I. J. Computer Network and Information Security, **2023, 6, 1-15**
Published Online on December 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2023.06.01

Classification of HHO-based Machine Learning Techniques for Clone Attack Detection in WSN

Ramesh Vatambeti*
School of Computer Science and Engineering, VIT-AP University, Vijayawada-522237, India
E-mail: v2ramesh634@gmail.com
ORCID iD: https://orcid.org/0000-0002-2611-4925
*Corresponding author

Vijay Kumar Damera
Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Hyderabad-500075, India
E-mail: vijay.kumar@klh.edu.in
ORCID iD: https://orcid.org/0000-0003-2445-4747

Karthikeyan H.
Computer Science and Engineering, CHRIST (Deemed to be University), Bangalore-560074, India
E-mail: karthikeyanh.a@gmail.com

Manohar M.
Computer Science and Engineering, CHRIST (Deemed to be University), Bangalore-560074, India
E-mail: manohar.m@christuniversity.in
ORCID iD: https://orcid.org/0000-0003-1110-3673

Sharon Roji Priya C.
Computer Science and Engineering, CHRIST (Deemed to be University), Bangalore-560074, India
E-mail: sharonroji@gmail.com
ORCID iD: https://orcid.org/0000-0003-1464-2090

M. S. Mekala
School of Communication Engineering, Yeungnam University, Republic of Korea, Gyeongsan-38541, Korea
E-mail: msmekala@yu.ac.kr
ORCID iD: https://orcid.org/0000-0002-1313-285X

Received: 20 November 2022; Revised: 18 January 2023; Accepted: 25 March 2023; Published: 08 December 2023

Abstract: Thanks to recent technological advancements, low-cost sensors with dispensation and communication capabilities are now feasible. As an example, a Wireless Sensor Network (WSN) is a network in which the nodes are mobile computers that exchange data with one another over wireless connections rather than relying on a central server. These inexpensive sensor nodes are particularly vulnerable to a clone node or replication assault because of their limited processing power, memory, battery life, and absence of tamper-resistant hardware. Once an attacker compromises a sensor node, they can create many copies of it elsewhere in the network that share the same ID. This would give the attacker complete internal control of the network, allowing them to mimic the genuine nodes' behavior. This is why scientists are so intent on developing better clone assault detection procedures. This research proposes a machine learning based clone node detection (ML-CND) technique to identify clone nodes in wireless networks. The goal is to identify clones effectively enough to prevent cloning attacks from happening in the first place. Use a low-cost identity verification process to identify clones in specific locations as well as around the globe. Using the Optimized Extreme Learning Machine (OELM), with kernels of ELM ideally determined through the Horse Herd Metaheuristic Optimization Algorithm (HHO), this technique safeguards the network from node identity replicas. Using the node identity replicas, the most reliable transmission path may be selected. The procedure is meant to be used to retrieve data from a network node. The simulation result demonstrates the performance analysis of several factors, including sensitivity, specificity, recall, and detection.

Index Terms: Wireless Sensor Network, Clone Attack, Horse Herd Metaheuristic Optimization Algorithm, Optimized Extreme Learning Machine, Battery Resource.

1. Introduction

The term "wireless sensor network" refers to a collection of sensors designed to share data from a monitored area through radio waves. Multiple gateway nodes will be used to relay the data. This information is intended for transfer to other systems, such as wireless Ethernet [1–2]. There are central hubs and a dispersed network of nodes. Using data transmitted via this network, we can get a better read on environmental factors like noise, humidity, and temperature [3]. The WSN is a subset of radio communication networks that features a wide variety of network topologies and addressing schemes. Varied assaults, such as the field of the assailants or the strategies that are to be employed in attacks, are to be offered for the wireless sensor network according to the different criteria [4]. Passive attacks and active attacks are the two main types of communication disruption that may be identified and classified. Software-defined networks are effective in this regard since they protect against attacks while preserving quality of service [5].

Clone Attack

The WSN is especially susceptible to the damaging clone assault. It is possible to identify clone attacks in WSN by using the following techniques: There are many centralised and decentralised methods for detecting clones in networks, such as detection of clone attacks, detecting hierarchical node replication attacks, compressed sensing-based clone identification, and fast recognition of node attacks in WSN through sequential investigation [6–8]. Conversely, in the clone node assault, the attacker first seizes a node, then creates clones or replicas of it, and finally deploys the clones at key locations across the WSNs.

Worryingly, attackers are so well-informed that they may quickly impersonate legitimate nodes and communicate with the freshly made clones. It's possible that this is the reason why the traditional secured routing system [9, 10] and validation techniques [11, 12] can't evaluate the impact of the clone or take steps to mitigate it. There have been several ineffective approaches described for finding clone nodes. As such, this study focuses mostly on distributed witness node-based approaches, which are the lone exception and have shown promising results so far. However, there are some drawbacks to these methods as well, including the fact that they select witness nodes in a predetermined fashion, that witness nodes aren't evenly distributed (the crowded centre problem), and that there's a trade-off between high detection likelihood and high communication and memory costs. Therefore, the purpose of this study was to develop an OELM model for predicting clone attacks where the kernel parameters are improved by HHO to boost the efficacy of ELM. In Section 2, we examine pre-existing methods, and in Section 3, we explain the clone attack using the suggested model. In Section 4, we give the results of our comparison of the proposed model to the state-of-the-art approaches in terms of a number of validation measures. Section 5 then illustrates the paper's scientific significance and potential for future research.

2. Related Works

In [13], Anitha S. et al. offer security techniques for IHM that may efficiently identify replication attacks and secure the system. Possible uses of the presented approaches, such as the EMABRD algorithm, the Secured Ant Colony Optimization (SACOP) algorithm, and the Fingerprint-based Zero Knowledge Authentication (FZKA) algorithm, are demonstrated in a real-time setting. Among the three methods, SACOP has the highest detection probability of malicious nodes, but it comes at the cost of greater storage and communication overheads compared to EMABRD and FZKA. In terms of detection probability, FZKA outperforms EMABRD, albeit at the expense of greater overhead. Therefore, EMABRD has lower overheads than SACOP but higher detection probabilities.

In [14], Devi, P.P. presents a reliable method for identifying clone nodes and classifying them accordingly. At first, undesirable information is filtered out by pre-processing the input data and normalising the pre-processed data. The best traits are chosen from this pool and used in the categorization procedure. The adapted particle swarm optimization approach is used to optimise the pre-processed data for maximum accuracy (MPSO). The K-means clustering procedure is then used to classify the information into groups. This MPSO and a modified artificial neural network classifier are used in the training process (MANN). Next, we use the MANN method to identify the optimised and trained characteristics as either normal or malicious, therefore detecting and labelling the clone assault. Also, the Trust Aware Intense Algorithm may be used to spot potential clones in a network (TAIGBRFCNIA). Last but not least, the effectiveness of the system is demonstrated by a study of both the suggested and current methodologies.

Meganathan, S et al. [15] investigated the security protocol to prevent the clone node from launching any successful attacks in the network. The security protocol is constructed such that the attacker cannot obtain the necessary cryptographic data. To accomplish this, we store unfinished cryptographic data in memory that can be completed by a trustworthy node. This node will use geolocation data to produce full cryptographic key material.

Clone node attack finding in stationary WSNs is the topic of Mohammad, S et al. [16]. We have shown that the sensor nodes are vulnerable to assaults like clone node attacks because of the characteristics of WSNs, such as their short battery life, limited processing, and lack of tamper-resistant hardware, memory, etc. Several methods have been developed to counteract this type of cyberattack, including the centralised detection strategy and the dispersed detection methodology. In addition to these major categories, we have also covered others that are often thought to fall under them, such as key

establishment, node-to-network distribution, clustering-based, witness node-based approaches, key usage-based, base station-based, neighbour ID-based, etc.

Therefore, the study presented by Devi, P.P., et al. [17] aims to employ an SDN-based mechanism to perform network-level analysis methodologies with the goal of identifying and avoiding duplicate nodes created by a cloning assault at a low cost and in a timely manner. Therefore, cyber-security applications built on top of a software-defined network are invaluable here. When applied to WSN, this SDN-based technique aids in the upkeep and enhancement of QoS restrictions. Finding a clone node in a wireless network is made easier with the aid of hybrid clone node detection (HCND) technology. The goal is to identify clones effectively enough to prevent cloning attacks from happening in the first place. Use a low-cost identity verification process to identify clones in specific locations as well as around the globe. The wireless sensor network may be shielded from identity forgeries thanks to this technique, which makes use of a superimposed SDIS junction code. Using the node identity replicas, the most reliable transmission path may be selected. For data retrieval from network nodes, the layered approach is recommended. As a preventative measure against a clustered assault launched from the clones, their removal is advisable. The simulation results demonstrate that several metrics, including the ratio, accuracy, recall, and detection, have been analysed for performance.

In order to identify a clone node assault on mobile IoT networks, Hameed et al. [18] propose a strategy that takes advantage of the semantic information of IoT devices, also known as context information, to safely pinpoint their locations. To expedite the confirmation process at selected trusted nodes, we build the location proof method by combining location proofs with an algorithm. We also provide a model for selecting reliable IoT devices according to their profile capabilities, allowing them to be selected above the other proof-verification processes. The performance analysis and practical findings show that our suggested approach greatly decreases the computational, communication, energy, and storage overheads compared to previous research while maintaining a high level of finding accuracy with little detection time.

3. Clone Node Attack

In general, there are two kinds of WSNs: stationary ones and mobile ones. Unlike mobile WSN, in which nodes are not constrained to a single location after deployment, sensor nodes in static WSN do not alter locations after they have been set up. Simply put, mobile WSNs rely on dynamic routing to distribute data, whereas static WSNs rely on flooding or fixed routing. Clone node attacks affect both types of WSN.

One of the most dangerous kinds of assaults on WSNs is the clone node attack. First, the attacker isolates and takes control of a legitimate node, and then, in under a minute, they use specialized tools to steal the node's credentials. The attacker then uses the credentials to make clones, which are subsequently distributed over the network and used to launch internal assaults such as wormhole attacks [19]. The attacker can then take over the network and possibly even abolish the node withdrawal mechanism [20] by isolating the obtained lawful node from the network and implementing the clones.

As a result, detecting clones quickly is crucial for limiting collateral damage, but doing so is challenging owing to features like nodes having legal identifiers, information, etc. In contrast to mobile WSN, however, it is simpler to identify if there are any cloned or replica sensor nodes in a static WSN due to their stationary placements. In most cases, this is checked by checking to see if the same logical ID is used by more than one legal node in the network. However, with mobile WSNs, the issue is different, since the nodes travel throughout the network, making it impossible to verify that there is a clone if the ID is retrieved again at a different location, as the node may be moving. Methods for identifying clones in mobile WSNs are discussed in more depth in [21]. First, an adversary must physically seize control of the legitimate nodes in the network.

- The attacker then takes possession of the node's private credentials (IDs, information, data, etc.).
- After gathering this data, the attacker can create new nodes with the legitimate IDs of the nodes they stole.
- Mounting the clone nodes in strategic locations around the network is the final stage.

Once the above steps have been taken, the attacker is free to use the clones in any way they see fit, including further internal attacks on the network.

3.1. Proposed System

To identify a clone node in a wireless network, the ML-CND technique is used. When it comes to applying security attack techniques and, by extension, maintaining or improving QoS limitations, an ML-based paradigm is significantly more effective. The goal is to identify clones effectively enough to prevent cloning attacks from happening in the first place. Use a low-cost identity verification process to identify clones in specific locations as well as around the globe. This technique uses OELM to prevent spoofing in a wireless sensor network by impersonating other nodes. Using the node identity replicas, the most reliable transmission path may be selected. Information must be retrieved from each node in the network using the OELM technique. As a preventative measure against a clustered assault launched from the clones, their removal is advisable. Its suggested model's flow is depicted in Fig. 1.

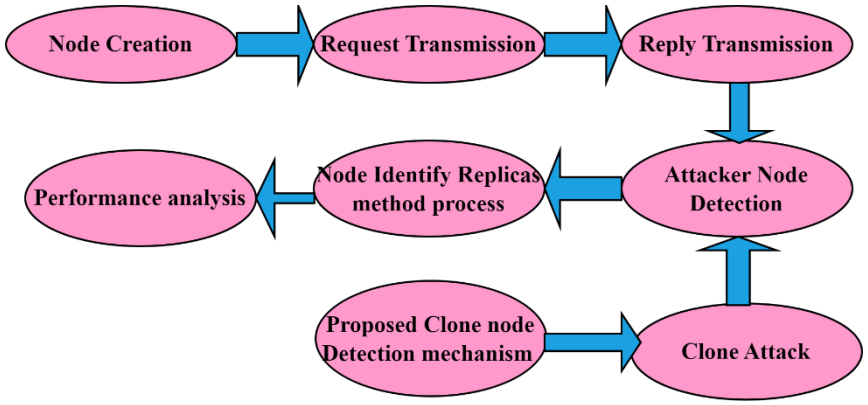

Fig.1. Projected flow diagram

3.2. Detecting the Clone Attack

After capturing the node, an attacker can re-program it and even make a copy of it. Each and every part of the network is going to get its own copy of the clone. As a result of the prevalence of clone nodes, assaults on sensor networks tend to have a negative impact on overall efficiency. Sensor nodes in a WSN are expected to be relocated on their own when deployment is complete. One might use a centralized or decentralized approach to solving problems. When a new node enters a network, the centralized method's node detection has to be duplicated. Distributed methods aid in the detection of clone nodes, the detection of which is reliant on the location information for a node being saved on the network. For a network to function, its nodes must be moved successively around the network, and several methods exist for identifying and eliminating duplicate nodes in a static network.

3.3. Key Distribution Procedure of Clone Attack

The evaluation of the cryptographic secret key between the accelerating Alice and the designating Bob is made possible thanks to the key distribution procedure. This encoding method is dependent on an optical process that occurs in a quantum state, and it is meant to be used for the safe transmission of sensitive data. The channel-like depolarization is defined below using the quantum state transition unit operator Dc, which we symbolize as

$$D_c(|\varphi) = (1-c)|\varphi)(\varphi| + \frac{c}{3}\sum_{p=1}^{3}\sigma_p|\varphi)(\varphi|\sigma_p) \tag{1}$$

where c is the depolarizing station limitation and σ_p is the Pauli metrics (p = x, y, z).

Below, we provide an assessment of the bit error in the absence of the attacker, which depends on the depolarizing value.

$$Quant_{BER} = 2c/3 \tag{2}$$

Based on the evaluation below, we can determine how this BER relates to the channel transparency $Vt\ as\ Quant_{BER}$,

$$Vt\ as\ Quant_{BER} = (1-Vt)/2 \tag{3}$$

The $Ak_p(p = 1, 2, \dots, RN)$, each attacker Ak_p clones with a cloning transformation C_p and A is the Alice,

$$C_p(|0)A|0)Ak_p = |0)Ak_p|0)Ak_p \tag{4}$$

and

$$C_p(|1)A|0)Ak_p = |1)Ak_p|1)Ak_p$$

Ak_p will use C_p in the basis out distinct as shadows,

$$C_p(|0)\ out_A\ |0)out_{Ak} = |0)out_{Ak}|0)out_{Ak_p} \tag{5}$$

$$C_p(|0)\ \ out_A\ |0)out_{Ak} = \cos(\theta_p)\,|1)out_A|0\ out_{Ak_p} + \sin(\theta_p)\,|0)out_A|1)out_{Ak_p} \tag{6}$$

Here $\theta_p(0 \le \theta_p \le \frac{\pi}{2})$ is the viewpoint of attack of the attacker Ak_p

The photon that was previously part of Ak p's state space is retained after the cloning assault. Depolarizing channel model applications allow the transmission of information between Alice and the attacker while maintaining the

polarization of individual photons.

3.4. ML-based Clone Node Detection Process (ML-CND)

To train a single hidden layer feed-forward ANN, the extreme learning machine (ELM) uses a learning technique that randomly produces input weights and biases. The weights for the final output are then calculated analytically. Traditional ANN is slower than ELM because of the need to tune all network parameters, but the main differences are that ELM can be used with non-differentiable or discrete transfer functions and that it does not require the optimization parameters that are crucial of ANN.

When x_i represents an input, y_k represents an output, and m neurons reside in the hidden layer, n reside in the input layer, and k represents the total number of outputs, as indicated by the following Eqn.

$$y_k = \sum_{j=1}^{m} \beta_{j,k} g(\sum_{i=1}^{m} w_{i,j} x_i + b_j) \tag{7}$$

where $w_{i,j}$,is input weights, $\beta_{j,k}$ is output weights, b_j is threshold values of the neurons in the hidden layer and $g(.)$ is the activation function. Eqn. 7 can be written as follows;

$$H\beta = y \tag{8}$$

where H can be expressed as follows.

$$H(w_{i,j}, b_j, x_i) = \begin{bmatrix} g(w_{1,1}x_1 + b_1) & \cdots & g(w_{1,m}x_m + b_m) \\ \vdots & \ddots & \vdots \\ g(w_{n,1}x_n + b_1) & \cdots & g(w_{n,m}x_m + b_m) \end{bmatrix} \tag{9}$$

where, H is hidden layer output matrix. Since the input weights $(w_1 \cdots n, 1 \cdots m)$, bias $(b_{1\ldots,m})$ and the inputs from $(x_{1\ldots,n})$ and the outputs $(y_{1\ldots,k})$ train dataset are known and the only unknown parameters are output weights $(\beta_{1\ldots.m,1\ldots k})$ then output weights can be found by Moore–Penrose generalized inverse method. Such as:

$$\hat{\beta} = H \dagger y \tag{10}$$

where $H\dagger$ is the generalized Moore–Penrose widespread inverse matrix of H. The output weights can be optimally selected by using HHO, which is described as follows.

Horse Herd Optimization Algorithm

Metaheuristic algorithms have been used to address many types of optimization issues in recent years. This is due, in part, to the fact that metaheuristic algorithms may be used to mathematically represent and solve a wide variety of practical issues. The goal of this research was to use a unique metaheuristic algorithm to address the issue of feature selection for spam email detection. That's why the Horse Herd Optimization Algorithm (HHO) was the main tool employed. MiarNaeimi et al. [22] propose HHO, a strong metaheuristic algorithm motivated by the horses' herding tendencies across age groups. HHO has remarkable performance in tackling complicated high-dimensional issues because of the large sum of control elements based on the behaviour of horses of varying ages. Using standard test functions, we were able to assess its performance at high dimensions (up to 10,000) and find that it is highly effective at both exploration and exploitation. It beats several well-known metaheuristic optimization algorithms in terms of accuracy and efficiency, and it has the capacity to identify the optimal solution in the quickest time, at the lowest cost, and with the least amount of difficulty. In the next part, we'll go deeper into the specifics of this algorithm.

Age-related changes in horse behaviour are evident [22]. A horse may live for up to 30 years at the very most. In HHO, horses are split into four age groups denoted by zero through five, five through ten, ten through fifteen, and sixteen and up, respectively. The herd orientation algorithm (HOA) models the social lives of horses based on six broad characteristics typical of those ages. "Grazing," "hierarchy," "sociability," "imitation," "defence mechanism," and "roaming" are all examples of such behaviours.

At each repetition, the horse's motion is described by Eqn. (11).

$$X_m^{Iter,AGE} = \vec{V}_m^{Iter,AGE} + X_m^{(Iter-1),AGE}, \ AGE = \alpha, \beta, \gamma, \delta \tag{11}$$

where $X_m^{Iter,AGE}$ is the site of the m[th] horse, $\vec{V}_m^{Iter,AGE}$ is the velocity vector of the m[th] iteration.

Each iteration of horse age estimation should have a comprehensive response matrix. The best answers are used to sort the matrix, and the first 10 horses are selected at random. The next 20%, 30%, and 40% of the remaining horses were categorized as, and. All six of these behaviors are modelled mathematically, and their simulated steps are used to detect the velocity vector. Eqn. (12) [22] describes the motion vector of horses of varying ages throughout each iteration of the procedure.

$$\vec{V}_m^{Iter,\alpha} = \vec{G}_m^{Iter,\alpha} + \vec{D}_m^{Iter,a}$$
$$\vec{V}_m^{Iter,\beta} = \vec{G}_m^{Iter,\beta} + \vec{H}_m^{Iter,\beta} + \vec{S}_m^{Iter,\beta} + \vec{D}_m^{Iter,\beta}$$
$$\vec{V}_m^{Iter,\gamma} = \vec{G}_m^{Iter,\gamma} + \vec{H}_m^{Iter,\gamma} + \vec{S}_m^{Iter,\gamma} + \vec{I}_m^{Iter,\gamma} + \vec{D}_m^{Iter,\gamma} + \vec{R}_m^{Iter,\gamma}$$
$$\vec{V}_m^{Iter,\delta} = \vec{G}_m^{Iter,\delta} + \vec{I}_m^{Iter,\delta} + \vec{R}_m^{Iter,\delta} \quad (12)$$

In particular, HHO takes cues from the six general and social behaviors that horses exhibit across their lifespan. Here, we'll go over the math behind those six behaviors.

Grazing: In all phases of their life, horses spend between 16 and 20 hours a day grazing. This behavior in HHO is modelled mathematically by Equations (13) and (14), which can be found in [22].

$$\vec{G}_m^{Iter,AGE} = gIter(\breve{u} + \breve{p}),\ AGE = \alpha, \beta, \gamma, \delta \quad (13)$$

$$g_m^{Iter,AGE} = g_m^{(iter-1),AGE} \times \omega_g \quad (14)$$

For the aforementioned equations, $\vec{G}_m^{Iter,AGE}$ represents the grazing propensity of the ith horse. This factor, _g, decreases linearly with each repetition. If you want to maximise your grazing area, set u to 1.05, the optimal value. The optimum value for l, the lower limit of the grazing space, is 0.95. I is an arbitrary integer between zero and one. It is suggested that the g coefficient be fixed at 1.5 across the board.

Hierarchy: Horses are herd animals that need a leader to follow. This leader might be a human, an adult stallion, or a mare. In a herd of horses, the leader is usually the oldest and strongest animal. Between the ages of five and fifteen (or/ and), horses were found to adhere to the hierarchy law. To put it mathematically, the hierarchy is built using Eqns. (15) and (16) below [22]:

$$\vec{H}_m^{Iter,AGE} = h_m^{Iter,AGE}[X_*^{(Iter-1)} - X_m^{(Iter-1)}] AGE = \alpha, \beta, \gamma \quad (15)$$

$$h_m^{Iter,AGE} = h_m^{(iter-1),AGE} \times \omega_h \quad (16)$$

where $\vec{H}_m^{Iter,AGE}$ is the impact of the site velocity, and $X_*^{(iter-1)}$ designates the site of that horse.

Sociability: HHO also encouraged a sociable nature in horses. Horses benefit from interacting with other animals and may live peacefully alongside them. This improves their odds of survival even further. There are horses that seem to be fine even around cattle and sheep. In horses, this pattern of behavior emerges between the ages of five and fifteen. Equations (17) and (18) [22] are used to implement the socialization in HHO, which is defined as the movement toward herd.:

$$\vec{S}_m^{Iter,AGE} = S_m^{Iter,AGE}\left[\left(\frac{1}{N}\sum_{j=1}^{N} X_j^{(Iter-1)}\right) - X_m^{(Iter-1)}\right] AGE = \beta, \gamma \quad (17)$$

$$S_m^{Iter,AGE} = s_m^{(Iter-1),AGE} \times \omega_s \quad (18)$$

where $\vec{S}_m^{Iter,AGE}$ is the ith horses' motion, and $S_m^{Iter,AGE}$ is the similar horse's orientation towards the herd at the Iterth iteration. $S_m^{Iter,AGE}$ decreases with every iteration by a factor of s. N is ages represented by those horses. In the sensitivity analysis, the s coefficient for and horses is determined.

Imitation: In the same way that humans learn from imitation, horses pick up both positive and negative traits from their peers [7]. The other HHO-inspired horse behaviour is this kind of mimicry. Young horses, even later in life, continue to try to mimic their elders. We can characterize the imitation with Eqns. (19) and (20) [22]:

$$\vec{I}_m^{Iter,AGE} = i_m^{Iter,AGE}\left[\left(\frac{1}{pN}\sum_{j=1}^{pN} \hat{X}_j^{(Iter-1)}\right) - X^{(Iter-1)}\right] AGE = \gamma \quad (19)$$

$$i_m^{Iter,AGE} = i_m^{(Iter-1),AGE} \times \omega_i \quad (20)$$

I _m (Iter,AGE) represents the ith horse's motion vector in the direction of the best horses' average at X in the aforementioned formulae. It is recommended that p be set to 10% of the total number of horses in the herd, and pN displays the total sum of horses that have the best locations. Each cycle has a reduction factor, denoted by ω_i for i_{iter}.

Defense: It's important for horses to be able to protect themselves, thus they exhibit the "fight or flight" response. They want to get away from it at first. They are also known to buck when cornered. The competition for resources, like as food and water, helps to maintain the status quo of rivalry. They battle not only to protect themselves against predators like wolves, but also to avoid them altogether. The other type of behavior utilized in HHO is the horses' defensive mechanism, which is typified by the horses' flight from situations when their reactions are suboptimal. The defensive mechanism is described by the following eqns. [22]:

$$\vec{D}_m^{Iter,AGE} = -d_m^{Iter,AGE}\left[\left(\frac{1}{qN}\sum_{j=1}^{pN}\check{X}_j^{(Iter-1)}\right) - X^{(Iter-1)}\right] AGE = \alpha,\beta,\gamma \tag{21}$$

$$d_m^{Iter,AGE} = d_m^{(Iter-1),AGE} \times \omega_d \tag{22}$$

$d_m^{Iter,AGE}$ denotes "the escape vector of ith horse from the average of several horses with worst positions, which are indicated by the X vector," in the aforementioned equations. It may be expressed as qN, where q is the number of horses in the poorest positions. 20 percent of the total number of horses is a good amount to use for q. This cycle-by-cycle reduction factor for d^{Iter} is denoted by ω_d.

Roaming: HHO mimics all save the horse's tendency of wandering. Wild horses, if not confined to stables, will wander from one pasture to another in search of forage. It's possible for a horse to switch grazing areas at the last minute. Horses are naturally inquisitive animals who go from pasture to pasture to learn about their environment. Eqns. (23) and (24) characterize the Roaming behavior, which is defined as the erratic motion of an individual horse within the herd.

$$\vec{R}_m^{Iter,AGE} = r_m^{Iter,AGE} pX^{(Iter-1)}, AGE = \gamma \; and \; \delta \tag{23}$$

$$r_m^{Iter,AGE} = r_m^{(Iter-1),AGE} \times \omega_r \tag{24}$$

$\vec{R}_m^{Iter,AGE}$ is ‘‘the In order to do a local search and to break out of a local minimum, the ith horse's velocity vector is chosen at random. The cycle-by-cycle reduction factor of $r_m^{Iter,AGE}$ is denoted by ω_r.

You can figure out the horses' average speed by replacing Eq. (13) with the results from Eqns. (21) through (24). (12). Equations (25)-(26) may be used to determine the velocities of horses at ages 1, 2, 3, and 4. (28).

$$\vec{V}_m^{Iter,\delta} = \left[g_m^{(Iter-1),\delta}, \omega_g\left(\breve{u} + \rho\breve{l}\right) + \left[X_m^{(Iter-1)}\right]\right] + \left[i_m^{(Iter-1),\delta}\omega_i\left[\left(\frac{1}{pN}\sum_{j=1}^{pN}\hat{X}_j^{(Iter-1)}\right) - X^{(Iter-1)}\right]\right] + \left[r_m^{(Iter-1),\delta}\omega_r pX^{(Iter-1)}\right] \tag{25}$$

where $\vec{V}_m^{Iter,\delta}$ m is the d horses’ velocity (horses at the age of 0–5).

$$\vec{V}_m^{Iter,\delta} = \left[g_m^{(Iter-1),\gamma}\omega_g\left(\breve{u} + \rho\breve{l}\right) + X_m^{(Iter-1)}\right] + \left[h_m^{(Iter-1),\gamma}\omega_h\left[X_*^{(Iter-1)}\right]\right] + \left[s_m^{(Iter-1),\gamma}\omega_s\left(\frac{1}{N}\sum_{j=1}^{N}X_j^{(Iter-1)}\right) - X_m^{(Iter-1)}\right] + \left[i_m^{(Iter-1),\gamma}\omega_i\left[\left(\frac{1}{pN}\sum_{j=1}^{pN}\hat{X}_j^{(Iter-1)}\right) - X^{(Iter-1)}\right]\right] - \left[d_m^{(Iter-1),\gamma}\omega_d\left[\frac{1}{qN}\sum_{j=1}^{pN}\begin{matrix}(Iter-1)\\ X\\ j\end{matrix}\right] - X^{(Iter-1)}\right] + \left[r_m^{(Iter-1),\gamma}\omega_r pX^{(Iter-1)}\right] \tag{26}$$

where $\vec{V}_m^{Iter,\gamma}$ is the γ horses’s speed (the age of 5–10).

$$\vec{V}_m^{Iter,\beta} = \left[g_m^{(Iter-1),\beta}\omega_g\left(\breve{u} + \rho\breve{l}\right) + \left[X_m^{(Iter-1)}\right]\right] + \left[h_m^{(Iter-1),\beta}\omega_h\left[X_*^{(Iter-1)} - X_m^{(Iter-1)}\right]\right] + \left[s_m^{(Iter-1),\beta}\omega_s\left[\frac{1}{N}\sum_{j=1}^{N}X_m^{(Iter-1)}\right]\right] - \left[d_m^{(Iter-1),\beta}\omega_d\left[\left(\frac{1}{qN}\sum_{j=1}^{pN}\begin{matrix}(Iter-1)\\ X\\ j\end{matrix}\right) - X^{(Iter-1)}\right]\right] \tag{27}$$

where $\vec{V}_m^{Iter,\beta}$ is the β horses’ velocity (horses at the age among 10 and 15 years).

$$\vec{V}_m^{Iter,\alpha} = \left[g_m^{(Iter-1),\alpha}\omega_g\left(\breve{u} + \rho\breve{l}\right) + \left[X_m^{(Iter-1)}\right]\right] - \left[d_m^{(Iter-1),\alpha}\omega d\left[\left(\frac{1}{qN}\sum_{j=1}^{pN}\begin{matrix}(Iter-1)\\ X\\ j\end{matrix}\right) - X^{(Iter-1)}\right]\right] \tag{28}$$

where $\vec{V}_m^{Iter,\alpha}$ is the horses' velocity.

The results confirmed HHO's ability to handle multifaceted problems with many open variables in high-dimensional spaces. The adult horse starts a highly accurate global optimum. When an adult a horse is in the area, the c horses are less interested in approaching the horses, but the horses are always on the lookout for other near situations so that they can approach the horses. They are highly motivated to travel to new places and learn about the best places in the world to live.

4. Results and Discussion

Witness Attack Detection is modelled in NS2. Table 1 summarizes the simulation parameters and settings.

Table 1. Simulation settings

Parameters	Values
Initial Energy	10.3J
Transmission Power	0.660
Getting Power	0.395
Sum. of Nodes	200
Area Size	500m X 500m
Simulation Time	50 sec
Traffic Source	CBR
Packet Size	512
Rate	50,100,150,200 and 250kb
MAC	IEEE 802.11
Routing Protocol	AODV
Communication Range	250m
Attackers	1,2,3,4 and 5

4.1. Performance Metrics

Results from both the proposed ML-CND algorithm and the system's implementation of the existing technique known as TAIGBRFCNIA [14] are averaged. The two protocols are compared and contrasted in terms of their detection.

4.2. Average Detection Delay (ADD)

It's the sum of time taken by the nodes to convey the data packets.

$$ADD = \sum_{\forall i} \sum_{\forall j} \left(\frac{Td_{ij} - Ts_{ij}}{n} \right) \tag{29}$$

where, Td_{ij} attack detected time of j[th] packet of node i, Ts_{ij} packet distribution period of packet j for node I and n - at node i.

4.3. Average Packet Delivery Ratio (APDR)

The ratio among the sum of packets sent and conventional.

$$APDR = \sum_{\forall i,j} \frac{N_{rj}}{N_{Si}} \tag{30}$$

where, N_{rj} No. of packets conventional at each destination (j) and N_{Si} No. of packets directed from.

4.4. Communication Overhead (CO)

It's overall amount of control packets swapped by the overall quantity of received packets. Each source (i).

$$CO = \sum_{\forall i,j} \left(\frac{Np_{rj}}{N_{ri}} \right) \tag{31}$$

where, Np_{rj} sum of routing packets received and N_{ri} sum of conventional data packets.

4.5. Average Energy Consumption (AEC)

This is defined as nodes N_{i} over the period of time $j = 1,2 \ldots n$

$$AEC = \sum_{\forall i} \sum_{\forall j} \left(\frac{EN_{ij}}{n} \right) \quad (32)$$

where, EN_{ij} - energy consumed by N_i of time j and n - No. of nodes spent the energy

4.6. Case-1 Results for Varying Number of Attackers

Table 2. Assessment of existing structure with proposed model

Cloned Nodes	Delay (Sec)		Packet Delivery Ratio (%)		Packet Drop (Packets)		Energy (Joules)	
	OELM	TAIGBRFCNIA	OELM	TAIGBRFCNIA	OELM	TAIGBRFCNIA	OELM	TAIGBRFCNIA
1	1.755	5.856	0.565	0.467	1113	3872	8.37	9.092
2	1.836	5.190	0.760	0.587	456	3784	7.93	9.18
3	1.895	5.630	0.724	0.559	846	3868	8.47	9.19
4	2.210	7.707	0.671	0.370	424	6666	7.91	9.99
5	2.351	8.510	0.673	0.250	327	8435	7.82	10.0

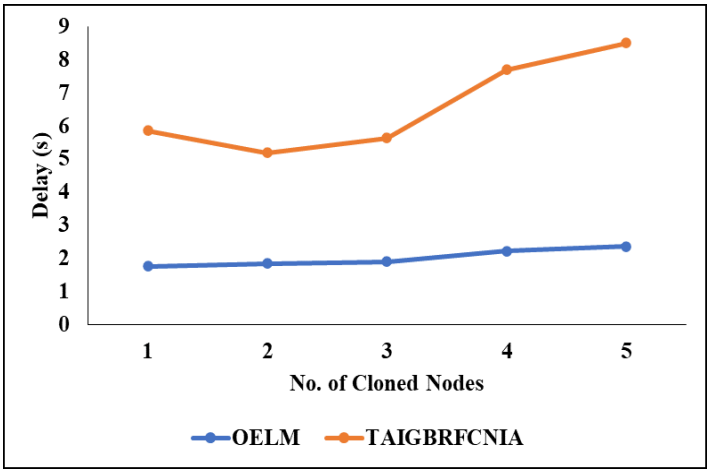

Fig.2. Delay comparison

Fig. 2 depicts the time lag assessed for both the current and proposed methods as the number of attackers is varied. It's obvious that more attackers mean a longer period before they're discovered. Delays in both the proposed and existing models increase; for the former, it goes from 1.75 to 2.35 seconds, and for the latter, it goes from 5.8 to 8.5 seconds. The detection delay of OELM is 68% less than that of the existing technique, which does not use monitoring nodes, because in OELM nodes.

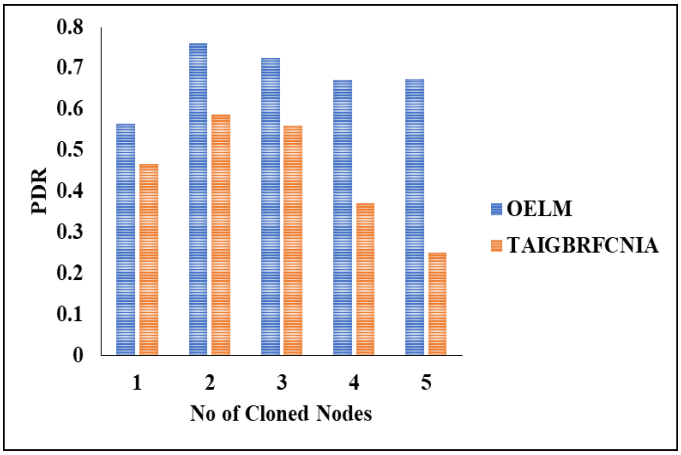

Fig.3. PDR comparison

As shown in Fig. 3, when the number of attackers is varied, the PDR for both the current and proposed models shifts accordingly. More packet dropouts are the result of an increase in attackers, lowering the delivery ratio. Figure 3 displays the consistent decline from 0.76 to 0.67 in OELM's delivery ratio. However, for up to three attackers, the delivery ratio

of existing techniques consistently decreases from 0.56 to 0.55. Beyond three attackers, it drops precipitously to 0.37 and 0.25 per target. This is because when there are more than three attackers, they are able to share information about their attacks and create more compromised nodes. Fig. 4 depicts the packet drop assessed for various strategies following an alteration in the number of attackers.

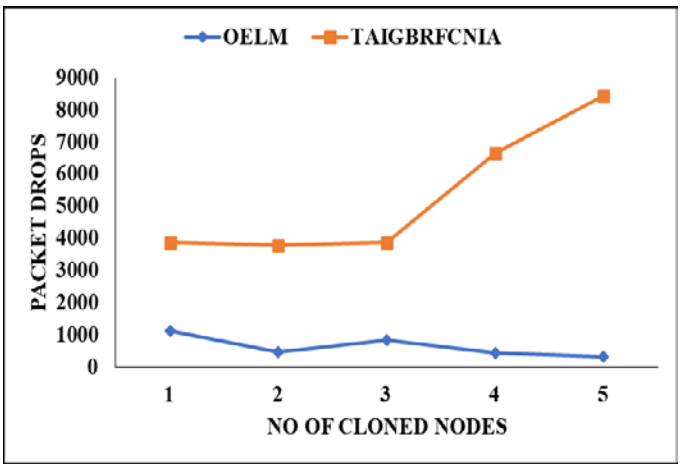

Fig.4. Packet drops comparison

Since there are now more potential targets, the rate of packet loss has increased. OELM experiences a consistent rise in packet loss, from 313 to 2327. In contrast, with the current method, the number of dropped packets steadily rises from 3772 to 3868 for a maximum of three attackers. As soon as there are more than three attackers, the number jumps to 6666 packets, and then 8435. This is because when there are more than 3 attackers, they are able to share knowledge about their attacks and create additional compromised nodes. Fig. 5 shows the average amount of energy used for each strategy after the number of attackers has been varied.

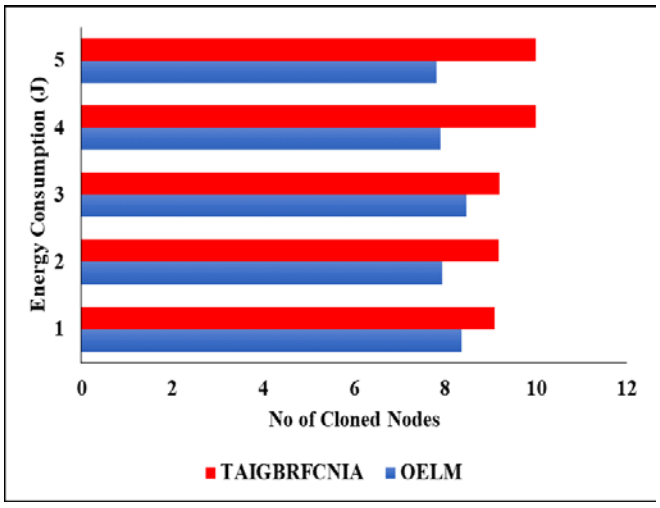

Fig.5. Energy consumption

Due to an increase in the number of attackers, there will be more packet exchanges and verification operations, which will increase energy usage. The energy needed to run OELM grows from 7.37 to 7.82 joules, whereas the energy needed to run the current approach goes from 9.09 to 10.5. With OELM, the impact of clone attacks is mitigated and the number of monitoring nodes is decreased. As a result, it uses 19% less power than the TAIGBRFCNIA method.

Case-2 Results for Communication Rate

In the second experiment, we try out several values for the data transfer rate, from 50 to 250Kb. The outcomes of adjusting the transmission rate using the OELM and TAIGBRFCNIA methods are displayed in Table 3.

Table 3. Transmission rate

Cloned Nodes	Delay (Sec)		Packet Delivery Ratio (%)		Packet Drop (Packets)		Energy (Joules)	
	OELM	TAIGBRFC NIA	OELM	TAIGBRFC NIA	OELM	TAIGBRFC NIA	OELM	TAIGBRFC NIA
50	1.83	5.190	0.960	0.587	456	3784	7.93	9.18
100	5.56	7.205	0.578	0.271	2414	10492	7.84	9.52
150	7.11	8.236	0.3571	0.152	5620	18406	8.08	9.39
200	8.18	10.884	0.226	0.112	9046	26201	7.93	9.84
250	10.54	13.250	0.0907	0.059	15671	35759	8.038	10.02

The impact of a transmission rate shift on the Detection Delay for the present and OELM methods is depicted in Fig. 6. Unfortunately, increased traffic loads are a side effect of faster transmission rates. The data depicts an increase in detection latency from 5.1 seconds to 13.2 seconds for DDCA, and from 1.8 seconds to 10.5 seconds for OELM. The OELM approach is 29% quicker than the conventional method since only the monitoring nodes are responsible for confirming claims. The evaluation of the packet delivery ratio for the existing and OELM methods after the rate change is depicted in Fig. 7. The delivery ratio drops as the data transfer rate increases because to buffer overflow. From 0.58 to 0.05 and 0.96 to 0.09, respectively. Due to its ability to detect cloning attacks, OELM improves delivery ratio by 47% compared to the industry norm. For both the baseline and OELM techniques, the results of a study of packet drop at the increased rate are shown in Fig. 8.

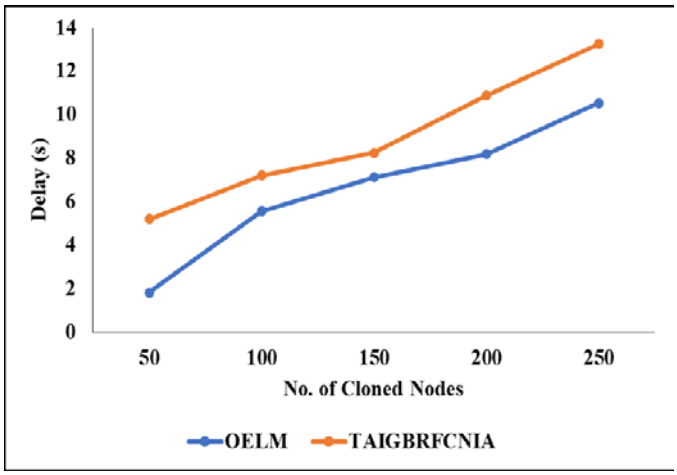

Fig.6. Delay rate

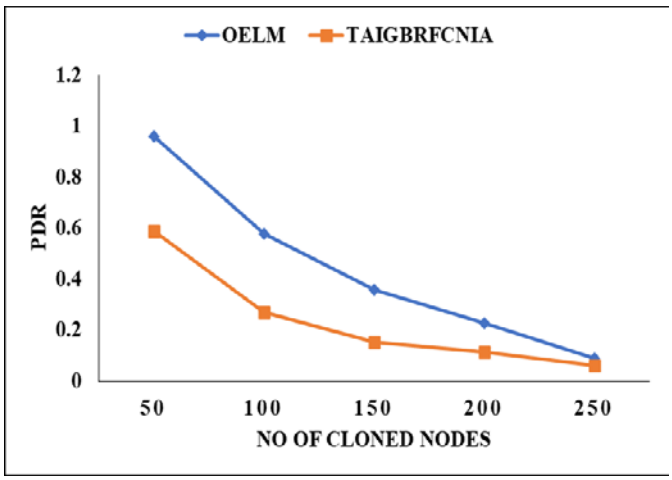

Fig.7. Packet delivery ratio (PDR)

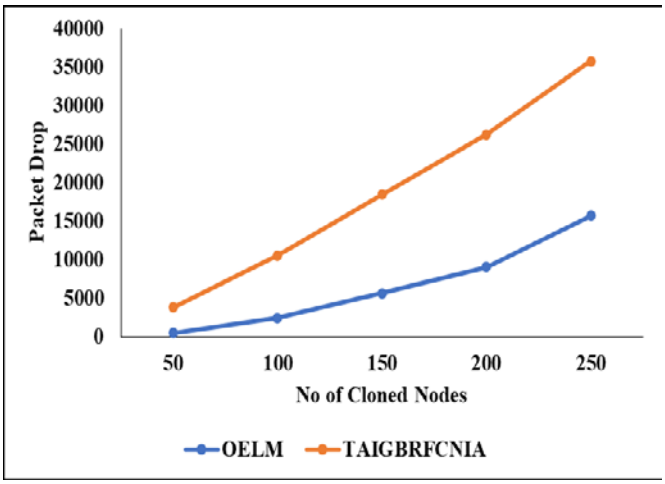

Fig.8. Packet drop

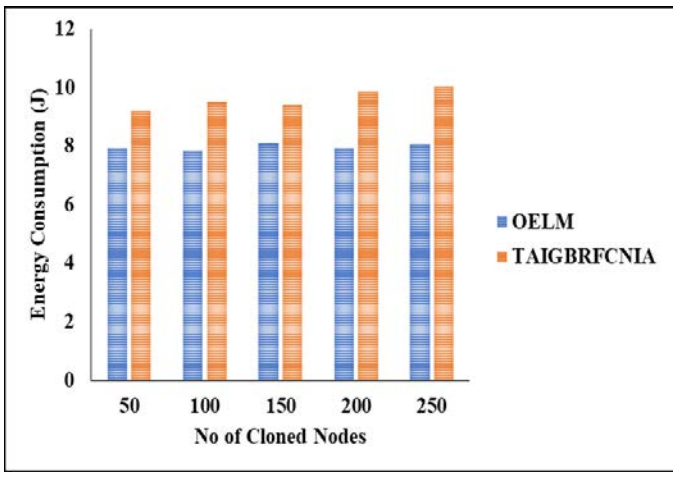

Fig.9. Energy consumption

Increased throughput leads to buffer overflow and more missed packets. The existing packet losses have grown from 3784 to 35759, while OELM packet drops have increased from 456 to 15671 throughout the course of the observations. Since OELM reduces packet dropouts due to clone attacks, it has 71% lower packet loss than the present technique. Because of the increased pricing, there has been a little rise in energy consumption due to the higher amount of traffic. Fig. 9 displays an increase from 9.18 to 10.13 joules in total energy consumption for the current building, and an increase from 7.9 to 8.03 joules for the OELM structure. OELM requires fewer monitoring nodes and is less vulnerable to clone attacks than conventional logging approaches. This means that OELM saves 17% of the energy required by the conventional approach.

5. Conclusions

An ML-based clone node detection (ML-CND) approach that uses OELM may be used to locate a clone node in a wireless network. To stop cloning assaults, OELM will be used as a cloud-based platform for efficient clone identification. Clones can be detected locally, nationally, and internationally using a low-cost identity verification method. The QoS limitations must be maintained or enhanced using an ML-based method. This method uses the OELM to stop fake nodes from wreaking havoc on a wireless sensor network. Nodes employ duplicates of their identities to determine the best secure path for sending data. The suggested method may be applied to any node in a network to retrieve data from it. One strategy for stopping the coordinated attacks is to destroy the clones that are hosting them. The results shed light on the novel mechanism's efficacy in relation to those of similar systems. The proposed strategy provides more reliable outcomes than the status quo. In the future, the suggested approach can be enhanced with a deep learning-based model to increase attack detection.

Data Availability Statement

The data that support the findings of this study are available upon reasonable request from the authors.

Ethics Approval

The submitted work is original and has not been published elsewhere in any form or language.

Disclosure of Potential Conflicts of Interest

There is no potential conflict of interest.

Research Involving Human Participants and/or Animals

NA

Funding

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Competing Interests

The authors have no relevant financial or non-financial interests to disclose.

References

[1] Numan, M., Subhan, F., Khan, W.Z., Hakak, S., Haider, S., Reddy, G.T., Jolfaei, A. and Alazab, M., "A systematic review on clone node detection in static wireless sensor networks", IEEE Access, 8, pp.65450-65461, 2020. DOI:10.1109/ACCESS.2020.2983091.

[2] Mohindru, V., Singh, Y. and Bhatt R., "Hybrid cryptography algorithm for securing wireless sensor networks from Node Clone Attack", Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering)", 13(2), pp.251-259, 2020. https://doi.org/10.2174/2352096512666190215125026.

[3] Mohindru, V. and Singh, Y, "Node authentication algorithm for securing static wireless sensor networks from node clone Attack", International Journal of Information and Computer Security, 10(2-3), pp.129-148, 2020. DOI:10.1504/IJICS.2018.091462.

[4] Lalar, S., Bhushan, S. and Surender, M., "Hybrid encryption algorithm to detect clone node attack in wireless sensor Network", In Proceedings of the International Conference on Innovative Computing & Communications (ICICC), 2020.

[5] Shaukat, H.R., Hashim, F., Shaukat, M.A. and Ali Alezabi, K., "Hybrid multi-level detection and mitigation of clone attacks in mobile wireless sensor network (MWSN)", Sensors, 20(8), p.2283, 2020. https://doi.org/10.3390/s20082283.

[6] Jane Nithya, K. and Shyamala, K., "A Systematic Review on Various Attack Detection Methods for Wireless Sensor Networks", In International Conference on Innovative Computing and Communications (pp. 183-204). Springer, Singapore., 2022.

[7] Dora, J.R. and Nemoga, K., "Clone node detection attacks and mitigation mechanisms in static wireless sensor networks", Journal of Cybersecurity and Privacy, 1(4), pp.553-579, 2021. https://doi.org/10.3390/jcp1040028.

[8] Mohindru, V., Singh, Y. and Bhatt, R., "Securing wireless sensor networks from node clone attack: a lightweight message authentication algorithm", International Journal of Information and Computer Security, 12(2-3), pp.217-233, 2020. DOI:10.1504/IJICS.2020.105174.

[9] Lalar, S., Bhushan, S. and Surender, "An efficient tree-based clone detection scheme in wireless sensor network", Journal of Information and Optimization Sciences, 40(5), pp.1003-1023, 2019. https://doi.org/10.1080/02522667.2019.1637998.

[10] Sherubha, P., Amudhavalli, P. and Sasirekha, S.P., "Clone attack detection using random forest and multi objective cuckoo search classification", In 2019 International Conference on Communication and Signal Processing (ICCSP), pp. 0450-0454, IEEE, 2019.

[11] Tang, C. and Han, D., "A low resource consumption clone detection method for multi-base station wireless sensor networks", IEEE Access, 8, pp.128349-128361, 2020, DOI: 10.1109/ACCESS.2020.3007388.

[12] Jaballah, W.B., Conti, M., Filè, G., Mosbah, M. and Zemmari, A., "Whac-A-Mole: Smart node positioning in clone attack in wireless sensor networks", Computer Communications, 119, pp.66-82, 2018. https://doi.org/10.1016/j.comcom.2018.01.010

[13] Anitha, S., Jayanthi, P., & Chandrasekaran, V., "An intelligent based healthcare security monitoring schemes for detection of node replication attack in wireless sensor networks", Measurement, 167, 108272, 2021. https://doi.org/10.1016/j.measurement.2020.108272.

[14] Devi, P.P. and Jaison, B., "Optimal Scheme for the Detection and Classification of Clone Node Attack in WSN Using TAIGBRFCNIA", Wireless Personal Communications, 125, pp.1615-1629, 2022. https://doi.org/10.1007/s11277-022-09623-z

[15] Meganathan, S., Rajesh Kumar, N., Sheik Mohideen Shah, S., Sumathi, A. and Santhoshkumar, S., "Wireless Sensor-Based Enhanced Security Protocol to Prevent Node Cloning Attack", In Computational Vision and Bio-Inspired Computing (pp. 603-615). Springer, Singapore., 2022.

[16] Mohammad, S.; Sultanul Kabir, A.F.M., "Hierarchical Design Based Intrusion Detection System for Wireless Ad Hoc Sensor Network", Int. J. Netw. Secur. Appl., 2, 102–117, 2022.

[17] Devi, P.P. and Jaison, B., "Protection on wireless sensor network from clone attack using the SDN-enabled hybrid clone node detection mechanisms", Computer Communications, 152, pp.316-322, 2020. https://doi.org/10.1016/j.comcom.2020.01.064

[18] Hameed, K., Garg, S., Amin, M.B., Kang, B. and Khan, A., "A context-aware information-based clone node attack detection scheme in Internet of Things", Journal of Network and Computer Applications, 197, p.103271, 2022. https://doi.org/10.1016/j.jnca.2021.103271

[19] Vatambeti R, Supriya KS, Sanshi S, "Identifying and detecting black hole and gray hole attack in MANET using gray wolf Optimization", Int J Commun Syst., 33(18): e4610, 2020. https://doi.org/10.1002/dac.4610.
[20] S. G. Thakur, ''Cinora: Cell based identification of node replication attack in wireless sensor networks'', in Proc. IEEE Int. Conf. Commun. Syst. (ICCS), pp. 1–8, 2008.
[21] H. R. Shaukat, F. Hashim, A. Sali, and M. F. Abdul Rasid, ''Node replication attacks in mobile wireless sensor network: A survey'', Int. J. Distrib. Sensor Netw., 10, 12, 2014. https://doi.org/10.1155/2014/402541
[22] Mirkarimi F, Azizyan G, Rashki M, "Horse herd optimization algorithm: a nature-inspired algorithm for high-dimensional optimization problems", Knowledge-Based Systems, 213:106711, 2021. https://doi.org/10.1016/j.knosys.2020.106711.

Authors' Profiles

Dr. Ramesh Vatambeti received his B.Tech from Sri Venkateswara University, Tirupati in Computer Science and Engineering and M.Tech in IT and PhD in CSE from Sathyabama University, Chennai. He has around 18 years of teaching experience from reputed Engineering Institutions. He works as Professor in the School of Computer Science and Engineering at VIT-AP University, Amaravati, India. He has published 4 books, 5 book chapters and more than 70 papers in refereed journals and conference proceedings. He is the reviewer for several refereed International Journals and acted as Session Chair and technical committee member for several International Conferences held in India and abroad. He has successfully guided 3 Ph.D. scholars. His research interests include Computer Networks, Mobile Ad-Hoc and Sensor Networks and Machine Learning.

Vijay Kumar Damera recently submitted his Doctoral Thesis in the Area of Cloud Computing. He has 14 years of Teaching and 1.5 years of Research Experience. His research interests include Cloud Computing, Artificial Intelligence, Machine Learning and Software Defined Networks. He has more than 28 research publications to his credit in various International Journals and Conferences. He is currently working in the department of Computer Science and Engineering at K L Deemed to be University off campus Hyderabad.

Karthikeyan. H. received his BE (Computer Science and Engineering) degree in the year 2009 from Anna University, Chennai and M.E (Computer Science and Engineering) degree in the year of 2011 from Anna University, Chennai. He completed his PhD in Anna University. He is currently working as an Assistant Professor in the Department of Computer Science and Engineering, CHRIST (Deemed to be University). His research interest is in Data Analytics, Healthcare Analytics, cyber security.

Dr. Manohara M. is an Associate Professor in the Computer Science and Engineering Department at School of Engineering and Technology of CHRIST (Deemed to be University), Bangalore. He is an educator by choice and vocation, with an experience of 22 years in Teaching. He is qualified in Bachelor and Master Degrees in Computer Science & Engineering, and Ph.D. in Computer Science & Engineering in the area of Data Mining and Big Data. His areas of interest are data mining, computer vision, machine learning, artificial intelligence, internet of things, and image processing. He has published several papers in peer reviewed journals and international conferences.

Sharon Roji Priya. C. is an Assistant Professor in Computer Science and Engineering Department at school of Engineering and Technology of CHRIST (Deemed to be University), Bangalore. She is having 11 years of experience in Teaching. She is qualified in Bachelor and Master Degrees in Computer Science and Engineering. Her area of interest are Automata Theory, Machine Learning, Deep Learning and Natural Language Processing. She has published papers in journals and international conferences.

Mahammad Shareef Mekala is a post-Doctoral researcher in School of Communication Engineering, Yeungnam University, Republic of Korea. He received his B.Tech. (CSE) degree in 2011 and his M.Tech. (CSE) Degree in 2013 from JNTU University, received Ph.D. from VIT University, Vellore, India. He had also received a VIT research award in 2017 and C. V. Raman research award in 2019. His research interests include IoT, cloud computing, wireless networks, cyber-physical systems, soft computing, machine learning.

How to cite this paper: Ramesh Vatambeti, Vijay Kumar Damera, Karthikeyan H., Manohar M., Sharon Roji Priya C., M. S. Mekala, "Classification of HHO-based Machine Learning Techniques for Clone Attack Detection in WSN", International Journal of Computer Network and Information Security(IJCNIS), Vol.15, No.6, pp.1-15, 2023. DOI:10.5815/ijcnis.2023.06.01