International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Chained Hash Table-Based Filtering Model for Detection of Flooding and Dos Attack in Manet

[1*]Shyamily P. V., [2]Dr. Anoop B. K.

Submitted: 17/01/2024 **Revised**: 25/02/2024 **Accepted**: 03/03/2024

Abstract: Mobile Ad Hoc Network (MANET), which connects mobile devices to all devices in the network where security is an essential task caused by Malicious Nodes (MNs), is a wireless communication technology. One of the crucial attacks that aim to exhaust network resources by flooding it with numerous fake packets as well as messages is the flooding attack. Hence, to detect flooding attacks, many models were developed. However, the models did not detect the sequences of flooding attacks. Therefore, this work proposes a soft kernel Swish-based Recurrent Neural Network (SS-RNN)-based sequential flooding attack detection model in MANET. Primarily, the nodes are initialized. Thereafter, the signature is created for the packet information utilizing the Chinese Remainder Theory-based Digital Signature Algorithm (CRT-DSA). Afterward, the packets are transmitted to nodes for connection establishment. The signed packet is verified, and to eliminate the attacked packets, the legitimate user's packets are filtered utilizing Boltzmann Gibbs Trapezoidal Entropy based Fuzzy (BGTE-Fuzzy). By utilizing XNOR-HAVAL, the packet information is hashed and added into the chaining grounded on time series to identify flooding attacks, namely HELLO, Internet Control Message Protocol (ICMP), Synchronize an Acknowledgement (SYN-ACK), Acknowledgement (ACK), and Data flooding. After that, using SS-RNN, Denial-of-Service (DoS) attacks are detected. Lastly, for data transmission, the optimal path is created utilizing Correlation Coefficient-based Galactic Swarm Optimization (CC-GSO). To prevent security in MANET, the developed model classifies attacks with an accuracy of 98.41%.

Keywords: *Denial-of-Service (DoS), Mobile Ad Hoc Network (MANET), Chinese Remainder Theory based Digital Signature Algorithm (CRT-DSA), Boltzmann Gibbs Trapezoidal Entropy based Fuzzy (BGTE-Fuzzy) Algorithm, Correlation Coefficient based Galactic Swarm Optimization (CC-GSO), Soft kernel Swish based Recurrent Neural Network (SS-RNN), Particle Swarm Optimization (PSO).*

1. Introduction

A proposed filtering mechanism for detecting flooding and denial-of-service attacks in MANETs is based on chained hash tables. The model maps flooded regions based on severity levels to enhance community resilience and decision making in catastrophe scenarios (Qanbar & Tasdemir, 2019). The algorithm divides flood photos into three categories: mild, moderate, and severe using image classification approaches, namely Convolutional Neural Networks (CNNs) with transfer learning (Kamoji et al., 2022). When various pretrained CNN models are compared in the study, the optimized pretrained ResNet50 model performs best for classifying flood images (Kamoji et al., 2015). The suggested approach, which makes use of statistical characteristics to accurately forecast cyberattacks, can be easily included into current cyberdefense systems (Işık et al.,2016). Furthermore, a novel concept for background modeling in surveillance systems—the Common Vector Approach (CVA)—is presented, demonstrating effective foreground detection outcomes (Aytekin, 2018).

A self-configuring communication network of mobile nodes linked by wireless radio nodes is named MANET. Since every single node has the capability of moving autonomously in any direction, it can change its position

*1*Research Scholar, Computer Science & Engineering Department, College of Engineering & Technology, Srinivas University, Mukka, Mangalore, Orcid-Id:0000-0002-2502-8937 shyamilyamritha@gmail.com*
2Professor, Srinivas Institute of Technology, Mangalore, Orcid-Id:0000-0003-4288-5065 dranoopbk@sitmng.ac.in

or link to other nodes frequently (Bouyeddou et al., 2020). MANET is quickly replacing other technologies in many areas of daily life owing to its low cost, low power consumption, and ease of implementation (Islabudeen& Devi, 2020). Even though MANETs are necessary in all spheres of life, their communication is very susceptible to security threats and is liable to interference, eavesdropping, and jamming (Karthigha et al., 2020). Owing to a lack of centralized control, a lack of a defined boundary, adversaries that are primarily within the network, a lack of energy resources, and regularly shifting positions, Adhoc wireless communications are inherently vulnerable to security assaults (Srinivas & Manivannan, 2020). In general, reactive routing protocols, namely Ad Hoc On-Demand Distance Vector (AODV) are utilized by the MANETs to fulfill their routing requirements (Dilipkumar & Durairaj, 2023).

The ad hoc network routing protocols' primary goal is correct as well as efficient route establishment betwixt a pair of nodes so that messages can be delivered on time (Gautam & Tokekar, 2020) (Ravi et al., 2020). The MN utilizes this protocol feature and sends bulk fake messages and route requests for constructing a route and making a transmission with them (Elhigazi et al., 2020) (Khalfaoui et al., 2023). As a result, legitimate network nodes are affected by flooding attacks, which are also known as jamming and network resource unavailable problems (Salunke & Ragavendran, 2021) (Banga et al., 2021). The flooding attacks include data flooding, error flooding, HELLO flooding, Synchronize (SYN) flooding, ACK flooding, et cetera. This sort of attack is also called a DoS attack, which happens at the transport, network, and application level of the Open Systems Interconnection (OSI) model (Shafi et al., 2022) (Kshirsagar& Kumar, 2022). Previous solutions for flooding and DoS attack detection are designed grounded on supervised approaches, including Random Forest (RF), Long Short-Term Memory (LSTM), Naive Bayes, Deep Neural Network (DNN), Convolutional Neural Network (CNN), et cetera (Talukdar et al., 2021) (Mittal et al., 2023). In some cases, deep learning techniques are complex in learning and are also time-consuming. Therefore, the paper proposes an XNOR-HAVAL and SS-RNN-based flooding attack detection model in MANET.

1.1. Problem Statement

The shortcomings of existing flooding attack detection models are,

- The patterns of sequence flooding attacks, such as hello flooding, SYN-ACK flooding, SYN flooding, ICMP flooding, ACK flooding, and data flooding were not focused on any of the existing systems.
- Existing models utilized many Route Requests (RREQs) originated by a node per unit of time as the threshold for detecting flooding attacks, which leads to high misdetection and reduced network performance.
- In MANET, any node might join or else leave the network at any time, making it challenging to distinguish between trustworthy and MNs.
- The nodes themselves are responsible for delays in propagating the messages in the network. MNs can take advantage of this and misuse the messaging traffic.

The research methodology's major objectives are described as,

- The sequences of the flooding attacks are detected in two phases, namely sequential flooding attack detection and DoS attack detection by utilizing the proficient techniques in the proposed research.
- An XNOR-HAVAL-centric hashing algorithm is employed to securely add the packets into the chaining for identifying the sequential flooding attacks.
- Unregistered nodes are prohibited by using the CRT-DSA-based signature verification function.
- The network path features are extracted here for enhancing the CC-GSO-based shortest path creation operation.

The remaining paper is structured as: Section 2 examines associated research on flooding attack detection in MANET. Section 3 displays the proposed detection of the flooding attack sequence. The proposed mechanism's efficacy is examined in Section 4. The paper is concluded in section 5.

2. Related Literature Review

Narmadha et al., (2023) propounded a Deep Neural Learned Projective Pursuit Regression-centered Watchdog MN Detection and Isolation (DNLPPR-WMNDI) system for enhancing the MANETs' security feature. The control command RREQ was utilized in multicasting for establishing the route pathways across the intermediary node. The attack detection rate in experiments was greatly increased by the DNLPPR-WMNDI approach. Yet, the developed model took a longer time for the detection owing to the complex parameters of the classifier.

Hai et al., (2023) propounded Secure Cryptography-centric Clustering Mechanism (SCCM) for rendering security for MANETs. Here, by utilizing an Elliptic-Curve Cryptography (ECC)-based encryption system, the original packet was encrypted before being sent to the receiver. The outcome revealed that multipath routing allowed the network to find additional routes that connected the source and destination. Yet, the model did not consider the nodes' behavior, which resulted in complex trust level identification.

Kamoji et al., (2022) created a map of flooded areas, categorizing them according to the intensity of the flooding. This mapping intends to enhance the ability of communities to withstand and recover from disasters, as well as to make informed decisions in such scenarios. The researchers employed picture classification methodologies and a specifically curated dataset comprising of flood photographs classified into three distinct categories based on their severity levels: mild, moderate, and severe. Convolutional Neural Networks (CNNs) with a transfer learning strategy were employed to enhance the classification task. Convolutional Neural Networks (CNNs) are highly effective in extracting distinctive characteristics from visual data and leveraging semantic knowledge. Transfer learning was employed instead of constructing and training a convolutional neural network (CNN) from the beginning. Specifically, pre-existing and pre-trained networks such as VGG16, MobilNet, and ResNet50 were utilized. The evaluation of the models was conducted based on the mean values of recall, precision, and F1-score. The fine-tuned pretrained ResNet50 model outperformed other cutting-edge algorithms in classifying flood images.

Gadzama & Saidu (2022) established a highly secure Hybrid Trust-centric Anonymous Authenticated Routing Protocol (HTBAARP) for mitigating flooding assaults. The node's status and trust level were broadcast throughout the network utilizing HTBAARP and the Trust-Based Management Scheme. The experimental analysis exemplified that the developed method had high network performance. Nevertheless, the trained model's complexity was high while testing a lot of data.

Sbai & Elboukhari (2022) established a Knowledge-Based Intrusion Detection System (KBIDS) for securing MANETs. By utilizing a DNN, the attacks in this developed model were classified in which the input features were the packet features. As per the outcomes, the designed architecture model might achieve highly interesting and encouraging performance and results. However, the developed scheme was limited by the high-power consumption.

Rahouti et al., (2021) proffered an adaptive threshold-centered kernel-level Intrusion Detection and Prevention System (IDPS) in MANET. Utilizing an IDPS solution, attacks on the data control plane saturation were efficiently detected and mitigated. The system's efficacy in identifying and thwarting SYN flood attacks in an SDN environment was proved by the experimental outcomes. However, the model lacked accuracy in its feature discrimination, rendering it to be unreliable.

Farahani (2021) exhibited an approach in MANETs for detecting black hole attacks utilizing the K-Nearest Neighbor (KNN) technique. Grounded on reputation and remaining energy, the cluster head was selected by fuzzy inference. According to the outcomes, in contrast to recent black hole detection approaches, the suggested mechanism has enhanced throughput, total network delay, packet loss rate, packet delivery ratio, and normalized routing load parameters. Nevertheless, the data's unbalanced distribution of classes led to an uncertain detection rate.

Nishanth & Mujeeb (2021) proffered modeling as well as detection of flooding-centric DoS Attacks in MANET utilizing Bayesian Inference. To detect the flooding attacks, an Exponential Weighted Moving Average (EWMA) was utilized. Based on the simulation, the developed model could effectively defend against any sort of flooding-centric DoS attack with higher detection accuracy. Yet, it was hard to judge packet-dropping

statistics to avoid flooding attacks owing to perceived congestion in its transmission.

Alam (2020) proffered a system for black-hole and flooding attack detections in MANET. Here, to identify and stop DoS flooding and black-hole attacks, a mixed routing method grounded on Bayesian classification was employed. The developed model's throughput was increased by the experimental findings. Nevertheless, the developed classifier had less memory design to store the trained data. However, the system had less adaptability when considering the mobility of sensor nodes in MANET.

Zalte et al., (2020) presented the mitigation of DDOS attacks in MANET. Here, using threshold timestamps, the DDOS attacks and attackers were detected, and hostile attackers were successfully blocked from transmitting via the network. The AODV routing protocol's Packet Delivery Ratio was slightly improved after the DDOS attack was mitigated. Therefore, the constructed model illustrated the viability of DDOS defense mechanisms in MANET. Nevertheless, the developed model was not applicable to large-scale networks.

Ramesh et al., (2020) presented a machine-learning technique for secure communication in MANET. A complexity minimal encryption was utilized by the encrypted protocol during the trust value trade in the clusters accommodating with message verification code. The results obtained ensured the developed work's efficacy when the support vector machine system was amalgamated with an encryption system. Yet, the trust parameters's confidentiality remained a challenge since it may be overheard by the impostor.

Qanbar & Tasdemir (2019) proposed a deep learning method that utilizes the Residual Attention Network (RAN) to accurately classify and diagnose cases of malaria. The RAN model demonstrated a superior classification accuracy rate of 95.79% in the analysis and categorization of blood cell pictures, surpassing other algorithm types such as Support Vector Machine (SVM) which reached a mere 83.30% accuracy. Their study emphasizes the benefits of utilizing deep learning techniques to automatically extract features from input data, hence reducing the need for specialized input from experts in automated malaria diagnosis. Deep learning approaches are proposed as effective tools for classifying blood cell pictures and diagnosing malaria.

Study	Methodology/Approach	Strengths	Weaknesses
Narmadha et al., (2023)	DNLPPR-WMNDI system, Deep Neural Learned Projective Pursuit Regression, RREQ multicasting, MANETs security enhancement.	Utilizes Deep Neural Network for MANET security, enhanced attack detection.	Longer detection time, complex classifier parameters.
Hai et al., (2023)	SCCM, Secure Cryptography-centric Clustering Mechanism, ECC-based encryption, multipath routing.	Utilizes ECC for secure clustering, multipath routing for additional routes.	Does not consider nodes' behavior, complex trust level identification.
Kamoji et al., (2022)	Flood area mapping, picture classification, CNNs with transfer learning, VGG16, MobilNet, ResNet50.	Effective use of CNNs for image classification, transfer learning for enhanced performance.	Limited to flood image classification, reliance on pre-trained models.
Gadzama & Saidu (2022)	HTBAARP, Hybrid Trust-centric Anonymous Authenticated Routing Protocol, Trust-Based Management Scheme.	Utilizes trust-centric routing protocol, high network performance.	High model complexity during testing with large data.
Sbai & Elboukhari (2022)	KBIDS, Knowledge-Based Intrusion Detection System, DNN, packet features for attack classification.	Utilizes DNN for attack classification, promising performance.	Limited by high-power consumption.
Rahouti et al., (2021)	Adaptive threshold-centered kernel-level IDPS, detection and prevention of SYN flood attacks.	Effective in identifying and thwarting SYN flood attacks, adaptive thresholding.	Lacks accuracy in feature discrimination, unreliable results.

Farahani (2021)	Black hole attack detection using KNN, fuzzy inference for cluster head selection.	Improved network parameters, fuzzy inference for cluster head selection.	Uncertain detection rate due to data imbalance.
Nishanth & Mujeeb (2021)	Flooding-centric DoS Attacks detection using Bayesian Inference, EWMA for detection.	Defends against flooding attacks, utilizes Bayesian Inference.	Challenging to judge packet-dropping statistics.
Alam (2020)	Black-hole and flooding attack detection using Bayesian classification, mixed routing.	Increased throughput, Bayesian classification for attack detection.	Less adaptability to sensor node mobility, classifier design limitations.
Zalte et al., (2020)	Mitigation of DDOS attacks using threshold timestamps, improved Packet Delivery Ratio.	Demonstrates viability of DDOS defense, improved Packet Delivery Ratio.	Not applicable to large-scale networks.
Ramesh et al., (2020)	Machine learning for secure communication in MANET, minimal encryption, SVM for trust value trade.	Minimal encryption, SVM for trust value trade, effective secure communication.	Challenges in trust parameter confidentiality.
Qanbar & Tasdemir (2019)	Deep learning method using Residual Attention Network (RAN) for malaria diagnosis.	High classification accuracy, Residual Attention Network for feature extraction.	No specific weaknesses mentioned.

3. Research Methodology

Here, an efficient flooding attack and DoS attack detection system in MANET is proposed that aims to mitigate security threats. Figure 1 displays the proposed system's structure.

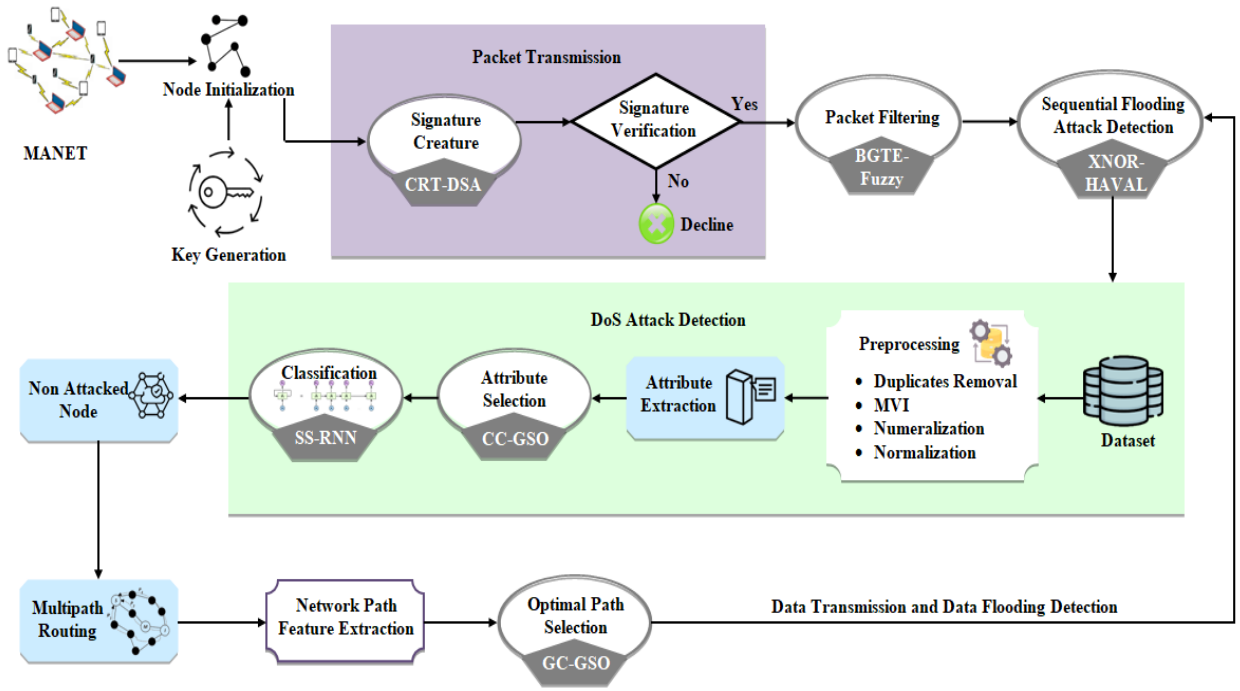

Fig 1: Block Diagram of the proposed methodology

3.1. Node Initialization and key generation

In the beginning, the mobile nodes are initialized for forwarding and receiving the packets. The initialized mobile nodes $(\aleph_n)$ are elucidated as,

$$\aleph_n = \aleph_1, \aleph_2, \aleph_3, ... \aleph_N, \text{ where } i = 1,2,3,...N \quad (1)$$

Wherein, the number of initialized mobile nodes is exemplified as $\aleph_n$, the node's number is implied as i, and the number of final nodes is notated as N. By utilizing the CRT-DSA, the public key $\left(K_{\aleph^i}^{public}\right)$ and private key $\left(K_{\aleph^i}^{private}\right)$ for each node are generated during the initialization. The steps followed to generate the keys are explained further.

- The private key $\left(K_{\aleph^i}^{private}\right)$ is selected randomly.
- Afterward, the public key $\left(K_{\aleph^i}^{public}\right)$ is generated by using,

$$K_{\aleph^i}^{public} = qK_{\aleph^i}^{private} \bmod X \quad (2)$$

Where, X is the prime number and q is a random integer, such that $1 < q < X$.

3.2. Packet Transmission

Here, to confirm network adjacency relationships with other nodes, the HELLO packets are transmitted from one node to another node, while ICMP, SYN, and SYN-ACK packets are transmitted from one node to another node to establish the connection between the nodes. The data packet can be sent by establishing the connection between the nodes. The packets are commonly signified as P. The packet information is signed before the transmission and then, it is transferred to the receiver node. In the receiver node, the signed packet information is verified whether that is transmitted from a legitimate node or not to avoid an unregistered node's participation.

(a) Packet signature Creation: Here, using CRT-DSA, the packet information is signed. Primarily, the Chinese Remainder Theory (CRT) spoofed the Packet's source and destination IP address. Thereafter, the packet information along with the spoofed address is signed utilizing a Digital Signature Algorithm (DSA). Therefore, the technique is known as CRT-DSA. Here, the conventional DSA provides an efficient outcome in integrity verification.

At first, by utilizing CRT, the source $\left(I_P^{source}\right)$ and destination Internet Protocol (IP) addresses $\left(I_P^{des}\right)$ are spoofed. The spoofed address $\left(ID^{spoof}\right)$ is mathematically represented as,

$$ID^{spoof} = \left(I_P^{source}\right) \bmod \left(I_P^{des}\right) \quad (3)$$

Following the spoofing operation, the steps in DSA to generate the signature are,

- The signer is responsible for generating the signature (S_1, S_2) for packet information $\left(P^{\inf}\right)$ and spoofed address $\left(ID^{spoof}\right)$.
- Thereafter, the random number (R) is generated, such that $0 < R < Y$.
- The signatures S_1 and S_2 are computed by utilizing,

$$S_1 = (q * R \bmod X) \bmod Y \quad (4)$$

$$S_2 = I(\left(P^{\inf} * ID^{spoof}\right) + S_1 * K_{\aleph^i}^{private}) \bmod Y \quad (5)$$

Where, Y is the prime number, and I is the modular multiplicative inverse of R modulo Y. The pair (S_1, S_2) epitomize the signature of the packets, which is to be transmitted.

(b) Signature Verification: After signature creation, the packet is transmitted from the sender to the receiver node. To avoid MN interference, signature

verification is done using CRT-DSA in the receiver node. The verification process is,

$$S_1 = (q * R \bmod X) \bmod Y \tag{6}$$

$$S_2 = I((P^{\inf} * ID^{spoof}) - S_1 * K_{\aleph^i}^{public}) \bmod Y \tag{7}$$

The node is recognized as a legitimate (registered) node if the receiver node can sign the signature using the source node's public key. Else, the node is an unregistered one, and it should be prohibited. Lastly, the obtained legitimate nodes are notated as $\aleph^{legi}$.

3.3. Packet Filtering

The packets (P) are filtered from $\aleph^{legi}$ to detect and eliminate the attacked packets, which come from the attacker node. The attacker node can attack the packets in two ways. The first way is that the attacker node may act as the parent node by sending the HELLO packet with the base node's characteristics. The second way is that the attacker may send the packets through an unreliable protocol, which degrades the network's security. This will prompt to increment in delay. Hence, the non-attacked packets are filtered in this section. For this filtering, the BGTE-Fuzzy algorithm is utilized in this proposed paper. The conventional Fuzzy algorithm could provide the most effective solution to complex issues. Yet, the Fuzzy algorithm has the tuning difficulty of scaling factor, membership function, and control rules. Therefore, the results are based on assumptions. Thus, the proposed paper uses Boltzmann Gibbs Trapezoidal Entropy (BGTE)-based membership function, which significantly overcomes the problem owing to tuning difficulties.

Primarily, the number of packets from every single legitimate node is fed into the fuzzy control system and mapped by sets of BGTE membership functions. The fuzzification function is exemplified as,

$$(P) \xrightarrow{fuzzification} (P)^{BGTE} \tag{8}$$

The membership function $((P)^{BGTE})$ is calculated utilizing BGTE as,

$$(P)^{BGTE} = k \log \tau(P) \tag{9}$$

Where, k is the constant and τ symbolizes the tuning factor. The BGTE membership function efficiently maps all the non-fuzzy data into fuzzy linguistic terms and vice versa. This fuzzy set aids in making the rules in the proposed work. Thereafter, the fuzzy rule is evaluated for the fuzzy sets. The rules are defined based on three factors, namely the node's energy value and transmission power $(\aleph^{e,p})$, packet count (P^{count}), and protocol name (P^{name}). The pseudo-code of the proposed BGTE-Fuzzy is,

---

Input: Number of packets from legitimate nodes (P)

Output: Filtered packets (P_{fil})

Begin
Initialize k , τ ,
For all the packets, do
Perform fuzzification process.
Evaluate BGTE membership function
Apply interference engine.
If $\{\aleph^{e,p} > B^{e,p} \,\&\, P^{name} \,\&\, P^{count} = \omega\}$
THEN {Consider as attacked packet.**}**
Else {Consider it as non-attacked packet.}
End if
Perform defuzzification function.
End for
Return Filtered non-attacked packet.
End begin

Rule generation is grounded on a collection of logic rules in the form of IF-THEN statements. The rules (ς) for HELLO packets are,

$$\varsigma = \begin{cases} If\ \aleph^{e,p} > B^{e,p} \,\&\, P^{count} \neq \omega,\ then\ P^{HELLO} = A \\ If\ \aleph^{e,p} < B^{e,p} \,\&\ P^{count} = \omega,\ then\ P^{HELLO} = NA \end{cases} \tag{10}$$

Here, $B^{e,p}$ is the base station's energy value and transmission power; P^{HELLO} implies the HELLO packet; ω is the defined network packet count; A and NA are the attacked and non-attacked packets.

The packets are identified when it is sent through reliable and unreliable protocols. The rules (ς) for filtering the remaining packets by considering the protocols are expressed as,

$$\varsigma = \begin{cases} If\ P^{name} = \omega^{TCP}, then\ P = NA \\ If\ P^{name} = \omega^{UDP}, then\ P = A \end{cases} \quad (11)$$

The transmission will be a reliable service and the packet is considered as non-attacked packet if the packet is sent through the Transmission Control Protocol (TCP). Also, the transmission will be an unreliable service and the packet is considered as an attacked packet if the packet is sent through the User Datagram Protocol (UDP). Here, ω^{TCP} and ω^{UDP} signify the reliable and unreliable protocols. Lastly, the non-attacked filtered packets are implied as P_{fil} .

3.4. Sequential Flooding Attack Detection

Here, for making the chaining structure, the packet information of P_{fil} , such as Sender Node ID and Receiver Node ID is hashed. Afterward, the hashed address is added into hash chaining as the index. Under the index, a list of request packets is linked based on timestamps by ignoring the flooding packets. For this hashing, the proposed paper utilizes XNOR-HAVAL. The prevailing HAVAL hashing, which has a large hashcode with complexity, is a one-way hashing algorithm. Nevertheless, the skeleton of the hashing process can be easily hacked owing to the fixed length of the consonant string. To solve this issue, the X-NOR technique is utilized for finding the bit length. This will provide the complex structure of consonant strings for enhancing security.

Primarily, the Sender Node ID and Receiver Node ID are combined, and it is notated as χ . Thereafter, χ is padded into multiples of 1024. The padding data is written as follows:

$$\chi^{pad} = \left\{\chi^1, \chi^2, \chi^3, \ldots\ldots, \chi^l\right\} \quad (12)$$

Here, χ^{pad} symbolizes the number of padded data. The last padded part (χ^l) is represented as the length of the combined sender and receiver's ID. The padded data is given in each round of the hashing function of XNOR-HAVAL.

The padded data is combined with the eight-word consonant string from the previous round of hashing. The merging operation is displayed as,

$$\Im^{r+1} = h\left(\Im^r, \chi^{pad}\right) \quad (13)$$

Where,

$$r = 1,2,3,\ldots\ldots\ldots\ldots l \quad (14)$$

Using the X-NOR technique, the consonant string $(\Im^r)$ is computed and is elucidated as,

$$\Im^r = \left(S^{odd}.S^{even}\right) + \left(S^{odd} + S^{even}\right) \quad (15)$$

Here, r signifies the number of hashing rounds, l epitomizes the final hashing round, $\Im^{r+1}$ symbolizes the hashing output in the current function, S^{odd} implies the odd strings, and S^{even} epitomizes the even string. This X-NOR-based consonant string provides a complex string. Using each string value, the padded message is combined to form the hashed data. Hence, a strong chaining function can be provided to link the list of packet requests. Similarly, the hashing rounds are performed till l .

- Adjusting the 256-bit value acquired from the final round to the specified length of the combined ID is the final step. According to the compressed data,

$$\chi^{hash} = C \otimes \left(\Im^{b+1}\right) \quad (16)$$

Here, χ^{hash} signifies the hashed data and C illustrates the compression function.

The hashed address is added as the index (head of packet request lists) in the chained hash table. Figure 2 exemplifies the hash chaining structure .

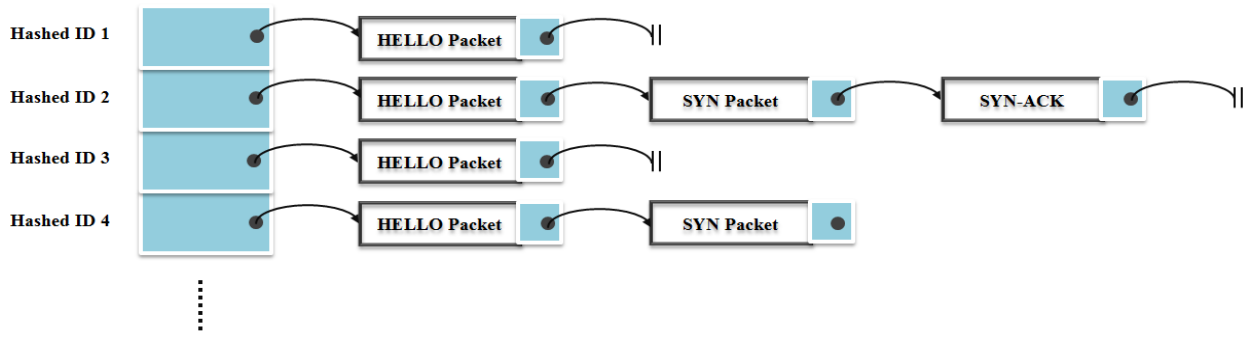

Fig 2: Chained Hash Table

The chained hash table is highly efficient in making the array with the linked lists. A chained hash table essentially comprises an array of linked lists. Every single list forms a bucket, where all elements are hashed to a specific position in the array. For inserting an array, the data key is first passed to a hash function. According to that, the proposed model hashed the sender and receiver node ID and added it as the array in the hash table. The lists of packet requests are added one by one on a time basis in this hashed array. The process of linking requests below the array starts with the HELLO packet. By analyzing the HELLO flooding packets, the HELLO packet request between the hashed sender and receiver ID is added on a time basis. Once the packet is added to the array, the remaining flooding packet between the hashed sender and receiver ID is ignored. Following the HELLO packet linked with the index, the SYN packet, SYN-ACK packet, and Data packet are added one by one by analyzing the flooding packets. Similarly, each sender ID and receiver ID between two nodes are hashed and added into the chaining to eliminate the flooding message. Lastly, the obtained packets after eliminating the flooding packets are notated as $\Re^m$.

3.5. DoS attack detection

Following the flooding attack detection, the other DoS attacks, namely R2L, U2R, Probe, et cetera are identified from $\Re^m$.

3.5.1. Preprocessing

The input data from the NSL-KDD dataset are preprocessed, where the raw input data is transformed into machine understandable. The input data $\left(\Re^m\right)$ is,

$$\Re^m = \left[\Re^1, \Re^2, \Re^3, \ldots\ldots\ldots \Re^M\right] \tag{17}$$

Here, $\Re^M$ epitomizes the M^{th} data packet.

(a) Duplicate removal: Here, the redundant copies of the packet data from $\Re^m$ are removed.

(b) Missing Value Imputation (MVI): To retain most of the information, the missing data are replaced with a substitute. Afterward, the imputation $\left(\Re^*\right)^{miss}$ is,

$$\left(\Re^*\right)^{miss} = \overset{imput}{\Re}\left\{mean\left(d_1^*, d_2^*, \ldots\ldots, d_z^*\right)\right\} \tag{18}$$

Where, $\overset{imput}{\Re}$ symbolizes the imputation function, $\left(d_1^*, d_2^*, \ldots\ldots, d_z^*\right)$ implies the number of data in the packet, and d_z^* specifies the z^{th} number of data.

(c) Numeralization: Here, to provide the efficient classification, numeralization is done to convert all

those strings into numerical values, and the resultant numeralized data $\left(\left(\Re^{*}\right)^{num}\right)$ is represented as,

$$\left(\Re^{*}\right)^{num} = numeralize\left(\sum_{a=1}^{z} d_{a}^{*}\right) \tag{19}$$

Here, $numeralize$ epitomizes the numeralization function and d_{a}^{*} exemplifies the a^{th} number of data.

(d) Normalization: Here, $\left(\Re^{*}\right)^{num}$ is further normalized in the standard format. Z-score is one of the most efficient normalization approaches. The Normalization $\left(\left(\Re^{*}\right)_{nor}^{num}\right)$ is expressed as,

$$\left(\Re^{*}\right)_{nor}^{num} = \frac{\left(\Re^{*}\right)^{num} - \varsigma}{\chi} \tag{20}$$

Here, ς, χ describe the mean and standard deviation of $\left(\Re^{*}\right)^{num}$. Lastly, the preprocessed data is signified as $\Re^{pre}$.

3.5.2. Attribute Extraction

Here, from $\Re^{pre}$, the attributes are extracted for gathering the information about the data to be transmitted. Therefore, the key attributes, namely protocol type, flag, dst bytes, dst host count, duration, service, src bytes, srv count, et cetera are extracted. The extracted data attributes $\left(E^{\Re}\right)$ are,

$$E^{\Re} = \left\{E^{\Re}{}_{1}, E^{\Re}{}_{2}, E^{\Re}{}_{3}, \ldots\ldots\ldots E^{\Re}{}_{\Phi}\right\} \; where, \; j = 1,2,3,\ldots\ldots\Phi \tag{21}$$

Here, the number of data attributes is exemplified as j, and the Φ^{th} attribute of data is depicted as $E^{\Re}{}_{\Phi}$.

3.5.3. Attribute Selection

Here, to reduce the computational and training time of the classifier, the most significant attributes are chosen from $E^{\Re}$. By using CC-GSO, the attribute selection is done. The traditional Galactic Swarm Optimization (GSO) gives the possibility of finding the global optimum with greater precision. Yet, the random replacement for the position updation leads to create the local optimal solution. Therefore, for the position updation, the proposed paper utilizes a Correlation Coefficient (CC), which significantly eliminates the uncertainty caused by randomness and reduces the exploration complexity.

In this implementation, the best solutions associated with every single group are acquired by the PSO at the 1st level. At the 2nd level, a superior population i.e. superswarm is generated among the groups' best solutions; afterward, by utilizing the PSO approach, the optimum solution is investigated.

Here, the extracted attributes are considered as particles, which are initialized randomly in the search space. The complete swarm framework is expressed as,

$$E^{\Re}{}_{j} \in E^{\Re} : j = 1,2,3,\ldots\ldots\Phi \tag{22}$$

$$E^{\Re}{}_{j}{}^{u} \in E^{\Re}{}_{j} : u = 1,2,3,\ldots\ldots U \tag{23}$$

$$\cup_{j=1}^{\Phi} E^{\Re}{}_{j} = E^{\Re} \tag{24}$$

Here, $E^{\Re}{}_{j}$ signifies the subset of particles belonging to the subswarm; $\cup$ symbolizes the summation; u and U specify the sizes of the swarm; $E^{\Re}{}_{j}{}^{u}$ is the position (subswarm members) of the particle $E^{\Re}{}_{j}$; the number of particles is specified as j and Φ. Thereafter, PSO is independently performed for every single subswarm. Each subswarm $E^{\Re}{}_{j}$ contains an associated global best;

also, it is updated based on the fitness value. Grounded on attaining the classifier's higher accuracy, each swarm's fitness value is computed. The fitness function (F) is,

$$F = \underset{acc}{Max}\left[E^{\Re}{}_{j}\right] \tag{25}$$

Here, $\underset{acc}{Max}$ symbolizes the maximum classifier accuracy function. The position updation of each sub-swarm is,

$$\left(E^{\Re}{}_{j+1}\right)^{*} = \begin{cases} \left(E^{\Re}{}_{j+1}\right)^{*}, if\ F\left(\eta^{j}\right) < F\left(G^{j}\right) \\ E^{\Re}{}_{j}, if\ F\left(\eta^{j}\right) > F\left(G^{j}\right) \end{cases} \tag{26}$$

Here, $F\left(\eta^{j}\right)$ notates the personal bests, and $\left(E^{\Re}{}_{j+1}\right)^{*}$ is the updated position of $E^{\Re}{}_{j}$. Using the CC technique, the position is updated, and is represented as,

$$\left(E^{\Re}{}_{j+1}\right)^{*} = \frac{\sum\left(E^{\Re}{}_{j} - \left(E^{\Re}{}_{j}\right)^{mean}\right)\left(F\left(G^{j}\right) - F\left(G^{j}\right)^{mean}\right)}{\sqrt{\sum\left(E^{\Re}{}_{j} - \left(E^{\Re}{}_{j}\right)^{mean}\right)^{2}\left(F\left(G^{j}\right) - F\left(G^{j}\right)^{mean}\right)^{2}}} \tag{27}$$

Here, $\left(E^{\Re}{}_{j}\right)^{mean}$ epitomizes the mean of particles in $E^{\Re}{}_{j}$, and $F\left(G^{j}\right)^{mean}$ signifies the mean value of global best. Instead of random movement towards the best solution, the CC provides the optimal movement grounded on the evaluation of the distance between the global best solution and the subswarm. Grounded on considering the mean value of the particles in the subswarm, the distance is evaluated. Hence, each particle's movement towards the global best will be appropriate. This optimal updation aids in generating the super swarm. Following the updation of the position for each subswarm, the super swarm is generated by updating the position of each subswarm from the global best solution to the galactic best solution. The updation procedure is,

$$\left(E^{\Re}{}_{j+1}\right)^{\exp} = \begin{cases} \left(E^{\Re}{}_{j+1}\right)^{\exp}, if\ F\left(G^{j}\right) < F(G) \\ E^{\Re}{}_{j}, if\ F\left(G^{j}\right) > F(G) \end{cases} \tag{28}$$

Here, $F(G)$ symbolizes the galactic best solution. Every subswarm freely and independently scours the search space. The first step in the iteration is to calculate the position and velocity. The position and velocity (ϑ_{j}) update expressions are,

$$\vartheta_{j} \leftarrow \omega_{1}E^{\Re}{}_{j} + A_{1}R_{1}\left(\eta^{j} - E^{\Re}{}_{j}\right) + A_{2}R_{2}\left(G^{j} - E^{\Re}{}_{j}\right) \tag{29}$$

$$E^{\Re}{}_{j}{}^{u} \leftarrow E^{\Re}{}_{j}{}^{u} + \vartheta_{j} \tag{30}$$

Here, the inertial weight is implied as ω_{1}, the acceleration coefficients are exemplified as A_{1} and A_{2}, the random numbers are illustrated as R_{1} and R_{2}, the personal best is notated as η^{j}, and the global best is epitomized as G^{j}. The generation of a super swarm is done till the maximum iteration. The important features are selected as similar to the generation of a super swarm by updating the position of the sub swarm. The selected features are signified as E^{sel}.

3.5.4. Classification

Here, to detect whether the data is attacked or not, E^{sel} is inputted into the classifier called SS-RNN. The conventional Recurrent Neural Network (RNN) has a higher efficiency in detecting the output even in complex parameters. However, RNNs suffer from the problem of vanishing gradients. Hence, the proposed paper utilizes the Soft Kernal Swish activation function, which can prevent neurons from saturating and avoid the vanishing gradient problem. Figure 3 displays the proposed SS-RNN's architecture,

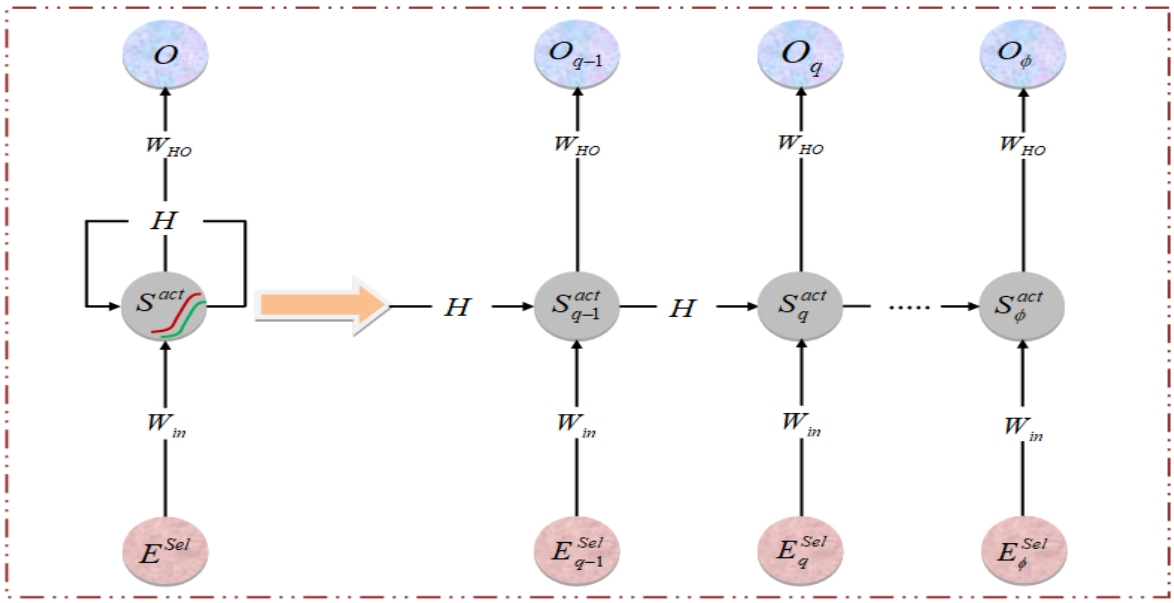

Fig 3: SS-RNN Architecture

Input layer: This layer is responsible for getting the inputs and feeding them to the hidden layer. It can be represented as,

$$E^{sel} = \{E^{sel}{}_1, E^{sel}{}_2, E^{sel}{}_3, \ldots\ldots\ldots, E^{sel}{}_\phi\} \quad (31)$$

Here, $E^{sel}{}_\phi$ indicates the ϕ^{th} selected attribute.

Hidden layer: This layer trains the input by aggregating with weight value (W) and bias value (B), and the hidden layer is overall activated by the Soft Kernal Swish activation. The hidden layer (H) is computed as,

$$H = S^{act}[W_{E^{sel}{}_q H} E^{sel}{}_q + W_{HH} H_{q-1} + B] \quad (32)$$

Where, the weights are exemplified as W (for example, the input-hidden weight value is indicated as $W_{E^{sel}{}_q H}$ and the hidden layer weight value is denoted as W_{HH}), H_{q-1} symbolizes the output of the previous hidden layer, and $E^{sel}{}_q$ specifies q^{th} selected attributes. The expression for the Soft Kernal Swish activation is given by,

$$S^{act} = \tfrac{1}{2} * \log\left((1 + E^{sel})/(1 - E^{sel})\right) \quad (33)$$

Output layer: The outcome from the hidden layer is further given to the output layer, where the attacked and non-attacked data are classified. It is represented as,

$$O = \sigma[W_{HO} H + B] \quad (34)$$

Wherein, the hidden-output weight value is explicated as W_{OH}, σ signifies the sigmoid activation function, and O symbolizes the classified data. The sigmoid activation function is defined as,

$$\sigma = \frac{1}{1 + e^{H}} \quad (35)$$

The proposed SS-RNN's pseudocode is,

Input: Number of Selected Attributes (E^{sel})

Output: Classification of attacks (O)

Begin

Initialize inputs (E^{sel}), weights (W), bias (B), $iter_{\max}$

For $(iter_{\max} = 1 to \phi)$

Evaluate the hidden layer operation using,

$$H = S^{act}[W_{\alpha H}\alpha_i + W_{HH}H_{i-1} + B]$$

Perform activation function.

$$S^{act} = 1/2 * \log\left((1+E^{sel})/(1-E^{sel})\right)$$

Evaluate the output layer operation using,

$$O = \sigma[W_{HO}H + B]$$

If $(Av == Pv)$ {

Accurate classification of attacks

} Else {

Perform Backpropagation by updating weights.

}

End if

End for

End

Loss function: Afterward, by computing the difference between the actual value (Av) and the predicted value (Pv), the loss value is assessed. The error value is computed as,

$$Loss = (Av - Pv)^2 \tag{36}$$

The attacks will be accurately detected by the model if the model's loss value or the error value is zero $(Loss = 0)$. In this case, if the loss values $Loss \neq 0$, then the backpropagation takes place by updating the weight values. Lastly, the classifier considerably detects the type of attacks. The obtained attacked data are implied as O^{att} and $O^{non-att}$.

3.6. Multipath Routing and Network Path Feature Extraction

After the attack detection, the path that is traveled by the non-attacked packets $\left(O^{non-att}\right)$ is considered as the attack-free path. Here, the nodes presented in the multiple paths are considered for extracting important features and selecting the optimal path for data transmission. As per that, the node's features, such as Response Time, Latency, and Throughput are extracted. The extracted features (ξ) are,

$$\xi = \left[\xi^1, \xi^2, \xi^3, \ldots\ldots\xi^{\delta}\right] \tag{37}$$

Here, the final path feature is notated as ξ^{δ}.

3.7. Optimal Path Selection

Here, the extracted path features (ξ) are given as the input to choose the optimal path for transmitting the data securely. For this optimal path selection, CC-GSO is utilized by the proposed work, which is explained in section 3.5.3. The network path features are assumed as the particles of the swarm in this optimization technique. Primarily, the position of each subswarm is updated to the global best fitness value. The fitness value is computed grounded on attaining the minimum response time. The fitness value $\left(F^{path}\right)$is,

$$F^{path} = \min_{res}\left[\xi^{\delta} \Leftrightarrow \xi^{\delta-1}\right] \tag{38}$$

The subswarm's position will be updated to the global best solution if the global best solution of each subswarm is higher than the personal best solution. After updating the position, the superswarm is generated by updating the position for multiple subswarms into the galactic best solution. Like the super-warm generation, the optimal path is evaluated for securely transmitting the data.

4. Result and Discussion

This phase assesses the proposed mechanism's performance. The implementation is done in the PYTHON platform.

4.1. Database Description

The publically available NSL-KDD dataset, which is an enhanced version of the KDD cup 99 gathered at the MIT Lincoln Laboratory, is utilized by the proposed system for performance evaluation. The NSL-KDD encompasses 148481 records, which are divided into training and testing data. Every single record comprises 41 features of which 34 are numerical and the remaining 7 are symbolic. From the collected data, 80% (125973) are utilized for training and 20% (22544) are used for testing.

4.1.1. Performance analysis of Hash code generation

Here, the proposed XNOR-HAVAL's performance is assessed and compared with the conventional models, namely HAVAL, TIGER, Secure Hash Algorithm (SHA)-512, and Message-Digest (MD-5).

Table 1: Hash code generation Time analysis

Techniques	Hash code Generation Time (ms)
Proposed XNOR-HAVAL	874
HAVAL	1254
TIGER	1565
SHA 512	1875
MD5	2124

Concerning hash code generation time, the performance assessment of the proposed and existing models is exemplified in Table 1. The system's efficient performance is represented by the lower time. The proposed XNOR-HAVAL is modified with the efficient technique, which aids in evaluating the small-length consonant string with a complex structure. This assists in generating the hash code quickly. However, lengthy consonant strings are used in the existing models that intend to take more time for the hash code generation. Here, the proposed model takes 874ms, which is lower than the existing models, such as HAVAL (1254ms), TIGER (1565ms), and MD5 (2124ms). Therefore, the proposed model is highly suitable to generate the hash code. As per the analysis, the proposed hashing is highly efficient.

4.1.2. Performance analysis of DoS attack Detection

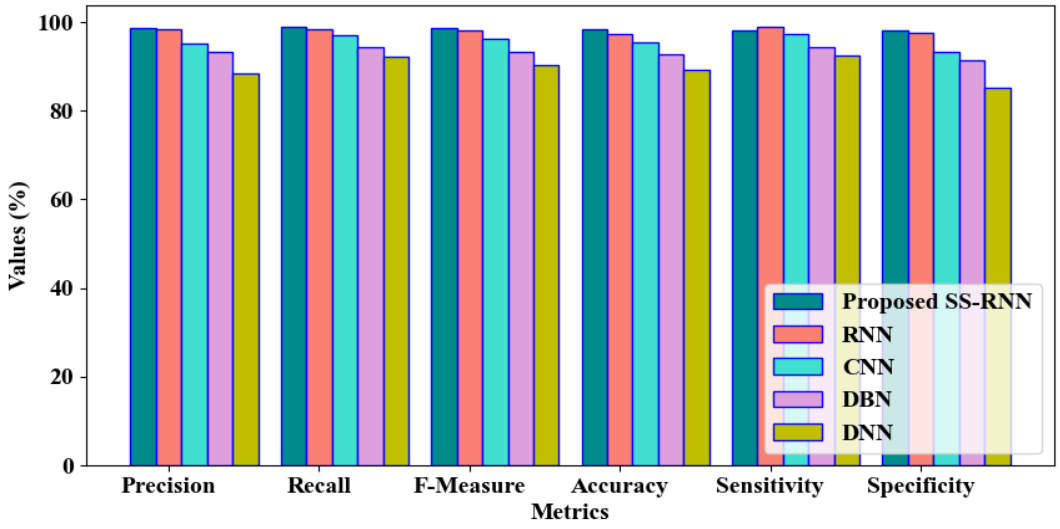

Fig 4: Performance analysis of proposed SS-RNN and existing models

Concerning precision, sensitivity, F-measure, True Negative Rate (TNR), False Negative Rate (FNR), Positive Predictive Value (PPV), accuracy, recall, specificity, True Positive Rate (TPR), False Positive Rate (FPR), Negative Predictive Value (NPV), and Training Time, the proposed SS-RNN is contrasted with RNN, Deep Belief Network (DBN), CNN, and DNN. Figure 4 displays the proposed SS-RNN and existing models' performance analysis. The proposed SS-RNN is modified with the efficient activation function, thus preventing the neurons from dying. Therefore, the proposed SS-RNN achieves accuracy, precision, recall, sensitivity, specificity, and F-measure of 98.41%, 98.56%, 98.85%, 98.14%, 98.25%, and 98.68%, respectively. Yet, the existing techniques obtain 93.61% accuracy, 93.8% precision, 95.4% recall, 95.7% sensitivity, 91.8% specificity, and 94.5% F-measure. Hence, the proposed SS-RNN classifies the attacks precisely.

Table 2: Performance analysis of proposed SS-RNN and existing methods

Metrics/ Techniques	Proposed SS-RNN	RNN	CNN	DBN	DNN
TPR (%)	98.36	98.96587	97.3254	94.20366	92.65205
TNR (%)	98.68	97.87542	93.21066	91.87453	85.2561
FPR (%)	1.324587	2.478552	6.645004	8.360247	14.2145
FNR (%)	1.239878	1.875423	2.216508	5.653201	7.350245
PPV (%)	98.14	98.54876	95.2036	93.65025	88.65783
NPV (%)	98.56	97.5466	96.54102	92.45781	89.32453
Training Time (ms)	37125	43265	47158	53147	58632

The performance assessment of proposed and conventional approaches is illustrated in Table 2. In SS-RNN, the neurons remain active for a maximum number of iterations by enhancing the existing RNN; therefore, it significantly learns the inputs. Owing to this, the proposed SS-RNN withstands lower FPR and FNR rates of 1.324587% and 1.239878%, respectively, and a higher TPR, TNR, PPV, and NPV of 98.36%, 98.68%, 98.14%, and 98.56%. The training time attained by the proposed model is 37125ms. However, the existing techniques achieved very low performance. For instance, the TPR and TNR of the existing RNN are 98.96587% and 97.87542, respectively, where the training time is 43265ms. Therefore, the proposed SS-RNN is the error-prone model.

4.1.3. Performance analysis of optimal path selection

Here, the proposed CC-GSO's performance is assessed and contrasted with the conventional approaches, namely Monarch Butterfly Optimization (MBO), Harries Hawaks Optimization (HHO), GSO, and Satin Bowerbird Optimization (SBO).

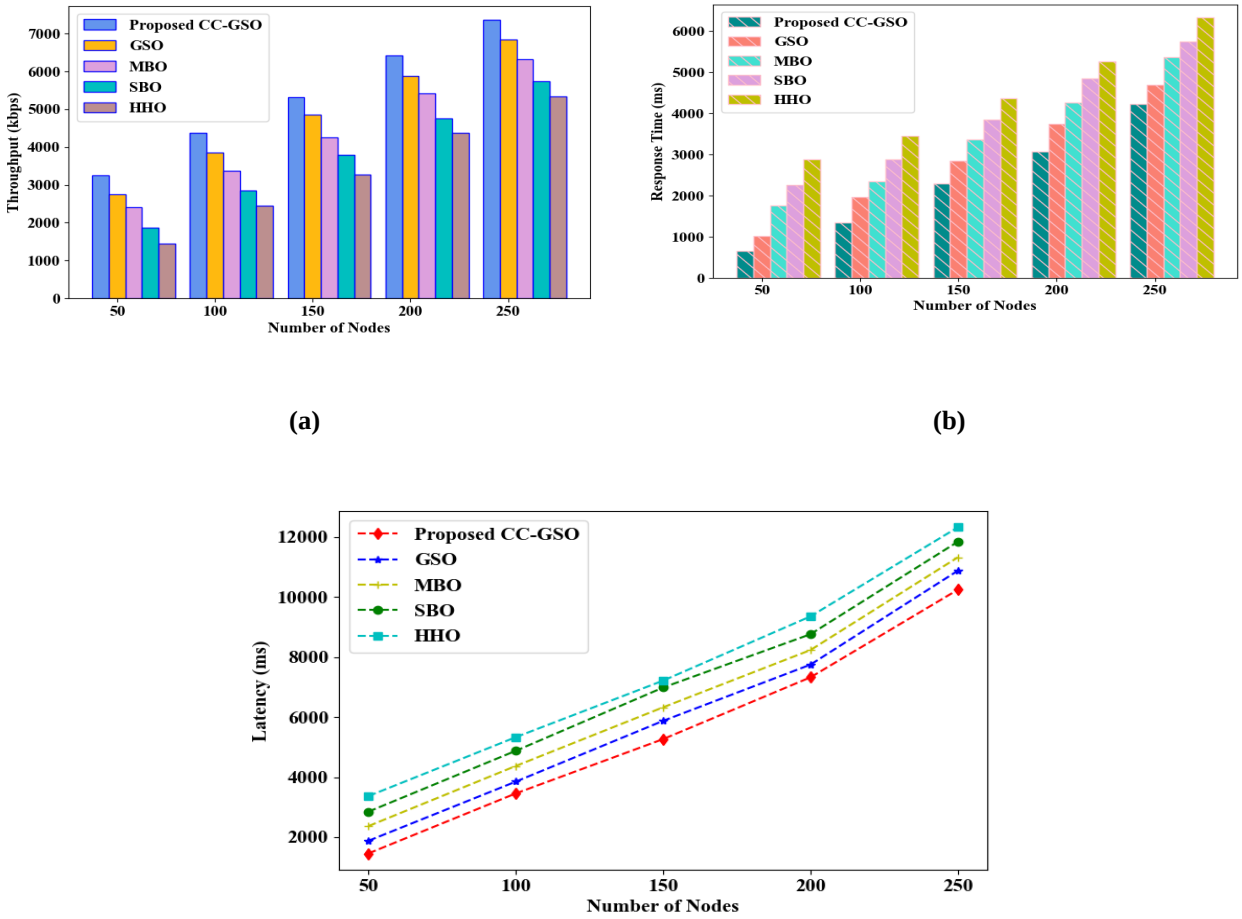

Fig 5: Performance analysis of proposed CC-GSO and existing models (a) Throughput analysis (b) Response time analysis and (c) Latency

The performance analysis of the proposed and existing models is exemplified in Figure 5. Figure 5(a) signifies the throughput, which is the ratio of the total data that reaches a receiver from the sender. The efficient performance is shown by a higher throughput value. Here, the throughput of the proposed CC-GSO is 3256kbps for 50 nodes, 4368kbps for 100 nodes, 5314kbps for 150 nodes, and 7358kbps for 250 nodes. Nevertheless, the existing model attains a throughput in the range of 1443 kbps-2748 kbps. Figure 5 (b) epitomizes the response time of the packet request. The response time of the proposed CC-GSO is 648ms for 50 nodes, 1354ms for 100 nodes, and 3065ms for 200 nodes. But the existing models take longer time, which is in the range of 1024ms to 2874ms for 50 nodes. Figure 5 (c) displays the latency analysis, which is the total time to travel the packet from the sender to the receiver node. Higher efficiency is shown by a lower time. Here, the proposed model attains 1454ms for 50 nodes, 3454ms for 100 nodes, and 5265ms for 150 nodes. However, the latency of the existing model was high. The optimal evaluation of fitness function gives the proposed system's higher performance. Hence, the proposed one is more efficient in contrast to the conventional approaches.

Table 3: Load balancing analysis in ms

Techniques/ Number of Nodes	50	100	150	200	250
Proposed CC-GSO	784	1665	2584	3452	4532
GSO	1365	2365	3256	3874	5236
MBO	1748	2874	3741	4258	5741
SBO	2265	3365	4259	4769	6245
HHO	2685	3674	4756	5362	6875

The performance assessment of the proposed and prevailing approaches concerning load balancing is exemplified in Table 3. The proposed model is modified with the efficient position updation technique, which aids in selecting the appropriate position of the subswarm toward the global best solution. This results in the generation of wide Superswarm, which handles the packets for the transmission without any network traffic. The proposed model takes 784ms for 50 nodes, 1665 ms for 100 nodes, and 4532ms for 250 nodes. However, the existing models take in the range of 1365ms to 2685ms for 50 nodes. As per the analysis, the proposed technique is highly superior in path selection.

4.1.4. Performance analysis of Attributes selection

The proposed CC-GSO's performance is evaluated concerning fitness Vs iteration, and its outcomes are compared with the GSO, Flower Pollination Optimization (FPO), Tunicate Swarm Optimization (TSO), and Ant Colony Optimization Algorithm (ACO).

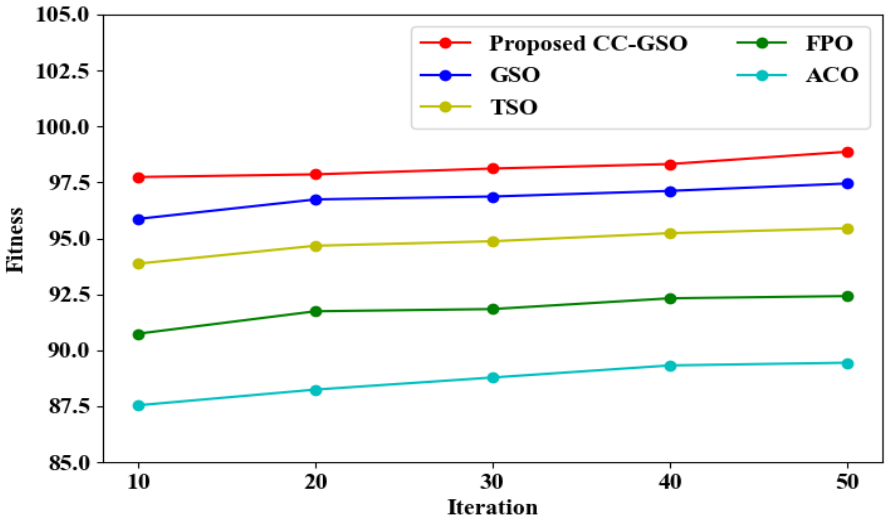

Fig 6: Convergence analysis

The optimization capability of the proposed CC-GSO and conventional approaches is contrasted in Figure 6. By using the CC technique, the proposed CC-GSO enhanced the position updation process. Thus, the proposed CC-GSO renders the optimal outcome with a minimum number of iterations, and it is converged at the iteration of 50 with the fitness value of 98.87; however, more iterations are required by the existing techniques to attain convergence. For instance, the existing GSO converged at the fitness value of 97.45%. This proves the efficacy of the proposed CC-GSO in attribute selection.

4.2. Comparative analysis with literature papers

Here, the proposed DoS attack detection model is compared with the prevailing papers.

Table 4: Comparative Analysis

Author Name	Technique Used	Evaluation Metrics			
		Throughput (kbps)	Delay (s)	Detection Rate (%)	PDR (%)
Proposed Model	CC-GSOA	7358	0.001454	97	95
(Nandi &Kannan, 2022)	HEERP	1049	0.54	-	85
(Veeraiah& Krishna, 2022)	BSWOA	0.000672	0.00377	67.1	-
(Yamini et al., 2022)	ML-AODV	140	0.34	-	80
(Shafi et al., 2023)	ETERE	16.523	0.57	-	-
(Wang et al., 2021)	CRT	11	0.42	-	77

The performance of the proposed and existing HEERP (Nandi &Kannan, 2022), BSWOA (Veeraiah& Krishna, 2022), ML-AODV (Yamini et al., 2022), ETERE (Shafi et al., 2023) (Wang et al., 2021) methods are exemplified in Table 4. As per the table, the performance attained by the proposed model for throughput, delay, detection rate, and PDR is improved by 6309kbps, 0.002s, 29.9%, and 10% than the existing methods. This is owing to the fact that existing methods utilize a predefined threshold value to classify an attacked node and lack centralization for the monitoring of the existence and arrival of any node in the network. Moreover, the methods concentrated on establishing the shortest route regardless of traffic attributes. These limitations in the prevailing methods cannot guarantee reliable network performance and create more vulnerability in the network. Therefore, the proposed CC-GSOA grounded on secure packet transmission and packet filtering for network monitoring, sequential filtering for reducing the misdetection rate, and optimal path selection based on traffic attributes reveals its superiority over the existing techniques.

5. Conclusion

Efficient time stamp-based hash chaining and SS-RNN techniques for detecting various patterns of flooding attacks in MANET have been proposed in this article. The different plans in the proposed system for identifying the patterns of attacks are the prevention of attack node entry in the network, packet filtering to detect flooding attacks, and the sequential detection and filtering of MNs. The performances of the proposed SS-RNN, BGTE-Fuzzy, and CC-GSOA methods have been assessed with existing techniques in the experimental analysis. As per the outcomes, the proposed approach performed well in contrast to the existing techniques. By avoiding flooding attacks, the proposed design displays maximal throughput and detection accuracy of 7358 kbps and 98.41%. Moreover, the proposed model ensures secure communication in MANETs by enhancing the network performance in terms of 3256kbps throughput for 50 nodes. By considering network attributes in optimal path selection, this work provides reliable data transmission. However, it does not provide data privacy in the network.

Recommendation for the future Direction: The work will be extended in the future with some enhanced cryptographic techniques to improve data security.

References

[1] Abaeian, V., Abdullah, A., Pillai, T., & Cai, L. Z. (2015). Intrusion detection forecasting using time series for improving cyber defence. *International Journal of Intelligent Systems and Applications in Engineering*, *3*(1), 28-33.

[2] Alam, T. (2020). Efficient and secure data transmission approach in cloud-MANET-IoT integrated framework. *Journal of Telecommunication, Electronic and Computer Engineering*, *12*(1), 1-10.

[3] Aytekin, T. (2018). Reservoir Sampling Based Streaming Method for Large Scale Collaborative Filtering. *International Journal of Intelligent Systems and Applications in Engineering*, *6*(3), 191-196.

[4] Banga, S., Arora, H., Sankhla, S., Sharma, G., & Jain, B. (2021). Performance Analysis of Hello Flood Attack in WSN. *Proceedings of International Conference on Communication and Computational Technologies*, 335–342. https://doi.org/10.1007/978-981-15-5077-5_30

[5] Bouyeddou, B., Kadri, B., Harrou, F., & Sun, Y. (2020). DDOS-attacks detection using an efficient measurement-based statistical mechanism.*Engineering Science and Technology, an International Journal*, *23*(4), 870–878. https://doi.org/10.1016/j.jestch.2020.05.002

[6] Dilipkumar, S., &Durairaj, M. (2023).Epilson Swarm Optimized Cluster Gradient and deep belief classifier for multi-attack intrusion detection in MANET.*Journal of Ambient Intelligence and Humanized Computing*, *14*(3), 1445–1460. https://doi.org/10.1007/s12652-021-03169-x

[7] Elhigazi, A., Razak, S. A., Hamdan, M., Mohammed, B., Abaker, I., &Elsafi, A. (2020). Authentication Flooding DOS Attack Detection and Prevention in 802.11.*IEEE Student Conference on Research and Development*, 325–329. https://doi.org/10.1109/SCOReD50371.2020.9250990

[8] Farahani, G. (2021). Black Hole Attack Detection Using K-Nearest Neighbor Algorithm and Reputation Calculation in Mobile Ad Hoc Networks.*Security and Communication Networks*, *2021*, 1–15. https://doi.org/10.1155/2021/8814141

[9] Gadzama, E., &Saidu, I. (2022). Developing an Efficient Hybridized Routing Protocol for Mitigation of Flooding Attacks in Mobile Ad-Hoc Networks.*Nigerian Defence Academy Journal of Military Science and Interdisciplinary Studies*, *1*(1), 1–8.

[10] Gautam, D., &Tokekar, V. (2020). A novel approach for detecting DDoS attack in MANET.*Materials Today: Proceedings*, *29*, 674–677. https://doi.org/10.1016/j.matpr.2020.07.332

[11] Hai, T., Zhou, J., Lu, Y., Jawawi, D., Wang, D., Onyema, E. M., &Biamba, C. (2023). Enhanced security using multiple paths routine scheme in cloud-MANETs.*Journal of Cloud Computing*, *12*(1), 1–23. https://doi.org/10.1186/s13677-023-00443-5

[12] Işık, Ş., Özkan, K., Gerek, Ö. N., & Doğan, M. (2016). A new subspace based solution to background modelling and change detection. *International Journal of Intelligent Systems and Applications in Engineering*, *4*(Special Issue-1), 82-86.

[13] Islabudeen, M., & Devi, M. K. K. (2020). A Smart Approach for Intrusion Detection and Prevention System in Mobile Ad Hoc Networks Against Security Attacks. In *Wireless Personal Communications* (Vol. 112, Issue 1). Springer US.https://doi.org/10.1007/s11277-019-07022-5

[14] Kamoji, S., Kalla, M., & Shamshi, I. (2022). A Framework for Flood Extent Mapping using CNN Transfer Learning. *International Journal of*

Intelligent Systems and Applications in Engineering, *10*(3s), 150-157.

[15] Karthigha, M., Latha, L., &Sripriyan, K. (2020). A Comprehensive Survey of Routing Attacks in Wireless Mobile Ad hoc Networks. *Proceedings of the 5th International Conference on Inventive Computation Technologies*, 396–402. https://doi.org/10.1109/ICICT48043.2020.9112588

[16] Khalfaoui, H., Farchane, A., & Safi, S. (2023). An Overview of the Security Improvements of Artificial Intelligence in MANET.*International Conference on Cybersecurity, Cybercrimes, and Smart Emerging Technologies*, *4*, 135–146. https://doi.org/10.1007/978-3-031-21101-0_11

[17] Kshirsagar, D., & Kumar, S. (2022). A feature reduction based reflected and exploited DDoS attacks detection system. *Journal of Ambient Intelligence and Humanized Computing*, *13*(1), 393–405. https://doi.org/10.1007/s12652-021-02907-5

[18] Mittal, M., Kumar, K., &Behal, S. (2023). Deep learning approaches for detecting DDoS attacks: a systematic review. *Soft Computing*, *27*(18), 13039–13075. https://doi.org/10.1007/s00500-021-06608-1

[19] Nandi, M., &Kannan, A. (2022). An Optimized and Hybrid Energy Aware Routing Model for Effective Detection of Flooding Attacks in a Manet Environment. *Wireless Personal Communications*, *127*(3), 2515–2533.https://doi.org/10.1007/s11277-021-09079-7

[20] Narmadha, A. S., Maheswari, S., &Deepa, S. N. (2023). Watchdog malicious node detection and isolation using deep learning for secured communication in MANET.*Automatika*, *64*(4), 996–1009.https://doi.org/10.1080/00051144.2023.2241766

[21] Nishanth, N., &Mujeeb, A. (2021). Modeling and Detection of Flooding-Based Denial-of-Service Attack in Wireless Ad Hoc Network Using Bayesian Inference. *IEEE Systems Journal*, *15*(1), 17–26. https://doi.org/10.1109/JSYST.2020.2984797

[22] Qanbar, M. M., & Tasdemir, S. (2019). Detection of malaria diseases with residual attention network. *International Journal of Intelligent Systems and Applications in Engineering*, *7*(4), 238-244.

[23] Rahouti, M., Xiong, K., Ghani, N., &Shaikh, F. (2021).SYNGuard: Dynamic threshold-based SYN flood attack detection and mitigation in software-defined networks. *IET Networks*, *10*(2), 76–87. https://doi.org/10.1049/ntw2.12009

[24] Ramesh, S., Yaashuwanth, C., &Muthukrishnan, B. A. (2020). Machine learning approach for secure communication in wireless video sensor networks against denial-of-service attacks. *International Journal of Communication Systems*, *33*(12), 1–12. https://doi.org/10.1002/dac.4073

[25] Ravi, N., Shalinie, S. M., &Theres, D. D. J. (2020). BALANCE: Link Flooding Attack Detection and Mitigation via Hybrid-SDN. *IEEE Transactions on Network and Service Management*, *17*(3), 1715–1729. https://doi.org/10.1109/TNSM.2020.2997734

[26] Salunke, K., &Ragavendran, U. (2021). Shield Techniques for Application Layer DDoS Attack in MANET: A Methodological Review. *Wireless Personal Communications*, *120*(4), 2773–2799.https://doi.org/10.1007/s11277-021-08556-3

[27] Sbai, O., &Elboukhari, M. (2022). Deep learning intrusion detection system for mobile ad hoc networks against flooding attacks.*IAES International Journal of Artificial Intelligence*, *11*(3), 878–885. https://doi.org/10.11591/ijai.v11.i3.pp878-885

[28] Shafi, S., Mounika, S., &Velliangiri, S. (2022). Machine Learning and Trust Based AODV Routing Protocol to Mitigate Flooding and Blackhole Attacks in MANET. *Procedia Computer Science*, *218*, 2309–2318. https://doi.org/10.1016/j.procs.2023.01.206

[29] Shafi, S., Mounika, S., &Velliangiri, S. (2023). Machine Learning and Trust Based AODV

Routing Protocol to Mitigate Flooding and Blackhole Attacks in MANET. *Procedia Computer Science*, *218*, 2309–2318. https://doi.org/10.1016/j.procs.2023.01.206

[30] Srinivas, T. A. S., &Manivannan, S. S. (2020). Prevention of Hello Flood Attack in IoT using combination of Deep Learning with Improved Rider Optimization Algorithm.*Computer Communications*, *163*, 162–175. https://doi.org/10.1016/j.comcom.2020.03.031

[31] Talukdar, M. I., Hassan, R., Hossen, M. S., Ahmad, K., Qamar, F., & Ahmed, A. S. (2021). Performance Improvements of AODV by Black Hole Attack Detection Using IDS and Digital Signature.*Wireless Communications and Mobile Computing*, *2021*, 1–13. https://doi.org/10.1155/2021/6693316

[32] Veeraiah, N., & Krishna, B. T. (2022). An approach for optimal-secure multi-path routing and intrusion detection in MANET. *Evolutionary Intelligence*, *15*(2), 1313–1327. https://doi.org/10.1007/s12065-020-00388-7

[33] Yamini, K. A. P., Stephy, J., Suthendran, K., & Ravi, V. (2022). Improving routing disruption attack detection in MANETs using efficient trust establishment.*Transactions on Emerging Telecommunications Technologies*, *33*(5), 1–16.https://doi.org/10.1002/ett.4446

[34] Zalte, S.., Kamat, R.., &Ghorpade, V.. (2020). Mitigation of DDoS Attack in MANET.*International Journal of Engineering and Advanced Technology*, *9*(6), 410–413. https://doi.org/10.35940/ijeat.e95400.089620