially wait, observing and evaluating the search space know by the upper bound (ub) and lower bound (lb) of the problem domain. They then engage in random searches for prey using two different strategies. The position update during the iteration is influenced by a probability parameter (q), determining the likelihood of a particular movement. The mathematical expression for this update process is utilized to guide the hawks in their exploration and exploitation efforts [8].

$$(t+1) = \begin{cases} X_{\text{rand}}(t) - r_1 \cdot |X_{\text{rand}}(t) - 2r_2 X(t)| & \text{if } q \geq 0.5 \\ (X_{\text{rabbitt}}(t) - X_m(t)) - r_3(LB + r_1(UB - LB)) & \text{if } q < 0.5 \end{cases}$$

LB and UB upper and lower bounds of variables; Xm represents the average position of the current hawk population; Xrand(t) represents a randomly selected hawk from the population; Xrabbit (t) rabbit position; X(t) current position of hawks; and r1, r2, r3, r4, and q are random numbers between (0,1), which are updated in each iteration. Eq. 4 is used to get the hawks' average position:

$$X(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t) \tag{2}$$

2.2 Exploration to Exploitation Transition

In swarm optimization algorithms like the HHO, maintaining a balance between exploration and exploitation is crucial for effective problem-solving. The Harris hawk is renowned for its flexible hunting tactics, since it can transition between various forms of predation based on the energy levels of its victims. If the prey is in a state of escape, its energy, which is symbolized by the symbol E, will gradually diminish. The HHO included the concept Escape energy to make the transition easy between the exploitation and exploration stages. The escape energy equation was relied upon in the algorithm to regulate this conversion process [9].

$$E = 2E_0(1 - t) \tag{3}$$

The symbol E stands for the prey's escape energy, the symbol T for the maximum number of repeats, and the symbol E0 for the prey's initial energy condition [10].

2.3 Exploitation Phase

When Harris' hawks execute a surprise pounce, they target prey that was identified during the preceding phase. Because prey frequently flee from hazardous circumstances, numerous pursuit techniques have developed. In the HHO, four potential tactics are put forth to simulate the offensive stage [11-12].

• Soft Besiege Because of the energy it has, the prey in this scenario can flee when $|e| \geq 0.5$ and $r \geq 0.5$. The Harris hawk then relies on a soft siege strategy for the purpose of gradually depleting the energy of the prey. The primary objective of this strategy is to select the optimal position from

which to launch raids and dives, effectively capturing the prey. The following equation controls the location update during the soft siege strategy [13]:

$$X(t+1) = \Delta X(t) - E\,|JX_{rabbit}(t) - X(t)| \tag{4}$$

$$\Delta X(t) = X_{rabbit}(t) - X(t) \tag{5}$$

$$X(t+1) = \Delta X(t) - E\,|JX_{rabbit}(t) - X(t)| \tag{4}$$

$$\Delta X(t) = X_{rabbit}(t) - X(t) \tag{5}$$

where $\Delta X(t)$ is the variation between the rabbit's position vector and its current location in the iteration, and $J = 2(1 - r_5)$ indicates the rabbit's random leap strength throughout the escape phase. During each repetition, the J value fluctuates at random to mimic the characteristics of rabbit motions [**?**].

- **Hard Besiege**

The prey's energy is greatly reduced when —E— ¡ 0.5 and r ¿ 0.5, at which point the Harris hawks execute a surprise pounce attack. At this stage, the Harris hawks no longer engage in extensive encircling maneuvers but instead make a sudden represented by Z in this equation, while the Levy function is represented by LF(d). A random vector of size 1 * D is called S. Y denotes the position ascertained by the gentle siege approach. The following formula is used to calculate the Levy function:

$$LF(x) = 0.01 \times \left(\frac{u \times \sigma}{|v|^{1/\beta}}\right) \tag{8}$$

Here, σ is a calculated value, β is a constant with a value of 1.5, and v, u are random numbers that range from zero to one. The HHO method simulates the prey's escape behavior by adding a stochastic element to the location updates through the use of the Levy function. This enables the Harris Hawks to modify their positions in response. This dynamic location update approach improves convergence towards optimal solutions and the algorithm's ability to catch elusive prey [**?**].

- **Hard besiege with progressive rapid dives**

The prey still has a chance to escape when —E— ¡ 0.5 and r ¡ 0.5, but its escape energy E is insufficient. The Harris hawk uses a hard besiege tactic in this instance, which is typified by increasingly quick dives. This approach involves initiating a hard besiege prior to launching an attack, gradually decreasing the distance between

and decisive attack [15]: The position is updated using the following equation

$$X(t+1) = X_{rabbit}(t) - E\,|\Delta X(t)| \tag{6}$$

- **Soft besiege with progressive rapid dives**

The HHO method models prey escape patterns and leapfrog movements statistically by utilizing the levy flight (LF) idea. LF imitates the irregular, sudden, and fast dives of hawks around the fleeing prey, as well as the true zigzag misleading maneuvers of rabbits during the escaping phase. The hawks circle the rabbit quickly as a team, making multiple attempts to adjust their position and trajectory. Real data from various competitive scenarios in nature provide evidence for this mechanism [17]. Hawks use the following rule to choose their next step when executing a gentle besiege:

$$Z = Y + S + LF(d) \tag{7}$$

The Harris Hawks' updated position vector is the hawk and the prey[19]. The position update equation governing this phase is as follows:

$$X(t+1) = \begin{cases} Y : X_{rabbit,t} - E\left|JX_{rabbit,t} - X_{m,t}\right| & \text{if } F(Y) < F(X(t)) \\ Z : Y + S \cdot LF(D) & \text{if } F(Z) < F(X(t)) \end{cases} \tag{1}$$

The Harris hawk executes a hard besiege with progressive rapid dives, continuously adjusting its position towards the prey to increase the chances of capturing it successfully. By employing different attack mechanisms based on the prey's escape energy and the factor r, the HHO algorithm effectively solves optimization problems [20-23].

3 Materials and Methods

3.1 Network Topology

Network topology refers to the arrangement and connectivity of nodes in a network. Traditionally, network topology has been created using either completely random or completely regular methods of layout and connection paths. However, in order to simulate real network environments, this research utilizes a random graph generator. The topology design involves placing v nodes within a square of size $M \times M$ and then randomly connecting them with a certain probability to establish the network topology. The probability of connecting two nodes, i and j, is determined using a formula that takes into account their Euclidean distance and a maximum possible distance between nodes. By adjusting the control variables η and γ within specific intervals, the average number of nodes connected and the average distance between node connections can be influenced. Unlike previous network topologies used for IP traceability, where the attacker and victim were positioned on the periphery, this study randomly selects the locations of the two ends to closely resemble real network scenarios. In the context of this paper, the establishment of the attack path analysis model involves generating a network topology between the attacking node and the victim node by randomly placing nodes within a square region and connecting them based on the probability equation (10).

$$P_{ij} = \frac{1}{1 + e^{-(d_{ij} - \eta)/\gamma}} \tag{2}$$

The selection of the attacking and victim nodes within the topology is also randomized to reflect real-world scenarios.

3.2 Reconstruction and statistical characteristics of attack path

The reconstruction of the attack path involves generating 30 sets of random topologies with different numbers of nodes. Monte Carlo Simulation is employed to simulate hackers performing one-to-one attacks on the victim, generating attack paths. The path, nodes, and number of packets are recorded as judgment criteria for backtracking the attack path. The reconstruction component of the attack path utilizes the HHO algorithm. Equation (11) serves as the state transition rule for path exploration, while formulas (6) and (7) are used to update the path. To effectively guide the Harris search along the correct attack path, certain research parameters are set.

$$p_{ij} = \sum_{k \in \text{neighbor}(ij)} \frac{[r_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{ij \cdot \eta_{ij}(t)} [r \cdot \eta]^{\beta} \tag{11}$$

3.3 Detection of fake IP

The detection of fake IP addresses involves identifying their distinct behavior compared to normal IP addresses. To modify the HHO algorithm to prevent counterfeiting IP, the following steps are taken:

• **Non-existent fake IP:** Internet Control Message Protocol (ICMP) command tracers are used to detect the attacker and victim by sending requests to all nodes along the path. Alternatively, network security and intrusion detection policies are implemented using the Challenge Handshake Protocol (CHAP) authentication process can be employed. If there is no response or the authentication fails, it is determined to be a fake IP [24]. • **Existence of fake IP:** The HHO algorithm is used for path backtracking, and the "Amount of attack information" at each node is considered for determining the presence of counterfeit IP. The detection of counterfeit IP can be categorized into two main categories: a. Inconsistent path: If there is no direct link between a node in the path and the destination (predicted attacker), or if the connection path leads to an unreachable node, it can be avoided depended on HHO characteristics. b. Abnormal amount of node attack information: This is primarily determined based on a sharp drop in the number of attacks. While the attack volume is set during the attack, in a real network environment, it is challenging to determine an upper-bound threshold for attack volume. Therefore, a fixed threshold should not be used to judge abnormal attack volume. Instead, the ratio of node attack amounts is examined. If the attack amount at a particular Harries-passing node is lower than that of the previous path, a penalty function is applied by multiplying the amount of node attack by a threshold and adding it to the cost of the path node. For example, if the threshold is set to 0.4 and the previous path node has an attack volume of 1000, but the selected path node has an attack volume of 350, it is considered lower than 1000×0.4 = 400. Therefore, if the algorithm determines that this point is an attacker, it is classified as a fake IP or an incorrect attacker. Otherwise, a penalty function is added to the exploration at this point.

$$r_{ij}(t+1) = (1-\rho) \cdot r_{ij}(t) + \mu_{ij} \tag{12}$$

$$\mu_{ij} = \begin{cases} \rho \Delta r & \text{if } \Delta r > 0 \\ 0 & \text{if } \Delta r \leq 0 \\ \text{if AttPackets}_i - \text{AttPackets}_j & \text{otherwise} \\ -\rho \Delta r & \text{otherwise} \end{cases}$$

Where μ_{ij} is the penalty function and AttPackets_i is the number of attack packets transmitted by node i. Its value depends on the number of attack packets collected. This adjustment in the algorithm helps in identifying nodes where the attack activity significantly decreases, indicating a wrong path or a fake node. The control threshold can be determined using a cubic spline function, where the cubic-spline interpolation formula predicts a reasonable value for AttPackets_m, and ± 2 standard deviations represent a 95% confidence interval. The upper and lower confidence limits $[\lambda, \omega]$ can be calculated using Equation (13):

$$[\lambda, \omega] = \text{AttPackets}_m \mp 2\sigma_{\text{AttPackets}}$$

If the amount of attack information falls below the lower limit ω, it can be determined that the attacker is a fake IP.

4 Testing and Verification of Proposed Approach

The simulation network topology was generated using a random graph, where η and γ are weight variables representing two types of control topologies. The value of γ plays a crucial role in controlling the average distance of the path. Increasing γ results in stronger connections between nodes, thereby reducing the average distance. The concept of average distance is based on network models. To ensure consistency with real network environments and avoid excessively large path distances, a parameter setting similar to other studies [10] fixes $\gamma = 0.1$. Conversely, the average number of connections for every node is determined by the value of η. Higher values of η increase the average connections, while too low values lead to insufficient paths, potentially causing excessive bottlenecks and reducing the number of feasible paths. Tables 1 to 3 provide insights into the influence of these parameter settings on the network topology. Figure 2 showcases a series of topology diagrams generated by setting the number of nodes to 100 and configuring $\gamma = 0.1$ and $\eta = 1.5$. In this figure, point A represents the randomly generated victim end, while point B represents the attack end.

The attack path reconstruction part is mainly built by the HHO algorithm. This algorithm performs a backtracking of the attacker based on the generated topology and monitors the

Table 1: Topology generates average data (number of nodes=50)

Topology setting	Max. No. of connections	Min. No. of connections	Average No. of connections	Maximum Distance(m)	Average Distance
η=0.5	6.30	0.00	1.2	46.64	6.12
η=1.0	5.85	0.05	2.39	59.77	8.61
η=1.5	7.75	0.15	3,47	59.94	8.71
η=2.0	10.15	0.55	4.47	59.11	8.72
η=2.5	12.05	0.70	5.52	61.98	10.01

Table 2: Topology generates average data (number of nodes = 100)

Topology setting	Max. No. of connections	Min. No. of connections	Average No. of connections	Maximum Distance(m)	Average Distance
η=0.5	6.65	0.00	2.37	61.88	8.77
η=1.0	11.35	0.35	5.10	64.17	8.60
η=1.5	13.85	1.30	6.95	68.39	8.61
η=2.0	18.05	1.75	9.15	70.11	8.75
η=2.5	21.05	2.70	11.05	76.10	9.43

Table 3: Topology generates average data (number of nodes=200)

Topology setting	Max. No. of connections	Min. No. of connections	Average No. of connections	Maximum Distance(m)	Average Distance
η=0.5	8.35	0.05	3.44	67.12	8.77
η=1.0	14.95	0.85	7.10	75.24	8.63
η=1.5	19.95	2.25	10.52	76.01	8.71
η=2.0	24.50	3.00	13.82	76.88	8.81
η=2.5	28.65	4.70	16.44	80.85	9.49

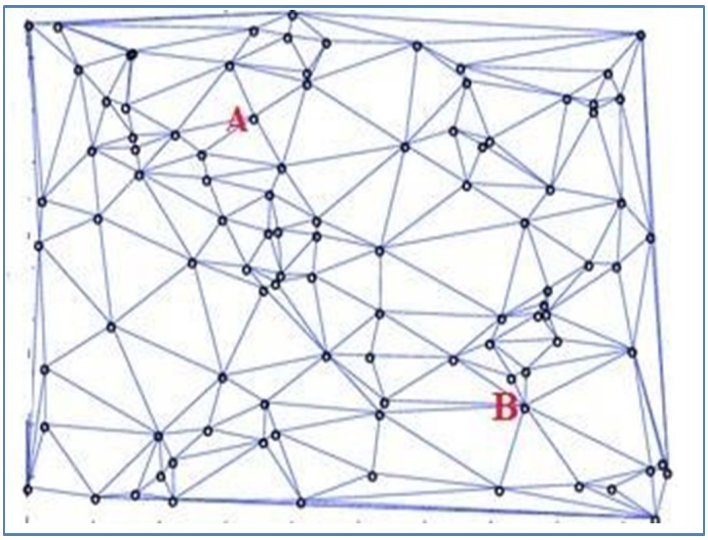

Figure 2: Simulation topology (100 nodes)

parameters. α and changes on the attack path convergence. The Harris size of 30, a decay rate of 0.5, α set to 0.7, β set to 1.3, and performs 27 generations of execution. The HHO algorithm uses the number of transmissions of the assault packet as its search criterion. Additionally, the inference criterion for finding the victim is utilized to judge the selection of the attack path regarding the statistical characteristics of the attack.

The judgment of counterfeit IP can be classified into two categories: path inconsistency and abnormal node attack information. Among them, if there is no direct path between node in path and predicted attacker or node is unreachable, it can be calculated by HHO calculation. The Hurries of the law cannot reach and avoid being Fake IP spoofing; and the abnormal amount of node attack information is part of this study. It is verified by equation (12) and equation (13), and the threshold is set as 0.5. For attack detection part of counterfeit IP, this research is based on. There are 20 sets oftopologies with different numbers of nodes to execute 5 times each, select Set any node in the topology on the non-attack path as the node of the fake IP Point to test whether HHO algorithm will be counterfeited when searching for a path, the attacking end interferes, and correct attacking end cannot be found. Figures 3 illustrate performance of HHO algorithm modes simulation with node size of 100 points that which used in analysis with execution is perform for 30 times, as seen the criterion of execution performance is based on the probability of the average number of attack packets on the path searched by the algorithm. The performance is when increase number of executed that increase probabilities fake IP. The different topology sizes are used for analysis, Figure 3 and Figure 4 illustrate execution algebra and average of topological size 100 and 200 nodes respectively. Searching the relationship graph of the error rate, it can be observed that as the execution algebra increases .Therefore, it can be observed from Figure 4 and Figure 5 that regardless of the topology, it can converge to a search error rate, and the topology is smaller Because of feasible solution is less, which will increase the probability of resetting. The test of counterfeit IP through this step shows that at the algorithm initial stage of the execution, the algorithm not be able to find correct attacker due to influence of the counterfeit IP. However, as the execution algebra increases, the algorithm will still refer to the correct attack path. Attack the number of packets, and pull the search path back to the correct attack path. It is evident that the HHO algorithm with adjustment of formulas (11) and (12) is effective in preventing counterfeit IP It has a very high implementation effect.

5 Conclusion

This paper analyzes the botnet attack path traceability model based on the HHO. Aiming at the shortcomings of the HHO algorithm that is not easy to escape after converging to the best solution in the area. To analyze the attack path of the botnet, and explore the computing resources required by the botnet control center in reverse traceability, a completely

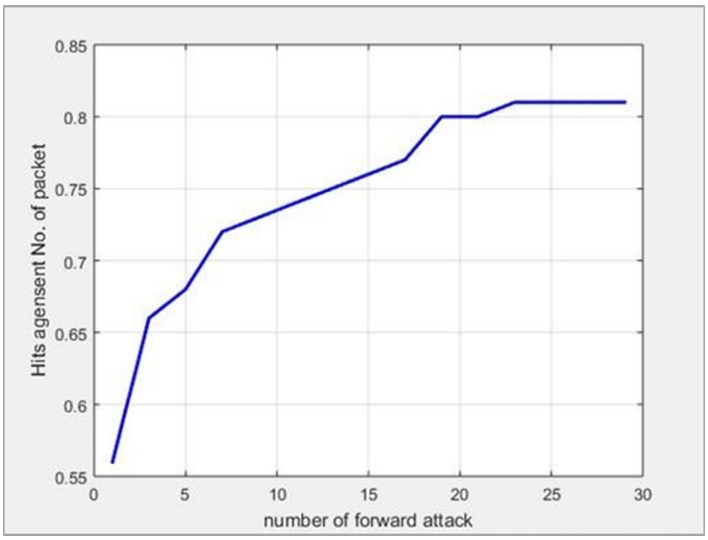

Figure 3: HHO probabilities for explored IP fake

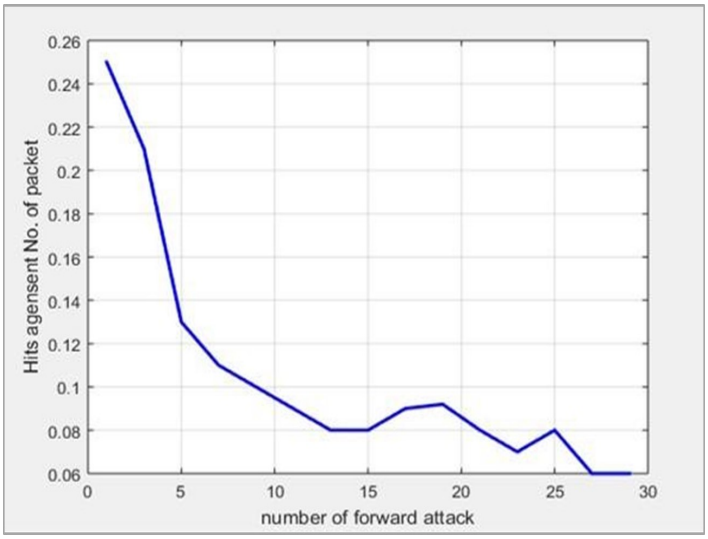

Figure 4: Algorithm search error rate of fake IP (100 nodes)

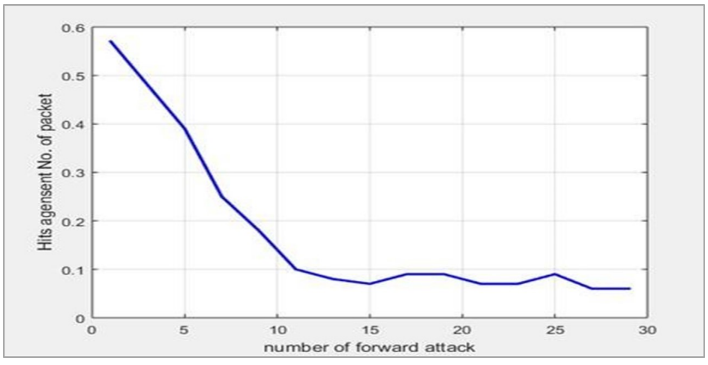

Figure 5: Algorithm search error rate of fake IP (200 nodes)

random communication method is used. The proposed approach has proven effective in preventing fake IP interference in the exploration and exploitation phases, and it has the ability to adapt and respond to the dynamic nature of bot attacks, strengthening cybersecurity measures against advanced cyber threats, as proven by experimental results. The paper also showed the importance of making modifications to the algorithm based on a set of information about the attack package to be able to identify nodes with significantly low activity, which results in improving the accuracy of the attack path.

References

[1] Sokkalingam, Sumathi, and Rajesh Ramakrishnan. ”An intelligent intrusion detection system for distributed denial of service attacks: A support vector machine with hybrid optimization algorithm based approach.” Concurrency and Computation: Practice and Experience 34.27 (2022): e7334.

[2] Baker, S. A., Nori, A. S. “Internet of things security: a survey”. Advances in Cyber Security: Second International Conference, ACeS 2020, Penang, Malaysia, December 8- 9, 2020, Revised Selected Papers 2. Springer Singapore, 2021.

[3] Baker, S. A., Nori, A. S. “A secure proof of work to enhance scalability and transaction speed in blockchain technology for IoT”. In AIP Conference Proceedings (Vol. 2830, No. 1). AIP Publishing, 2023.

[4] Alabool HM, Alarabiat D, Abualigah L, Heidari AA. Harris hawks optimization: a comprehensive review of recent variants and applications.Neural Comput Appl. 2021;33(15):8939-8980.

[5] Heidari, Ali Asghar, et al. ”Harris hawks’ optimization: Algorithm and applications.” Future generation computer systems 97 (2019): 849-872.

[6] H. Moayedi, A. Osouli, H. Nguyen and A. Rashid, “A novel Harris hawks’ optimization and k-fold cross-validation predicting slope stability”, Engineering with Computers, 2019.

[7] H. Moayedi, M. Abdullahi, H. Nguyen and A. Rashid, “Comparison of dragonfly algorithm and Harris hawk’s optimization evolutionary data mining techniques for the assessment of bearing capacity of footings over two-layer foundation soils”, Engineering with Computers, 2019.

[8] Alabool, Hamzeh Mohammad, et al. ”Harris hawks’ optimization: a comprehensive review

of recent variants and applications.” Neural Computing and Applications 33 (2021): 8939-8980.

[9] Shehab, Mohammad, et al. ”Harris hawks optimization algorithm: variants and applications.” Archives of Computational Methods in Engineering 29.7 (2022): 5579-5603.

[10] Li, ChenYang, et al. ”Enhanced Harris hawks optimization with multi-strategy for global optimization tasks.” Expert Systems with Applications 185 (2021): 115499.

[11] Al-Betar, Mohammed Azmi, et al. ”A hybrid Harris Hawks optimizer for economic load dispatch problems.” Alexandria Engineering Journal 64 (2023): 365-389.

[12] Dhawale, Dinesh, Vikram Kumar Kamboj, and Priyanka Anand. "An improved Chaotic Harris Hawks Optimizer for solving numerical and engineering optimization problems." Engineering with Computers 39.2 (2023): 1183-1228.

[13] Fan, Qian, Zhenjian Chen, and Zhanghua Xia. "A novel quasi-reflected Harris hawks optimization algorithm for global optimization problems." Soft Computing 24 (2020): 14825-14843.

[14] ÇetınbaŞ, İpek, Bünyamın Tamyürek, and Mehmet Demırtaş. "The hybrid Harris hawks optimizer-arithmetic optimization algorithm: A new hybrid algorithm for sizing optimization and design of microgrids." IEEE Access 10 (2022): 19254-19283.

[15] Li, Wenyu, Ronghua Shi, and Jian Dong. "Harris hawks optimizer based on the novice protection tournament for numerical and engineering optimization problems." Applied Intelligence 53.6 (2023): 6133-6158.

[16] Yüzgeç, Ugur, and Meryem Kusoglu. "Multi-objective harris hawks optimizer for multiobjective optimization problems." BSEU Journal of Engineering Research and Technology 1.1 (2020): 31-41.

[17] Fan, Qian, Zhenjian Chen, and Zhanghua Xia. "A novel quasi-reflected Harris hawks optimization algorithm for global optimization problems." Soft Computing 24 (2020): 14825-14843.

[18] Gupta, Shubham, et al. "Opposition-based learning Harris hawks optimization with advanced transition rules: Principles and analysis." Expert Systems with Applications 158 (2020): 113510.

[19] Zhang, Yang, Xizhao Zhou, and Po-Chou Shih. "Modified Harris Hawks optimization algorithm for global optimization problems."Arabian Journal for Science and Engineering 45 (2020): 10949-10974.

[20] Maray, A. H., Alsaif, O. I., & Tanoon, K. H. (2022). Design and implementation of low- cost medical auditory system of distortion otoacoustic using microcontroller. J. Eng. Sci. Technol, 17(2), 1068-1077.

[21] Mohammed, N. L., Aziz, M. S., & AlSaif, O. I. (2020). Design and implementation of robot control system for multistory buildings. TELKOMNIKA (Telecommunication Computing Electronics and Control), 18(5), 2682-2689.

[22] Thanoon, K. H., Q Hasan, S.,& I Alsaif, O. (2020). Biometric information based on distribution of arabic letters according to their outlet. International Journal of Computing and Digital Systems, 9(5), 981-991.

[23] Yuan, Yongliang, et al. "An adaptive instinctive reaction strategy based on Harris hawks optimization algorithm for numerical optimization problems." AIP Advances 11.2 (2021).

[24] Baker, S. A., Mohammed, H. H., & Alsaif, O. I. (2024). Docker Container Security Analysis Based on Virtualization Technologies. International Journal for Computers & Their Applications, 31(1).

BIOGRAPHY / BIOGRAPHIES

Ali Ibrahim Ahmad earned his bachelor's and master's degrees from the University of Mosul, Faculty of Computer Science and Mathematics, Department of Software Engineering in 2019 and 2022 respectively. Currently, he works as a Lecturer at the Al-Noor University College, Mosul, Iraq. His research areas encompass artificial intelligence applications, security, and image processing. Email: ali.ibrahim@alnoor.edu.iq

AbdulSattar M. Khidhir (Born 9th Jan. 1959) is an assistant professor at Electronics Technology Department - Mosul Technical Institute - Northern Technical University in Iraq. He obtained his B.Sc. (1981) and M.Sc. (1989) both in Electronics and Communications Engineering from University of Mosul. His Ph.D. (2000) was obtained in Communications Engineering from University of Mosul too. He supervised many Ph.D. and M.Sc. theses in different scientific and engineering areas. He was a member of scientific committees for many Ph.D. and M.Sc. students. He published many researches in various fields of science and engineering (see google scholar). He reviewed many scientific papers for journals and conferences. Email : abdulsattarmk@ ntu.edu.iq, abdulsattarmk@gmail.com

Shatha A. Bakr has a Bachelor's degree in Computer Science from the University of Mosul, which she obtained in 1997. In 2013, she earned a master's degree in Computer Science from the same university. Later, in 2022, she completed her Ph.D. from the University of Mosul. Dr. Bakr worked as a Lecturer at the Northern Technical University in Mosul, Iraq. Her research interests encompass mobile phone programming, information security, multimedia communications, and artificial intelligence.

Omar I. Alsaif is currently a lecturer in the Mosul Technical Institute/ Northern Technical University in Mosul, Iraq. He received his B.Sc. in electrical engineering from the University of Mosul in 1992. In 2005 and 2018, he obtained his M.Sc. and Ph.D. degrees in Electronics and Microelectronic Engineering from Mosul University, respectively. His research interests encompass microelectronic and solid-state systems, renewable energy, and nanotechnology devices. Email: omar.alsaif@ntu.edu.iq.

Ibrahim Ahmed Saleh was born in Mosul - Iraq in 1963. He received his MSc. degree (in signal and image processing) from the University of Mosul, Iraq in 2003 and in 2013 he received his PhD inartificial techniques and computer networking from Mosul University. He became professor in 2021. From 1997 to 2005, he worked at computer center in Mosul University/Iraq. Currently he is lecturer at the Dept. of Software Engineering, College of Computer Sciences and Math, and University of Mosul, Iraq. He can be contacted at email: i.hadedi@uomosul.edu.iq.