Received: February 21, 2024. Revised: April 6, 2024.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Collaborative Attackers Detection and Route Optimization by Swarm Intelligent-based Q-learning in MANETs

Ammaiyappan Kavitha[1]* **Valavandan Srinivasan Meenakshi[1]**

[1]*PG and Research Department of Computer Science, Chikkanna Government Arts College, Affiliated to Bharathiar University, Tiruppur - 641602, Tamil Nadu, India*
*Corresponding author's Email: kavivksphd@gmail.com

Abstract: Mobile Adhoc Networks (MANETs) are vulnerable to various attacks such as Black Hole Attack (BHA), Gray Hole Attack (GHA), and Wormhole Attacks (WHA). While researchers have focused on detecting and mitigating individual attacks, protection against collaborative attacks is limited. Therefore, this article introduces a new Tunicate Swarm Optimization Q-learning-based Collaborative Attacker Detection Algorithm (TSOQCADA) to identify and prevent collaborative attackers like BHA, GHA, and WHA, thereby improving routing efficiency. This algorithm utilizes feedback about node properties such as energy, reputation, buffer space, transmission delay, and packet transfer rate from all nodes to determine efficient packet routing. In the TSOQ-learning algorithm, the TSO is adopted to set the Q-table values, resulting in faster convergence of Q-learning. First, a Q-table with prior knowledge is trained to enhance searchability. Additionally, a novel selective search mechanism is adopted to improve exploration efficiency and reduce unwanted explorations by considering the correlation between current and target locations. Furthermore, a nonlinear function is designed to achieve a tradeoff between search and use abilities in Q-learning, dynamically changing ε value in the ε-greedy method according to the number of iterations. Thus, the TSOQ-learning can efficiently obtain a routing path by isolating collaborative attackers with low reputation values. Simulation results show that the TSOQCADA achieves a Packet Delivery Ratio (PDR) of 94.8%, Packet Loss Rate (PLR) of 5.2%, energy consumption of 2.53J energy/packet, throughput of 355Kbps, and End-to-End (E2E) delay of 35ms for a network of 100 nodes with 20 malicious nodes in MANET, outperforming the Efficient Trust-based Routing Scheme (ETRS), Hybrid Trust-based Reputation Mechanism (HTRM) and Deep Neural Learned Projective Pursuit Regression-based Watchdog Malicious Node Detection and Isolation (DNLPPR-WMNDI) algorithms.

Keywords: MANET, Routing, Collaborative attacks, Reputation, Q-learning, Path selection, Tunicate swarm optimization.

1. Introduction

MANET runs as a point-to-point transmission utilizing self-managing and self-configuring nodes. It operates without infrastructure [1-2]. It is ideal for critical applications like combat and emergency response [3]. Routing protocols in MANETs involve four major conventions [4]: proactive, reactive, hybrid, and geographic. Proactive protocols update routing tables periodically [5]. Reactive protocols [6] establish routes on-demand. Hybrid protocols merge proactive and reactive elements [7]. Geographic protocol uses location information [8]. However, these protocols may face challenges from external or internal nodes that may disrupt packet routing [9].

1.1 Collaborative attackers in MANET

Collaborative attackers in MANET are malicious nodes that work together to compromise network security, disrupt communication, compromise data integrity, or gain unauthorized access. Some common types of attacks are discussed below.

BHA: A complete packet drop assault occurs when a BHA node sends a fake Route Request (RREQ) to bait data traffic (See Fig. 1). The BHA node accepts the RREQ packet and sends a fake

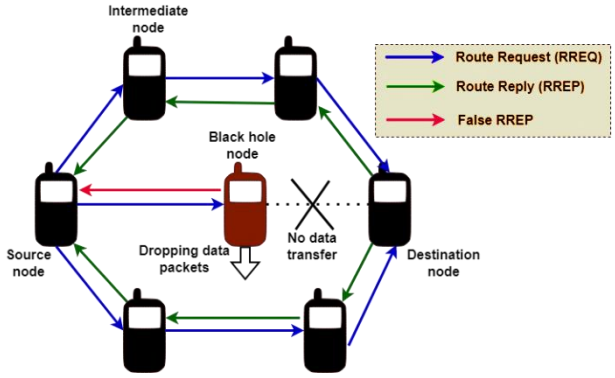

Figure. 1 Packet dropping in BHA scenario

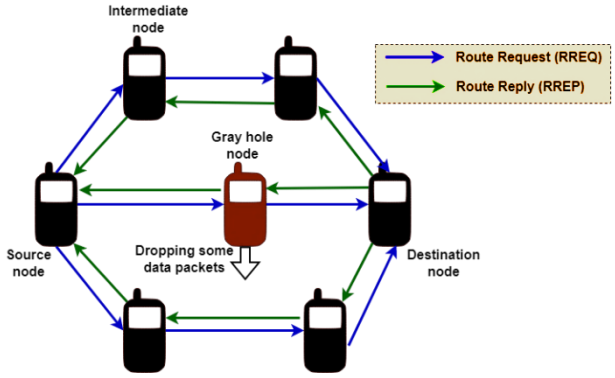

Figure. 2 Partial packet dropping in GHA scenario

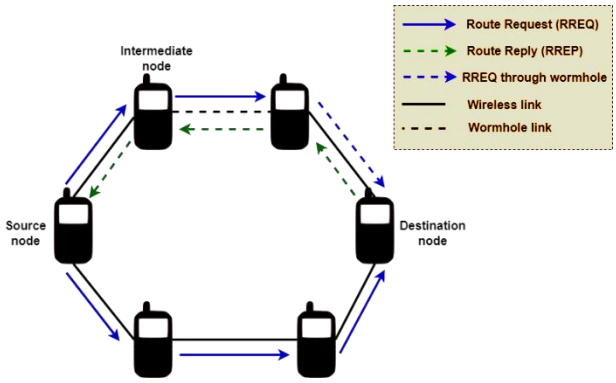

Figure. 3 WHA scenario

Route Reply (RREP) packets, reducing network throughput [10].

GHA: Partial packet drop attack, an extension of BHA, involves data packet drops (see Fig. 2) during transmission, making GHA nodes hard to detect [11].

WHA: In WHAs, attackers intercept and tunnel packets to replay them (see Fig. 3), which can be harmful in reactive protocols. This allows the attacker to control the discovered routes and potentially launch network attacks [12].

1.2 Problem statement

Scholars have proposed various protocols to detect MANET attacks, but most focus on single types like either BHA, WHA, or GHA. Detecting collaborative attacks is difficult. Researchers are exploring security mechanisms [13-14] like intrusion detection systems, reputation-based models, and secure routing protocols to address this. Cooperation among network nodes in sharing information can also help detect and mitigate these threats. Ongoing research is essential to keep up with evolving attacks and network security.

1.3 Major contributions of the paper

This manuscript presents a new CADA scheme for MANETs, using Q-learning and swarm intelligence. The goal is to identify and prevent collaborative attackers such as BHA, GHA, and WHA in the network, leading to more efficient routing. The key contributions of this study include:

1. A new TSOQCADA is proposed to identify and prevent collaborative attackers. This is achieved by taking into account feedback on node properties such as energy, reputation, buffer space, transmission delay, and packet transfer rate from all nodes to determine efficient packet routing.
2. The TSOQ-learning algorithm utilizes the TSO to set the Q-table values, resulting in faster convergence of Q-learning. Before the exploration stage, the algorithm learns from previous experience to enhance its efficiency. Additionally, a novel selective search mechanism is adopted to improve exploration efficiency and reduce unwanted explorations by considering the correlation between current and target locations.
3. Furthermore, a nonlinear function is designed to achieve a tradeoff between search and use abilities in Q-learning, dynamically changing ε value in the ε-greedy method according to the maximum iteration. This allows the TSOQ-learning algorithm to efficiently obtain a routing path by isolating collaborative attackers with low reputation values.
4. The TSOQCADA is evaluated through simulations, comparing its performance with conventional routing algorithms in the presence of collaborative attackers. The results demonstrate that the TSOQCADA outperforms other algorithms regarding different network QoS measures.

1.4 Outline of the paper

The rest of the article is structured as follows: Section 2 studies related works. Section 3 outlines the TSOQCADA in MANETs. Section 4 demonstrates

its performance effectiveness. Section 5 précises the study and offers ideas for further enhancements.

2. Literature survey

Various routing protocols have been developed to identify and counteract malicious nodes such as BHA, GHA, and WHA [15]. This section reviews recent studies related to these protocols in MANET.

In [16], the Gray Wolf Optimization (GWO) was used to detect BHA and GHA based on the node's trust in wireless adhoc networks. But the throughput was low due to the presence of malicious nodes in some cases. In [17], the ETRS was created to address malicious behavior in MANET. However, it was found that the E2E delay increased with node density.

In [18], a new technique was proposed for MANETs to detect BHAs using K-Nearest Neighbor (KNN) for grouping and fuzzy inference for CH selection. The algorithm assesses the trust level of each node using beta distribution and Josang mental logic, selects CH based on reputation and remaining energy. However, the throughput was low due to some nodes with high trust values behaving maliciously, leading to packet loss.

In [19], a 2-level feedback-based trust strategy was applied to identify and isolate cooperative blackmailing nodes. However, it had low PDR and high PLR when the number of nodes was increased. In [20], the HTRM was developed to find the best-trusted path and detect misbehaving nodes based on trust values. However, the throughput and energy consumption values were not effective in detecting collaborative attacks. In [21], an AODV-based defense strategy was proposed to prevent BHAs in MANET. However, low throughput was caused by inaccurate detection of malicious nodes.

In [22], a lightweight anomaly detection system was developed using a Support Vector Machine (SVM) to identify BHAs. But it was not suitable for detecting collaborative attacks involving more than one attacking node, as the simulation was limited to seven nodes and one attacker. This results in low PDR and high PLR in large-scale networks.

In [23], the DNLPPR-WMNDI scheme was used, which selects adjacent nodes using DNLPPR and establishes routes for multicasting. The WMNDI identifies malicious nodes based on information exchange periods and isolates them from the network. However, the unstable routes have a negative impact on the throughput, PDR, and energy consumption. The literature suggests that existing algorithms are not effective at identifying collaborative attackers in MANETs. This leads to unstable routes and poor network performance in terms of PDR, PLR, throughput, E2E, and energy consumption. So, a new reputation-based routing approach is proposed using Q-learning to detect and isolate collaborative attackers in MANETs.

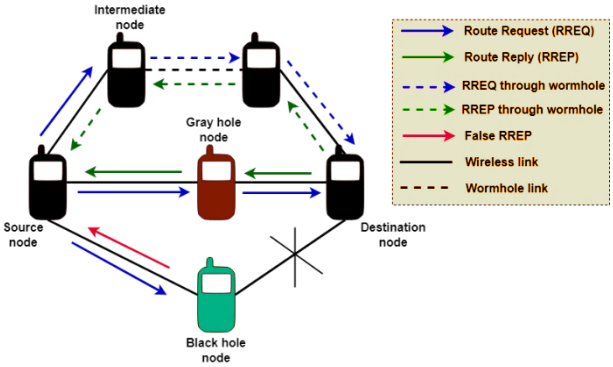

Figure. 4 MANET with collaborative attackers' scenario

3. Proposed methodology

This section explains the TSOQCADA in MANET. Table 1 presents the lists of notations used in this study.

3.1 Network model

Consider the MANET topology illustrated in Fig. 4, which includes normal and collaborative attackers, namely BHA, GHA, and WHA nodes. MANET is represented as an undirected graph $G = (N, E)$, where N, E denote the collection of nodes (mobile devices) and edges (wireless connections), respectively. When two nodes a and b have an edge $e(t) = (a, b) \in E$, a and b can interact immediately with each other in interval t.

All nodes use Global Positioning System (GPS) to define their position, broadcasting hello packets and receiving feedback to locate adjacent nodes, allowing them to interact with each other. A set of nodes that node a can interact with at t is called $adjacent_t(a)$.

Malicious nodes can disrupt the network by dropping packets, wasting energy, and degrading its lifetime, so it's crucial to identify and isolate suspected malicious nodes during transmission.

3.2 Energy model

The Friis free-space formula in Eq. (1) is used to compute the transmission power between nodes.

$$P_r(d) \propto \frac{\lambda^2}{4\pi d^2} P_t \tag{1}$$

In Eq. (1), P_t and P_r denote the transmitted and received powers in Watts (W), correspondingly, d is

Table 1. Lists of notations

Notations	Description
G	Undirected graph
N	Collection of nodes
E	Collection of edges
a, b	Two nodes
t	Time
$e(t)$	Edge between a and b at t
$adjacent_t(a)$	Set of nodes that node a can interact with at t
P_t, P_r	Transmitted and received powers, correspondingly
d	Distance between the transmitter and receiver
λ	Radio frequency range
C_{ab}	Standardized energy utilization
$P_{max}(a)$	Highest transmission power of a
$P_{t-min}(a, b)$	Lowest transmission power between a and b
$e_{level}(a)$	Energy level of a
$e_{rem}(a)$	Remaining energy of a
$e_{ini}(a)$	Initial energy of a
Rep_{ini}	Initial value of reputation
$Rep(a)$	Reputation of a
$Rep_{old}(a)$	Old reputation value of a
$Rep_{new}(a)$	Reputation that a receives according to the outcomes of transmission
α	Weight between previous and modified reputations
$D(a, b)$	Euclidean distance between a and b
$dest$	Destination
$D(dest, b)$	Euclidean distance between the $dest$ and b
TTL	Packet's Time-To-Live value
T_{ab}	Duration taken to forward a packet between a and b
β	Weight value
$ANode_t(a)$	Remaining set of adjacent nodes for a after isolation
$Rep_{threshold}$	Reputation threshold
s_t	Present state
u_t	Action executed in s_t
r_{t+1}	Reward obtained after s_t is performed
s_{t+1}	Successive state
γ	Discount factor
$\pi(s_t, u_t)$	Routing policy
R	Total discounted payoff
$Q_a(dest, b)$	Q-value while a transmits a packet to $dest$ and transfers it to b
$adjacent(b)$	Set of nearby nodes of b
η	Training rate of Q-routing
q	Waiting period until a packet is forwarded from the queue of a
$e_{level}(b)$	Energy level of b
$Rep(b)$	Reputation of b
$TD(b)$	Transmission delay of b
$PTR(b)$	Packet transfer rate of b
$B_{idle}(b)$	Idle buffer space for b
$B_{max}(b)$	Maximum buffer space for b
$\vec{A}$	Position vector
$\vec{G}$	Gravitational power
$\vec{F}$	Water flow advection in deep sea
τ_1, τ_2, τ_3	Random integers

$\vec{M}$	Social forces among tunicates
S_{min}, S_{max}	Minimum and maximum speeds to create group contact, respectively
$\overrightarrow{PD}$	Distance between the food source and tunicate
i	Iteration number
$\overrightarrow{FS}$	Location of food source
$\vec{L}(i)$	Location of tunicate
$rand$	Arbitrary value
$\vec{L}(i')$	Modified position of tunicate with respect to the $\overrightarrow{FS}$
$s_t(x_t, y_t)$	Present location
$s_g(x_g, y_g)$	Target location
ρ	Initial value of ε at the start of the iteration
φ	Incremental range of ε
ω_1, ω_2	Coefficient factors of the adaptive arc
i_{max}	Maximum iteration

the distance between the transmitter and receiver, and λ is the radio frequency range. As per Eq. (1), the energy required for a node to transfer information to the other node increases with the square of d between them. Also, as the transmission energy available for each node varies, it must be standardized. In this scenario, the standardized energy utilization C_{ab} is represented by Eq. (2).

$$C_{ab} = \frac{P_{t-min}(a,b)}{P_{max}(a)} \quad (2)$$

In Eq. (2), $P_{max}(a)$ is the highest transmission power of a and $P_{t-min}(a,b)$ is the lowest transmission power between a and b. Nodes use energy for data transfer as represented in Eq. (2), reducing over intervals. Regulating the battery size, or energy level, is crucial due to potential variations in node size. The energy level of a is determined by Eq. (3).

$$e_{level}(a) = \frac{e_{rem}(a)}{e_{ini}(a)} \quad (3)$$

In Eq. (3), $e_{rem}(a)$ and $e_{ini}(a)$ are the remaining and initial energy of node a, correspondingly.

3.3 Reputation model for isolating collaborative attackers and selecting candidate forwarding set

A reward-based reputation scheme is developed to penalize malicious nodes for avoiding packet forwarding and minimizing energy usage during transmission. Each node begins with an initial reputation value of Rep_{ini}. When $a \in N$ successfully transmits a packet, its reputation increases. Nodes with higher reputations have more packets forwarded, increasing the chances of successful delivery. The reputation of a termed $Rep(a)$ is modified iteratively by Eq. (4).

$$Rep(a) = \alpha Rep_{old}(a) + (1-\alpha) Rep_{new}(a) \quad (4)$$

In Eq. (4), $Rep_{old}(a)$ is the old reputation value of a, $Rep_{new}(a)$ is the reputation that a receives according to the outcomes of transmission, and $\alpha \in [0,1]$ represents the weight between previous and modified reputations. If a succeeds in transferring to b, then Rep_{new} for a is computed by Eq. (5).

$$Rep_{new}(a) = \beta \frac{D(a,b)}{D(dest,b)} + (1-\beta)\frac{TTL}{T_{ab}} \quad (5)$$

In Eq. (5), $D(a,b)$ denotes the Euclidean distance between a and b, $D(dest,b)$ refers to the Euclidean distance between the destination $(dest)$ and b, TTL denotes the packet's Time-To-Live value, and T_{ab} is the duration taken to forward a packet between a and b. In Eq. (5), the initial term is the vicinity to b of a, which forwards the packet to the destination, and the 2nd term states how quick a forwards the packet; $\beta \in [0,1]$ is the weight of those 2 terms. When a fails to transmit, Rep_{new} is 0. So, the reputation of a is decreased through the update, as defined in Eq. (4).

Once the reputation of all nodes is determined, the candidate forwarding set for a at t is chosen, which is represented by $adjacent_t(a)$. Some nodes in this set, i.e., collaborative malevolent nodes (e.g., BHA, GHA, and WHA nodes), may not be suitable for data transmission. This study identifies and isolates undesirable adjacent nodes as collaborative attackers using the reputation model defined in Eqns. (4) and (5). The remaining set of adjacent nodes for

a after isolation is represented as $ANode_t(a)$, and this set is decided according to Algorithm 1.

Algorithm 1: Candidate Forwarding Set Selection by Detecting and Isolating Collaborative Attackers based on Reputation Model

Input: Network graph $G = (N, E)$
Output: Candidate forwarding set $ANode_t(a)$
1. Discover node a's adjacent set $adjacent_t(a)$;
2. $ANode_t(a) \leftarrow adjacent_t(a)$;
3. $\boldsymbol{for}(all\ b \in adjacent_t(a))$
4. $\quad \boldsymbol{if}(adjacent_t(b) \setminus \{a\} == \emptyset)$
5. $\quad\quad ANode_t(a) = ANode_t(a) - \{b\}$;
6. $\quad \boldsymbol{end\ if}$
7. $\quad \boldsymbol{if}(Rep(b) < Rep_{threshold})$
8. $\quad\quad ANode_t(a) = ANode_t(a) - \{b\}$;
9. $\quad \boldsymbol{end\ if}$
10. $\boldsymbol{end\ for}$

In this algorithm, $Rep_{threshold}$ is the reputation threshold. When $ANode_t(a)$ is empty, node a does not transmit packets at t. In the worst-case scenario, every node is densely occupied in the transmission range. In this case, $adjacent_t(a)$ becomes a group excluding a from N, which refers to the collection of each node. So, both time and space complexities of Algorithm 1 is $O(N)$, if $ANode_t(a) = N - \{a\}$.

3.4 Relay node selection based on tunicate swarm optimization Q-learning

After obtaining the candidate relay set for node a, optimal relay nodes are selected based on the TSOQ-learning algorithm to improve routing in MANETs.

3.4.1 Q-learning

Q-learning is an algorithm that uses trial and error in a Markov environment to determine optimal behavior and maximize rewards without an environmental model [24]. In Q-learning, all $Q(s, u)$ have a respective Q-value. In the successive training procedure, the successive action is chosen based on $Q(s, u)$. The total rewards acquired from performing a specific policy and executing the present action are described by the Q-value. The best Q-value is the total rewards obtained by performing allied actions based on the best policy, as described by Eq. (6).

$$Q(s_t, u_t) \leftarrow (1-\alpha)Q(s_t, u_t) + \alpha\left[r_{t+1} + \gamma \max_{u_{t+1}} Q(s_{t+1}, u_{t+1})\right] \quad (6)$$

In Eq. (6), $s_t \in \mathcal{S}$ is the present state, $u_t \in \mathcal{A}$ is the action executed in s_t, $r_{t+1} \in \mathbb{R}$ is the reward obtained after s_t is performed, s_{t+1} is the successive state, γ is a discount factor $(0 \le \gamma \le 1)$, and α is a training coefficient $(0 \le \alpha \le 1)$. The chance of selecting u_t for s_t is called a policy, and denoted by $\pi(s_t, u_t)$. The aim of Q-learning is to find the optimal policy for acquiring the highest reward in the long run. The long-run payoff is described by the total discounted payoffs R, as Eq. (7):

$$R = \sum_{k=0}^{\infty} \gamma^k r_{k+1} \quad (7)$$

In Q-routing, the Q-value when a sends a packet to $dest$ and sends it to b is denoted by $Q_a(dest, b)$. Once a transmits a packet to b, it has the residual transfer period t anticipated by b as Eq. (8):

$$t = \min_{c \in adjacent(b)} Q_b(dest, c) \quad (8)$$

In Eq. (8), $adjacent(b)$ denotes the set of nearby nodes of b. Let $adjacent(b) = ANode_t(b)$. If a receives this value from b, then a modifies its Q-table by Eq. (9):

$$Q_a(dest, b) = (1-\eta)Q_a(dest, b) + \eta(q + T_{ab} + t) \quad (9)$$

In Eq. (9), $\eta \in [0,1]$ is the training rate of Q-routing and q represents the waiting period until a packet is forwarded from the queue of a. The update equation utilized in this algorithm is adapted to fit the setting considered above by Q-routing as:

$$Q_a(dest, b) \leftarrow (1-\eta)Q_a(dest, b) + \eta\left(e_{level}(b) + Rep(b) + TD(b) + PTR(b) + \frac{B_{idle}(b)}{B_{max}(b)}\right) \quad (10)$$

In Eq. (10), $e_{level}(b), Rep(b), TD(b), PTR(b)$ are the energy level, reputation, transmission delay and packet transfer rate of node b, respectively. $B_{idle}(b)$ and $B_{max}(b)$ are the idle and maximum buffer spaces for node b, respectively.

The learning process for each agent starts with an arbitrary state and uses an ε-greedy approach to choose actions. The agent explores each action, modifies $Q(s, u)$ for each based on the highest Q-value and feedback. The agent continues to discover actions until reaching the end state.

3.4.2 Tunicate swarm optimization

Tunicates can find food in the sea through the use of jet propulsion and swarm intelligence. Jet propulsion entails avoiding conflicts, moving towards the optimal search agent (tunicate), and staying close to it. Swarm behavior modifies the locations of other tunicates [25]. These behaviors are mathematically modeled in below subsections.

A. Circumventing Conflicts among Tunicates

To evade the conflicts among tunicates, $\vec{A}$ is used to calculate the new tunicate position by Eqs. (11), (12), and (13):

$$\vec{A} = \frac{\vec{G}}{\vec{M}} \tag{11}$$

$$\vec{G} = \tau_2 + \tau_3 - \vec{F} \tag{12}$$

$$\vec{F} = 2 \times \tau_1 \tag{13}$$

In Eqs. (11)-(13), $\vec{G}$ is the gravitational power and $\vec{F}$ is the water flow advection in deep sea. The parameters τ_1, τ_2, τ_3 are random integers ranging between 0 and 1. Also, $\vec{M}$ is the social forces among tunicates, which is determined by Eq. (14):

$$\vec{M} = [S_{min} + \tau_1 \times S_{max} - S_{min}] \tag{14}$$

In Eq. (14), S_{min} and S_{max} reflect the minimum and maximum speeds to create group contact.

B. Movement towards the Direction of the Best Adjacent

Then, the tunicates are traveling towards the direction of optimal adjacent as Eq. (15):

$$\overrightarrow{PD} = \left|\overrightarrow{FS} - rand \times \vec{L}(i)\right| \tag{15}$$

In Eq. (15), $\overrightarrow{PD}$ is the distance between the food source and tunicate, i is the iteration number, $\overrightarrow{FS}$ is the location of food source, i.e., optimal, and $\vec{L}(i)$ is the location of tunicate and $rand$ is an arbitrary value between 0 and 1.

C. Converge towards the Optimal Tunicate

The tunicate can sustain its location towards the food source, as Eq. (16):

$$\vec{L}(i') = \begin{cases} \overrightarrow{FS} + \vec{A} \times \overrightarrow{PD}, & if\ rand \geq \frac{1}{2} \\ \overrightarrow{FS} - \vec{A} \times \overrightarrow{PD}, & if\ rand < \frac{1}{2} \end{cases} \tag{16}$$

In Eq. (16), $\vec{L}(i')$ denotes the modified position of tunicate with respect to the $\overrightarrow{FS}$.

D. Swarm Behavior

To statistically model the collective nature of a tunicate swarm, the top 2 optimal solutions are kept and the locations of others are updated based on $\overrightarrow{FS}$. The tunicate swarm behavior is described by Eq. (17).

$$\vec{L}(i+1) = \frac{\vec{L}(i) + \vec{L}(i+1)}{2 + \tau_1} \tag{17}$$

The final location would be entirely random within the boundaries of a cylinder or cone, based on the orientation of the tunicate.

3.4.3 TSO-based Q-learning algorithm

TSOQ-learning is a route optimization scheme that uses the TSO to set the initial Q-value for an enhanced ε-greedy Q-learning, rather than starting with Q-values of zero. Its primary objective is to address the slow convergence issue caused by initialization in standard ε-greedy Q-learning.

During the initialization stage, TSOQ-learning generates n tunicate populations in a 15×15 search (grid) space and utilizes the Q-value computation in Eq. (10) to determine the fitness value of all tunicates. Grid locations correspond to coordinates in the area. The Q-table of ε-greedy Q-learning maps the values of all grids. A reward of -1 indicates a problem, +1 indicates free space, and 100 indicates the desired location. The grid with the highest Q-value is the optimal location. TSO is used to adjust the Q-value of all locations, and the process stops after a set number of iterations. Enhanced ε-greedy Q-learning is then applied for route optimization using the updated Q-table.

Enhanced ε-greedy Q-learning uses a selective exploration scheme based on the target location to improve convergence and reduce unnecessary searches. In every search, the agent assesses the correlation between its present location $s_t(x_t, y_t)$ and the target location $s_g(x_g, y_g)$ to determine the most promising directions to search, rather than randomly exploring all four directions. This targeted approach aims to optimize the agent's exploration strategy and ultimately enhance its learning process. The exploration rules of this scheme include:

- If $x_t \leq x_g$ and $y_t < y_g$, then $z = rand(1,2)$; if $x_t < x_g$ and $y_t \geq y_g$, then $z = rand(2,3)$.
- If $x_t > x_g$ and $y_t \leq y_g$, then $z = rand(3,4)$; if $x_t \geq x_g$ and $y_t > y_g$, then $z = rand(1,4)$.

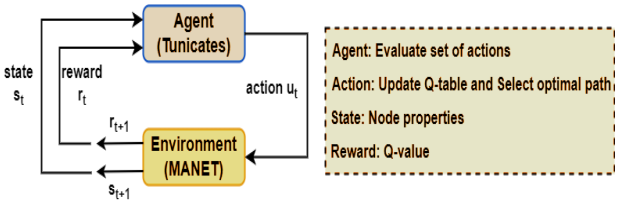

Figure. 5 Principle of TSOQ-learning

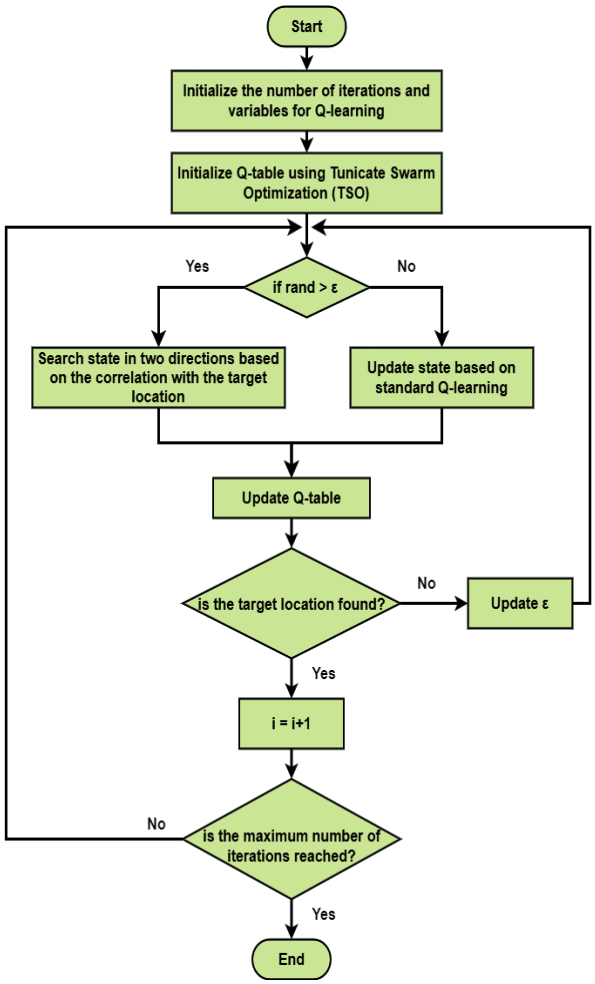

Figure. 6 Flow diagram of TSOQ-learning algorithm

Table 2. Parameters used in TSO and Q-learning

Parameters		Value
TSO	No. of tunicates	80
	S_{min}	1
	S_{max}	4
	No. of generations	100
	No. of iterations	200
Q-learning	α	0.2
	γ	0.8
	No. of iterations	200

After that, to flexibly switch between search and use procedures, the ε value in ε-greedy Q-learning is modified adaptively. The value of ε is calculated by Eq. (18):

$$\varepsilon = \rho + \varphi/(1 + e^{\omega_1 - \omega_2 t}) \quad (18)$$

In Eq. (18), ρ represents the initial value of ε at the start of the iteration, while φ represents the incremental range of ε. The addition of ρ and φ is assigned to one. The parameters ω_1 and ω_2 are coefficient factors of the adaptive arc, and their values are calculated by the number of iterations. The parameter t represents the present iteration number. The enhanced ε-greedy Q-learning algorithm optimizes routes using a Q-table. The Q-value computation updates the Q-table iteratively, and the optimal route is determined by finding the route with the maximum Q-value.

This approach allows the TSO to learn from previous experiences, reducing computation time and accelerating convergence speed. Fig. 5 illustrates a principle of TSOQ-learning, and Fig. 6 shows a flow diagram of the TSOQ-learning algorithm. Algorithm 2 presents its pseudocode for route optimization. As well, Table 2 shows the parameters used for TSO and Q-learning algorithms.

Algorithm 2: Pseudocode for TSOQ-learning

Input: Network graph $G = (N, E)$

Output: Best routing policy π

1. Initialize the tunicate population $\vec{L}$, variables $\vec{A}, \vec{G}, \vec{F}, \vec{M}$, and maximum iteration i_{max};
2. ***while***$(i < i_{max})$
3. Compute the fitness of all tunicates using Eq. (10);
4. Find the optimal tunicate in the given search space;
5. Modify the location of all tunicates utilizing Eq. (17);
6. Change the new tunicate in a particular search space that crosses the margin;
7. Calculate the new tunicate fitness value;
8. Modify L when the best solution exists than the past optimal one;
9. ***end while***
10. Obtain the optimal solution (optimal Q-values) which is obtained so far;
11. Initialize the optimal Q-values in Q-table, start location, and target location;
12. Choose a starting state $Q(s_1, u_1)$;
13. ***while***$(i < i_{max})$
14. ***while***$(s_t \; is \; not \; target \; location)$
15. ***if***$(Probability < \varepsilon)$
16. Select u_t based on the max $Q(s_t)$;
17. Take action u_t, and get reward r;

18. Update Q-value by Eq. (10);
19. Update reputation by Eqs. (4) and (5);
20. Shift to new state;
21. ***else if*** $(Probability \geq \varepsilon)$
22. Select u_t randomly in two directions based on the correlation with the target;
23. ***end if***
24. Update ε by Eq. (18);
25. ***end while***
26. ***end while***

Algorithm 2 has a time complexity of $O(N)$ and a space complexity of $O(N^2)$ in the worst-case scenario. Thus, the TSOQCADA can detect collaborative attackers and ensure the selection of optimal paths for efficient data transmission.

4. Simulation results

This section provides the efficacy of the TSOQCADA and compares it with the existing algorithms in MANETs. The simulations are carried out on a system with an Intel ® Core TM i5-4210 CPU @ 2.80 GHz platform. Table 3 presents the simulation parameters in this study. The existing algorithms such as ETRS [17], HTRM [20], and DNLPPR-WMNDI [23] are also simulated using these parameters to compare their performance with the proposed algorithm. The evaluation metrics include PDR, PLR, energy consumption, throughput, and E2E delay. An experimental test is conducted to study how changes in node density affect the performance of routing algorithms.

4.1 PDR

It is the percentage of total packets transferred by the source node and delivered at the destination node, calculated by Eq. (19):

Table 3. Simulation parameters

Parameters	Value
Simulation tool	NS2.34
Simulation region	1000×1000 m^2
No. of nodes	100
No. of malicious nodes	20
Attack types	BHA, GHA, and WHA
Transmission range	100 m
Packet generation rate	1 packet/s
Buffer size	100 MB
Packet size	1 MB
Packet TTL	10 sec

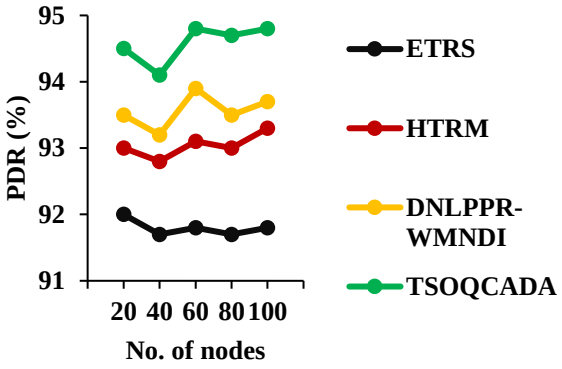

Figure. 7 PDR vs. No. of nodes

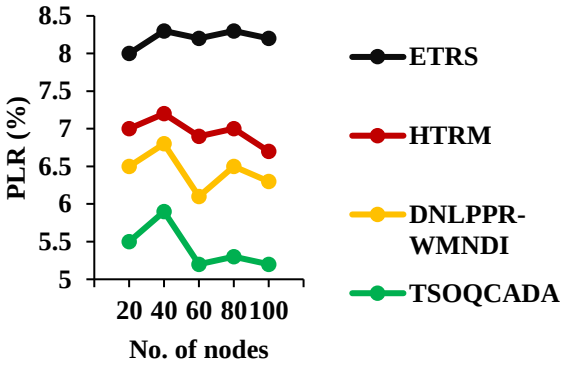

Figure. 8 PLR vs. No. of nodes

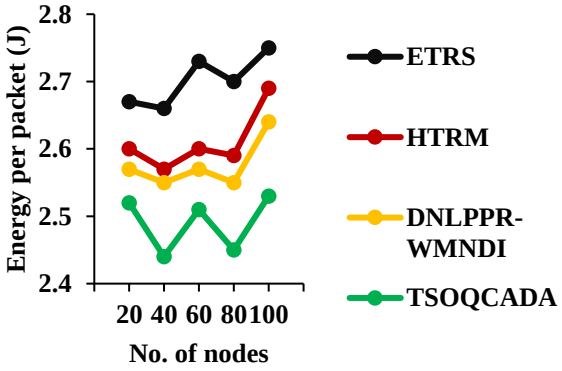

Figure. 9 Energy utilization vs. No. of nodes

$$PDR = \frac{Number\ of\ packets\ delivered\ at\ the\ target\ node}{Number\ of\ packets\ sent\ from\ the\ source\ node} \times 100 \quad (19)$$

Fig. 7 displays the PDR for the proposed and existing routing algorithms. The results indicate that the TSOQCADA can improve PDR values for 100 nodes by 3.3%, 1.6%, and 1.2% compared to the

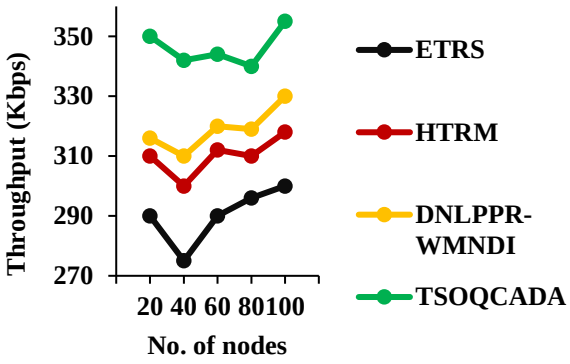

Figure. 10 Throughput vs. No. of nodes

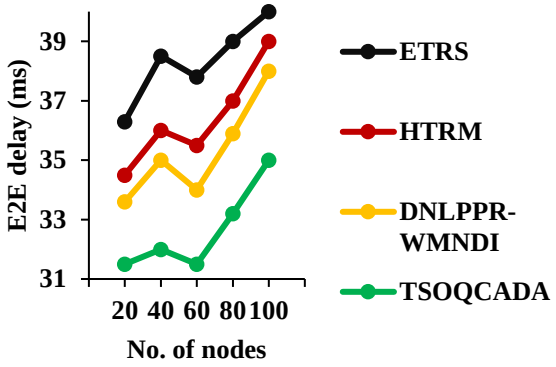

Figure. 11 E2E delay vs. No. of nodes

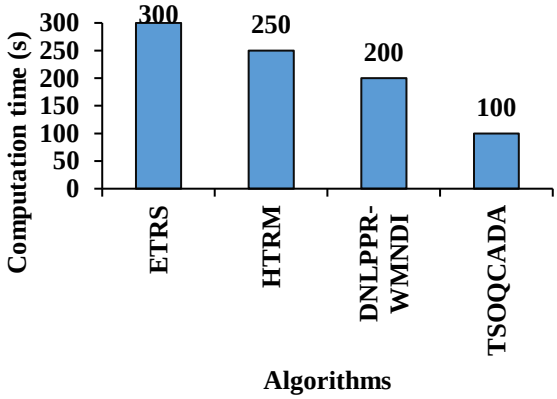

Figure. 12 Computation time vs. different routing algorithms

ETRS, HTRM, and DNLPPR-WMNDI algorithms, respectively. This enhancement is due to the isolation of collaborative malicious nodes, ensuring that only reliable nodes are used in the best route.

4.2 PLR

It is the percentage of data packets that don't reach the destination node, calculated by Eq. (20):

$$PLR = \frac{Number\ of\ packets\ lost}{Number\ of\ packets\ sent\ from\ the\ source\ node} \times 100 \quad (20)$$

Fig. 8 depicts the PLR for the proposed and existing routing algorithms. The findings show that the TSOQCADA can decrease PLR values for 100 nodes by 36.6%, 22.4%, and 17.5% when compared to the ETRS, HTRM, and DNLPPR-WMNDI algorithms, respectively. This is due to the mitigation of collaborative malicious nodes from the network, leading to a decrease in packet dropping.

4.3 Energy consumption

The energy utilized by each node is determined by Eq. (21):

$$Energy\ used = \sum_{i=1}^{n} ini(i) - e(i) \quad (21)$$

In Eq. (21), n is the node number, $ini(i)$ is the initial energy level of node i and $e(i)$ is the energy level of node i after packet transfer. Fig. 9 illustrates the energy utilization of various routing algorithms. The data confirms that the energy utilization per packet of the TSOQCADA for 100 nodes is reduced by up to 8%, 5.9%, and 4.2% compared to the ETRS, HTRM, and DNLPPR-WMNDI, respectively. Isolating collaborative attackers leads to a decrease in unwanted energy dissipation, as these nodes do not join in choosing the relay nodes and optimal paths.

4.4 Throughput

It is the quantity of packets efficaciously delivered at the target node from the origin node in a given interval, calculated using Eq. (22). Fig. 10 presents the throughput of proposed and existing routing algorithms. The throughput of the TSOQCADA for 100 nodes is increased by 18.3%, 11.6%, and 7.6% compared to the ETRS, HTRM, and DNLPPR-WMNDI, respectively. This is due to the isolation of collaborative attack nodes from the routing based on the node's reputation.

4.5 E2E delay

The time taken for a packet to be sent from the origin node to the target node is calculated by Eq. (23).

$$E2E\ delay = \frac{\sum_{x=1}^{p}(R_x - S_x)}{n} \quad (23)$$

In Eq. (23), p is the successful packets that are delivered to the target nodes, R_x is the receiving period of the packet x and S_i is the transmitting period of x. In Fig. 11, the E2E delay for various routing algorithms are shown. The E2E delay of the TSOQCADA for 100 nodes is reduced by 12.5%, 10.3%, and 7.9% compared to the ETRS, HTRM, and DNLPPR-WMNDI, respectively. This is because collaborative suspected nodes do not participate in reliable node and optimal path selection procedures, ensuring that data transfer only occurs between reputable nodes, leading to a decrease in E2E delay.

4.6 Computation time

It is the time required to execute various algorithms for detecting and mitigating collaborative attacks in the network. Fig. 12 shows the computation time for different routing algorithms with 100 nodes. The TSOQCADA has 66.67%, 60%, and 50% reductions in computation time compared to ETRS, HTRM, and DNLPPR-WMNDI, respectively. This indicates that TSOQCADA can detect collaborative attacks more quickly than the other algorithms.

5. Conclusion

This article presents the TSOQCADA algorithm for detecting and mitigating collaborative attacks in MANETs. It fine-tunes Q-table initialization and uses a nonlinear function to update ε value dynamically. This leads to optimal routing paths, reducing packet loss and energy consumption. Simulation outcomes demonstrate that the TSOQCADA outperforms existing routing algorithms in MANETs facing collaborative attackers, thereby enhancing routing performance and network security. Specifically, the TSOQCADA achieved a 94.8% PDR, 5.2% PLR, 2.53J energy/packet, 355Kbps throughput, and 35ms E2E delay for 100 nodes with 20 malicious nodes in the network, outperforming other routing algorithms.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, methodology, software, validation, Kavitha; formal analysis, investigation, Meenakshi; resources, data curation, writing—original draft preparation, Kavitha; writing—review and editing, Kavitha; visualization, supervision, Meenakshi.

References

[1] S. N. Mahapatra, B. K. Singh, and V. Kumar, "A Survey on Secure Transmission in Internet of Things: Taxonomy, Recent Techniques, Research Requirements, and Challenges", *Arabian Journal for Science and Engineering*, Vol. 45, pp. 6211-6240, 2020.

[2] O. I. Khalaf, F. Ajesh, A. A. Hamad, G. N. Nguyen, and D. N. Le, "Efficient Dual-Cooperative Bait Detection Scheme for Collaborative Attackers on Mobile Ad-Hoc Networks", *IEEE Access*, Vol. 8, pp. 227962-227969, 2020.

[3] M. A. Al-Absi, A. A. Al-Absi, M. Sain, and H. Lee, "Moving Ad Hoc Networks—A Comparative Study", *Sustainability*, Vol. 13, No. 11, pp. 6187, 2021.

[4] S. Kalaivanan, "Quality of Service (QoS) and Priority Aware Models for Energy Efficient and Demand Routing Procedure in Mobile Ad Hoc Networks", *Journal of Ambient Intelligence and Humanized Computing*, Vol. 12, pp. 4019-4026, 2021.

[5] M. E. M. Dafalla, R. A. Mokhtar, R. A. Saeed, H. Alhumyani, S. Abdel-Khalek, and M. Khayyat, "An Optimized Link State Routing Protocol for Real-Time Application over Vehicular Ad-Hoc Network", *Alexandria Engineering Journal*, Vol. 61, No. 6, pp. 4541-4556, 2022.

[6] P. Sarao, "Ad Hoc On-Demand Multipath Distance Vector Based Routing in Ad-Hoc Networks", *Wireless Personal Communications*, Vol. 114, No. 4, pp. 2933-2953, 2020.

[7] S. Satheeshkumar and N. Sengottaiyan, "Defending against Jellyfish Attacks Using Cluster Based Routing Protocol for Secured Data Transmission in MANET", *Cluster Computing*, Vol. 22, pp. 10849-10860, 2019.

[8] M. T. Sultan, K. N. Yasen, and A. Q. Saeed, "Simulation-Based Evaluation of Mobile Ad Hoc Network Routing Protocols: Ad Hoc On-Demand Distance Vector, Fisheye State Routing, and Zone Routing Protocol", *Cihan University-Erbil Scientific Journal*, Vol. 3, No. 2, pp. 64-69, 2019.

[9] S. Kalime and K. Sagar, "A Review: Secure Routing Protocols for Mobile Adhoc Networks (MANETs)", *Journal of Critical Reviews*, Vol. 7, pp. 8385-8393, 2021.

[10] N. Khanna and M. Sachdeva, "A Comprehensive Taxonomy of Schemes to Detect and Mitigate Blackhole Attack and Its

Variants in MANETs", *Computer Science Review*, Vol. 32, pp. 24-44, 2019.

[11] K. Ourouss, N. Naja, and A. Jamali, "Defending against Smart Grayhole Attack within MANETs: A Reputation-Based Ant Colony Optimization Approach for Secure Route Discovery in DSR Protocol", *Wireless Personal Communications*, Vol. 116, pp. 207-226, 2021.

[12] N. Panda, B. Patra, and S. Hota, "MANET Routing Attacks and Their Countermeasures: A Survey", *Journal of Critical Reviews*, Vol. 7, pp. 2777-2792, 2020.

[13] W. Li, W. Meng, and L. F. Kwok, "Surveying Trust-Based Collaborative Intrusion Detection: State-of-the-Art, Challenges and Future Directions", *IEEE Communications Surveys & Tutorials*, Vol. 24, No. 1, pp. 280-305, 2021.

[14] B. U. I. Khan, F. Anwar, F. D. B. A. Rahman, R. F. Olanrewaju, M. L. B. M. Kiah, M. A. Rahman, and Z. Janin, "Exploring MANET Security Aspects: Analysis of Attacks and Node Misbehaviour Issues", *Malaysian Journal of Computer Science*, Vol. 35, No. 4, pp. 307-338, 2022.

[15] S. M. Muzammal, R. K. Murugesan, and N. Z. Jhanjhi, "A Comprehensive Review on Secure Routing in Internet of Things: Mitigation Methods and Trust-Based Approaches", *IEEE Internet of Things Journal*, Vol. 8, No. 6, pp. 4186-4210, 2020.

[16] R. Vatambeti, K. S. Supriya, and S. Sanshi, "Identifying and Detecting Black Hole and Gray Hole Attack in MANET Using Gray Wolf Optimization", *International Journal of Communication Systems*, Vol. 33, No. 18, p. e4610, 2020.

[17] A. A. Mahamune and M. M. Chandane, "An Efficient Trust-Based Routing Scheme against Malicious Communication in MANET. *International Journal of Wireless Information Networks*, Vol. 28, No. 3, pp. 344-361, 2021.

[18] G. Farahani, "Black Hole Attack Detection Using K-Nearest Neighbor Algorithm and Reputation Calculation in Mobile Ad Hoc Networks", *Security and Communication Networks*, Vol. 2021, pp. 1-15, 2021.

[19] S. V. Simpson and G. Nagarajan, "A Fuzzy Based Co-Operative Blackmailing Attack Detection Scheme for Edge Computing Nodes in MANET-IOT Environment", *Future Generation Computer Systems*, Vol. 125, pp. 544-563, 2021.

[20] S. N. Pari and K. Sudharson, "Hybrid Trust Based Reputation Mechanism for Discovering Malevolent Node in MANET", *Computer Systems Science and Engineering*, Vol. 44, No. 3, pp. 2775-2789, 2023.

[21] I. Moumen, N. Rafalia, J. Abouchabaka, and Y. Chatoui, "AODV-Based Defense Mechanism for Mitigating Blackhole Attacks in MANET", In: *Proc. of E3S Web of Conf.*, Vol. 412, p. 01094, 2023.

[22] A. Abdelhamid, M. S. Elsayed, A. D. Jurcut, and M. A. Azer, "A Lightweight Anomaly Detection System for Black Hole Attack. *Electronics*, Vol. 12, No. 6, p. 1294, 2023.

[23] A. S. Narmadha, S. Maheswari, and S. N. Deepa, "Watchdog Malicious Node Detection and Isolation Using Deep Learning for Secured Communication in MANET", *Automatika*, Vol. 64, No. 4, pp. 996-1009, 2023.

[24] J. Ryu and S. Kim, "Reputation-Based Opportunistic Routing Protocol Using Q-Learning for MANET Attacked by Malicious Nodes", *IEEE Access*, Vol. 11, pp. 47701-47711, 2023.

[25] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, "Tunicate Swarm Algorithm: A New Bio-Inspired Based Metaheuristic Paradigm for Global Optimization", *Engineering Applications of Artificial Intelligence*, Vol. 90, p. 103541, 2020.