SINGLE LEADER SINGLE FOLLOWER GAME FOR DETECTING NETWORK ATTACKS IN WSN

S. Suganthi

Assistant Professor, Department of Computer Science, G.Venkataswamy Naidu College (Autonomous), Kovilpatti

ABSTRACT

The paper introduces a game theory-based approach for detecting network attacks in wireless sensor networks (WSN), employing a Stackelberg game model with single leaders and single followers. The model utilizes quadratic programming and a backtracking search optimization algorithm to determine optimal strategies for the leader (cluster head) and followers (agent nodes). The study focuses on addressing energy depletion and security breaches in WSN by optimizing major upper-level problems and handling lower-level problems. Simulation results, executed in Matlab, demonstrate the effectiveness of the proposed method in detecting black hole and warm hole attacks. Notably, the integration of the Stackelberg game with a backtracking search optimization algorithm contributes to improved optimization results compared to previous approaches. The paper concludes by discussing the potential application of the proposed model in large-scale IoT devices, emphasizing its significance in enhancing network security.

Keywords: Stackelberg Game, Black hole attack, warm hole attack, backtracking search, and optimization.

Introduction

Game theory is one of the interactive decision-making methodologies that follows the techniques mathematically. Generally, a formal game should consist of three elements such as the players of the game, the techniques available for each player, and the optimal payoffs of each player. A proposed Stackelberg game contains single leaders and single followers [1]. The follower makes decisions based on their leader’s decision. In a proposed game the cluster head acts as a single-leader which can optimize the major upper-level problem and all other agent nodes act as followers who join with leaders which can handle the lower-level problems. Generally, the follower decides after observing the leader's decision and the leader

expects the response from the follower and selects their optimal strategy for making a decision. At the same time, all followers select their optimal responses by competing with each other by the leader's choice. Many researchers do their research in [2] Stackelberg games and apply wide applications in various areas. In a proposed game in this paper, several cluster heads can act in the position of leaders and the remaining players can act in the position as followers, it becomes a single-leader-follower game. The Single-leader-follower game [3] takes place in some oligopoly markets [4]. For example, large care companies called the leaders to have to produce new-fashioned cars when they decide to produce the cars and quantities [5]. After observing the decisions of the large car companies (leaders), all other smaller companies (followers) choose their optimal strategies to produce the quantities and qualities of the cars followed by the leaders [6]. Generally, the characteristics of the attack of a warm hole are two malicious or attacker nodes combined to make a tunnel [7]. The attacker node automatically receives the packet and sends it to another destination node through the tunnel without the knowledge of the network [8][9]. The introduction part of the paper discusses a network model and a game between players in the context of a black hole attack. Section 2 of the paper is expected to describe the proposed network model, while section 3 formulates the game that is played between the players. Section 4 likely provides an overview of the strategic space of the game, while section 5 presents the results of experiments conducted using MATLAB. Finally, section 6 concludes the paper and provides future directions.

Network Model

Network Topology

The network topology of the proposed system for analyzing Black hole and warm hole attacks is given in Figure 1.

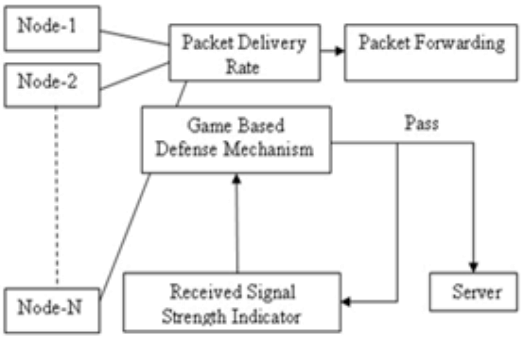

Fig. 1. Network Topology

Received Signal Strength Indicator (RSSI) is the measured power of a received radio signal [9]. The attacks are detected depending on the Packet Delivery Rate (PDR) and Packet Forwarding Rate (PFR). THE threshold RSSI value for a black hole attack is T_(rssi_bh)=-40. The threshold RSSI value for a warm hole attack is T_(rssi_wh)=-55.

The Threshold Packet delivery rate in a black hole attack isT_(pdr_bh)=90. The threshold value of the packet delivery rate in a warm hole attack is T_(pdr_wh)=80. The cluster head has been selected using particle swarm optimization refer to Figure 2. During the transmission of data from the cluster head to all other agents, nodes refer to Figure 3, some attacker nodes may indicate the highest Received Signal strength value and packet delivery rate[10].

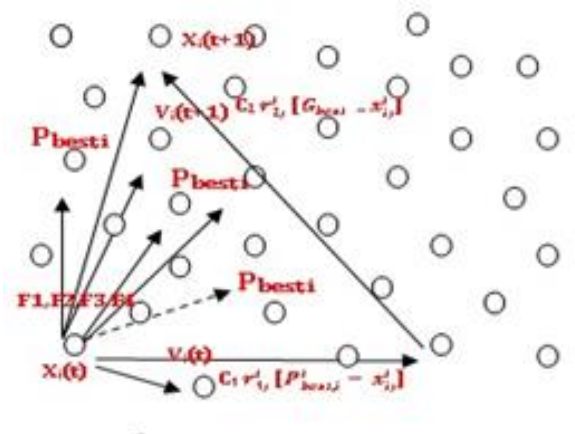

Fig. 2. Cluster Head Selection

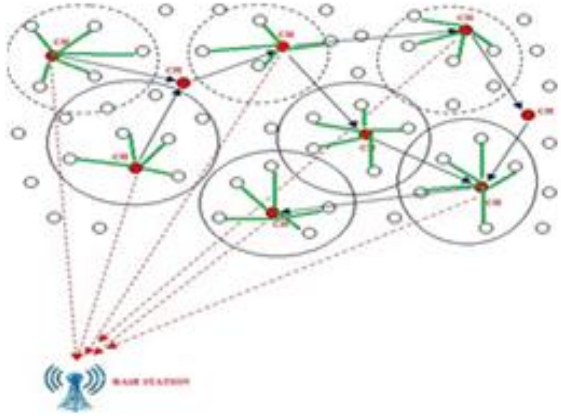

Fig. 3. Node Transmission

As an AI language model, detecting black hole and warm hole attacks in wireless networks by monitoring RSSI values and packet delivery rates is a reasonable approach. However, it is important to note that this method alone may not be sufficient for detecting all types of attacks, as attackers may use sophisticated techniques to avoid detection. Therefore, it is recommended to implement a combination of security measures, including cryptographic techniques and secure routing protocols, in addition to monitoring RSSI values and packet delivery rates. Cryptographic techniques can provide authentication, confidentiality, and integrity of the data transmitted in the network, while secure routing protocols can help ensure that packets are delivered through legitimate nodes.

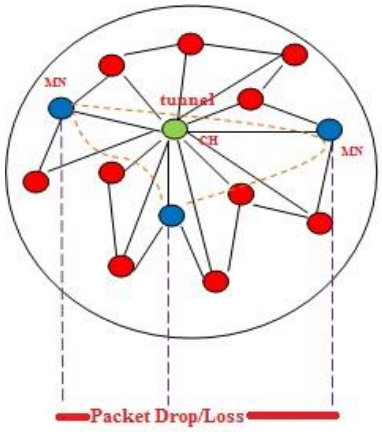

Figure 4 – Packet Drop / Loss

The proposed algorithm can compute the threshold Received Signal Strength

Indicator(RSSI) value and Threshold Packet Delivery Rate (PDR)value. The threshold value is based on bandwidth, data flow rate, link strength, and the outcome of the defender. The threshold value T automatically passes to the indicators of the game. Based on the threshold value the game decides whether to allow the packets or redirect or drop.

Game Formulation

Game theory is the formal, mathematical methodology for interaction between players like people, agents, or robots which has been applied in various areas such as business, economics, and management to solve problems. In this section, we have identified the attacked nodes through some gaming strategies. Competition among the players is one of the significant topics. Here to use the Stackelberg game model that is applied to find the normal nodes in a sensor environment which is shown in Fig. 5.

In a Stackelberg model, the leader node chooses a strategy first, and then the follower monitors the decision and makes his own decision. Generally, the game includes the following three comprise:

In game theory, players are assumed to be rational decision-makers who choose their strategies based on their beliefs about the other players' strategies, to maximize their own utility or payoff function. The leader node is mentioned by N and the follower node is mentioned by M. The utility function represents the outcome or payoff that a player receives from a particular combination of strategies, taking into account the strategies of the other players.

The strategy set A is a set of all possible combinations of actions that the players can take in the game. Each player i has a strategy a_i, which is a subset of A that specifies the action that player i takes given the other players' strategies. The set of all other players' strategies, denoted by a−i, is the set of all strategies of the other players except for player i.

The utility function ui(ai,a−i) is a function that maps the combination of strategies (ai,a−i) to a real-valued outcome or payoff for player i. The payoff function takes into account the strategies of all players and the outcomes that result from their strategies. The goal of each player is to choose a strategy that maximizes their expected utility or payoff, given their beliefs about the other players' strategies. Overall, game theory provides a framework for analyzing the strategic interactions between rational decision-makers, and the strategies and outcomes that result from these interactions.

Players and the Strategic Space

In this paper, use a Stackelberg game between the leader and follower. The cluster head plays the role of the game leader and takes the decision first whereas all other agent nodes act as the follower that monitors the leader's strategy and chooses their best responses to it in terms of attack detection strategies as illustrated in the following Figure 5.

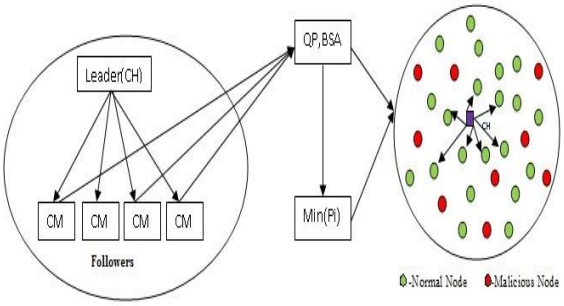

Figure 5 – Stackelberg Game

Quadratic Programming

Quadratic programming is a type of nonlinear optimization problem where the objective function and constraints are quadratic. In this case, the problem has two players, a leader and a follower, and their objectives are given by QL(PL, PF) and QF(PL, PF), respectively.

The objective function for the leader, QL(PL, PF), can be written as a quadratic form in terms of the decision variables PL and PF. The matrix Q is the Hessian of the quadratic function, and the vector C is the linear coefficient. The constraints are given by the linear system AX=b, where A is a matrix of coefficients and b is a vector of constants. The non-negativity constraint x≥0 is also included.

The linear system can be transformed into a block matrix form where the decision variables are arranged as [PL, PF] and the coefficients are arranged in blocks. This makes it easier to apply optimization algorithms to the problem. The quadratic programming problem for the leader and follower can then be expressed as the block matrix form

$$Q_L = \frac{1}{2}\begin{bmatrix} P_L \\ P_F \end{bmatrix}^T \begin{bmatrix} C_L^{LL} & C_L^{LF} \\ C_L^{LF} & C_L^{FF} \end{bmatrix} \begin{bmatrix} P_L \\ P_F \end{bmatrix} + \begin{bmatrix} C_{LL} \\ C_{LF} \end{bmatrix}^T \begin{bmatrix} P_L \\ P_F \end{bmatrix}$$

$$Q_F = \frac{1}{2}\begin{bmatrix} P_L \\ P_F \end{bmatrix}^T \begin{bmatrix} 0 & C_L^{LF} \\ C_L^{LF} & C_L^{FF} \end{bmatrix} \begin{bmatrix} P_L \\ P_F \end{bmatrix} + [C_{FF}]^T [P_F]$$

$Q_L, Q_F \geq 0$

where X=[PL, PF] and Y is an auxiliary variable. The matrices Q, E, and C are defined as before, and d is a vector of constants.

The solutions to the quadratic programming problem are the optimal values of PL and PF that minimize their respective objective functions subject to the given constraints. The solutions must satisfy the non-negativity constraints QL, QF ≥ 0.

In a single leader-follower game, the algorithm for the Leader's Choice is as follows:

Algorithm 1 describes a leader-follower game, where the leader makes a decision based on the follower's response. The leader has a set of decision variables XL, which are independent of the follower's decision y. The objective function ΘL is dependent on the leader's decision xL as well as the follower's decision y. The algorithm starts with the input of the leader's variables and outputs the objective function. The leader's strategy is to choose XL based on x-L (the set of decision variables of all other leaders) and independent of the follower's decision y. The leader then solves the optimization problem Min QL (xL, x-L, y) subject to the constraint xL ε XL, where XL is the feasible set of decision variables for the leader.

The overall objective is to find the optimal values of the decision variables xL that minimize the objective function QL, which is a function of xL, x-L, and y. The objective function represents the leader's utility or payoff, which is dependent on the decisions of both the leader and the follower. Overall, the Leader's Choice algorithm is a game-theoretic approach that allows the leader to decide while taking into account the decisions of other leaders and the follower's response.

Algorithm 2: Follower's Choice is a mathematical algorithm that describes how a follower makes decisions based on the leader's input. The algorithm takes in the follower's variables as input and outputs the objective function of the follower. The algorithm assumes that there are M followers, each with their response variable yF in ℝmF. The response variables are grouped into a vector tuple y:= (y1….ym) ε ℝm, and the total number of response variables is n. The algorithm also assumes that the followers' decision variables are (xF,x-F) ε ℝmF + m –F, where m-F := m-mF. The objective function or utility function of the follower is ΘF : ℝn+m → ℝ, which depends on the follower's response variables yF, and the leader's decision variables x.

The follower's strategy is to use a function YF (y-F , x) ℝmF that depends on the leader's decision variables x to determine its response variables yF. The follower then solves the optimization problem:

Min QF (x, yF, y-F)

Subject to yF ε YF

F(x,y):= (∇_(y^') ,ΘF)

where QF is a function of the leader's decision variables x, the follower's response variables yF, and the other followers' response variables y-F. The function QF is minimized subject to the constraint that the follower's response

variables yF are in the function YF. In summary, the Follower's Choice algorithm describes how a follower makes decisions based on the leader's input. The follower solves an optimization problem to determine its response variables based on the leader's decision variables and the other followers' response variables.

Backtracking Search Optimization

After applying quadratic programming we can get the optimized nodes. Generally, a backtracking search is applied to have single solutions and need all those solutions. Backtracking algorithm which is mainly used to execute singleple sequences of decisions, which is performed recursively until satisfying certain constraints.

IG_{min}= Global Minimizer Game matrix

$G_{\min_{zer}}$= Minimum of Minimum Game matrix

N= Number of agent nodes

D= population size

$P_{i,j}$= Player of i,j strategy

Pre $_{i,j}$ = All other players i,j strategy

fit$_{pi}$ = Fitness function player with strategy i

Algorithm 3:

I Input:

$O_{func}, N, D, max_{cycle}, mix_{rate}, low\ , up, epoches$

ut: $G_{min}, G_{\min_{zer}}$

Step1: Initialization

1. $P_{i,j} \sim U(low_j - up_j)$ *// i=1..N , j=1..D,U-is the uniform distribution*

Step 2: Selection –IBSA has the option of redefining the old population, it is used to randomly change the order of the individuals in the old population

2. $G_{min} = inf$,D=30,N= $Agent_{node}$;
3. $toP_{i,j} = \lim_{i \to 1\ to\ N} \lim_{j \to 1\ to\ D} (up_j - low_j) + low_j$
4. $Pre_{i,j} = \lim_{i \to 1\ to\ N} \lim_{j \to 1\ to\ D} (up_j - low_j) + low_j$
5. $fit_{Pi} = \lim_{\text{i}\to 1 \text{ to N}} O_{func}(P_i)$
6. $For\ \ y = 1\ to\ max_{cycle}$

   $If (a < b)\ then\ \ Pre = P$

   $Pre = permuting(Pre)$ *// permuting function is a random shuffling function*

$Step\ 3: Mutation$

The mutation process generates the initial form of the trial population called M_{mut}

$M_{mut} = P + F\ (Pre - P)$ *// where F controls the amplitude of the search direction matrix*

$(oldp - p)$*, the historical population is used in the calculation of the search-direction matrix.*

7. $M_{mut} = P + 3\ (Pre - P)$ *// 3 is random number ~ N(0,1)*

Step 4: Crossover

BSA is a type of swarm intelligence algorithm inspired by movement and behavior. In BSA, a population of candidate solutions is iteratively evolved to optimize a given objective function. The crossover process you described is a way to combine individuals from the population to create new candidate solutions with potentially better fitness values. The first step of the crossover process creates a map that pairs individuals from the trial population T with individuals from a separate population P. The second step uses this map to update the individuals in T by replacing them with corresponding individuals from P. The mix rate parameter controls how many individuals from T are updated in each iteration. Compared to other evolutionary algorithms (EA), BSA's crossover process is more complex and uses different parameters to control the rate of crossover. However, it is difficult to compare the effectiveness of different optimization techniques without more context about the specific problem being solved and the performance of different algorithms on that problem. A few people of the preliminary populace toward the finish of BSA's hybrid cycle can flood the permitted search space limits because of BSA's transformation methodology. The people past the pursuit space limits are recovered utilizing Algorithm 3. $\mathrm{Cr}_{1:\mathrm{N},1:\mathrm{D}} = 1$

9. $If (c < d)\ then$

For k=1 from N $\quad Cr_{i,1:[mix_{rate} * rand(\ D)]]} = 0$

$End\ for$

Else

$For\ k = 1\ from\ N$

$Cr_{k:rand(\ D)]]} = 0$

$End\ for$

$$T = M_{mut}$$

//*Boundary control mechanism*

$$T_{i,j} = \lim_{i \to 1\ to\ N} \lim_{j \to 1\ to\ D} \begin{cases} rand(up_j - low_j) + low_j & if\ (T_{i,j} < low_j)\ OR\ (T_{i,j} > up_j) \\ T_{i,j} & else \end{cases}$$

$End\ if$

Step 5: Selection – II

10. $fit_T = O_{func}(T) fit_{Pq} = \lim_{q \to 1 \text{ to } N} \begin{cases} fit_{T_q} & if(fit_{T_q} < fit_{Pq}) \\ 0 & else \end{cases}$

11. $fit_{best} = \min(fit_P)$

12. If $fit_{best} < G_{min}$ *then*

$G_{min} = fit_{best}$ *// the global minimum value is updated to be the fitness value of the population*

$G_{\min_{zer}} = P_{best}$

End if

13. *End for*

//detection of normal as well as malicious node

14. $label_{label} = \max(Best_{pos)} > mean(G_{\min_{zer}})$

$$Nor_{node} = \begin{cases} 1 & if \left(\max(Best_{pos)} > mean\left(G_{\min_{zer}}\right) \right) \\ 0 & otherwise \end{cases}$$

The follower employs the same strategy to find best nodes or any strategy with the same payoff.

Attack Model

Blackhole and Warmhole Attacks

A black hole attack involves a malicious node that advertises itself as having the shortest path to a destination node or a packet that is being transmitted. This is done by the attacker node intercepting the route request and replying with a forged route based on its routing table. The attacker node may provide false information about its signal strength and packet delivery rate, which can deceive the routing protocol into selecting the attacker node as the next hop. Once the forged route is established, the attacker node can either drop all the packets or forward them to an unknown address, effectively blocking the communication between the source and destination nodes. A wormhole attack, on the other hand, involves two or more attacker nodes that create a shortcut between distant parts of the network by relaying packets through a tunnel. This tunnel may be created using a direct physical connection or by using a higher-powered radio transmission that can reach far distances. The wormhole attackers can selectively drop or modify packets to manipulate the network's behavior. Both black hole and wormhole attacks are serious security

threats in wireless networks, and various countermeasures have been proposed to detect and mitigate these attacks. These countermeasures include the use of authentication and encryption mechanisms, secure routing protocols, and intrusion detection systems.

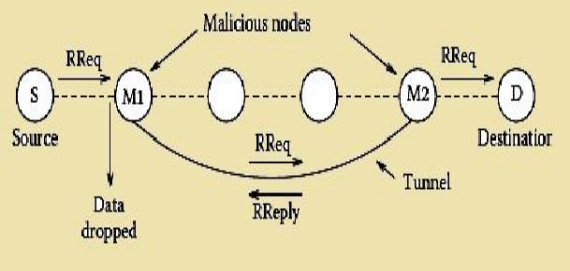

Figure 6 – Blackhole & Warmhole Attack

A wormhole attack is a type of attack in wireless networks where two attackers collaborate to create a shortcut tunnel between two distant points in the network. This tunnel can be created using a wired link, a high-quality wireless band link, or a logical link via packet encapsulation. Once the wormhole tunnel is established, the malicious nodes can easily intercept, modify, or drop packets that are transmitted between other nodes in the network. In a wormhole attack, one malicious node receives packets from its neighbors and sends them to another malicious node through the wormhole tunnel, where they are received and forwarded to their destination. This attack can have serious consequences for the network, as it can lead to the disruption of communication, the injection of false data, and the compromise of sensitive information. The effectiveness of a wormhole attack can depend on the quality of the link used to create the tunnel. For example, if a wired link or a high-quality wireless out-of-band link is used, the attackers can communicate quickly and easily, making the attack more effective. However, even with a lower-quality link, a wormhole attack can still be launched without the need for any special hardware or routing protocols.

To detect a wormhole attack, several techniques can be used, including monitoring the received signal strength value and packet delivery rate. If these values exceed a certain threshold, it may indicate that some nodes in the network are being affected by a wormhole attack. Other techniques for detecting wormhole attacks include hop count analysis, time of arrival analysis, and network topology analysis. Overall, a wormhole attack can pose a serious threat to wireless networks, and it is important for network administrators to take measures to prevent and detect such attacks. These measures may include the use of encryption, authentication, and intrusion detection systems, as well as the deployment of physical security measures to prevent unauthorized access to the network.

Algorithm 4: Detection of various attacks

Input: $PDR\_node\_Id,$
$RSSI\_node\_Id, DPR\_node\_Id$

Output: Malicious and normal node Attack ID

PDR- Packet Delivery Rate

RSSI-Received Signal Strength Indicator

DPR=Duplicate Packet Rate

For i=1:to size of N_x

$if\ (PDR\_node\_Id(i)$
$> T_{pdr_{bh}} \&\& RSSI\_node\_Id\ (i)$
$> T_{rssi\_bh}\ )$
$BH = i$
$elseIf\ (RSSI\_node\_Id\ (i) >$
$T_{rssi\_wh} \&\& PDR\_node\_Id(i) > T_{pdr\_wh}\ )$
$WH = i$
else
$Nor\_node = i$
End if
End for

After applying all the above algorithms, the optimum results can be obtained.

Simulation Results

An algorithm that has been implemented in Matlab code and evaluated using various performance measures. The results have been presented in graphical form using different figures to show accuracy, sensitivity, detection rate, F-score, false positive rate, and false-negative rate. The X-axis represents the number of nodes used, while the Y-axis shows the number of rounds the algorithm was implemented for. The evaluation measures are presented in Table 1,2,3,4,5,6, and the results are depicted graphically in Figures 7-12. Overall, this approach provides a comprehensive evaluation of the algorithm's performance, allowing for a detailed analysis of its strengths and weaknesses. These results can be used to improve the algorithm further and optimize its performance for specific use cases.

Table 1. Accuracy

	20	40	60	80	100
Overall	94.52518	97.15385	98.89155	98.74127	98.9011
BH	84.80255	89.28571	92.89041	95.38462	97.77419
WH	80.31325	87.2069	95.59459	96.93939	97.74468

Table 2. Sensitivity

Stackelberg Game, Black hole attack, warm hole attack, backtracking search, and optimization.

	20	40	60	80	100
Overall	93.43697	95.59459	96.56098	97.2973	98.53922
BH	82.29197	85.36585	93.52326	94.30769	95.33962
WH	78.34247	83.39535	91.90909	93.58	924.5926

Table 3. Detection Rate

	20	40	60	80	100
overall	98.9	98.1	99.1	100	100
BH	97.4	98.2	98.3	100	100
WH	96.6	98.5	99.2	99.4	100

Table 4. F-Score

	20	40	60	80	100
overall	91.534	92.817	95.836	99.974	98.738
BH	79.633	94.329	94.659	90.341	94.794
WH	74.796	84.818	93.009	87.722	92.312

Table 5. False Positive

	20	40	60	80	100
overall	0.006	0.003	0.082	0.064	0.052
BH	0.005	0.001	0.073	0.014	0.002
WH	0.004	0.095	0.061	0.02	0.085

Table 6. False Negative

	20	40	60	80	100
overall	0.072	0.082	0.095	0.074	0.063
BH	0.063	0.061	0.093	0.002	0.066
WH	0.068	0.04	0.072	0.085	0.092

Fig.7. Accuracy

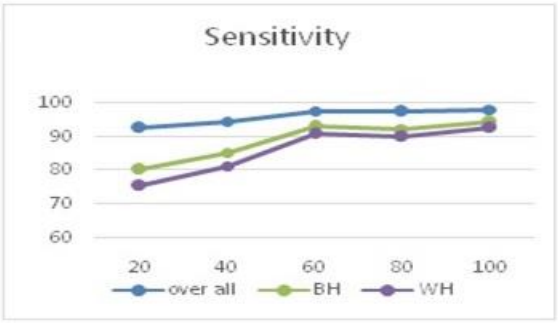

Fig.8. Sensitivity

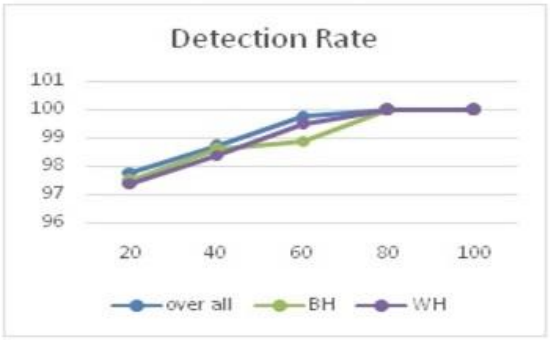

Fig.9.Detection Rate

Fig.10. F-Score

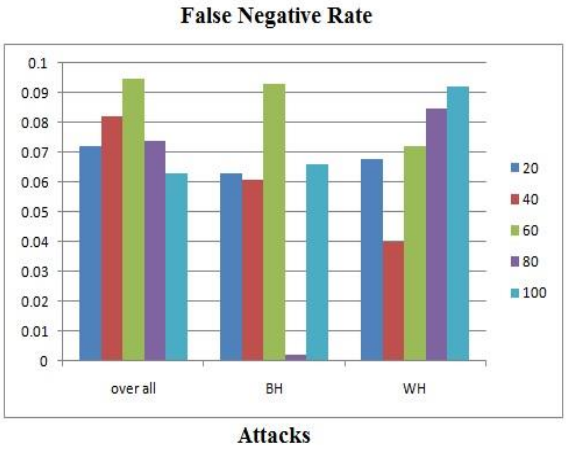

Fig.11. False Positive Rate

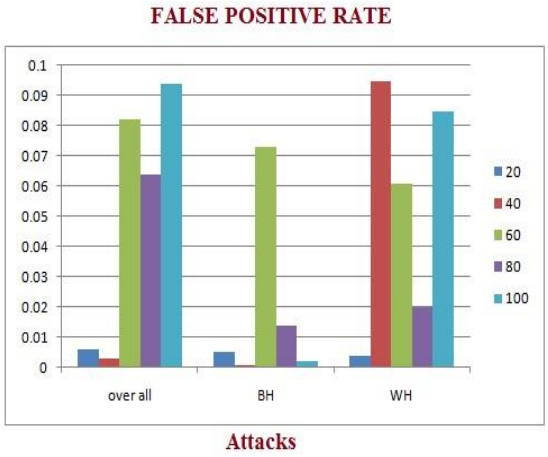

Fig.12. False Negative Rate

Stackelberg Game, Black hole attack, warm hole attack, backtracking search, and optimization.

Conclusion

The proposed game theory-based attacks identification model for improving the accuracy and detection rate in sensor networks. Security concerns are indeed a critical issue in sensor networks, and a model seems to be addressing this issue effectively.

By using a Stackelberg game model and backtracking search optimization, the proposed model offers optimal performance in detecting attacks and reducing the number of dropped packets, promoting efficiency, and increasing the security level in the network. Rapid attack detection and dealing with the communication between nodes are also some of the noteworthy outcomes of your proposed model.

It's good to know that are planning to embed the proposed game with specific routing protocols in large-scale IoT devices. This will help address accuracy detection, energy consumption, and network delay issues. Overall, the proposed model has significant potential in improving the security of sensor networks.

References

Amutha S, "Energy-efficient cluster manager-based cluster head selection technique for communication networks", Int J Commun Syst. 2020; e4427. wileyonlinelibrary.com/journal/dac © 2020 John Wiley & Sons, Ltd. https://doi.org/10.1002/dac.4427.

Jothikumar C. and Venkataraman R., "EODC: An Energy Optimized Dynamic Clustering Protocol for Wireless Sensor Networks using PSO Approach", International Journal Of Computers Communications & Control ISSN 1841-9836, e-ISSN 1841-9844, 14(2), 183-198, April 2019.

Kashif Naseer Qureshi et.al, "Optimized Cluster-Based Dynamic Energy-Aware Routing Protocol for Wireless Sensor Networks in Agriculture Precision", Hindawi Journal of Sensors Volume 2020, Article ID 9040395, 19 pages https://doi.org/10.1155/2020/9040395.

Mann, P.S., Singh, S., 2017. Energy efficient clustering protocol based on improved metaheuristic in wireless sensor networks. J. Network Comp. Appl. 83, 40-52.

Morteza Biabani, "An Energy-Efficient Evolutionary Clustering Technique for Disaster Management in IoT Networks", Sensors 2020, 20, 2647; doi:10.3390/s20092647.

Noureddine Moussa et.al, “A novel approach of WSN routing protocols comparison for forest fire Detection”, wireless networks, springer,https://doi.org/10.1007/s11276-018-18723.

Pathak Aruna, “A Proficient Bee Colony-Clustering Protocol to Prolong Lifetime of Wireless Sensor Networks”, Journal of Computer Networks and Communications Volume 2020, Article ID 1236187, 9 pageshttps://doi.org/10.1155/2020/1236187

Pradeep J. et.al, “Distributed Entropy Energy-Efficient Clustering Algorithm For Heterogeneous Wireless Sensor Network Based Chaotic Firefly Algorithm Cluster Head Selection”, Journal of Critical Reviews, Vol 7, Issue 8, 2020.

Yi-Han Xu et.al, “An Environmentally Aware Scheme of Wireless Sensor Networks for Forest Fire Monitoring and Detection”, Future Internet 2018, 10, 102; doi:10.3390/fi10100102

Wang, J., Gao, Y., Liu, W., Sangaiah, A. K., & Kim, H. J. (2019). An Improved Routing Schema with Special Clustering Using PSO Algorithm for Heterogeneous Wireless Sensor Network. Sensors, 19(3), 671.