Discuss the Prevention of Jelly Fish Attack by AODV and APD-JFAD in MANET

Dr.Sumathi Ganesan[1], Dr.Manikandan.R[2],

[1]Assistant Professor, Department of Computer science and engineering, Annamalai University,Chidambaram

[2]Assistant Professor, Department of Computer science and engineering, Government college of Engineering, Tanjore.

ABSTRACT** Mobile ad hoc networks (MANETs) re vulnerable to various types of attacks due to inherently in-secure wireless communication medium and multihop routing communication process. In this paper, we analyze the behavior and impact of JellyFish attack over MANETs. We have implemented and evaluated all three variants of JellyFish attack namely JF-reorder, JF-delay and JF-drop through simulation processes. These attacks exploit the behavior of closed loop protocols such as TCP and disturb the communication process without disobeying any protocol rules, thus the detection process becomes difficult. Consequently, traffic is disrupted leading to degradation in network throughput. Through extensive simulation results that d are obtained using an industry standard scalable network simulator called NS2, impact of these attacks in terms of network throughput, overhead incurred and end-to-end delay is analyzed and used for devising detection and countermeasure. We have proposed a light-weight direct trust-based detection (DTD) algorithm which detect and remove a JellyFish node from an active communication route. Simulation results are provided, showing that in the presence of malicious-node attacks, the CBDS outperforms the existing and compared with proposed JF detection scheme in terms of packet delivery ratio and routing overhead.Mobile Adhoc Networks (MANETs) is surrounded by tons of different attacks, each with different behavior and aftermaths. One of the serious attacks that affect the normal working of MANETS is DoS Attack. A sort of DoS attack is Jellyfish attack, which is quite hard because of its foraging behavior. The Jellyfish attack is regarded as one of the most difficult attack to detect and degrades the overall network performance. In order to combat Jellyfish attack in MANETs, this paper proposes a novel technique called APD-JFAD (Accurate Prevention and Detection of Jelly Fish Attack Detection) and AODV. It is a fusion of authenticated routing-based framework for detecting attacks and Support Vector Machine. SVM is utilized for learning packet forwarding behavior. The proposed technique chooses trusted nodes in the network for performing routing of packets on the basis of hierarchical trust evaluation property of nodes. The technique is tested using NS-2 simulator against other existing techniques i.e. ABC, MABC and AR- AIDF-GFRS algorithms by various parameters such as throughput, PDR, dropped packet ratio and delay. The results prove that APD-JFAD is highly efficient in Jellyfish attack detection and also performs well as compared to other algorithms.

***INDEX TERMS** Jellyfish attack; Trust Evaluation; CBDS; malicious node; Packet forwarding behavior; Support Vector Machine; ABC.*

I. INTRODUCTION

Due to the widespread availability of mobile devices, mobile ad hoc networks (MANETs) have been widely used for various important applications such as military crisis operations and emergency preparedness and response operations. This is primarily due to their infrastructure less property. A mobile ad hoc network (MANET) is a continuously self- configuring, infrastructure-less network of mobile devices connected without wires. Ad hoc is Latin and means "for this purpose". Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. Each must forward traffic unrelated to its own use, and therefore be a router. The primary challenge in building a MANET is equipping each device to continuously maintain the information required to properly route traffic. Such networks may operate by themselves or may be connected to the larger Internet. The growth of laptops and 802.11/Wi-Fi wireless networking has made MANETs a popular research topic since the mid-1990s. Many academic papers evaluate protocols and their abilities, assuming varying degrees of mobility within a bounded space, usually with all nodes within a few hops of each other. Different protocols are then evaluated based on measures such as the packet drop rate, the overhead introduced by the routing protocol, end-to-end packet delays, network throughput, ability to scale, etc.

In a MANET, each node not only works as a host but can also act as a router. While receiving data, nodes also need cooperation with each other to forward the data packets, thereby forming a wireless local area network. These great features also come with serious drawbacks from a security point of view. Many research works have focused on the security of MANETs.

MANET has an ability to intelligently handle all sorts of topological changes as well as node malfunctioning issues via network re-configuration technique [1]. If any node in MANET leaves the network and causes breakage in links, affected nodes can immediately request for new routing paths in a matter of seconds so that network transmission continues [1]. This can cause some issues with regard to delay, but the network remains operational and work normally. In general terms, **MANETS are highly vulnerable to security attacks** because of the following reasons: i) No centralized administration for node authentication, no network management utility/provision and authorization of nodes entering or leaving the network; ii) Multi-Hop Communication; iii) Dynamic and Frequent Changing topology; iv) Limited resources in terms of non- implementation of secure routing protocol/algorithm because of limited processing power of nodes [1].

The basic operations of MANETs lack efficient security features in which all intermediary nodes from source to destination are assumed as trustworthy at different layers for packet transmission. The most critical issue faced by MANET is trusting intermediary nodes when operating in dynamic topology. It is highly easy for an attacker to eavesdrop the network, especially in wireless communication scenario and perform packet capturing and even break-in the network and compromise trustworthy nodes. Without strict security methodologies, all the layers, especially the network layer and transport are prone to serious threats which affect the overall MANET operational scenarios. UDP is used by most of the applications in MANET as the transport layer protocol, which is the prime reason of errors and unreliable communication process because of interference and dynamic changing topology. Various applications like FTP, HTTP requires end to end reliable communication and mostly relies on TCP protocol to reliable end-to-end packet delivery [2]. In MANETs, TCP does not perform well, and performance decreases gradually when network mobility increases . The reason is that TCP has no detection mechanism to detect whether any packet is dropped during transmission between source and destination. It may due to network properties or congestion.

The **paper** proposes a novel defense mechanism based on Support Vector Machine called Accurate Prevention and Detection of Jelly Fish Attack Detection (APD-JFAD) for Jelly-Fish attack, which is also regarded as sort of Denial of Service (DoS) attack on TCP based MANETs. Jellyfish attack is regarded as most crucial DoS, which is harder to detect than other wireless attacks in MANETs. This kind of attacks makes delay in network, and hence the overall throughput in the network decreases. In the new method, a node is assumed to launch Jellyfish Attack, which is hard to detect. Node property based hierarchical trust evaluation is carried out in the proposed technique. As a result to large extent, Jelly Fish Attack is defended in MANETs by choosing trusted paths for routing packets from source to destination. The proposed technique is highly efficient in precision detection as well as preclusion of jellyfish attack in MANET.

Related work is presented in Section II. Section III describes a detailed overview of the Jellyfish attack along with its variants. Section IV shows the proposed technique (ADP-JFAD). Section V highlights the experiments and performance comparison with other techniques namely ABC, MABC, AR-AIDF-GFRS with regard to various network parameters like PDR, Throughput, Packet Dropping Ratio and Delay. Section VI concludes the paper with future scope.

II. RELATED WORKS

In order to assure packets, reach the destination, the network has primary responsibility to provide a secure mechanism between all nodes (sender, destination as well as intermediary nodes). In MANETs, if any one malicious node enters the operating network, it can lead to incorrect network performance and network will show the following outcomes:

Worm Hole attack It is one of the most sophisticated and rigorous attacks in MANETs. Here, two attackers place themselves strategically in the network. Once strategically placed, the attacker pair advertises path through them as the shortest one. This is to ensure traffic diversion through these nodes. The attackers can eavesdrop the communication through them and record it for future use. The Worm Hole attacker creates a tunnel in order to record the ongoing communication and traffic at one network position and channels it to another position in the network.

Black Hole attack [2]: A malicious node (called blackhole) sends fake routing information, claiming that it has an

optimum route and causes other nodes to route data packets through itself. For example, in AODV (Ad-hoc On-demand Distance Vector) , the attacker can send a fake RREP1 (including a fake destination sequence number equal to or higher than the one contained in the RREQ2) to the source node, claiming that it has a sufficiently fresh route to the destination node. This causes the source node to select the route that passes through the attacker. Once paths have been established, blackhole simply drops all packets leading to a DoS attack.

Sybil attack: In this attack [4], the attacker assumes multiple identities and uses these identities to launch a distributed DoS attack, establish non-existent routes disrupting traffic, fabrication of control/data messages, etc. Multiple identities help the attacker in evading detection.

Grayhole attack: The attacker node drops some packets that pass through it.

Selfish Node Misbehaving: In MANETs, the nodes participate in a collaborative manner to forward packets to other nodes. A node refusing to forward packets in order to conserve its limited resources is termed a ‘selfish node’. This selfishness causes network and traffic disruptions.

In addition, some of these methods require specific environments or assumptions in order to operate. In general,

detection mechanisms that have been proposed so far can be grouped into two broad categories.

1) Proactive detection schemes that need to constantly detect or monitor nearby nodes. In these schemes, regardless of the existence of malicious nodes, the overhead of detection is constantly created, and the resource used for detection is constantly wasted. However, one of the advantages of these types of schemes is that it can help in preventing or avoiding an attack in its initial stage .

2) Reactive detection schemes are that trigger only when the destination node detects a significant drop in the packet delivery ratio.

Among the above schemes are the ones previously proposed, in which considered as benchmark schemes for performance comparison purposes. In 2ACK scheme for the detection of routing misbehavior in MANETs. In this scheme, two-hop acknowledgement packets are sent in the opposite direction of the routing path to indicate that the data packets have been successfully received.

The disadvantage of the system are described as follows : a) Lack of central point for authentication, network

TABLE I
OVERVIEW OF VARIOUS SORTS OF ATTACKS IN MANETS

NetworkLayer	AttackType
Application	
Transport Layer	SYN Flooding, Session Hijacking
Network Layer	Blackhole, Wormhole, Byzantine, Information Disclosure, Link Spoofing attack, Rushing Attack, Gray Hole, Flooding; Routing Attacks: Routing Table Poisoning, Routing Table Overflow, Route cache poisoning, Replay Attacks
MAC Layer	Denial of Service (DoS), Bandwidth Stealth, MAC Targeted attack, WEP targeted attack
Data Link Layer	DoS Attack, MAC Targeted attack, Traffic Analysis and Monitoring.
Physical Layer	Jamming, Device Tampering, Eavesdropping, Malicious Message Injection, Stolen or CompromisedAttack

Nayyar *et al.* proposed a defense mechanism to detect and combat Jellyfish attack in MANETs using Genetic Algorithm to improvise overall network performance regarding delay, throughput, PDR, and energy efficiency. The proposed technique is highly efficient to provide a defense mechanism against Jellyfish Periodic Dropping attack. In paper [1] analyzed performance of the AODV routing protocol with and without Jellyfish attack. They proposed an approach called Collaborative Intrusion Detection and Prevention Approach for detecting Jellyfish attack. It successfully detects attacker nodes as well as the number of infected packets and improvised the throughput and packet delivery ratio in MANETs.

In paper [1] proposed a non-cryptography approach which is resilient against JFDV attack for OLSR routing protocol. With this approach, a node is considered as a malicious node termed as originator of the Jellyfish attack and compared the network performance in terms of delay. Simulation of the proposed approach proves that it improvises packet delivery ratio and throughput in MANETs. In paper [1] proposed a defense mechanism to prevent MANETs from buffer overflow and Jellyfish attack by design of a secure routing protocol. Therein, attacker node makes use of hello flood technique to deploy attack, and buffer values get modified in trustworthy nodes. The proposed technique was analyzed on AODV and ODMRP routing protocols. Simulation shows that it is efficient to

combat Jellyfish attack and improvises throughput, PDR and delay in the network.Satheeshkumar and Sengottaiyan proposed ACO- CBRP (Ant Colony Optimization ased Clustered Routing protocol) for detecting and combating Jellyfish attack in MANETs. In this approach, clustering procedure was done by Ant Colony Optimization, and key management scheme was proposed for enhancing security. The performance of the proposed technique was determined using NS-2 simulator against the other methods namely CBDS (Collaborative Bait Detection Scheme). The results stated that ACO-CBRP is efficient regarding overall PDR, overhead and improvises network lifetime of the nodes. Thomas *et al.* proposed cure link establishment method to combat the Jellyfish reorder attack on MANETs based on ODMRP protocol. They analyzed serious vulnerabilities and backdoors in multicast routing protocols and proposed an algorithm for defense which is highly secure and robust. The proposed technique was tested using EXata-Cyber simulator using a combined network comprising MANET and UAV. The results showed that the proposed approach improvises throughput and packet delivery ration in MANETs. Kumar and Babu proposed DSMANET to detect malicious nodes and improvised the overall throughput and routing overhead.

Kalucha *et al.*applied Artificial Bee Colony (ABC) algorithm to solve a wide range of problems in MANETs with regard to attacks defense, mobility and high scalability. The authors highlighted ABC as one of the best optimization swarm intelligence techniques having simple and robust behavior to solve multimodal and multidimensional problems. ABC is highly efficient as compared to other swarm-based techniques like (PSO) and (ACO) for MANETs in terms of functional optimization. Sailaja *et al.* analyzed nature-based algorithms for MANETs based on Ant Colony and Bee Colony. They highlighted the importance of ABC as well as BeeAdhoc based routing protocol for MANETs in terms of attack counterfeiting, dynamic mobility, robustness, scalability, congestion avoidance and overall effective routing.

In paper [1] proposed a dynamic hybrid technique based on ABC and negative selection (NS) methods, known as BeeID, for detecting all sorts of intrusion in MANETs. The methodology has three stages: Training, detection and updating. In training, a niching ABC algorithm i.e. NicheABC, runs a negative selection technique several times to output a set of mature negative detections to cover the non-self space. During detection stage, mature negative detectors are utilized to distinct normal and malicious network activities. In the updating phase, mature negative detections are used for total updating. Prasad and Rao proposed a hybrid Improved Artificial Bee Colony and Simulated Annealing (HIASA) based algorithm for detecting various types of attacks in MANETs like sybil attack, wormhole attack and routing attacks. HIASAalgoriexamines the attack via Simulated Annealing (SA) initialization, self-adaptive mechanism for employed bees and onlooker bees steps and chaotic opposition based learning (OBL) for scout bee step. The initial search algorithm investigates the most hopeful search space regions while the exploitation's capability is enhanced via Simulated Annealing through auditing of surroundings of basic solutions. Self-adaption mechanism was used to equalize the analyzing capability and convergence speed of algorithm. OBL was used to enhance convergence implementation.

A novel defense mechanism for detecting and counterfeiting jellyfish attack in MANETs is presented in this research paper. It is the mix of authenticated routing-based framework for detecting attacks and genetic fuzzy rule-based system. The **difference between the proposed algorithm and the related ones** is that the new algorithm makes use of a machine learning technique namely SVM for learning packet forwarding behavior and chooses trusted nodes in the network for performing routing of packets by hierarchical trust evaluation property of nodes. It is indeed realized that this initiative could enhance packet delivery, less delay, less packet delay and overall best throughput in MANETs.

III. JELLY FISH ATTACK IN MANETS

In this section, we give a brief overview of Jellyfish attack along with classification .

A. *Overview of Jellyfish Attack*

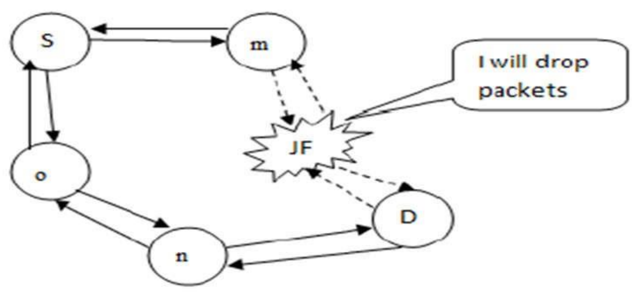


FIGURE 1. Jellyfish attack in MANETS

Jellyfish attack comes under the classification of passive attack and is regarded as a type of Denial of Service (DoS) attack. It maintains complete compliance with control and data protocols for making detection and prevention highly challenging tasks to work upon. Jellyfish attack introduces delay in network before any sort of transmission and receipt of packets happen between the communicating nodes [35]. Jellyfish attack degrades the performance of both TCP and UDP packets and performs in the same manner like Blackhole attack. The only difference is that, in black hole attack, the infected node drops all the packets whereas Jellyfish malicious node introduces delay during packet forwarding [35]. Attackers can also scramble packet ordering before delivering packets to the destination node. ACK based flow control mechanism generates duplicate ACK packets in the network . Jellyfish attack is primarily targeted towards closed loop flows with the ultimate goal to disrupt normal operation of the network by packet dropping. Jellyfish attack is highly vulnerable in TCP traffic in which cooperative nodes can hardly distinguish between attacks from network congestion.

B. *Jellyfish attack variants*

Fig. 2 highlights the variants of Jellyfish attack: Jellyfish Reordering attack, Jellyfish periodic dropping attack and Jellyfish Delay Variance attack.

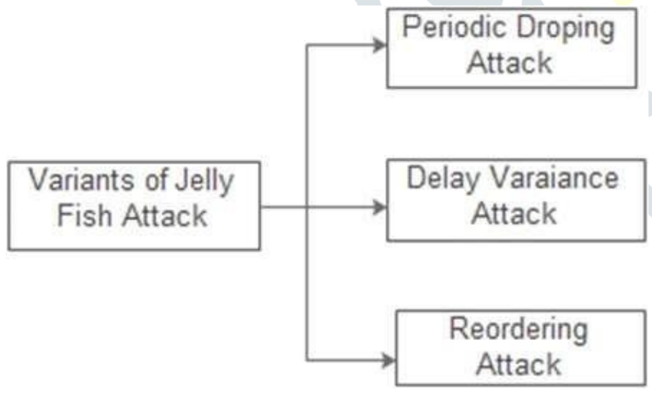
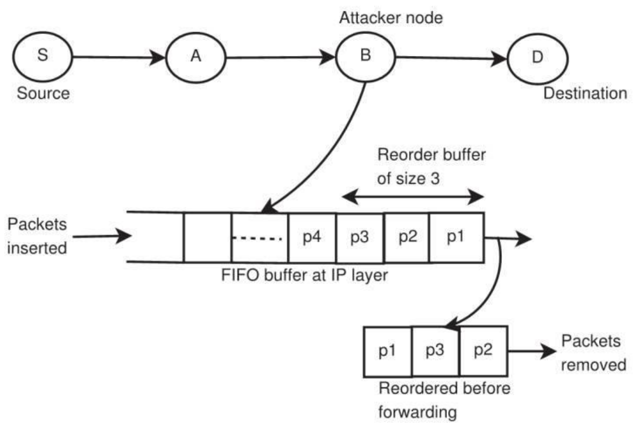


FIGURE 2. Jellyfish Attack Types

 Jellyfish Reordering Attack: In this attack, the malicious node performs packet reordering before transmitting packets to the destination node. Some of the ACKs of the reordered packets are not received by the destination node in pre-specified time so that the sender has to perform packet retransmission. Considering the receiver, every time a packet is received, ACK for the packet is automatically generated. In case of any fluctuations, the sender receives duplicate ACK packets. Duplicate ACK packets in turn create a threshold level, and TCP will initiate a flow control mechanism. In case of Jellyfish reordering packet, the Jellyfish attack node creates a buffer reordering before transmitting packets. The resulting reordering increases the number of ACK packets in the network, which decreases the overall throughput and impact the network utilization performance.

 Jellyfish periodic dropping attack: Under this, the jellyfish performs discarding of packets for a certain period of time, which makes the sender to enter into a timeout situation. In order to handle the timeout situation, TCP enters into the slow start phase of packet transmission with the impacts the throughput of the network. As a result, packet dropping increases and the overall network becomes unreliable and inefficient as packets do not reach the destination in the correct shape and



time.

 Jellyfish delay variance attack: Under this, the node impacted by jellyfish attack makes delay packet delivery at random intervals without changing the packet order. This in turn can impact the network via congestion.

III. PROPOSED METHOD

Jellyfish Attack

JellyFish attack maintains compliance with both the control and data protocols to make its detection and prevention

difficult. [12] Due to no functional distinction among mobile nodes in MANETs, an intermediate node can

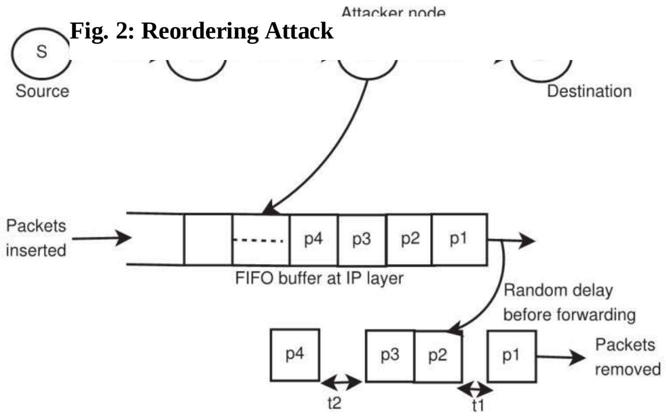


Fig. 2: Reordering Attack

introduce acritical vulnerability for TCP congestion control mechanism. Such a compromised/malicious node alters its forwarding behavior as described in following variants of JF attacks.

Jellyfish Reordering Attack

As the name suggests, an attacker node reorders some of the packets before forwarding them. As ACKs of some of

reordered packets are not received in time, the sender need to retransmit them again. From receiver's perspective, each time a packet is received, an ACK is generated. For out-of-order packets, sender shall received duplicate ACK messages. TCP initiates its flow control mechanism if these duplicate ACK messages exceed a threshold (3 in our case). In our implementation of JF reordering attack, the JF node creates a reordering buffer of size k in its input queue as shown in Fig. 2. The data packets in this buffer are reordered before being forwarded. This attack can be implemented in following two ways:

1. Reorder packets in batches of k packets each. Algorithm includes three steps e(1) Reorder current batch of k packets, (2) Forward the reordered batch and (3) Wait for next batch. In our implementation of JF-reorder attack, we have used this method.

2. Reordering is done using a sliding window of k size and each time a packet is sent, this window is shifted by one packet. Reordering is initiated on available k packets each time a packet is about to leave the reordering buffer.The packet reordering results in an increase of duplicate ACK packets sent by the destination node. When the source receives three consecutive duplicate ACKs, it initiates flow and congestion control mechanism, which eventually decreases the network throughput leading to under utilization of available network resources.***Jellyfish Periodic Dropping Attack***

In this attack, JF nodes randomly discard some packets over a specified period during communication process. In this way, incorrect route congestion information is conveyed to TCP, which uses dropping of packets as an indication of congestion on the route. The JF-node may either choose to discard a fraction of packets (e.g., 10 packets from every 100 packets) or may discard all the packets received during a slice of time (e.g., discarding data packets for few milliseconds every second near the TCP sender timeout). This forces TCP to enter the retransmission timeout (RTO) and to increase its RTO value. As the flow becomes stable, attacker repeats the above strategy to sustain the attack and keep the data flow rate low. An instance depicting the periodic drop attack is shown in Fig. 3.

Fig. 3: Periodic Dropping

As JF-node starts discarding packets for some duration, the sender will eventually enter in timeout. TCP handles the timeouts by entering in slow start phase leading to decrease in the network throughput. The throughput decreases as the frequency of packets dropped by the attacker node increases. To maximize the impact of the attack, a JF-node will drop packets as soon as the TCP sender exits its slow start phase. Due to this, the flow will always be in a fragile slow-start state.

Jellyfish Delay Variance Attack

Round trip time (RTT) of data packets vary considerably due to congestion. Though TCP has a flow control mechanism to adapt to the changes, it cannot determine if the change in RTT is due to dynamic wireless topology, network congestion or JellyFish attack. Also, changes in RTT force TCP to increase RTO. By delaying packets randomly, a JF node can initiate this attack resulting in.

Fig. 4: Delay Variance

- Self-clocking of TCP leading to increased collisions and data packet loss,
- Wrong estimation of the available bandwidth for delay based congestion control protocols such as TCP Westwood and TCP Vegas,
- Very high RTO estimate thus decreases network throughput due to delayed detection of congestion in the network.

In delay variance attack, JF nodes are selfishly delaying packets. Resultant increase in RTT misleads the sender TCP, which increases its congestion window size and sends traffic in bursts. It will eventually result in more collisions. Fig. 4 shows an instance of our implemented delay variance attack.

In order to defend the MANET network against Jellyfish attack, a novel methodology called Accurate Prevention and Detection of Jellyfish Attack Detection (APD-JFAD) is proposed. Node property based hierarchical trust evaluation is carried out in the proposed technique. As a result, the Jellyfish attacks are prevented by choosing only trusted nodes for route path construction. In the proposed technique, Support Vector Machine is utilized for packet forwarding behavior learning. This technique guarantees the detection of Jellyfish attack with high precision. Fig. 3 demonstrates the complete working cycle of the proposed technique.

direct trust is computed. Dependent upon information that is provided by neighbor nodes, indirect trust is computed

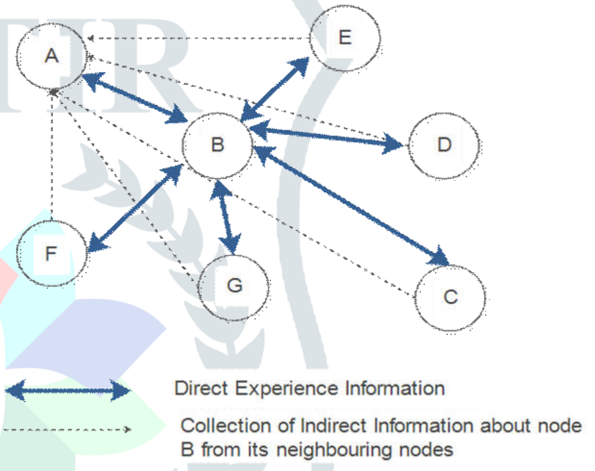

A
B
C
D
E
F
G
Direct Experience Information
Collection of Indirect Information about node B from its neighbouring nodes

FIGURE 4. Node A accesses the trust of node B

Metrics for Computing Trust in MANETs:

Table II enlists various metrics used for Trust Calculation of node in a MANET environment [27].

Overall Monitoring and jelly fish attack detection

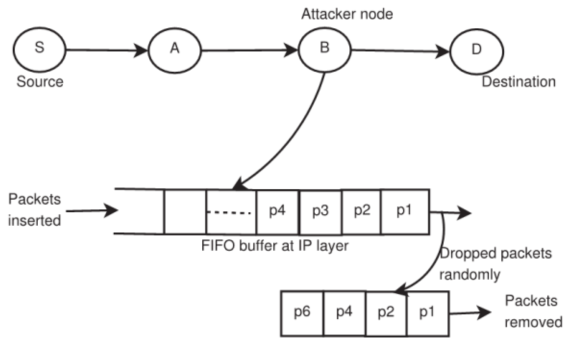


FIGURE 3. Mechanism of APD-JFAD for combating Jellyfish attack in MANETs

A. Node Property-based Hierarchical Trust Evaluation

Trust Computation of Nodes in MANETS:

For assessing the trust value of sensor node to determine intrusions in MANET, the trust calculation is dependent upon the node's properties and endorsements from neighbor technique. Any node in MANET can determine trust of neighboring nodes. Neighbor nodes are those in radio range of another. The trust is known as the confidence level, which is based on time. This value fluctuates with the time when any sort of transactions happen between MANET nodes. Trust is computed on the basis of previous experience with node and the endorsements, provided by neighboring nodes. Here, previous experience signifies the behavior of the node that is dependent upon diverse aspects i.e. trust metrics. Direct Trust (DT) is computed dependent upon trust metrics. Indirect Trust (IT) is computed dependent upon the indirect information provided via recommendation of neighbor nodes. Overall Trust (OT) is computed by direct as well as indirect trust dependent upon the individual effect of kind of trust. Fig. 4 demonstrates Node A assessing the trust of node B. Depending upon the unswerving experience information, Honesty Intimacy Unselfishness

Every node in MANET is anticipated to determine up-to-date trust metric values regarding its neighboring node for every activity happening in the network. In order to compute the Direct Trust (DT) of neighbor node, the record produced by observation of neighbor nodes is utilized. By the means of information acquired from all other neighboring nodes, Indirect trust (ID) of any neighbor node is computed. The following overviews the varied trust metrics:

- **Packet Forwarding**: This metric is utilized to identify the denial to transfer any packet that is forwarded from source node to the neighboring node for additional forwarding.
- **Availability to hello message**: Identification of nodes inside radio range and capable of sending the packets.
- **Packet Delay**: Detection of delay in time to reach the destination node by a transmitted packet.
- **Packet Integrity/Precision**: Verifying that no modification is made in the packet while transferring from source to destination.
- **Remaining Energy**: Even though energy is not clean metric of trust, considering energy enables balancing of the node

Reputation: In the trust calculation method, neighboring nodes are demanded to offer indirect information regarding node. This would be useful when there is no direct information exists regarding the trust of the node.

In this trust computation technique, trust metrics are divided into two categories: High Priority and Low Priority. High priority trust metrics identify the important node functionalities. Thus, these trust metrics are not considered to go below the level of trust threshold, e.g. values of trust metrics for instance data packets forwarded, control packet forwarded are not considered below the higher priority threshold as the functionality of nodes remains unseen within these metrics.

B. Hierarchical Trust Level Evaluation

Considering the real PSN, the total number of nodes is highly limited. As a result, a hierarchical evaluation system is required. In this system, nodes are clustered in 7 groups. Furthermore, the *GT* and *LT* assessment reunited to attain an efficient HTL evaluation system. The common trust evaluation on the node *i* from TS is specified by *GT(i)*, when *LT(i)* signifies the trust evaluation dependent upon the (7; 3;

$n = n7$;
p1 = p2 = … = p7 = n;
Grouping (p1), Grouping (p2), Grouping (p3), Grouping (p4),
Grouping (p5), Grouping (p6), Grouping (p7);
End if
End while

leaf$_{\text{number}}$= n;

C. Node Behavior and Attack Learning Using SVM Classifier

In the proposed technique, Support Vector Machine (SVM) classifier is used to detect and identify Jellyfish attack in MANETS. SVM is based on supervised learning and is highly useful for prediction in any type of dataset. Considering the concept of Intrusion Detection System, SVM is highly useful for predicting any sorts of threats and vulnerabilities.~~ In order to yield improvised outc~~omes, especially prediction, SVM based models are used. Here, we present the mathematical model of the problem:

The training dataset (D) is

Algorithm 1: Grouping algorithm


```
Input: Suppose , the total number of nodes in PSN are taken as n, p1; p2; p3; p4; p5; p6; p7 are child nodes in PSN, and TS provides assessment on the added nodes.
Output: Grouping Between i^th CWFU and j^th CSFU.
=================
Algorithm:
Initialization: $c = 0$; number of pairs formed.
While $n > 7$ do
        If $(n \% 7) = 0$ then
    $n = n = 7$;
    $p1 = p2 = … = p7 = n$;
    Grouping (p1), Grouping (p2), Grouping (p3), Grouping (p4),Grouping (p5), Grouping (p6), Grouping (p7);Else
    $n = n + (7 - n\%7)$;
    $r = 7 - n\%7$;
```

Algorithm 2 demonstrates the SVM algorithm to solve the above problem.

assessment on the added nodes.

Algorithm 2: SVM algorithm

```
---- Initialization: vector v=0, b=0; // v-vector and b-bias.
KD dataset is given by $D=(x_1,y_1), …, (x_n,y_n)$, C // C-class and
x and y – labeled samples.
Train an initial SVM and learn the model
For each $x_i$ in X do // $x_i$ is a vector containing features describing example i.
Classify $x_i$ using $f(x_i)$
If (yi f(xi) < 1) then   // prediction class label
Find w', b' for known data // w', b' for new features
Add $x_i$ to known data
Using Eq. (3) to minimize the error function and using Eq. (4) to estimate.
If(prediction is wrong) then
Endif
Retrain
Else
Repeat
Endif
Classify attributes as normalor abnormal
```

D. The APD-JFAD algorithm

Algorithm 3 shows the proposed method in details.-

Algorithm 3: SVM algorithm Input: N= {1………n}, where n is number of neighboring nodes the network, i=0;
Algorithm: For each node N, while (N[i] NULL) Node = N[i]

Calculate the trust value

If (Trust value =properties of N[i] + endorsement rovided by neighbors i.e. N – N[i])

Calculate the various trust metrics i.e. Packet forwarding, availability to hello messages, packet delay, packet integrity, precision, remaining energy, reputation.

DT Trust metrics calculated

IT indirect information via recommendation of neighbor node

OT (Overall Trust) = DT+ IT

Else

Form clusters of seven nodes

Efficient HTL Evaluation System is formed

GT + HT HTL Evaluation System

End If

Detect and Identify Jelly Fish attacks in MANETs using SVM

The systematic execution of the proposed approach is shown in Fig. 5.

V. EXPERIMENTS

Section V outlines the performance of APD-JFAD technique against Artificial Bee Colony (ABC) [28], Memetic Artificial Bee Colony (MABC) and AR-AIDF-GFRS using NS-all-in-one 2.35 in all simulation scenarios to investigate changes in varied performance metrics and efficient routing of packets from source to destination. Here, we use MANET scenarios for evaluating

In order to test the proposed technique on MANETs to determine its viability to detect Jellyfish attack, the following parameters are taken into consideration:

performance in the area of 1000m x 1000m and 100 mobile nodes. The objective here is to test whether APD-JFDA is

efficient to combat Jellyfish attack and how the technique

can improvise other performance metrics of the MANET network.

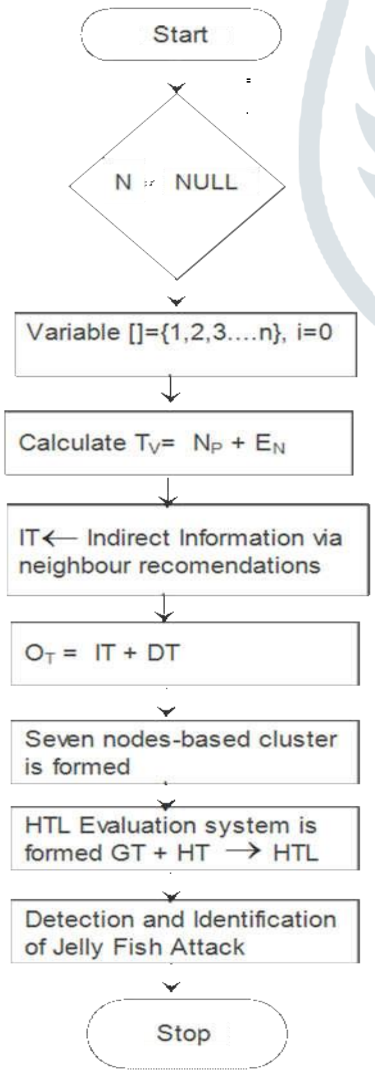

Start
N ≠ NULL
Variable []={1,2,3….n}, i=0
Calculate T_V= N_P + E_N
IT⟵ Indirect Information via neighbour recomendations
O_T = IT + DT
Seven nodes-based cluster is formed
HTL Evaluation system is formed GT + HT ⟶ HTL
Detection and Identification of Jelly Fish Attack
Stop

FIGURE 5. Execution of the proposed method

 Throughput (T): It is regarded as the ratio of total packets transmitted at particular time. It can be calculated as the difference between the packet transmission time of origin and time of receipt between source and destination node.

$$T = \frac{\sum_{i=1}^{n} N_i^r}{\sum_{i=1}^{n} N_i^s} \times 100\%$$

number of sent packets) * 100.

 Dropped Packet Ratio (DPR): Dropped Packet Ratio is regarded as the proportion of the number (5)

Packet Delivery Ratio (PDR): It can be calculatedas the total number of packets sent by source nodev/s number of packets received by the destination node.

PDR = (number of received packets / (6)

$$DPR = \sum_{i=1}^{n} (N_i^s - N_i^r) - \sum_{i=1}^{n} N_i^s$$

packets transferred by the source node, but not received by the destination node.

TABLE III
SimulationScenario

Parameter	Value
Operating System	Ubuntu 17.04
Simulator	NS-2.35-all-inone
Total No. of Nodes	100
Node Speed	1 to 5 m/s
Transmitted Packet Type	UDP
Time of Simulation	160 seconds
MAC Specification	802.11
Packet Size	100/300/500/700/80 2000 bytes
Simulation Area	1000m*1000m
Radio Type	802.11 a/g
Routing Protocol	AODV/DSR
Transport Protocol	TCP

and T3 sub-sets consist of 3 types of mixed attack data, comprising 100 SYN Flood, 100 UDP Flood and 100 ICMP Flood data correspondingly. There are 1300 data in the training set. Taking out RLT and TRA features from the training set and training SVM correspondingly, we acquire 2 × 6 SVMs, on account of utilizing 1-v-1 SVM.

$$\Delta = \frac{\sum_{i=1}^{N_{rcd}} \Delta_i}{N_{rcd}}, \tag{8}$$

where $Nrcd$ is the number of packets received by the destination node. throughput.

TABLE IV

Throughput value analysis with ABC, MABC and AR-AIDF-GFRS

Throughput (MBps / Seconds)

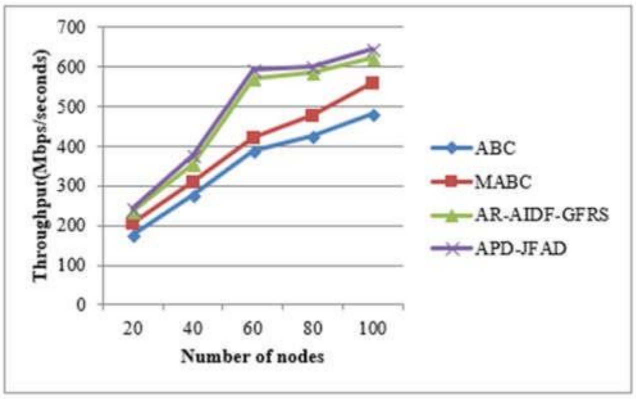


FIGURE 6. Throughput comparison of all algorithms

No. of nodes	ABC	MABC	AR- AIDF- GFRS	APD- JFAD
20	178	205	236	245
40	278	312	356	378
60	389	423	569	591
80	425	476	587	599
100	483	561	621	645
Avg.	**350.6**	**395.4**	**473.8**	**491.6**

PDR

Table V gives the comparison of PDR.

TABLE V

PacketDeliveryRatiovalueanalysis

Packet Delivery Ratio (PDR) (%)

20	75.89	82.56	89.52	92
40	76.93	83.87	90.51	93.5
60	77.52	84.79	91.45	94.6
80	77.86	85.31	91.87	95.1
100	78.52	86.52	92.53	96.8
Avg.	**77.344**	**84.61**	**91.176**	**94.4**

To determine the training outcomes, two experiments are done on two testing datasets. The first dataset consists of category labels and the second one is taken from MIT Lincoln Lab and has no category labels. The second dataset contains 4 different types of: "Normal, Light, Medium, Heavy", whilst the first dataset comprises 1200 records. Training data set comprised of four sub-sets: T0, T1, T2 and T3 that represent Normal, Light, Medium, Heavy data correspondingly. The dataset consists of 400 normal data in T0 and 300 attack data in T1, T2 and T3 respectively. T1, T2

End-to-End Delay (Δ)**:** Δ is calculated as the ratio of every packet tran

With a tabular description of data, it is analyzed that the average rate of packet delivery ratio in ABC is about 77%, 85% in MABC and 91% in AR-AIDF-GFRS which is less as compared to the APD-JFAD technique with a whooping packet delivery ration of almost 95%. The analysis demonstrates that in terms of throughput, APD-JFAD technique has better packet delivery ratio, almost 6% as compared to the AR-AIDF-GFRS, 12% as compared to MABC and 22% as compared to ABC. Fig. 7 highlights the graphical based analysis of APD-JFAD technique in terms of PDR.

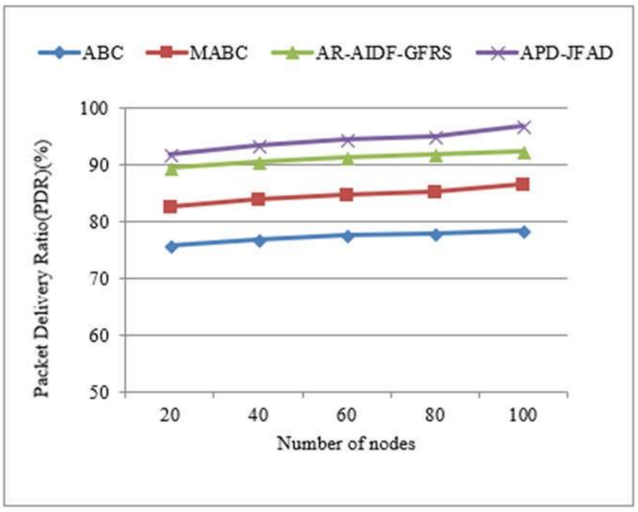
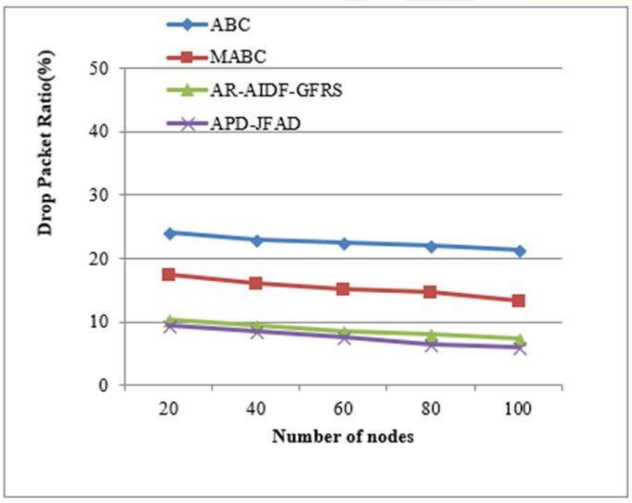


TABLE VII

End-To-EndDelaycomparison

End to End Delay (%age) No. of nodes

f nodes	ABC	MABC	AR-AIDF-GFRS	APD-JFAD
20	24.11	17.44	10.48	9.5
40	23.07	16.13	9.49	8.6
60	22.48	15.21	8.55	7.56
80	22.14	14.69	8.13	6.5
100	21.48	13.48	7.47	6.1



- *Dropped Packet Ratio*

FIGURE 8. Dropped Packet Ratio (DPR) comparison of all algorithms

It has been stated that dropped packet ratio which creates lots of packet dropping in the network is very high in the ABC algorithm which is near to about 22%, 14% in MABC and almost 8% in AR-AIDF-GFRS technique. The proposed APD-JFAD algorithm reduces the dropped packet ratio to almost 6%, which is ultimate to maintain overall network efficiency in MANET. The analysis demonstrates that APD-JFAD technique has better dropped delivery ratio. APD-JFAD outshines to almost 13% reduced packet drop rate as compared to the AR-AIDF-GFRS, 50% as compared to MABC and 66% as compared to ABC. Fig. 8 highlights the graphical based analysis of APD-JFAD technique in terms of Dropped Packet ratio.

- *Delay*

Table VII gives data analysis of End-To-End Delay values of all algorithms.

ABC MABC AR-AIDF- GFRS APD- JFAD

FIGURE 7. PDR comparison of all algorithms

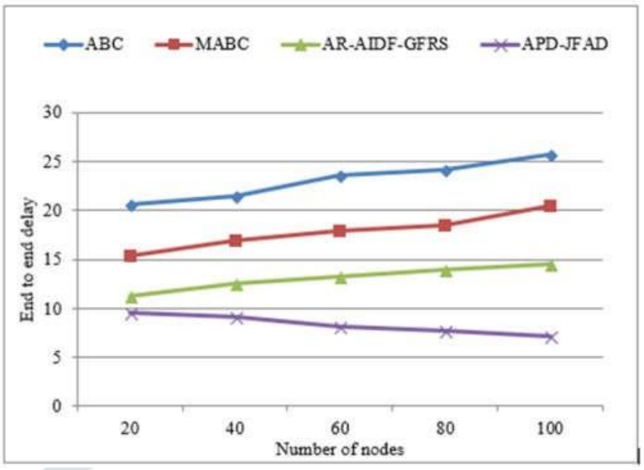
TABLE VI

DroppedPacketRatiovalueanalysis

Dropped Packet Ratio (%)

20	20.56	15.36	11.25	9.56
40	21.43	16.98	12.58	9.15
60	23.58	17.87	13.17	8.2
80	24.15	18.46	13.95	7.65
100	25.63	20.51	14.47	7.1
Avg.	**23.07**	**17.836**	**13.084**	**8.332**

FIGURE 9. End to End Delay based performance



comparison

It is clear that ABC algorithm has almost 23% delay in end-to-end delivery, 18% is with MABC, 13% with AR-AIDF-GFRS. The APD-JFAD technique has very less end-to-end delay about 8%, which means that the proposed technique is highly optimized for MANETs. It is observed that APD-JFAD outperforms other methods in terms of End-to-End delay. APD-JFAD displays 36% reduced end-to-end delay as compared to the AR-AIDF-GFRS, 53% reduced end-to-end delay as compared to MABC and 64% reduced end-to-end delay compared to ABC. Fig. 9 highlights the graphical based analysis of APD-JFAD technique in terms of End-To-End Delay.

VI. CONCLUSION

In this paper, we proposed a novel method for detecting and combating Jellyfish attack in MANET called the Accurate Prevention and Detection of Jellyfish Attack Detection (APD-JFAD). And AODV MANETs is surrounded by tons of different attacks, each with different behavior and aftermaths. The Jellyfish attack is regarded as one of the most difficult attack to detect and degrades the overall network performance. In the APD-JFAD, node property based hierarchical trust evaluation was carried out so that only trusted nodes are selected for route path construction. Support Vector Machine was used to perform packet forwarding learning. The proposed technique was validated using NS-2 simulator and compared with 3 other existing techniques i.e. ABC, MABC and AR-AIDF-GFRS algorithms by various parameters such as throughput, PDR, dropped packet ratio and delay.

Simulation results states APD-JFAD is better than AODV in terms of throughput with almost 4% of AR-AIDF-GFRS, 24% of MABC and 40% of ABC. APD-JFAD is better in terms of packet delivery ratio (almost 6% with regard to AR-AIDF- GFRS, 12 % with regard to MABC and 22% with regard to ABC). APD-JFAD is better in terms of dropped packet ratio almost to about 13% with regard to AR-AIDF-GFRS, 50% with regard to MABC and 66% with regard to ABC. It is also better in end-to-end delay (36% with regard to AR-AIDF- GFRS, 53% with regard to MABC and 64% with regard to ABC). The main conclusion is that APD-JFAD outshines ABC, MABC, AR-AIDF-GFRS and defends MANETs against jelly fish attack in precision manner.

In the future, detection accuracy will be enhanced by integrating deep learning in AI technology. In addition to that, we try to implement APD-JFAD technique on some real-time MANET scenarios using emulations.

REFERENCES

[1]. Srinath Doss, Anand Nayyar, G. Suseendran APD-JFAD: Accurate Prevention and Detection of Jelly Fish Attack in MANET

[2]. K. Naveeda, V. Nithya Poorani Defending Against Intrusion and Prevention of Jellyfish Attack Approach for Detecting Malicious Node in MANET.

[3]. G. M. Borkar, A. R. Mahajan, "A secure and trust based on- demand multipath routing scheme for self-organized mobile ad- hoc networks". *Wireless Networks*, *23*(8), 2455-2472, 2017.

[4]. Y. Wang, R. Chen, J. H. Cho, A. Swami, K. S. Chan, "Trust- based service composition and binding with multiple objective optimization in service-oriented mobile ad hoc networks," *IEEE Transactions on Services Computing*, *10*(4), 660-672, 2017.

[5]. W. K. Kuo, S. H. Chu, "Energy efficiency optimization for mobile ad hoc networks," *IEEE Access*, *4*, 928-940, 2016.