IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 14, 2018, accepted August 23, 2018, date of publication September 10, 2018, date of current version October 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2868544

APD-JFAD: Accurate Prevention and Detection of Jelly Fish Attack in MANET

SRINATH DOSS[1], ANAND NAYYAR[2], G. SUSEENDRAN[3], SUDEEP TANWAR[4], ASHISH KHANNA[5], LE HOANG SON[6], AND PHAM HUY THONG[7,8]

[1]Faculty of Computing, Botho University, Gaborone 501564, Botswana
[2]Graduate School, Duy Tan University, Da Nang 550000, Vietnam
[3]School of Computing Sciences, VELS Institute of Science, Technology & Advanced Studies, Chennai 600117, India
[4]Department of Computer Engineering, Institute of Technology, Nirma University, Ahmedabad 382481, India
[5]Department of Computer Science and Engineering, Maharaja Agrasen Institute of Technology, New Delhi 110086, India
[6]VNU Information Technology Institute, Vietnam National University, Hanoi 010000, Vietnam
[7]Division of Data Science, Ton Duc Thang University, Ho Chi Minh 700000, Vietnam
[8]Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh 700000, Vietnam

Corresponding author: Pham Huy Thong (phamhuythong@tdt.edu.vn)

ABSTRACT Mobile ad hoc networks (MANETs) are 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 accurate prevention and detection of jelly fish attack detection (APD-JFAD). It is a fusion of authenticated routing-based framework for detecting attacks and support vector machine (SVM). 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, packet forwarding behavior, support vector machine, ABC.

I. INTRODUCTION

In the past few years, significant advancements were observed towards availability of wireless networks in a number of handheld devices like portable computers, smart phones, Internet-of-Things (IoT) based wearable technologies [1]. The most common example of wireless communications is the availability of Wi-Fi Access points in bus stops, railway stations, hotels, cafes and even small shops in which people use these points to surf the Internet [2]. Wireless devices connect to gateways to access the Internet via infrastructure-based wireless network without any sort of relaying called Adhoc Network [3]. Mobile Adhoc Networks (MANETs) is regarded as a systematic organization of communication devices willing to communicate with each other for sharing information without any fixed infrastructure [4], [5]–[9]. MANET nodes are highly responsible for dynamic discovery of neighboring nodes to form a dynamic network for transferring packets from sources to destinations [10]. In MANETs, all the mobile nodes operate in self-organized manner connected in wireless manner and making random topology in dynamic manner. [11]. The nodes have the freedom to roam randomly in the network; organize with other nodes in an arbitrary manner, and in turn the topology of MANET network changes in random fashion and highly unpredictable manner [11]. Adhoc nodes in MANET should have the ability to detect other node's presence to allow break-free communication and sharing of information [12]. Along with that, the network should identify different types of services and other communication attributes [12]. Since during operation the number of wireless nodes also changes in real-time, routing information is indeed changed and topology change in MANET is more frequent than in wired networks [12].

Various parameters (e.g. topology change, dynamic mobility, unpredictable link changes, limited energy of nodes and security) raise challenges in normal operational scenarios of

MANETs [13]. Considering the above parameters, it is highly tricky and quite daunting task to propose an **efficient routing protocol for MANETs** for stable and break free operation. However, researchers have proposed certain protocols in terms of improvement in delay, latency, throughput, but other parameters such as **energy, reliability and mobility** were neglected [14]. In MANET, every node has wireless interface and communicates with other neighboring nodes for packet transmission via Radio Frequency [10]. Nodes in MANETS are mostly mobile, but some could be fixed like Wireless Access Points [10]. There may be semi-Mobile nodes which are regarded as relay nodes for transmitting information to remote nodes but work temporarily [10]. It operate without any requirement of centralized administration, which in turn makes MANETs a non-collapse network as some nodes at certain conditions can move out of transmitting range and new nodes can enter or leave the network as per their operational desire [15]. As the mobile nodes have limited transmitting range, multi-hops are required to traverse all nodes in the network [16]. Every node operating in MANET should be willing to perform packet forwarding so that packets reach the destination without any sort of hiccup [17].

MANET has an ability to intelligently handle all sorts of topological changes as well as node malfunctioning issues via network re-configuration technique [17]. 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 [16]. 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 [14].

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 [14]. 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 [2]. 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 [18]. 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 [19], [20].

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.

Complexity of the Problem: Finding the solution to Jellyfish attack in MANETs in entirely complex and jellyfish attack impacts the overall throughput, packet delivery ratio and connectivity among the sensor nodes. Therefore, to propose a solution, Machine Learning based technique comes to rescue. In this paper, we applied SVM based technique to detect the malicious behavior of nodes by observing the quality of packets reached at the destination. Furthermore, it is a real practical solution to observe all the complex behaviors and algorithm learns and become efficient at regular intervals and will be able to detect the jelly fish attack efficiently.

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:

- Tremendous increase in the number of the junk packets, in turn, preventing the trustworthy nodes to transmit data packets in the network.
- Generation of fake control packets carrying incorrect topology information and impacting routing table.

TABLE 1. Overview of various sorts of attacks in MANETs.

Network Layer	Attack Type
Application Layer	Repudiation, Malicious Code Injection in Nodes
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 Compromised Attack

- Delay in packet transmission and impacting overall throughput in the network.

Table 1 gives an overview of various sorts of attacks in MANETs along with the layer details [13]–[17].

Attacks regarding Blackhole, Sybil, and Wormhole impact normal working of the routing protocol by adding fake information, altering information and dropping information in control packets during the process of discovery of routing paths [16], [17]. These attacks are highly easy to detect because malicious node does not use any sort of protocol directions. On the other hand, attacks like Grayhole, Jellyfish, Rushing Attack and Malicious Infection is harder to detect as they follow all sorts of protocol rules and impact normal network functioning [14]. To deploy mechanisms to detect and combat these attacks is highly challenging and tricky task.

Kaur *et al.* [21] 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. Bhawsar and Suryawanshi [22] 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.

Sharma and Kaur [23] 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. Soni and Uikey [24] 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 [25] proposed ACO-CBRP (Ant Colony Optimization based 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.* [26] proposed a secure 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 [27] proposed DSMANET to detect malicious nodes and improvised the overall throughput and routing overhead.

Kalucha and Goyal [28] 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.* [29] 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.

Barani and Barani [30] 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 [31] 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. HIASA algorithm examines 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 [24], [32], [33].

A. OVERVIEW OF JELLYFISH ATTACK

Jellyfish attack comes under the classification of passive attack and is regarded as a type of Denial of Service (DoS) attack [25]. It maintains complete compliance with control and data protocols for making detection and prevention highly challenging tasks to work upon [25]. Jellyfish attack introduces delay in network before any sort of transmission and receipt of packets happen between the communicating nodes [34]. 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 [34]. 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 [34]. Jellyfish attack is primarily targeted towards closed loop flows with the ultimate goal to disrupt normal operation of the network by packet dropping [35]. Jellyfish attack is highly vulnerable in TCP traffic in which cooperative nodes can hardly distinguish between attacks from network congestion as shown in Fig. 1.

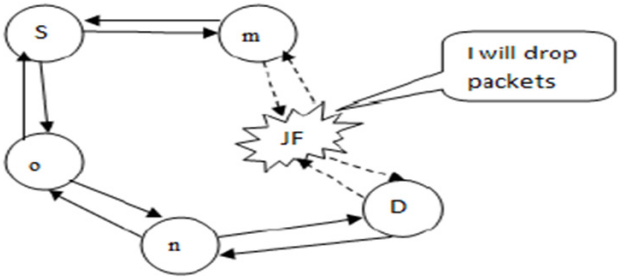

FIGURE 1. Jellyfish attack in MANETS.

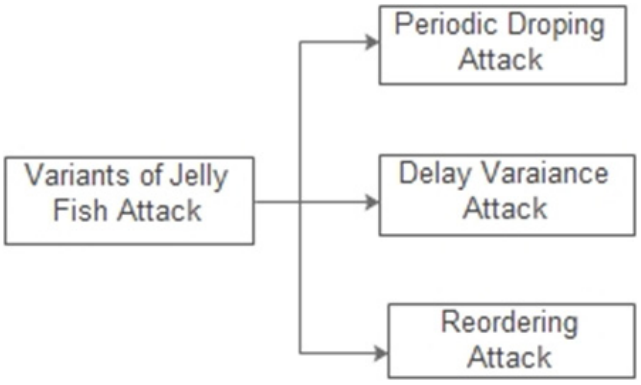

FIGURE 2. Jellyfish attack types.

B. JELLYFISH ATTACK VARIANTS

Fig. 2 highlights the variants of Jellyfish attack: Jellyfish Reordering attack, Jellyfish periodic dropping attack and Jellyfish Delay Variance attack [32], [36], [37].

- **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

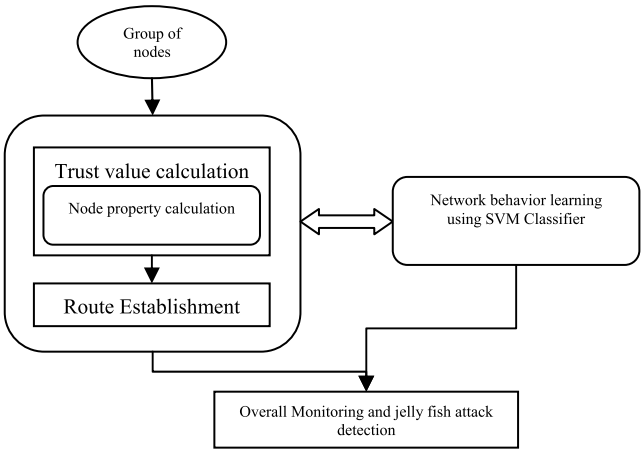

FIGURE 3. Mechanism of APD-JFAD for combating Jellyfish attack in MANETs.

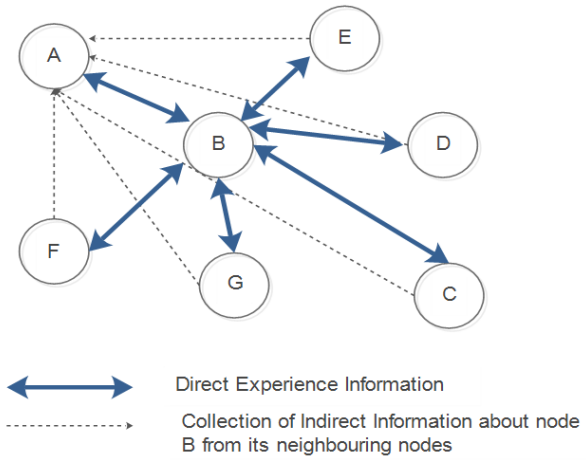

FIGURE 4. Node A accesses the trust of node B.

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 the packet delivery at random intervals without changing the packet order. This in turn can impact the network via congestion.

IV. PROPOSED METHOD

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.

A. NODE PROPERTY-BASED HIERARCHICAL TRUST EVALUATION

1) 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.

TABLE 2. Metrics for trust calculation battery lifetime/energy.

Delay	
Packets transmitted	
Reputation	
Sensing Communication	
Reputation	Packet
Precision/Integrity	
Honesty	
Intimacy	
Unselfishness	

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

2) METRICS FOR COMPUTING TRUST IN MANETs

Table 2 enlists various metrics used for Trust Calculation of node in a MANET environment [27].

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; 1) design. Here, I is a group of 7 nodes. Every node in the set I contain a better recognition of the remaining nodes.

Algorithm 1 is a recursive method to build tree structure dependent upon any number of nodes in PSN. In this algorithm, the number of nodes in the PSN is denoted as n, and p1; p2; p3; p4; p5; p6; p7 are the numbers of PSN nodes encompassed in the child nodes. In order to make sure that the (7; 3; 1) design is utilized in multiple nodes in PSN, every recursion must guarantee that the number of nodes is accurately divisible by 7. If not, certain nodes are added by the TS in order to come across the constraint. TS provides the assessment on the added nodes.

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. With the technique of SVM, the nodes cannot change the behavior and if any change comes in the behavior, it is immediately notified and that node leaves the routing path. In order to yield improvised outcomes, especially prediction, SVM based models are used. Here, we present the mathematical model of the problem:

Algorithm 1 Grouping Algorithm

Input: Suppose , the total number of nodes in PSN are taken as n, $p_1; p_2; p_3; p_4; p_5; p_6; p_7$ 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**
    $n = n = 7$;
    $p_1 = p_2 = \ldots = p_7 = n$;
    Grouping (p_1), Grouping (p_2), Grouping (p_3), Grouping (p_4),
    Grouping (p_5), Grouping (p_6), Grouping (p_7);
  Else
    $n = n + (7 - n\%7)$;
    $r = 7 - n\%7$;
    $n = n7$;
    $p_1 = p_2 = \ldots = p_7 = n$;
    Grouping (p_1), Grouping (p_2), Grouping (p_3), Grouping (p_4),
    Grouping (p_5), Grouping (p_6), Grouping (p_7);
  End if
End while
$leaf_{number} = n$;

The training dataset (D) is:

$$D = \{x_i, y_i\}_{i=1}^{N}, \quad x \in R^n, \ y \in \{-1, 1\} \tag{1}$$

Here, x and y is input variables and satisfy

$$y^i \left[\left| w^T x^i + b \right| \right] \geq 1 \quad i = 1 \text{ to } N. \tag{2}$$

w^T and b are identified as separated variables. In order to reduce errors, the following formula is utilized:

$$\Phi(w) = \frac{1}{2} \|w\|^2. \tag{3}$$

The objective is to estimate the below function:

$$F(x) = \sum_{i=1}^{nsv} (x_i, y_i) k(x_i, y_i) + b \tag{4}$$

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

D. THE APD-JFAD ALGORITHM

Algorithm 3 shows the proposed method in details.

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

Algorithm 2 SVM Algorithm

Initialization: vector $v = 0$, $b = 0$; // v-vector and b-bias. KD dataset is given by $D = (x_1, y_1), \ldots, (x_n, y_n)$, 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 (y_i $f(x_i) < 1$) **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**
Retrain
Else
Repeat
Endif
Endif
Classify attributes as normal or abnormal

Algorithm 3 SVM Algorithm

Input: N = {1 . . . . . n}, where n is number of neighboring nodes in 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 provided 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

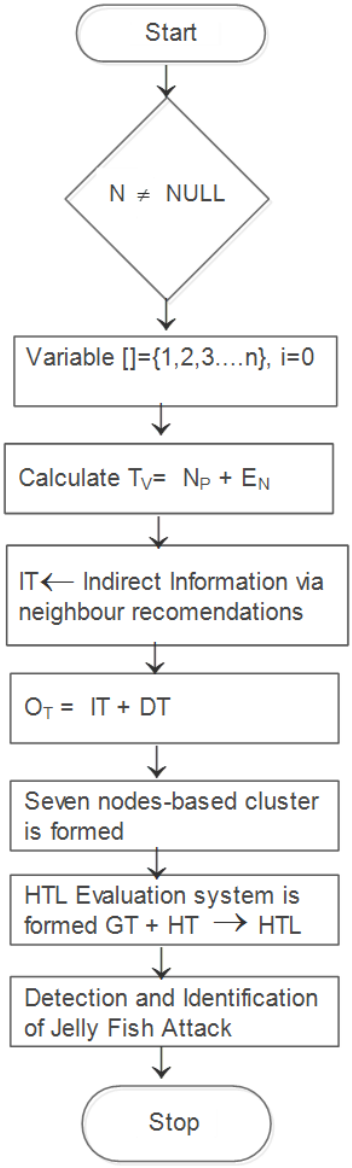

FIGURE 5. Execution of the proposed method.

Bee Colony (MABC) [30] and AR-AIDF-GFRS [38] 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 performance in the area of 1000m x 1000m (Low density Network) 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. The detailed simulation scenario is highlighted in Table 3.

Limitations: In terms of implementation, the simulation model has some limitations in terms of: (1) All the sensor nodes operating in the network have the same amount of energy level; (2) Random speed of packet transmission among mobile nodes; (3) All the nodes transceivers use wireless antenna and omnidirectional in nature and propagate isotropic signals in all directions.

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

- **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

TABLE 3. Simulation scenario.

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	100/300/500/700/800/1000/1500/ 2000 bytes
Simulation Area	1000m*1000m
Radio Type	802.11 a/g
Routing Protocol	AODV/DSR
Transport Protocol	TCP

time of origin and time of receipt between source and destination node.

$$T = \frac{\sum_{i=1}^{n} N_i^r}{\sum_{i=1}^{n} N_i^s} \times 100\% \quad (5)$$

- **Packet Delivery Ratio (PDR):** It can be calculated as the total number of packets sent by source node v/s number of packets received by the destination node.

$$\text{PDR} = (\text{number of received packets/ number of sent packets}) * 100. \quad (6)$$

- **Dropped Packet Ratio (DPR)**: Dropped Packet Ratio is regarded as the proportion of the number of packets transferred by the source node, but not received by the destination node.

$$DRP = \sum_{i=1}^{n} (N_t^s - N_t^r) - \sum_{t=1}^{n} N_i^s \quad (7)$$

- **End-to-End Delay (Δ):** Δ is calculated as the ratio of every packet transmitted from source node to the number of data packets received at the destination node. This metric is highly important to evaluate the Jellyfish attack impact on TCP-based MANET. It is calculated using the following formulae:

$$\Delta = \frac{\sum_{i=1}^{N_{rcd}} \Delta_i}{N_{rcd}} \quad (8)$$

where N_{rcd} is the number of packets received by the destination node.

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 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.

TABLE 4. Throughput value analysis with ABC, MABC and AR-AIDF-GFRS.

No. of nodes	Throughput (MBps / Seconds)			
	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**

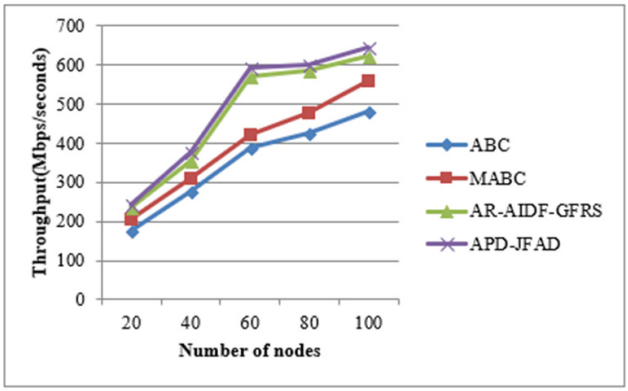

FIGURE 6. Throughput comparison of all algorithms.

A. THROUGHPUT

Table 4 shows data analysis of throughput values compared with ABC, MABC and AR-AIDF-GFRS techniques.

The results show that average throughput of ABC is 350.6, MABC is 395.4, AR-AIDF-GFRS is 473.8 which is extremely less as compared to APD-JFAD (∼492). Thus, APD-JFAD is the best technique for detecting the Jellyfish attack and maintains best throughput in the network. The analysis also state that throughput performance of APD-JFAD is accountable to almost 4% better as compared to the AR-AIDF-GFRS, 24% better as compared to MABC and 40% better as compared to ABC. Fig. 6 highlights the graphical based analysis of APD-JFAD technique in terms of throughput.

B. PDR

Table 5 gives the comparison of PDR.

With a tabular description of data, it is analyzed that the average rate of packet delivery ratio in ABC is about 77%,

TABLE 5. Packet delivery ratio value analysis.

No. of nodes	Packet Delivery Ratio (PDR) (%)			
	ABC	MABC	AR-AIDF-GFRS	APD-JFAD
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**

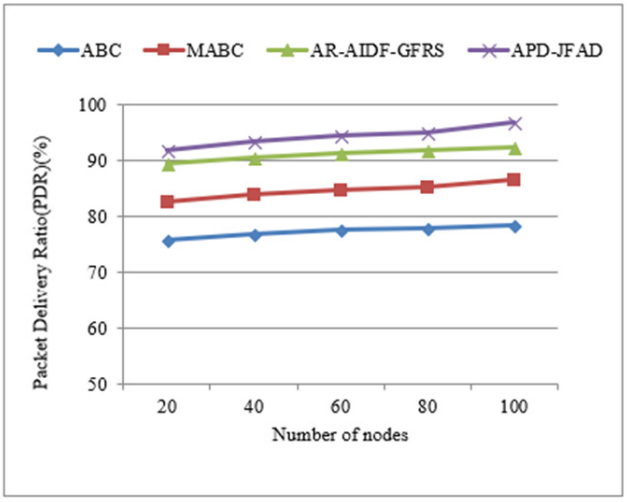

FIGURE 7. PDR comparison of all algorithms.

TABLE 6. Dropped packet ratio value analysis.

No. of nodes	Dropped Packet Ratio (%)			
	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

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.

C. DROPPED PACKET RATIO

Table 6 analyzes Dropped Packet Ratio of all algorithms.

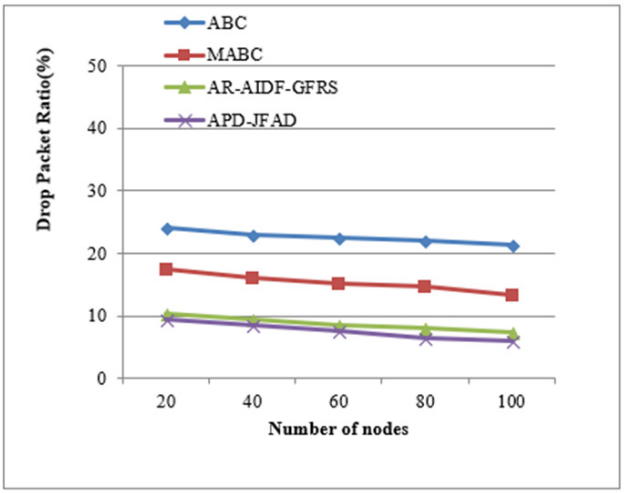

FIGURE 8. Dropped packet ratio (DPR) comparison of all algorithms.

TABLE 7. End-to-end delay comparison.

No. of nodes	End to End Delay (%age)			
	ABC	MABC	AR-AIDF-GFRS	APD-JFAD
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**

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.

D. DELAY

Table 7 gives data analysis of End-To-End Delay values of all algorithms. 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

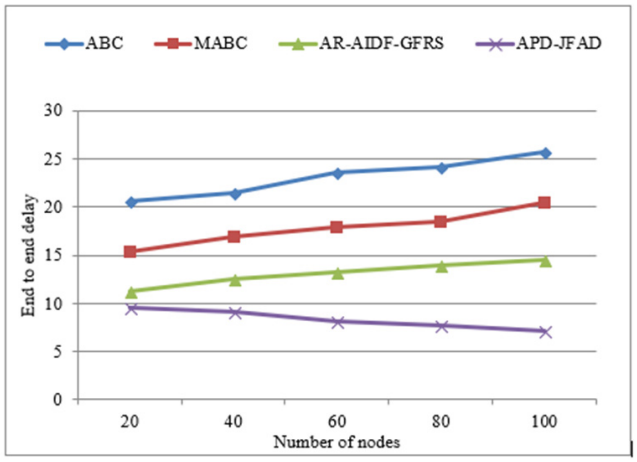

FIGURE 9. End to end delay based performance comparison.

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). 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 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 technology. In addition to that, we try to implement APD-JFAD technique on some real-time MANET scenarios using emulations.

REFERENCES

[1] X. Liu, Z. Li, P. Yang, and Y. Dong, "Information-centric mobile ad hoc networks and content routing: A survey," *Ad Hoc Netw.*, vol. 58, pp. 255–268, Apr. 2017.

[2] D. Ahmed and O. Khalifa, "An overview of MANETs: Applications, characteristics, challenges and recent issues," *Int. J. Eng. Adv. Technol.*, vol. 6, no. 4, p. 128, Apr. 2017.

[3] G. M. Borkar and A. R. Mahajan, "A secure and trust based on-demand multipath routing scheme for self-organized mobile ad-hoc networks," *Wireless Netw.*, vol. 23, no. 8, pp. 2455–2472, 2017.

[4] Y. Wang, I.-R. Chen, J.-H. Cho, A. Swami, and K. S. Chan, "Trust-based service composition and binding with multiple objective optimization in service-oriented mobile ad hoc networks," *IEEE Trans. Services Comput.*, vol. 10, no. 4, pp. 660–672, Jul./Aug. 2017.

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

[6] A. M. E. Ejmaa, S. Subramaniam, Z. A. Zukarnain, and Z. M. Hanapi, "Neighbor-based dynamic connectivity factor routing protocol for mobile ad hoc network," *IEEE Access*, vol. 4, pp. 8053–8064, 2016.

[7] A. Mehmood, A. Khanan, A. H. H. Mohamed, S. Mahfooz, H. Song, and S. Abdullah, "ANTSC: An intelligent Naïve Bayesian probabilistic estimation practice for traffic flow to form stable clustering in VANET," *IEEE Access*, vol. 6, pp. 4452–4461, 2018.

[8] A. Taha, R. Alsaqour, M. Uddin, M. Abdelhaq, and T. Saba, "Energy efficient multipath routing protocol for mobile ad-hoc network using the fitness function," *IEEE Access*, vol. 5, pp. 10369–10381, 2017.

[9] Y. Liu, J. E. Fieldsend, and G. Min, "A framework of fog computing: Architecture, challenges, and optimization," *IEEE Access*, vol. 5, pp. 25445–25454, 2017.

[10] R. H. Jhaveri and N. M. Patel, "Attack-pattern discovery based enhanced trust model for secure routing in mobile ad-hoc networks," *Int. J. Commun. Syst.*, vol. 30, no. 7, p. e3148, 2017.

[11] F. A. Khan, M. Imran, H. Abbas, and M. H. Durad, "A detection and prevention system against collaborative attacks in mobile ad hoc networks," *Future Gener. Comput. Syst.*, vol. 68, pp. 416–427, Mar. 2017.

[12] D. Hurley-Smith, J. Wetherall, and A. Adekunle, "SUPERMAN: Security using pre-existing routing for mobile ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 10, pp. 2927–2940, Oct. 2017.

[13] A. Aburumman, W. J. Seo, C. Esposito, A. Castiglione, R. Islam, and K.-K. R. Choo, "A secure and resilient cross-domain SIP solution for MANETs using dynamic clustering and joint spatial and temporal redundancy," *Concurrency Comput., Pract. Exper.*, vol. 29, no. 23, p. e3978, 2017.

[14] B. Pati, B. K. Pattanayak, and J. Swain, "A systematic study and analysis of security issues in mobile ad-hoc networks," *Int. J. Inf. Secur. Privacy*, vol. 12, no. 2, pp. 38–45, 2018.

[15] M. Wang and Z. Yan, "A survey on security in D2D communications," *Mobile Netw. Appl.*, vol. 22, no. 2, pp. 195–208, 2017.

[16] M. S. Pathan, N. Zhu, J. He, Z. A. Zardari, M. Q. Memon, and M. I. Hussain, "An efficient trust-based scheme for secure and quality of service routing in MANETs," *Future Internet*, vol. 10, no. 2, p. 16, 2018.

[17] T. Singh, J. Singh, and S. Sharma, "Energy efficient secured routing protocol for MANETs," *Wireless Netw.*, vol. 23, no. 4, pp. 1001–1009, 2017.

[18] M. Usman, M. A. Jan, X. He, and P. Nanda, "QASEC: A secured data communication scheme for mobile Ad-hoc networks," *Future Gener. Comput. Syst.*, May 2018, doi: 10.1016/j.future.2018.05.007.

[19] R. Hasan, M. Hossain, and R. Khan, "Aura: An incentive-driven ad-hoc IoT cloud framework for proximal mobile computation offloading," *Future Gener. Comput. Syst.*, vol. 86, pp. 821–835, Sep. 2018.

[20] P. Vijayakumar, V. Chang, L. J. Deborah, and B. S. R. Kshatriya, "Key management and key distribution for secure group communication in mobile and cloud network," *Future Gener. Comput. Syst.*, vol. 84, pp. 123–125, Jul. 2018.

[21] M. Kaur, M. Rani, and A. Nayyar, "A novel defense mechanism via genetic algorithm for counterfeiting and combating jelly fish attack in mobile ad-hoc networks," in *Proc. 5th Int. Conf.-Confluence Next Gener. Inf. Technol. Summit (Confluence)*, Sep. 2014, pp. 359–364.

[22] D. Bhawsar and A. Suryavanshi, "Collaborative intrusion detection and prevention against jellyfish attack in MANET," *Int. J. Comput. Appl.*, vol. 129, no. 13, pp. 37–42, 2015.
[23] A. Sharma and R. Kaur, "Non-cryptographic detection approach and countermeasure for JFDV Attack," in *Proc. ACM 7th Int. Conf. Secur. Inf. Netw.*, 2014, p. 367.
[24] J. Soni and K. Uikey, "Mitigation of jellyfish attack over AODV and ODMRP routing protocol in MANET," *Int. J. Innov. Advancement Comput. Sci.*, vol. 6, no. 8, pp. 158–161, 2017.
[25] S. Satheeshkumar and N. Sengottaiyan, "Defending against jellyfish attacks using cluster based routing protocol for secured data transmission in MANET," in *Cluster Computing*, 2017, pp. 1–12.
[26] A. Thomas, V. K. Sharma, and G. Singhal, "Secure link establishment method to prevent jelly fish attack in MANET," in *Proc. Int. Conf. Comput. Intell. Commun. Netw. (CICN)*, Dec. 2015, pp. 1153–1158.
[27] K. P. Kumar and B. R. P. Babu, "DSMANET: Defensive strategy of routing using game theory approach for mobile adhoc network," in *Proc. Comput. Sci. On-Line Conf.* Cham, Switzerland: Springer, 2018, pp. 311–320.
[28] R. Kalucha and D. Goyal, "A review on artificial bee colony in MANET," *Int. J. Comput. Sci. Mobile Comput.*, vol. 3, no. 7, pp. 34–40, 2014.
[29] M. Sailaja, R. K. Kumar, P. S. R. Murty, and P. K. Prasad, "A study on routing algorithms derived from the nature for MANETs," *Int. Mag. Adv. Comput. Sci. Telecommun.*, vol. 2, no. 1, pp. 31–37, 2011.
[30] F. Barani and A. Barani, "Dynamic intrusion detection in AODV-based MANETs using memetic artificial bee colony algorithm," in *Proc. 22nd Iranian Conf. Elect. Eng. (ICEE)*, May 2014, pp. 1040–1046.
[31] P. V. S. S. Prasad and S. K. M. Rao, "HIASA: Hybrid improved artificial bee colony and simulated annealing based attack detection algorithm in mobile ad-hoc networks (MANETs)," *Bonfring Int. J. Ind. Eng. Manage. Sci.*, vol. 7, no. 2, pp. 1–12, 2017.
[32] V. Laxmi, C. Lal, M. S. Gaur, and D. Mehta, "JellyFish attack: Analysis, detection and countermeasure in TCP-based MANET," *J. Inf. Secur. Appl.*, vol. 22, pp. 99–112, Jun. 2015.
[33] S. Kaur and R. Singh, "Review on jelly fish detection and prevention schemes in MANETS," *Int. J. Advance Res., Ideas Innov. Technol.*, vol. 3, no. 2, pp. 1192–1194, 2017.
[34] E. Mustikawati, D. Perdana, and R. M. Negara, "Network security analysis in vanet against black hole and jellyfish attack with intrusion detection system algorithm," *CommIT (Commun. Inf. Technol.) J.*, vol. 11, no. 2, pp. 77–83, 2017.
[35] B. P. Pooja, M. P. Manish, and B. P. Megha, "Jellyfish attack detection and prevention in MANET," in *Proc. 3rd Int. Conf. Sens., Signal Process. Secur. (ICSSS)*, May 2017, pp. 54–60.
[36] S. Korde and M. V. Sarode, "Review on network layer attacks detection and prevention techniques in mobile ad hoc networks," in *Proc. Int. Conf. Inventive Syst. Control (ICISC)*, Jan. 2017, pp. 1–5.
[37] S. K. Chowdhury and M. Sen, "Attacks and mitigation techniques on mobile ad hoc network—A survey," in *Proc. Int. Conf. Trends Electron. Inform. (ICEI)*, May 2017, pp. 11–18.
[38] G. Suseendran, E. Chandrasekaran, and A. Nayyar, "Defending jellyfish attack in mobile ad hoc networks via novel fuzzy system rule," in *Data Management, Analytics and Innovation*. Singapore: Springer, 2019, pp. 437–455.

SRINATH DOSS received the B.Tech. degree in information technology from the University of Madras in 2004, the M.Eng. degree in computer science and engineering from Anna University in 2006, and the Ph.D. degree in computer science and engineering from St. Peter's University in 2014. He was with various colleges in India, and also with Garyounis University, Libya. He is currently the Head of the Department with the Faculty of Computing, Botho University, Gaborone, Botswana. He has published and presented over 25 papers in international journals and a good number of prestigious conferences. His research interests include MANET, information security, network security and cryptography, cloud computing, wireless and sensor network, and mobile computing. He serves as the editorial member, a reviewer for reputed international journals, and an Advisor for the Research Scientific Publishing Development Organization. He received the Best Teacher Award during his service at the Panimalar Institute of Technology. He has been the session chair and an advisory member for various international conferences.

ANAND NAYYAR received the MCA degree (with Gold Medal and University Topper) from Punjabi University, and the Ph.D. degree in computer science in wireless sensor networks, swarm intelligence, and simulation from Desh Bhagat University, Mandi Gobindgarh, in 2017. He is currently with the Graduate School, Duy Tan University, Vietnam. He has published over 250 research papers and over 18 books on different areas of computer science. He has been associated with over 150 journals as an Editorial Board Member. He is involved in the areas of wireless sensor networks, cloud computing, network security, mobile ad hoc networks, machine learning, deep learning, algorithm design, big data, and wireless communications. He has been a recipient of over 25 awards from the National and International Societies for Research and Teaching.

G. SUSEENDRAN received the M.Sc. degree in information technology, the M.Phil. degree from Annamalai University, India, and the Ph.D. degree in information technology-mathematics from the Presidency College, University of Madras, India. His additional qualifications include DOEACC "O" Level AICTE Ministry of Information Technology and the Diploma (Hons.) in computer programming. He is currently an Assistant Professor with the Department of Information Technology, School of Computing Sciences, Vels University, Chennai, India, which is a well-known University. He has years of teaching experience in both UG and PG Level. His research interests include ad hoc networks, data mining, cloud computing, image processing, knowledge-based systems, and web information exploration.

SUDEEP TANWAR received the B.Tech. degree from Kurukshetra University, Kurukshetra, in 2002, the M.Tech. degree (Hons.) from Guru Gobind Singh Indraprastha University, New Delhi, India, in 2009, and the Ph.D. degree from Mewar University, Chittorgarh, India, in 2016, with the specialization in wireless sensor network. He is currently an Associate Professor in computer engineering with the Department at the Institute of Technology, Nirma University, Ahmedabad, India. He has authored over 50 technical research papers published in leading international conferences and peer-reviewed international journals from the IEEE, Elsevier, Springer, and John Wiley. Some of his research findings are published in top-cited journals, such as the *Journal of Network and Computer Application*, the *Pervasive and Mobile Computing* (Elsevier), the *International Journal of Communication System* (Wiley), the *Telecommunication System* (Springer), and the IEEE Systems Journal. He has also published three books on Role of Heterogeneity in WSN, Big Data Analytics, and Mobile Computing with International/National Publishers for the students of UG/PG. He has guided many students leading to M.E./M.Tech. and Ph.D. degrees. His current interest includes routings issues in WSN, integration of sensor with the cloud, computational aspects of smart grid, and assessment of fog computing in BASN. He is an Associate Editor of *Security and Privacy Journal* (Wiley).

ASHISH KHANNA received the B.Tech. degree from GGSIPU, New Delhi, India, in 2004, the M.Tech. degree in 2009, and the Ph.D. degree from the National Institute of Technology, Kurukshetra, in 2017. He is currently with the Maharaja Agrasen Institute of Technology, New Delhi. He has around 15 years of expertise in Teaching, Entrepreneurship, and Research & Development with the specialization in computer science engineering subjects. He has published around 30 research papers in reputed SCI, Scopus Journals, and conferences. He has papers in Springer, Elsevier, and IEEE journals. He has co-authored 10 text books of engineering courses like Distributed Systems and Java Programming. His research interest includes distributed systems and its variants like MANET, FANET, VANET, and IoT. He is a Post-Doctoral Fellow at the Internet of Things Lab, Inatel, Brazil. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Maharaja Agrasen Institute of Technology, under GGSIPU, New Delhi. He is a Convener and Organizer of the ICICC-2018 Springer Conference. He played a key role in the origination of a reputed publishing house ''Bhavya Books'' having 250 solution books and around 60 text books. He is also a part of the research unit under the banner of ''Universal Innovator.'' He also serves as the Guest Editor in various reputed International journals of Inderscience, IGI Global, and Bentham Science.

LE HOANG SON received the Ph.D. degree in mathematics–informatics from the VNU University of Science, Vietnam National University (VNU). Since 2017, he has been promoted as an Associate Professor in information technology. He is currently a Researcher and also the Vice Director of the Center for High Performance Computing, VNU University of Science, VNU. His major field includes artificial intelligence, data mining, soft computing, fuzzy computing, fuzzy recommender systems, and geographic information system.

Dr. Son is a member of the International Association of Computer Science and Information Technology, the Vietnam Society for Applications of Mathematics (Vietsam), and the Key Laboratory of Geotechnical Engineering and Artificial Intelligence, University of Transport Technology, Vietnam. He serves as an Editorial Board of the *International Journal of Ambient Computing and Intelligence*, in SCOPUS, and the *Vietnam Journal of Computer Science and Cybernetics* (JCC). He is an Associate Editor of the *Journal of Intelligent & Fuzzy Systems* in SCIE, the *Neutrosophic Sets and Systems* (NSS), the Vietnam Research and Development on Information and Communication Technology (RD-ICT), and the *VNU Journal of Science: Computer Science and Communication Engineering* (JCSCE).

PHAM HUY THONG is currently pursuing the Ph.D. degree with the Center for High Performance Computing, Hanoi University of Science, Vietnam National University (VNU). He is a Researcher at the Center for High Performance Computing, Hanoi University of Science, VNU. His research interests include data mining, soft computing, fuzzy computing, geographic information systems, and molecular dynamics simulation.