Multivariate Fuzzy Analysis for Mobile Ad hoc Network Threat Detection

Sathish Kumar Alampalayam, University of Louisville, USA
Essam F. Natsheh, King Faisal University, Saudi Arabia

ABSTRACT

Detection of malicious, compromised and selfish node attacks is urgently needed to protect Mobile Ad hoc Networks (MANET) and their applications from failures. Once a MANET component is affected by a malicious, compromised or selfish node its operational state shifts from normal state to vulnerable state. Online monitoring mechanisms can collect important MANET network data that can be effectively used to detect abnormal behaviors caused by attacks. In this article we develop an online multivariate analysis algorithm to analyze the behavior of network resources and protocols to proactively detect MANET attacks. This algorithm is based on multivariate fuzzy analysis technique. We have validated the algorithm and demonstrated how it can effectively detect well-known attacks such as Denial of Service in MANET.

Keywords: denial of service attacks; detection rate; intrusion detection; MANET; multivariate fuzzy analysis; processing time; threat index

INTRODUCTION

A MANET is a collection of wireless mobile nodes forming a temporary network without any established infrastructure or centralized authority. In a MANET, the nodes are free to move around and organize themselves into a network. MANET does not require any fixed infrastructure such as base stations; therefore, it is an attractive networking option for connecting mobile devices quickly and spontaneously. For instance, MANET can be used by first responders at a disaster site or soldiers...
in a battlefield to provide their own communications.

Until recently, the main research focus has been on improving the protocols for multihop routing, performance and scalability of the ad hoc networks. Although the performance and scalability have their place in wireless MANET research, the current and future applications of the ad hoc networks have forced the research community to look at dependability and security aspects of the ad hoc networks (Alampalayam, Kumar, & Srinivasan, 2005). Security in the mobile ad hoc network is essential even for basic network functions like routing, which are carried out by the nodes themselves rather than specialized routers. The intruder or malicious node in the ad hoc network can come from anywhere, along any direction, and target any communication channel in the network. Threat prevention techniques such as authentication and encryption are applicable in the wired and infrastructure-based cellular network. In the case of infrastructure-free mobile ad hoc networks these techniques are not applicable (Sun, Wu, & Pooch, 2006). The dynamic nature of the ad hoc network also means that trust between nodes in the network is virtually nonexistent. Without trust, preventive measures are unproductive and measures that rely on a certain level of trust between nodes are susceptible to attacks themselves. Hence, there is a need for intrusion detection because it provides a second line of defense. Intrusion detection and response is the process of detecting and responding to malicious activity that is aimed at attacking the network (Hariri, Qu, Modukuri, Chen, & Yousif, 2005).

There are basically two types of existing Intrusion Detection Approaches (IDA): anomaly detection and misuse detection. Anomaly detection approach analyzes the user’s current session and compares them to the profile representing the user’s normal behavior. As it catches sessions which are not normal, this model is referred to as an “anomaly” detection model. Anomaly detection bases its idea on statistical behavior modeling and anomaly detectors look for behavior that deviates from normal system use. A typical anomaly detection system takes in audit data for analysis. The audit data is transformed to a format statistically comparable to the profile of a user. The user’s profile is generated dynamically by the system (usually using a baseline rule laid by the system administrator) initially and subsequently updated based on the user’s usage. Thresholds are normally always associated to all the profiles. If any comparison between the audit data and the user’s profile resulted in deviation crossing a threshold set, an alarm of intrusion is declared. This type of detection system is well suited to detect unknown or previously not encountered attacks. Our methodology is based on the anomaly detection approach.

Misuse detection approach bases its detection upon a comparison of parameters of the user’s session and the user’s commands to a rule base of techniques.
used by attackers to penetrate a system. Known attack methods are what this model looks for in a user’s behavior. Because this model looks for patterns known to cause security problems, it is called a “misuse” detection model. This type of detection is also known as rule-based approach because it bases its idea on precedence and rules, and misuse detectors look for behavior that matches a known attack scenario. A typical misuse detection system takes in audit data for analysis and compares the data to large databases of attack signatures. The attack signatures are normally specified as rules with respect to timing information and are also referred to as known attack patterns. If any comparison between the audit data and the known attack patterns described resulted in a match, an alarm of intrusion is sounded. This type of detection systems is useful in networks with highly dynamic behavioral patterns but like a virus detection system, it is only as good as the database of attack signatures that it uses to compare with.

MANET systems have multiple measurement attributes that can be used to describe its behaviors. One measurement attribute is not sufficient to characterize accurately the MANET node’s operational states. Hence, multivariate analysis is needed for determining the system operational states that can be used to detect abnormal behaviors.

In this article, we propose an approach to perform online multivariate analysis based on fuzzy analysis for MANET attack detection. By using this multivariate analysis algorithm we can efficiently detect MANET attacks at their earliest stage of propagation and prevent further damages to MANET. The fuzzy-based multivariate analysis methods combine multiple aspects of a MANET node’s behavior and quantify the threat. If this threat is larger than the predetermined threshold value, the observed state of the MANET node is considered vulnerable and the appropriate recovery actions can be invoked. Otherwise, it is considered normal.

The article is organized as follows: First is a summary of the related approaches for MANET and their limitations. Next, our Multivariate Fuzzy Analysis Threat Detection Architecture is presented, followed by our overall algorithm for this methodology. Then, an example to illustrate the fuzzy logic methodology for Threat Index (TI) evaluation to detect MANET threats is presented, followed by the presentation of experimental results that demonstrate the effectiveness of our approach to detect MANET Denial of Service (DoS) attacks. Finally, the conclusion is presented.

RELATED WORK
The following are some of the related security models for MANET that we studied in our literature survey. Kachirski and Guha proposed an Intrusion Detection System (IDS) model which is efficient and bandwidth-conscious (Kachirski & Guha, 2003). It targets intrusion at multiple levels and fits the distributed nature of IDA for MANET.
The method has clusters and the IDA on cluster head employs independent detection decision-making after gathering information from other nodes. It utilizes mobile agent for communication among various nodes. This model provides a framework to work with multiple types of audit data. It is expandable, meaning, if the IDA needs to work with new types of audit data, it can do so by just incorporating extra agents that can monitor the new type of audit data. Unfortunately, its performance is not verified by any implementation. Once its performance is proved to be on an acceptable level, this framework can serve as a generic and expandable architecture for commercial products, because having a possibility to add in more functionality is an important property for successful products. Because it utilizes the cluster heads, it is supposed to make the MANET more efficient by limiting the resources usage for IDA purposes to only a few nodes. Such a framework can be applied in an environment where the security requirement is medium and efficiency requirement is high. Also, it may easily be expanded for multilayered MANET.

The IDS model for wireless mobile ad hoc networks proposed by Zhang and Lee implements local and collaborative decision making with anomaly detection (Zhang, Lee, & Huang, 2003). In this approach, individual IDA agents can work by themselves and also collaborate in decision making. Each IDA agent runs on a node and monitors local activities. If a node detects local intrusion with strong evidence, then the node concludes that intrusion has happened and initiates an alarm response. However, if the evidence is not strong enough, but needs investigation in a wider area in the network, then the IDA agent can start collaboration procedure which is a distributed consensus algorithm. This model provides a framework that fits the distributed nature of MANET. It also works with multiple types of audit data. If the IDA needs to work with new types of data, it can add in more data collection module in the IDA agent. It uses data mining as the local intrusion detection mechanism. The data mining is supposed to be superior in terms of both detection rate and false alarm rate. Also, because this IDA does not use mobile agents for communication, it can be designed for high security need, if it can find an effective way to protect from Byzantine nodes. This framework is designed for flat MANET. In a large multilayered MANET, it can work in a subsection of the MANET.

Tseng and Balasubramanyam (2003) have proposed an IDS system where the normal behavior of critical objects in the network is constructed with the normal specification first. Then the actual behavior is compared to the normal specification. It uses distributed network monitor to trace the request-reply flow in the routing protocol. The network monitor runs a specification-based detection algorithm to make decisions (Okazaki, Sato, & Goto, 2002; Sekar, 2002). This model is novel with no conventional local detection mecha-
nism, but has low efficiency because packet is checked at each hop.

Neighborhood Watch, an IDS protocol proposed by Sowjanya and Shah (2002), has two neighboring nodes of which one node is used to ensure that the packets are not modified while traveling in the network. This is done by comparing the information in each packet at each hop. It has two modes: passive mode—to protect a single host and active mode—to collaboratively protect the nodes in a cluster. In active mode, a cluster head starts a voting algorithm to determine whether intrusion really happens.

Puttini et al. have proposed an IDS architecture where information in the management information base (MIB) is used as input data (Puttini, Percher, Me, Camp, & De Souza, 2003). It also uses mobile agent and a collaborative decision-making mechanism. This model is distributed and efficient in use, with high scalability and can detect attack at multiple levels, but has security, computational cost and management problems related to mobile agents.

Most of the surveyed models use packets and network traffic-related information such as updates in routing table or request-reply flow in the network. Among the ones that use packets-related information, IDS approach proposed in Guha, Kachirski, Schwartz, Stoecklin, and Yilmaz (2002) and Tseng and Balasubramanyam (2003) uses the information inside the packets header directly, such as network address or port number. Other models using packet or network traffic-related information mainly use statistical data processes from packet information, such as the statistics of the number of packets received and sent or the statistics of change in routing table. IDS Model as described in Huang, Fan, Lee, and Yu (2003, pp. 478-487) utilizes the statistics derived from packet- or traffic-related statistics, for instance, the correlation between the number of packets dropped and the percentages of updates in the routing table. Intrusion Detection approaches illustrated in Kachirski and Guha (2003) and Zhang et al. (2003) allow the IDA to work on different types of audit data or the possibility to adapt to different types of audit data. This property is valuable and should be an important consideration for the future design of IDA. Most of the architectures detect only the fact that an intrusion happens. Some models go further to obtain more information, such as the type of attack and the location of the intruder. For instance, Zone-based IDA can detect both the type and location of the attack (Sun et al., 2006).

**Limitations of Related Models**

In Qu, Hariri, Hussain, Oh, Fayssal, and Yousif (2004) and Hariri et al. (2005), the authors proposed Multivariate Analysis for Network Attack detection algorithm (MANA) that uses abnormality distance (AD) metric to quantify how far the current operational state of a component from the normal state based on one or more measurement attributes.
However, one of MANA’s main weaknesses is that the anomaly status detection is sensitive to the number of observations (L) parameter. That is, if L is too short, the nodes will suffer from continually calculating normal region size and updating AD value. On the contrary, if L is too high, the nodes will suffer from unreliable normal region size and unreliable AD metric.

On the other hand, there is very little work that has used multivariate analysis in MANET IDA. The model proposed by Sun et al. (2006) uses Hotelling’s T2 multivariate statistical analysis for MANET IDA. Per our literature survey, there is only one other work that has used fuzzy logic for MANET threat detection. In Jin, Zhang, Lai, and Zhou (2006) and Nie, Wen, Luo, He, and Zhou (2006), the authors proposed a protocol that can determine the security-level of an individual mobile host in MANETs and use it to determine the routes with the best security-level. Although the authors used fuzzy logic to achieve this target, they used the secret key length and its frequency of change as the measurement to detect the security-leak behaviors. These two metrics are related to the secret key attributes (misuse detection approach) and not to the MANET nodes attributes. As we mentioned in the previous section, our target in this article is abnormal detection approach rather than misuse detection approach.

The misuse detection systems use patterns of known attacks to match and identify those intrusions (Sun, Wu, & Pooch, 2003). Although it can accurately and efficiently detect instances of known attacks, it lacks the ability to adapt in detecting new type of attacks. The anomaly detection systems, on the other hand, detect intrusions by finding deviations from the established user profiles. Anomaly detection should detect new types of intrusions but it could have a higher false positive rate (Guha et al., 2002). Traditionally, IDA are developed using expert knowledge of the system and attack methods (Huang et al., 2003). Due to the complexity of the modern network system and sophistication of attackers, expert knowledge engineering is often very limited and unreliable (Zhang et al., 2003). Some IDA schemes are very sensitive to the data representation. For instance, these schemes may fail to generalize an unseen data if the representation contains irrelevant information. In some instance, it has been observed that training of IDA requires a noise-free data (the data that is labeled “normal”) (Kachirski & Guha, 2003).

It has been observed that the existing IDA performs poorly in detection as well as the false positive rates at higher mobility rates (Brutch & Ko, 2003; Huang et al., 2003; Sun et al., 2006). It has recently been observed that Denial of Service (DoS) attacks are targeted even against the IDA (Zhang et al., 2003). Thus, IDA themselves need to be protected. An IDA should also be able to distinguish an attack from an internal system fault.

The current schemes thus have practical problems in intrusion detection.
and real time response. The proposed fuzzy multivariate fuzzy analysis-based Intrusion Detection model for Mobile Ad hoc Networks security model addresses many of these limitations. To demonstrate that the proposed approach does have these features and performs better in terms of the above proposed metrics, we conducted performance evaluation experiments using different metrics and parameters. These experiments and results are explained.

MULTIVARIATE FUZZY ANALYSIS DETECTION METHODOLOGY

Rationale of the Proposed Model
To strengthen our premise of using network parameters as the audit data to detect threat, we explain three different types of attacks and their effect on the network parameters in the following paragraphs of this section.

DoS Attack: Bombardment of Packets by an Intruder Node on MANET Host Node
In a type of DoS attack, an intruder bombards packets on a MANET host node servicing multiple mobile nodes. In this attack model, the intruder creates huge traffic in the link between the intruder and the host resulting in the quick exhaustion of the MANET hosts’ precious resources. This kind of DoS attack results in the inability of the host node to serve the other genuine nodes in MANET fairly.

DoS attack is depicted in Figure 1, where node B is a host node and C is the intruder. The intruder node C creates huge traffic, resulting in the exhaustion of the node B’s resources. This results in the inability of node B to serve genuine nodes A, D, E and F fairly. Thus, DoS attacks on the mobile ad hoc networks can lead to network performance degradation. Some of the critical parameters with respect to MANET that are affected by this type of DoS attacks are:

Figure 1. DoS attack

![Figure 1. DoS attack](image-url)
• **Packet drop**: Due to DoS attacks, packets in the link may be dropped due to exhaustion of the hosts’ resources.

• **Queue length**: The inability of the host to service the request from other nodes because DoS attacks result in an increase in queue length on the links between the host and other nodes.

• **Energy consumption**: The bombardment of packets due to traffic and servicing them results in the consumption of significant battery power (a constrained resource in MANET) in the link between intruder and host.

**Active Attack: “Poisoning” Routing Table by Intruder Causing Routing Loop**

In this kind of attack, an intruder injects incorrect routing information, which in turn poisons the routing table or protocol. The intruder may also update the routing table to create a loop, so that packets traverse in the network without reaching the destination (Alampalayam & Kumar, 2004).

In the MANET shown in Figure 2, let us assume that packets are supposed to traverse from source node A to destination node C. However, the intruder updates the routing table so that the packets traverse from B to D instead of C, and hence the packets from A never reach C. This also causes congestion on domains served by nodes A, D and E, due to the bombardment of packets whose actual destination was C. Thus, the attack can lead to network performance degradation. Some of the critical parameters of mobile ad hoc networks that are affected by this type of active routing attacks are:

• **Throughput**: Due to the poisoning of routing protocol or table, packets may never reach the destination. Thus, there is a relative increase in the number of lost packets from the genuine nodes during the attack.

• **Total number of packets dropped due to no routing information**: Due to poisoning of the routing protocol or table, packets may be dropped from the network for want of routing information. The drop count
of packets from genuine nodes increases due to the attack.

**Passive Attack: Packet Discarding by Selfish Nodes**

It has been demonstrated statistically that over 60% of the resources are taken up by packet routing function at the nodes (Nie et al., 2006). If a node has turned selfish, it does not route to conserve energy, thus affecting other nodes on the ad hoc network. A selfish node simply does not perform its intended function of forwarding the packet to a proper destination node and routes all packets to itself and later discards them. The motivation in these attacks by selfish nodes is to save significant battery power, instead of performing networking functions such as packet forwarding and routing.

As shown in Figure 3, the packets are supposed to traverse from source node A to destination node C. However, selfish node B discards the packets from A and hence the packets from A never reach C. This results in “black hole” attacks. This in turn may also result in DoS or deadlock issues that result in performance degradation. Some of the critical parameters that are affected by this kind of passive attack with respect to ad hoc networks are:

- **Throughput**: Due to packet mistreatment by selfish nodes, packets do not generally reach host nodes. This results in packet loss, and hence a significant decrease in the measurement of throughput for the destination nodes.
- **Packet drop rate**: Packets are discarded by selfish nodes and hence there is a significant increase in the packet drop rate for the collaborating selfish nodes in the ad hoc network.
- **Energy consumption**: Because battery power is a constrained resource in mobile ad hoc networks, the selfish nodes discard packets to save battery power by means of techniques like sleep deprivation (Nie et al., 2006). Hence, there will be a relative decrease in the measurement of power or energy consumed for groups of selfish nodes within the network.

*Figure 3. Packet mistreatment attack*
These examples of different attacks clearly indicate that for both active and passive attacks, the network parameters are significantly affected and deviate from their normal levels. Thus, identifying the appropriate network parameters for each attack, and continuously monitoring them, will be an effective method to detect attacks.

Model Architecture

The basic framework of the threat detection model is shown in Figure 4 and is explained as follows. In Figure 4, MANET is represented as a function: 

$$f(x_1(t), x_2(t), \ldots, x_n(t), v_1(t), v_2(t), \ldots, v_n(t), m_1(t), m_2(t), \ldots, m_n(t), k(t), u(t))$$

where $x_n(t)$ represents the significant attack sensitive network parameters, $v_n(t)$ represents the network parameters which are not significant in representing the node vulnerability, $m_n(t)$ represents the mobility parameters, $k(t)$ represents the attack and $u(t)$ represents the control input. $x_n'(t)$ represents the modified values of the significant attack sensitive network parameter due to the influence of the attack $k(t)$ and the control input $u(t)$.

Threat Index (TI) for a node is calculated by the threat detection framework from the attack sensitive network parameters, $x_n'(t)$ using fuzzy logic. The computed Threat Index $TI(t)$ is compared with the threshold values of the Threat Index $TI'$. The Threat Index thresholds ($TI'$) are obtained with the help of the training dataset where the state of each record is labeled. Data records collected from the simulation environment with and without attack are used as a training dataset for identifying the Threat Index thresholds.

Figure 4. Fuzzy multivariate analysis-based threat detection framework for MANET
In order to obtain the TI thresholds (TI'), we have set the TI metric to be a number between 1 and 10, where 10 represents the highest threat (Most Vulnerable State) and 1 represents the lowest threat (Most Normal State) to the node. For the value of TI to make sense, we need to classify which number indicates which state of the node: normal state, uncertain state, or vulnerable state. So we need to identify uncertain state threshold (P_1) and vulnerable state threshold (P_2) for TI from this range of 1 to 10, such that values of TI less than the uncertain state threshold (P_1) indicate a node is in normal state, TI values greater than the vulnerable state threshold (P_2) indicate a node is in vulnerable state and TI values between the uncertain state threshold (P_1) and the vulnerable state threshold (P_2) indicate a node is in the uncertain state.

Once obtained, these TI thresholds (TI') are used for the following purposes in the proposed model: The first purpose is to assign outputs (Y_j) in the fuzzy rule and the second purpose is to identify the node state as being normal, uncertain or vulnerable by comparing the computed TI with these thresholds.

Because P_1 and P_2 are the integers values between 1 and 10 and P_2 should be greater than P_1, 45 combinations of P_1 and P_2 are possible. The following brute force search algorithm is applied on the training sample space S to search for the pair (P_1, P_2) that gives the correct classification of TI.

**Algorithm**
The algorithm for the TI threshold training is illustrated in the Figure 5.

**Figure 5. TI threshold training algorithm**

1. Let there be N records in the training sample space S, where each record consists of the following elements: [x_1, x_2, ..., x_m, Outcome Label]. Here x_1, x_2, ..., x_m represent the values of the significant parameters and ‘Outcome Label’ represents ‘Normal’, ‘Uncertain’ and ‘Vulnerable’ to indicate the true state.
2. For each of the possible pair (P_1, P_2) where P_2 > P_1 and 1 <= P_1 <= 10 and 1 <= P_2 <= 10 assign a counter that is initialized to 0. This counter is incremented when the TI state calculated using that pair matches the Outcome Label.
3. For each record in the sample space S, perform the following steps.
   3.1 For each possible pair of values (P_1, P_2), substitute values (P_1, P_2) for uncertain threshold values and vulnerable threshold values respectively in the output of each fuzzy rule and calculate TI using x_1, x_2, ..., x_m and their membership functions as defined by the equation 4.
   3.2 If the TI value obtained is greater than P_2, classify ‘the calculated TI state’ as vulnerable state. If the TI value obtained is less than P_1, classify ‘the calculated TI state’ as normal state. If the TI value obtained is in between P_1 and P_2 classify ‘the calculated TI state’ as uncertain state.
   3.3 Compare the ‘calculated TI state’ with the outcome label of the record. If they are same then increment the counter for that pair (P_1, P_2).
4. Use the pair (P_1, P_2) that has the highest counter value to be the uncertain state and vulnerable state threshold value respectively for TI in the model.
5. The TI range 1 to P_1 represents the Normal State, P_1 to P_2 represents the Uncertain State and P_2 to 10 represents the Vulnerable State.
Based on the experimentation explained in Section 5, the TI threshold training implementation of the algorithm with experimental data resulted in the pair (4, 7) having the highest count. Hence, the TI threshold (TI') for the uncertain state of the network, P_1, is chosen to be 4, and the TI threshold (TI’) for vulnerable state of the network, P_2, is chosen to be 7.

As shown in Figure 4, the training data is derived from MANET and is used in the identification of significant parameters, thresholds of these significant parameters and the threat index. They are also used to optimize the fuzzy membership functions for fuzzy multivariate analysis. If the computed TI(t) of a node is greater than or equal to vulnerable state threshold reference TI’, the node is identified to be under threat. Threat Index is maintained or updated by continuously computing applying Multivariate Fuzzy Membership Function Optimization Algorithm and defuzzification equation 4 using the measured online significant network parameters.

The response and protection framework is executed once a node is identified to be under threat, in order to respond to the attacks and protect the MANET. This article does not deal with response and protection framework. Our work on the response and protection framework is explained in detail in Alampalayam and Kumar (2004).

Because multivariate fuzzy analysis-based MANET threat detection model is an online-based model, we need to determine the optimal execution time in order for the online algorithm to work properly. The desired minimum execution time can be estimated based on the various computation tasks in the multivariate fuzzy analysis-based intrusion detection algorithm. Based on the time complexity analysis, the algorithm’s time complexity is turned out to be O(N^3). The execution time also depends on the speed of the processor. If a processor executes 1 billion instructions per second, for O(N^3) algorithm, it will take one second for N = 1,000. Here, N indicates that there are 1,000 nodes and each node has 1,000 neighboring nodes and there are 1,000 parameters to analyze. Thus, the minimum execution time in the model can be set depending on the number of nodes in MANET and the number of significant parameters, as well as the speed of the processor.

**MULTIVARIATE FUZZY MEMBERSHIP FUNCTION OPTIMIZATION METHODOLOGY**

Figure 6 shows the algorithm for the multivariate fuzzy membership function optimization methodology. For a fuzzy variable, x with universe of discourse [0, U_x] and three fuzzy sets NS, US, and VS, the membership functions are as shown in Figure 7. Here, NS represents the normal state fuzzy set, US represents the uncertain state fuzzy set and VS represents the vulnerable state fuzzy set. The fuzzy variable, x (identified in the identification of the significant
network parameters block shown in Figure 4) represents the significant network parameter that detects a particular attack. In Figure 7, \( n_s \) and \( v_s \) are initialized with the threshold values of the significant parameters obtained from the training data in the identification of thresholds for the significant network parameters block shown in Figure 4. Let \( u_s \) be initialized to be the average of \( n_s \) and \( v_s \). The membership function optimization problem is the determination of the optimized \( n_s \), \( u_s \) and \( v_s \) points for the fuzzy variable, \( x \).

The membership function optimization methodology works by increasing or decreasing the membership value of each fuzzy set based on the multivariate input training data. The concept here is to increase/decrease the membership values by modifying the slope. This is achieved by multiplying \( v_s \) with decrease_ratio in order to shift \( v_s \) value to the left; \( n_s \) is multiplied with increase_ratio in order to shift \( n_s \) to the right and for \( u_s \), it is shifted either left or right depending on the input values.

Figure 8 shows the membership functions for fuzzy variable, \( x \) after \( v_s \) is shifted to left by multiplying \( v_s \) with decrease_ratio.

Figure 9 shows the membership functions for fuzzy variable, \( x \) after \( n_s \) is shifted to the right by multiplying \( n_s \) with increase_ratio.

Due to the triangle rule, the value of \( u_s \) must always be greater than \( n_s \) and lower than \( v_s \). The triangle rule

---

**Figure 6. Multivariate fuzzy membership function optimization algorithm**

```
decrease_ratio = 0.97;
increase_ratio = 1.03;
for i = 1: no_of_training_data
    Calculate output_TI using proposed fuzzy system with the new MFs;
    % For each fuzzy variable \( x \) [ec pd ql], do:
    % Test NS network %
    if((Target <=4) & (output_TI > 4))
        if ((ns \_x * increase_ratio) < us \_x)
            ns \_x = ns \_x * increase_ratio;  / / shifting ns \_x to right
        end
    end
    % Test VS network %
    if((Target >=7) & (output_TI < 7))
        if ((vs \_x * decrease_ratio) > us \_x)
            vs \_x = vs \_x * decrease_ratio;  / / shifting vs \_x to left
        end
    end
    % Test US network %
    if((Target > 4) & (Target < 7) & (output_TI < 4))
        if ((us \_x * increase_ratio) < vs \_x)
            us \_x = us \_x * increase_ratio;   // shifting us \_x to right
        end
    end
    if((Target > 4) & (Target < 7) & (output_TI > 7))
        if ((us \_x * decrease_ratio) > ns \_x)
            us \_x = us \_x * decrease_ratio;   // shifting us \_x to left
        end
    end
end
```
is given by: \( ns_x < us_x < vs_x \). This rule is taken care of by checking the result of multiplication and making sure the above rule is preserved before assigning the new values to \( ns_x, us_x \), and \( vs_x \).

In our membership function optimization methodology, the increase_ratio and decrease_ratio were selected to be 1.03 and 0.97, respectively. This is due to the reason that the selected training data is not continuous, and so we cannot choose values less than 3% and keep increasing or decreasing it. Also, the decrease/increase ratio cannot be in...
a very large range (say 60% or 40%). This is because the range between $n_s$ and $u_s$ is set to 50% and the range between $u_s$ and $v_s$ is set to 50% as $u_s$ is initialized to be the average of $n_s$ and $v_s$. The use of 3% for increment/decrement ratio also means that the membership function optimization methodology needs around 15 steps to make $n_s = u_s$ (the maximum value for $n_s$ using triangle rule), and around 15 steps to make $v_s = u_s$ (the minimum value for $v_s$ using triangle rule). To summarize, in recommending values for increment/decrement ratio, it is necessary to ensure that a single variation of the triangle points ($n_s$, $u_s$, and $v_s$) does not result in large variation in the membership functions. In our case, it takes a maximum of 15 intervals to increase from $n_s$ to $u_s$ or to decrease from $v_s$ to $u_s$.

Also, in our fuzzy membership function optimization methodology, it is not necessary to optimize all input variables at once. For example, let Packet Drop (PD), Queue Length (QL) and Energy Consumption (EC) be identified as the significant parameters for a DoS attack (using the identification of the significant parameters block in the Figure 4). Let the threshold ranges for the normal state (NS), uncertain state (US) and vulnerable state (VS) of these network parameters were identified in identification of thresholds for significant network parameters block shown in Figure 4, and they are then subjected to multivariate fuzzy membership optimization. After multivariate fuzzy membership optimization, for the Packet Drop (PD) parameter, let

$$NS = [0, 119],$$
$$US = [119, 208],$$
$$VS = [208, U_{PD}].$$

For Energy Consumption (EC) parameter, let

$$NS = [0, 1.33 \text{ Joules}],$$
$$US = [1.33 \text{ Joules}, 1.99 \text{ Joules}],$$
$$VS = [1.99 \text{ Joules}, U_{EC}].$$

For Queue Length (QL) parameter, let

$$NS = [0, 656].$$

**EXAMPLE TO ILLUSTRATE FUZZY LOGIC METHODOLOGY FOR TI EVALUATION TO DETECT THREAT**

Let the network parameters, Packet Drop (PD), Queue Length (QL) and Energy Consumption (EC) be identified as the significant parameters for a DoS attack (using the identification of the significant parameters block in Figure 4). Let the threshold ranges for the normal state (NS), uncertain state (US) and vulnerable state (VS) of these network parameters were identified in identification of thresholds for significant network parameters block shown in Figure 4, and they are then subjected to multivariate fuzzy membership optimization. After multivariate fuzzy membership optimization, for the Packet Drop (PD) parameter, let

$$NS = [0, 119],$$
$$US = [119, 208],$$
$$VS = [208, U_{PD}].$$

For Energy Consumption (EC) parameter, let

$$NS = [0, 1.33 \text{ Joules}],$$
$$US = [1.33 \text{ Joules}, 1.99 \text{ Joules}],$$
$$VS = [1.99 \text{ Joules}, U_{EC}].$$

For Queue Length (QL) parameter, let

$$NS = [0, 656].$$
US = ] 656, 1157[ and
VS = [1157, UQL].

Here \( U_{PD} \), \( U_{EC} \) and \( U_{QL} \) represent the upper limit of the universe of discourse of these parameters. The next step is to formulate the fuzzy relation of these three attack sensitive network parameters with the grade of membership. Figure 10 below represents the fuzzy relation of PD with the membership functions.

Figure 10 shows Z-shaped, triangular and S-shaped membership functions.

We have used the triangular membership functions due to the reason that the parametric and functional descriptions of these membership functions are efficient. In these membership functions, the designer needs only to define three parameters; \( n_{sx} \), \( v_{sx} \) and \( u_{sx} \). Here \( n_{sx} \), \( u_{sx} \), and \( v_{sx} \) are the normal state, uncertain state and vulnerable state threshold values. It has been proven that triangular MFs can approximate any other membership function (Pedrycz, 1994). This function is specified by three parameters (a, b, c) as shown in Box 1, where \( a = n_{sx} \), \( b = v_{sx} \), \( c = v_{sx} \) and \( x_{in} \) is the input to the fuzzy system of type fuzzy variable x. The remaining MFs are as follows: Z-shaped membership to represent the whole set of NS (Normal State) values and S-shaped membership to represent the whole set of US (Uncertain State) values. The Z-shaped membership function represents the fuzzy set

\[
\mu_{NS}(PD) = \begin{cases} 
1.0 & \text{for } 0 \leq x_{in} \leq 119 \\
0 & \text{elsewhere}
\end{cases}
\]

Figure 10. Fuzzy model for packet drop metric

Box 1.

\[
\text{triangle}(x; a, b, c) = \begin{cases} 
\frac{(x_{in} - a)}{(b - a)} & \text{for } a \leq x_{in} \leq b \\
\frac{(c - x_{in})}{(c - b)} & \text{for } b \leq x_{in} \leq c \\
0 & \text{elsewhere}
\end{cases}
\]
This fuzzy set indicates that membership function \( \mu_{NS} \) of PD is 1 for the value of PD from 0 to 119, and the membership function \( \mu_{NS} \) of PD is 0 when the value of PD is 163. The second fuzzy set is a triangle and represents the fuzzy set

\[
\mu_{US}(PD) = \{0.0/119, 1.0/163, 0.0/208\}.
\]

The S-shaped membership function represents the fuzzy set

\[
\mu_{VS}(PD) = \{0.0/163, 1.0/>=208\}.
\]

Similar to PD, the fuzzy models for EC and QL can be arrived to determine their MF points. Next, we assign the threshold values for the threat index of NS and VS (TI\(^{NS}\) and TI\(^{VS}\)). Based on the training data, TI\(^{NS}\) and TI\(^{VS}\) are 4 and 7, respectively. These thresholds are used to assign the threat index range for the node status to be:

\[
TI_{NS} = [0, TI^{NS}],
\]

\[
TI_{US} = [TI^{NS}, TI^{VS}], \text{ and}
\]

\[
TI_{VS} = [TI^{VS}, 10].
\]

The next step is to formulate all the combinations of rules possible for this system. Because the system has 3 network parameters and 3 grades of membership for each parameter, that is, \(n=3\) and \(k=3\), and hence the number of fuzzy rules, \(m = k^n = 3^3 = 27\), 27 combinations of rules are possible. The rules are presented in Table 1. For example, according to Table 1, the first rule is: If PD is NS and QL is NS and EC is NS, then TI is NS.

### Table 1. Rule-base for fuzzy threat index

<table>
<thead>
<tr>
<th>No.</th>
<th>PD</th>
<th>QL</th>
<th>EC</th>
<th>TI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>2</td>
<td>NS</td>
<td>NS</td>
<td>US</td>
<td>US</td>
</tr>
<tr>
<td>3</td>
<td>NS</td>
<td>NS</td>
<td>VS</td>
<td>NS</td>
</tr>
<tr>
<td>4</td>
<td>NS</td>
<td>US</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>5</td>
<td>NS</td>
<td>US</td>
<td>US</td>
<td>US</td>
</tr>
<tr>
<td>6</td>
<td>NS</td>
<td>US</td>
<td>VS</td>
<td>US</td>
</tr>
<tr>
<td>7</td>
<td>NS</td>
<td>VS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>8</td>
<td>NS</td>
<td>VS</td>
<td>US</td>
<td>US</td>
</tr>
<tr>
<td>9</td>
<td>NS</td>
<td>VS</td>
<td>VS</td>
<td>VS</td>
</tr>
<tr>
<td>10</td>
<td>US</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>11</td>
<td>US</td>
<td>NS</td>
<td>US</td>
<td>US</td>
</tr>
<tr>
<td>12</td>
<td>US</td>
<td>NS</td>
<td>VS</td>
<td>US</td>
</tr>
<tr>
<td>13</td>
<td>US</td>
<td>US</td>
<td>NS</td>
<td>US</td>
</tr>
<tr>
<td>14</td>
<td>US</td>
<td>US</td>
<td>US</td>
<td>US</td>
</tr>
<tr>
<td>15</td>
<td>US</td>
<td>US</td>
<td>VS</td>
<td>US</td>
</tr>
<tr>
<td>16</td>
<td>US</td>
<td>VS</td>
<td>NS</td>
<td>US</td>
</tr>
<tr>
<td>17</td>
<td>US</td>
<td>VS</td>
<td>US</td>
<td>US</td>
</tr>
<tr>
<td>18</td>
<td>US</td>
<td>VS</td>
<td>VS</td>
<td>VS</td>
</tr>
<tr>
<td>19</td>
<td>VS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>20</td>
<td>VS</td>
<td>NS</td>
<td>US</td>
<td>US</td>
</tr>
<tr>
<td>21</td>
<td>VS</td>
<td>NS</td>
<td>VS</td>
<td>VS</td>
</tr>
<tr>
<td>22</td>
<td>VS</td>
<td>US</td>
<td>NS</td>
<td>US</td>
</tr>
<tr>
<td>23</td>
<td>VS</td>
<td>US</td>
<td>US</td>
<td>US</td>
</tr>
<tr>
<td>24</td>
<td>VS</td>
<td>US</td>
<td>VS</td>
<td>US</td>
</tr>
<tr>
<td>25</td>
<td>VS</td>
<td>VS</td>
<td>NS</td>
<td>VS</td>
</tr>
<tr>
<td>26</td>
<td>VS</td>
<td>VS</td>
<td>US</td>
<td>US</td>
</tr>
<tr>
<td>27</td>
<td>VS</td>
<td>VS</td>
<td>VS</td>
<td>VS</td>
</tr>
</tbody>
</table>

---

**Fuzzification, Inference and Defuzzification**

The fundamental diagram of the fuzzy system is presented in Figure 11. Formally, the rule-base (Table 1) of the fuzzy-TI method can be represented in the following format:
If PD is $A_{i1}$ and QL is $A_{i2}$ and EC is $A_{i3}$ then TI is $B_i$.

where $A_{i1}$, $A_{i2}$, $A_{i3}$, and $B_i$ are the linguistic labels NS, US, and VS of the $i^{th}$ rule.

Fuzzification is a process where crisp input values are transformed into membership values of the fuzzy sets (as described in the previous section). After the process of fuzzification, the inference engine calculates the fuzzy output using the fuzzy rules described in Table 1. Mamdani method (Yager & Filev, 1994) is used as a fuzzy inference engine, where the Min ($\land$) operator is chosen as AND connective between the antecedents of the rules as follows:

$$\tau_i = A_{i1}(x_1) \land A_{i2}(x_2) \land A_{i3}(x_3)$$

(1)

where $\tau_i$ is called the degree of firing of the $i^{th}$ rule for the input values: PD = $x_1$, QL = $x_2$ and EC = $x_3$. The next step is determining the individual rule output $F_i$ (fuzzy set) which is obtained by:

$$F_i(y) = \tau_i \land B_i(y)$$

(2)

The third step is the aggregation of rules outputs to obtain the overall system output $F$ (fuzzy set), where the Max ($\lor$) operator is chosen as OR connective between the individual rules:

$$F(y) = \lor_i F_i(y) = \lor_i (\tau_i \land B_i(y))$$

(3)

For use in the ad hoc networks environment, a fourth step must be added. A crisp single value for LL is needed. This process is called defuzzification. Center of area (COA) (Yager & Filev, 1994) is chosen as the defuzzication method given in the following:
\[ TI = \frac{\sum_{j=1}^{m} F(y_j) \times y_j}{\sum_{j=1}^{m} F(y_j)}. \] (4)

Here, \( y_j \) is a sampling point in the discrete universe output \( F \), and \( F(y_j) \) is its membership degree in the membership function.

**SIMULATION EXPERIMENTATION AND RESULTS**

**Simulation Environment and Experiment to get the Training Data**

Figure 12 shows the part of simulated MANET used to get the training data. Nodes (denoted by \( N_i \)) are connected within the mobile ad hoc network environment, and mobile agents (denoted by \( M_i \)) are dispatched by a source node to a destination node for service purposes. The simulation of the MANET was carried out using NS2.

The parameters for the MANET to be simulated were specified using the OTcl configuration script. The routing protocol used in NS2 for mobile ad hoc networks was the Ad hoc On Demand Distance Vector (AODV) routing protocol. The network parameters considered for analysis were: packet drop rate, energy consumption, and queue length based on the identification of significant parameters experiments using the classification trees methodology and training dataset (Alampalayam & Kumar, 2004).

In order to study the feasibility and performance of the proposed MANET intruder identification and response framework, we carried out extensive simulation experiments using various MANET parameters. In our simulation, the channel type was set to wireless channel type and TwoRayGround model was used as the propagation model. The radio model was modeled as a shared-media radio with all nodes having the same channel capacity of 2 Mb/s and a transmission range of 250 m.

In the simulation, 100 mobile nodes were set to move in a 1,000 meter x 500 meter simulation environment and experiment to get the training data.
meter rectangular region. Each node was set to move independently with the same average speed. The mobility model we used was the Random Waypoint (RW) model. In RW model, a node randomly selects a destination from the physical terrain. It moves in the direction of the destination in a speed uniformly chosen between the minimal and the maximal speed. After it reaches its destination, the nodes stay there for a pause time and then move again to a newly selected destination. In our simulation, the speed was set to 5 m/s. The pause time, which affects the relative speeds of the mobile nodes, was varied. Simulations were run for 1,000 simulated seconds. Constant Bit Rate (CBR) traffic sources were used. The source-destination pairs were spread randomly over the network to generate CBR traffic. The size of all data packets was set to 512 bytes. The hardware used for the simulation was the Pentium 3, 512 MB machine. The operating system used for the simulation was the Redhat 9.0, Linux Kernel 2.6.

In our simulation experiments, to generate training data we considered the Denial of Service (DoS) attack on a host node by a set of mobile agent-based nodes in MANET. These attacks by mobile agent-based mobile nodes cause the network to be loaded excessively, thus causing enormous retransmissions, which consumes an excessive amount of host resources and hence the host cannot service genuine agents properly (Alampalayam & Kumar, 2004). The basic steps of the simulation experimentation were as follows: Agent M\textsubscript{x} dispatched by node N\textsubscript{x} was configured to send heavy traffic to Node N\textsubscript{y}, while all other nodes received normal traffic from their agents. This is simulated in TCL Script by setting the set interval parameter for CBR traffic. For all the parameters (EC, PD and QL) monitored at all links between neighboring nodes and the host node, these parameters’ values are accumulated for every 20 seconds. For instance, from second 1 to second 20, the accumulated PD at node 2 for every second before response is 2,000. This methodology is followed for all the parameters that are monitored for all nodes to collect the training data. The initial normal and vulnerable state thresholds for the network parameters are calculated by means of the six-sigma methodology using this training data. They are used as initial thresholds for the fuzzy multivariate analysis and membership function optimization performed using the training data sourced from simulated MANET.

**Experimental Results**

This section presents the experimental results for the fuzzy multivariate analysis performed to optimize the fuzzy membership functions using our algorithm and methodology. The testing data for these experimental results is sourced from the MANET simulation environment. Table 2 shows the threshold values of the significant network parameters for DoS attacks with and without multivariate fuzzy optimization. In Table 2 NS, US and VS refer to the Normal State, Uncertain State and Vulnerable State of the network parameter, respectively.
Figure 13 shows the results of TI results calculated with and without membership function (MF) optimization, when the output label is normal.

As seen in Figure 13, the TI calculated using MF optimization follows the actual TI closely, whereas the TI calculated without MF optimization deviates from the actual normal TI. Figure 14 shows the results of TI calculated with and without membership function (MF) optimization, when the output label is abnormal. As seen in Figure 14, the TI calculated using MF optimization follows the actual TI closely, whereas the TI calculated without MF optimization deviates from the actual abnormal TI.

**Performance Evaluation Analysis**

This section explains the performance evaluation of the Multivariate Fuzzy Analysis for Mobile Ad hoc Network Threat Detection when more mobile devices are present.

<table>
<thead>
<tr>
<th>Significant Network Parameter</th>
<th>Threshold Value Without MF optimization</th>
<th>Threshold Value With MF optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy Consumption (EC) NS</td>
<td>2.1 Joules</td>
<td>2.1 Joules</td>
</tr>
<tr>
<td>Energy Consumption (EC) US</td>
<td>2.15 Joules</td>
<td>2.15 Joules</td>
</tr>
<tr>
<td>Energy Consumption (EC) VS</td>
<td>2.2 Joules</td>
<td>2.2 Joules</td>
</tr>
<tr>
<td>Packet Drop (PD) NS</td>
<td>109</td>
<td>125</td>
</tr>
<tr>
<td>Packet Drop (PD) US</td>
<td>117</td>
<td>191</td>
</tr>
<tr>
<td>Packet Drop (PD) VS</td>
<td>126</td>
<td>194</td>
</tr>
<tr>
<td>Queue Length (QL) NS</td>
<td>122</td>
<td>220</td>
</tr>
<tr>
<td>Queue Length (QL) US</td>
<td>134</td>
<td>249</td>
</tr>
<tr>
<td>Queue Length (QL) VS</td>
<td>146</td>
<td>255</td>
</tr>
</tbody>
</table>

*Figure 13. TI with and without MF optimization when output label is normal*
nodes and more malicious nodes are added into the network at varied mobility speed. The results of this section demonstrate the effectiveness of the proposed model from the scalability perspective with respect to the size of the mobile network as well.

Simulation environment and scenario for this experiment is the same. However, the number of mobile nodes is varied from 10 to 100 in steps of 10 with the minimum speed of 3m/s and the maximum speed of 15 m/s. Also, the number of malicious nodes is also varied from 10% to 70%. AODV is used as the routing protocol.

**Metrics for Performance Evaluation with Respect to Scalability**

The following metrics were chosen for evaluating the performance of the proposed model from the scalability perspective:

- **Detection rate**: It is defined as the percentage or fraction that a mobile node under threat is correctly detected. This is chosen because accuracy is one of the most important characteristics of threat detection. A high detection rate for a large-sized MANET is desirable for a scalable MANET IDA.

- **Total Processing Time**: It is defined as the total time the system takes to analyze parameters and detect that the node is under threat. This is an important metric because effective intrusion detection should happen quickly even when there are a large number of mobile nodes, so that the response can be applied before significant damage occurs to the MANET.
Experimental Results
The detection rate metric experimental results with and without Multivariate Fuzzy (MVF) optimization is shown in Figures 15 through 19. Figure 15 and Figure 16 shows the detection results of our model when more mobile nodes are added into MANET at varying mobile speed. Figure 17 and Figure 18 show the detection results of our model when more malicious mobile nodes are added into MANET at varying mobile speed.

As seen in Figure 15 and Figure 16 for detection rate metric, our approach works slightly better when there are

Figure 15. Detection rate metric results for varied number of mobile nodes when TI is calculated using MVF optimization

![Detection Rate analysis of Network Size parameter with varied mobility when TI is calculated using MVF optimization](image1)

Figure 16. Detection rate metric results for varied number of mobile nodes when TI is calculated without MVF optimization

![Detection Rate analysis of Network Size parameter with varied mobility when TI is calculated without MVF optimization](image2)
less mobile nodes compared to a higher number of mobile nodes. Also, the detection rate is slightly better when TI is calculated using MVF optimization compared to when MVF optimization is not used.

When MVF optimization is used as seen in Figure 15, the detection rate is close to 100% when the number of mobile nodes is less than 20. But as the number of mobile nodes increases, the detection rate falls to 98% when the number of nodes is 20, then to 95% when the number of nodes is 40 and finally to 82% when the number of nodes is 100. The detection rate is slightly better when the node mobility speed is less.

When MVF optimization is not used, as seen in Figure 16, the detection rate is slightly worse compared to when MVF optimization is used. Detection rate is close to 97% when the number of mobile nodes is less than 20. But as the number of mobile nodes increases, the detection rate falls to 90% when the number of nodes is 20, then to 85% when the number of nodes is 40 and finally to 77% when the number of nodes is 100. The detection rate is slightly better when the node mobility speed is less.

As seen in Figure 17 and Figure 18 for detection rate metric, our approach works slightly better when there are less malicious mobile nodes compared to a higher number of malicious mobile nodes. Also, the detection rate is slightly better when TI is calculated using MVF optimization compared to when MVF optimization is not used.

When MVF optimization is used, as seen in Figure 17, the detection rate is close to 100% when there are 10% malicious nodes in MANET. But as the number of malicious mobile nodes increases, the detection rate falls to 98% when there are 20% malicious nodes in MANET, then to 95% when there are 40% malicious nodes in MANET and finally to 82%, when there are 70% malicious nodes in MANET. The detection rate is slightly better when the node mobility speed is less.

When MVF optimization is not used, as seen in Figure 18, the detection rate is close to 92% when there are 10% malicious nodes in MANET. But as the number of malicious mobile nodes increases, the detection rate falls to 90% when there are 20% malicious nodes in MANET, then to 87% when there are 40% malicious nodes in MANET and finally to 80%, when there are 70% malicious nodes in MANET. The detection rate is slightly better when the node mobility speed is less.

The processing time metric experimental results with and without multivariate fuzzy (MVF) optimization are shown in Figures 19 through 22. Figure 19 and Figure 20 show the processing time of our model when more mobile nodes are added into MANET at varying mobile speed. Figure 21 and Figure 22 show the processing time of our model when more malicious mobile nodes are added into MANET at varying mobile speed.

Figure 19 and Figure 20 show the processing time metric of proposed
model when more mobile nodes are added into the MANET at varying speed. They show the results for the proposed model when the number of mobile nodes is increased from 10 to 100 in steps of 10.

As seen in Figure 19 and Figure 20 for processing time metric, our approach works slightly better when there are less mobile nodes compared to a higher number of mobile nodes. Also, the processing time is less when TI is calculated without MVF optimization compared to when MVF optimization is used.
When MVF optimization is used, as seen in Figure 19, the processing time is less than 2 seconds when the number of mobile nodes is less than 40. But as the number of mobile nodes increases, the processing time increases to 4 seconds when the number of nodes are 60, then to 8 seconds when the number of nodes are 70 and finally to about 14 seconds when the number of mobile nodes increase to 100. Also, the processing time is slightly less when the node mobility speed is less.

When MVF optimization is not used, as seen in Figure 20, the process-
ing time is less than 1.5 seconds when the number of mobile nodes is less than 40. But as the number of mobile nodes increases, the processing time increases to 4 seconds when the number of nodes are 60, then to 6 seconds when the number of nodes are 70 and finally to about 13 seconds when the number of mobile nodes increase to 100. Also, the processing time is slightly less when the node mobility speed is less.

Figure 21 and Figure 22 show the processing time metric of the proposed model when more malicious mobile

Figure 21. Processing time metric results for varied number of malicious mobile nodes when TI is calculated using MVF optimization

![Graph showing processing time for varying number of malicious nodes when TI is calculated using MVF optimization.](image)

Figure 22. Processing time metric results for varied number of malicious mobile nodes when TI is calculated without MVF optimization

![Graph showing processing time for varying number of malicious nodes when TI is calculated without MVF optimization.](image)
nodes are added into the MANET at varying speed. They show the results for the proposed model when the number of malicious mobile nodes is increased from 10% to 70% of size of MANET in steps of 10%.

As seen in Figure 21 and Figure 22 for processing time metric, our approach works slightly better when there are less malicious mobile nodes compared to higher number of malicious mobile nodes. Also, the processing time is less when TI is calculated without MVF optimization compared to when MVF optimization is used.

When MVF optimization is used, as seen in Figure 21, the processing time is close to 1 second when there are 10% malicious nodes in MANET. But as the number of malicious mobile nodes increases, the processing time increases to 1.5 seconds when there are 20% malicious nodes in MANET, then to 2 seconds when there are 40% malicious nodes in MANET and finally to 8 seconds, when there are 70% malicious nodes in MANET. The processing time is slightly better when the node mobility speed is less.

When MVF optimization is not used, as seen in Figure 22, the processing time is less than 1 second when there are 10% malicious nodes in MANET. But as the number of malicious mobile nodes increases, the processing time increases to 1 second when there are 20% malicious nodes in MANET, then to 1.75 seconds when there are 40% malicious nodes in MANET and finally to 6.5 seconds, when there are 70% malicious nodes in MANET. The processing time is slightly better when the node mobility speed is less.

**CONCLUSION**

In this article, we have developed an effective multivariate analysis algorithm based on multivariate fuzzy analysis technique to analyze the behavior of MANET and detect its threat for well-known MANET attacks. Our online monitoring and analysis algorithm builds an adaptive behavior profile of system resources that can be used to detect any abnormal MANET behavior and detect the threat caused by MANET attacks. Normalization of TI (security level) from the arbitrary fuzzy sets is performed with the help of our multivariate fuzzy membership function optimization algorithm. This algorithm identifies the optimized membership function points of all the input significant parameters by training them together. These trained and optimized membership function points normalize the input parameters even though they are in multidimensional space, and hence generate the normalized output TI. We have validated our approach to detect Denial of Service attacks in MANET.

**REFERENCES**


Sathish Kumar Alampalayam obtained his PhD in computer science and engineering from University of Louisville, Kentucky, USA in 2007. Currently, he is working as IT Solution Delivery Manager in IT industry and as an Adjunct Faculty at the Department of Computer Science, California State University, Los Angeles, California USA. Alampalayam has more than ten years of industry, teaching and research experiences in USA. Also, he has more than 10 publications in refereed journals and conferences at the international level. His research interest is in mobile ad-hoc networks, in particular, the security architectures and algorithms for this type of networking.

Essam Natsheh obtained his PhD in communications and networks engineering from University Putra Malaysia in 2006. Currently, he is an assistant professor at the computer information systems department, College of Applied Studies and Community Services, King Faisal University (Saudi Arabia). Natsheh has more than ten years of teaching and research experiences in Malaysia and Saudi Arabia. Also, he has more than 15 publications in refereed journals at international level. His research interest is mobile ad-hoc networks, in particular, the development of a new routing algorithm for this type of networking.