Chapter 3

Evolution of Soft Computing Based Intrusion Detection System in MANETs

This chapter presents the problem of intrusion detection in MANETs and also introduces the techniques that form the background of our proposed solution. The soft computing techniques such as artificial neural network, fuzzy logic and neuro-fuzzy are explained in subsections 3.2.1, 3.2.2 and 3.2.3 respectively. The proposed intrusion detection solutions based on the soft computing techniques in the literature for MANETs are also discussed in subsection 3.2.5. A discussion is made on why soft computing is desirable for the development of an IDS in MANETs is described in subsection 3.2.6. The applications of M-FIS and S-FIS systems to intrusion detection and the evolution of neuro-fuzzy classifier based IDS are presented in section 3.3.

3.1 Threat Modeling in MANETs

MANETs routing needs new approaches because of its cooperative nature of routing protocols that have become targets of new attacks. For MANETs operations, collaborative routing is very essential so that we have determined to concentrate on the detection of attacks on those routing protocols that enforce collaborative routing. Particularly as an exemplar, we will concentrate on AODV [4] routing protocol in this research. The AODV basic operations and AODV attacks that we focus to detect are discussed in the subsequent sections.

3.1.1 Ad Hoc On-Demand Distance Vector (AODV)

Many routing protocols have been proposed for MANETs to fulfil its different needs. However, mostly routing protocols do not think about the security. One of the most popular routing protocol in MANETs is Ad Hoc On-Demand Distance Vector (AODV) routing protocol. This research is employed the Ad Hoc On-Demand Distance Vector (AODV) routing protocol. The AODV basic operations are now explained in respect to allow a good understanding of the routing attacks described subsequently.
AODV routing protocol comes under the category of reactive routing protocols. It discovers the routes when they are required. AODV routing protocol is able to adjust dynamic link conditions, handle the overheads of low processing and memory, low network employment and find out the unicast routes for the destinations in the ad hoc network [4]. It is arrogated that AODV can deal with the different mobility rates (low, medium and high) together with different data traffic rates. Moreover, AODV routing protocol does not consider any security provision.

AODV routing protocol is having three types of messages, i.e., Route Request (RREQ), Route Reply (RREP) and Route Error (RERR). When a node does not have a fresh route to communicate with another node, in this case a route discovery process starts from this node through broadcasting a RREQ message with Time to Live (TTL) field set to 1 in the network for the destination node. A node uses TTL field to wait for a specific time for receiving a RREP or RERR on the behalf of sent RREQ message, if TTL time expires, then the mobile node sends again a RREQ message with an incremented value of TTL field, after receiving the RREQ message, intermediate nodes send either a RREP message to the source node if they have a fresh route for the destination or forward the RREQ message to the other nodes in the network.

If the RREQ message has previously been forwarded by this intermediate node, then it drops the packet silently. Every node in AODV contains its destination sequence number to ensure the loop freedom property of all routes for that node. A fresh root is an unexpired root whose sequence number is greater or equal than that existed in the RREQ message. When the destination node gets a RREQ message for itself, then this node sends a RREP message through the reverse path and on behalf of the new route, nodes (requesting node and nodes receiving RREP messages) update their routing tables.

In AODV, Hello messages are used to maintain the local connectivity of nodes. After each Hello interval, a mobile node checks that it is heard from his neighbour, if it could not hear then a RREP Packet sends with TTL=1. This shows a Hello message and waits for a particular time and if it does not get a reply, then the node presume that link is lost to the neighbour. Due to the nodes mobility or transmission error, wireless ad hoc networks may have the frequent link breakage. In AODV, mobile nodes respond in case of link breakage and changes in the respect of time [4].
### 3.1.2. AODV Attacks

This subsection of the thesis explains the three attacks on AODV routing protocol that are given below. For analyzing our proposed approach, we have taken into account the detection of these attacks in this research.

#### 3.1.2.1 Packet Dropping Attack (PDA)

In the packet dropping attack scenario, malicious nodes drop all the incoming data packets for disrupting the network services [78]. For dropping the data packets, malicious nodes need to be a part of routing path so malicious nodes have little reason to drop control packets such as RREQ, RREP and RERR messages that are used in route discovery and route maintenance phases of AODV. In this research, control packets are not dropped by the malicious nodes.

Dropping data packets can prevent the end to end communication between the mobile nodes and also decrease the network performance due to the retransmission of data packets or discovery of new routes. During the simulation of PDA, malicious nodes continuously drop the data packets at each 1 sec interval.

Generally in wired networks, the packet losses happen due to congestion. In MANETs, due to its complex characteristics there are some other reasons such as congestion, mobility and wireless links transmission error to drop the packets. As in MANETs, mobility is the major cause to lose the data packets on AODV [79]. That’s why this research concentrates to differentiate the packet dropping due to the mobility from the packet dropping due to malicious nodes in MANETs.

#### 3.1.2.2 Sleep Deprivation Attack through Malicious RREQ Flooding (SDMF)

The sleep deprivation attack [80] is a kind of distributed DoS attack. During this attack, in the route discovery phase of AODV an attacker node broadcasts the route request (RREQ) packets to their neighbour with a destination IP address that exists in the network address range but in actually, it does not present in the network. Thus, all presented nodes in the network will compel to forward these route request (RREQ) packets because no one is having the route for this fake destination address in the network. The main aim of SDMF attack is to consume the battery power of nodes and discard them to perform the network operations.
3.1.2.3 Route Disruption Attack (RDA)

In RDA, attacker node sends RREP (route reply) packet to the neighbour node (that has selected as a victim node) without receiving of any RREQ packet from that victim node. AODV routing protocol does not have any mechanism for checking the flow of route request-reply messages even nodes can update their routing table based on the overhearing of the RREP message which contains fresh routes. Since, attacker node can get the information about the active routes of its neighbour nodes through the broadcasted routing control packets or promiscuous monitoring so that attacker selects one of their neighbour nodes as a victim node. Moreover, the attacker sends the fresh RREP message to this selected victim node and, based on this fresh RREP message victim node gets update his routing table. Thus, an attacker node is capable to interrupt the routing table entries of the victim’s routes. Sun et al. [17] stated that one or few control packets could impose the damage in the system. Here in the simulation, attacker node sends the 4-8 RREP packets to the victim node at each 1 sec interval.

In this research, soft computing techniques are applied for the detection of these attacks efficiently and effectively. These soft computing techniques are described in the subsequent sections.

3.2 Introduction of Soft Computing

Soft Computing is a group of methods for constructing the intelligent systems. Here, we mean ‘intelligent systems’ as those that are able to imitate the human reasoning process and also handle the qualitative and quantitative knowledge. It is well recognized that the intelligent systems can present the human like expertness i.e., uncertain reasoning, adjustment with time-varying and noisy environments, and tackle the practical computing problems. Soft computing is an inhibit candidate for developing these knowledge based systems. In recent years, soft computing has raised the growing interest of researchers from several scientific communities.

Fundamentally, soft computing is viewed as an emerging approach for computing which presents the notable potentiality of the human mind to understand and learn in a situation of imprecision and uncertainty. In respect of the fuzzy logic, Dr. Zadeh, has commented that the leading principle of soft computing is to effort the tolerance for uncertainty, partial truth to achieve tractability, imprecision, better rapport with reality and
low solution cost [81]. Furthermore, soft computing methods deal with uncertainty, imprecision, low solution cost and also helps at acquiring competitive solutions, whereas hard computing methods only deal with certainty, precision and rigor. In abruptly, as of its unequalled nature in dealing with the real world problems, e.g., optimization, intelligent control and nonlinear programming, soft computing has shown the increasing research attention of researchers from different backgrounds [82].

Generally, soft computing techniques comprise of three requisite paradigms such as neural networks [83], fuzzy logic [84] and Genetic algorithms [85]. However, soft computing is receptive rather than conservative conceptions. It is acquiring those applicable techniques together with the significant progresses in other new computational methods i.e., artificial immune systems [86]. In the trinity of soft computing: neural networks are related with nonlinear function approximation, universal generalization and adaptive learning; fuzzy logic with approximate reasoning, uncertainty and imprecision; and genetic algorithms are concerned with propagation of belief. Table 3.1 presents these three soft computing techniques together with their strengths.

<table>
<thead>
<tr>
<th>Table 3.1: Soft computing components</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Techniques</strong></td>
</tr>
<tr>
<td>Neural Networks</td>
</tr>
<tr>
<td>Fuzzy Logic</td>
</tr>
<tr>
<td>Genetic Algorithms</td>
</tr>
</tbody>
</table>

Soft computing techniques are complementary to each other rather than competitive. More incisively, it is advantageous to apply soft computing techniques i.e. neural networks, fuzzy logic and genetic algorithms in respect of combination rather than exclusively. For supporting this argument, a distinctive example is fuzzy-neural model that takes the capabilities of both fuzzy logic and neural networks [87]. The fuzzy-neural model is built to combine the fuzzy inference mechanism and neural networks into an integrated system so that the individual weaknesses of fuzzy logic and neural network are defeated.

The fuzzy-neural model holds the approximate inference features and imprecise information treating capability of fuzzy logic, in the mean time it has the strength of
generalization and adaptation by using the learning algorithms of neural networks. It has applied in many applications such as speech recognition [88] and image processing [89]. However, the neural-fuzzy model have the same procedure with the feed forward neural network, i.e., nodes and layers where node functions internally are substituted with the fuzzy membership functions and commonly utilized back-propagation learning algorithm is employed to adjust these fuzzy membership functions parameters. Many consumer products such as air conditioner and washing machine are produced with the embedded neuro-fuzzy model.

Many hybrid systems of neural networks, fuzzy logic and genetic algorithms are already available [90]. Figure 3.1 presents the hybrid of soft computing techniques such as neuro-fuzzy, fuzzy-genetic and genetic-neural. These hybrid systems take the advantage of both the exclusive techniques.

![Diagram of Soft Computing Framework](image-url)

**Figure 3.1: Soft computing framework**

In this thesis, we are concentrating to develop neuro-fuzzy classifier based IDS for MANETs so that we focus only on neural networks and fuzzy logic techniques. The detailed description of neural networks, fuzzy logic and neuro-fuzzy is given in the subsequent subsections.
3.2.1 Neural Networks

Artificial neural networks [91, 92] are usually mentioned as ‘neural networks’. The neural networks were first examined desirably to understand and imitate the human brain functions [93]. A human mind comprises a tremendous amount of nerve cells, i.e., neurons. Every cell is connected with another similar cells and creating a really complex network for signal transmitting. As a similar structure of the tree passes from every neuron cell that consents the inputs from other neurons. They are known as dendrites. The conveyor of outputs from one neuron to the dendrites in other neurons is known as an axon. In simple form, neural network can be viewed as a directed graph which has nodes and weighted connections amongst the nodes [83]. In respect of the biological neuron system, the cell body looks like a node and connections between nodes are identical to the synapses (the interconnection amongst the neurons) and connection weights are similar to the synaptic efficiency.

Moreover, the classification of networks is based on factors such as: how these connections are set up, how the connection weights are changed for minimizing some error evaluate on the response of the network to a stimulus. Few networks such as Hopfield net or Hamming net [83] are having the fixed weights. Furthermore, most of the other networks are updated with their connection weights in the training phase. The procedure for minimizing the error measure through the weight update is called the learning. Learning could be supervised or un-supervised. Those networks have the capability to learn on their own, then these are called the un-supervised networks. Otherwise, supervised networks need a training set with possible inputs and corresponding outputs for learning. Examples of unsupervised networks include Self organizing feature maps, ART, Neocognitron etc. while ADALINE, Perceptron and Multiple Perceptrons etc. are the examples of supervised networks [83, 94].

Thus, neural networks are built as simplified mathematical models, which are able to match the organizational principles of the brain of human on the behalf of microscopic biological models. By accepting advantage of these principles, neural networks are believed as a form of new propagation intelligent information processing systems [95]. They have the good potentiality of learning from the training data set, generalizing to the unobserved patterns and retrieving memorized information. These capabilities present the great use of neural networks in many applications such as pattern recognition [96] and signal processing [97].
There are entirely more than one hundred algorithms and structures suggested by people in respect of changing viewpoints [98]. Therefore, the widely employed neural networks are very limited to merely a few. Back-Propagation neural network is one of the very powerful neural network, which is also known as feed-forward or multi-layer perceptron network. It is a significant class of neural networks because of its powerful approximation potentiality and simple topology. It is very well known that the simple perceptron can just solve the linearly independent or linearly separable problems [99].

For the requirement to solve the linearly non-separable problems, neural networks should have allowed an intermediate representation of input patterns by presenting the nonlinear hidden layers. The structure of back propagation neural network is motivated by this idea. A basic structure of back propagation neural network with a single hidden layer is presented in Figure 3.2, although back propagation neural network can have more numbers of hidden layers.

![Figure 3.2: Back propagation neural network with a single hidden layer](image)

The design of the back-propagation learning algorithm for the back propagation neural network is the turning point in the development history of neural networks [100]. The
back-propagation learning algorithm is a very popular algorithm that is proposed by Rosenblatt in 1961 and was modified in 1974 by Werbos [101].

Here, we derive the back propagation learning algorithm based on the back propagation neural network with only single hidden layer [102]. In Figure 3.2, let \( v_{qj} \) show the weight connecting node \( j \) in the input layer with node \( q \) in the hidden layer and, \( w_{iq} \) show the weight that connects \( q \) in the hidden with node \( i \) in the output layer. There are nodes, i.e. \( m, l \) and \( n \) that exist in the input layer, hidden layer and output layer. Suppose \( (x, d) \) presents a pair of training samples, where \( x = \{x_1, x_2, ..., x_m\} \) is input patterns and \( d = \{d_1, d_2, ..., d_n\} \) is desired output. Node \( q \) in the hidden layer obtains the input as:

\[
net_q = \sum_{j=1}^{m} v_{qj} x_j
\]  

(3.1)

It devotes the transformed output as:

\[
z_q = a( net_q ) = a \left( \sum_{j=1}^{m} v_{qj} x_j \right)
\]  

(3.2)

\( a(.) \) is referred as the node activation function of the back propagation neural network. We can calculate the node \( i \) input in the output layer:

\[
net_i = \sum_{q=1}^{l} w_{iq} z_q = \sum_{q=1}^{l} w_{iq} a \left( \sum_{j=1}^{m} v_{qj} x_j \right)
\]  

(3.3)

However, the corresponding output is:

\[
y_i = a( net_i ) = a \left( \sum_{q=1}^{l} w_{iq} z_q \right) = a \left[ \sum_{q=1}^{l} w_{iq} a \left( \sum_{j=1}^{m} v_{qj} x_j \right) \right]
\]  

(3.4)

The above mentioned equations present the feedforward propagation process from the input to output of back propagation neural network. The basic back propagation learning algorithm is based on these above mentioned equations. The detailed description of learning algorithms and structures is given in [98].
3.2.2 Fuzzy Logic

Some aspects of real world problems cannot be effectively dealt under the framework of exact mathematical models or precise models, and could be solved by using the fuzzy logic [103]. The fuzzy logic concept was first presented by Dr. Zadeh to deal with the uncertainty and complexity which is derived from human reasoning [104]. Fuzzy logic is a scheme for presenting the communication procedure and information processing in the natural human mind. Fuzzy logic is able to deal with the multivalued logic of fuzzy set theory within the range 0 to 1. Fuzzy logic based decisions are in the form of degrees instead of yes or no conditions. In these days, there are many applications of fuzzy logic available in the various fields such as pattern recognition [96], consumer electronic products [105], fault diagnosis [106], reliable engineering [107] and signal processing [97], while early applications of fuzzy logic admit the industrial process control [108]. Basically, a fuzzy logic system has four components and the configuration of these components is presented in Figure 3.3.

![Figure 3.3: The basic architecture of fuzzy inference system](image)

These four fuzzy components are a fuzzification interface, a fuzzy rule base (knowledge base), a fuzzy inference engine (decision making logic) and a defuzzification interface. Furthermore, the fuzzification interface executes the fuzzification function, which transforms the crisp (numeric) value to fuzzy value. Moreover, it converts the input data into linguistic values that are considered as labels of fuzzy sets. Fuzzy sets are specified in the relating universes of discourse. The fuzzy rule base (knowledge base) is specified in the form of “IF-THEN” rules in that the preconditions and consequents take linguistic variables. The rule base for a fuzzy logic system has collection of these fuzzy rules.

The fuzzy inference engine plays a significant role in the fuzzy logic system. The fuzzy inference engine is used to take the decisions on the basis of inputs and fuzzy rules.
Normally, four compositional operators are employed in the compositional rule of inference, they are Max product operation [109], Max-min operation [104], Max drastic product operation [110] and bounded product operation [110].

However, many fuzzy inference systems (FISs) are proposed in the literature [82] for playing the role of fuzzy inference engine. Here, we will discuss the models based on Mamdani fuzzy inference system [82, 111] and Sugeno fuzzy inference system [82, 112]; these are used later in the thesis. At First, two-input single output based Mamdani fuzzy inference system (M-FIS) proposed to map an input space to an output space which is presented in Figure 3.4.

![Figure 3.4: The Mamdani fuzzy inference system with min and max operators [82]](image)

Max and min as the select for T-norm and T-conorm operators are followed here, respectively. For more acquaintance of T-norm and T-conorm and M-FIS, readers may refer to [82]. M-FIS used defuzzification module that converts the fuzzy values to crisp values in terms of output. There is an example of M-FIS based on two rules that can be given as:

if x is A1 and y is B1 then z is C1,

if x is A2 and y is B2 then z is C2,

Where A1, A2 and B1, B2 are the membership functions of inputs for A and B fuzzy sets, and C is the fuzzy sets for output respectively [82].
Takagi, Sugeno, and Kang proposed an approach to generate fuzzy rules from input output dataset [82]. Two-input and single output based first order of the Sugeno fuzzy inference system (S-FIS) is presented in Figure 3.5. Sugeno model is more efficient towards computation and also more suitable with adaptive techniques. There is an example of a fuzzy rule in S-FIS has the form:

\[
\text{if } x \text{ is } A \text{ and } y \text{ is } B, \text{ then } z = f(x, y),
\]

![Figure 3.5: Fuzzy reasoning of Sugeno model [82]](image)

Where A and B are the inputs fuzzy sets, and \( z = f(x, y) \) is zero or first order polynomial function for crisp output respectively. Here, due to the time consuming procedure of defuzzification in the Mamdani fuzzy model is replaced by the procedure of weighted average.

The defuzzification strategy is a mapping from fuzzy actions that is presented over an output universe of discloser to a space of non-fuzzy actions. Since, the fuzzy inference engine output is generally a fuzzy set while in practical applications such as control engineering, require the crisp (numeric) actions to drive the external actuators. Hence, the main aimed of a defuzzification scheme is to produce a crisp action that presents the possibility distribution of an inferred fuzzy action.

Generally, the three defuzzification methods are the Centre of Area (COA), the Max Criterion and the Mean of Maximum (MOM). In these three methods, the Max Criterion method is very simple to the implementation point of view. The Max Criterion method gives the point where possibility distribution of the action achieves a maximum value. The COA method gives the centre of gravity of the possibility distribution of an action.

Furthermore, the MOM method yields the crisp action that presents the mean value of all local actions whose membership functions attain the maximum*. A comparison between

62
COA and MOM defuzzification methods has been presented in [113]. It was demonstrated that the COA method had the superior capableness during the simulation experiments.

### 3.2.3 Neuro-Fuzzy

So far, we have discussed two main soft computing techniques: neural networks and fuzzy logic. Both the techniques have their own advantages and disadvantages as mentioned above. For example, it is proved that back propagation neural network has very good non-linear function approximation capabilities [114] and, the ‘black box’ type data processing structure and slow convergence speed are the main drawbacks of the back propagation neural network. On the other side, fuzzy logic has the same inference mechanism to the human brain, but it lacks of learning capability.

Therefore, it is hard to tune the membership function and fuzzy rules in a fuzzy logic system on the basis of the training data set. Precisely, neural network is considered as a model of free numeric methods, and fuzzy logic treats the rules and inference on the linguistic level. Thus, it is natural to combine the neural networks and fuzzy logic for making a hybrid system to overcome the individual drawbacks of both the methods. This combination is presented in the Figure 3.6 that is recognized as “neuro-fuzzy systems”.

![Figure 3.6: Combination of neural network and fuzzy logic as a “neuro-fuzzy systems”](image)

There are many possibilities for the fusion of both the methods i.e., neural networks and fuzzy logic [115]. The hybrid of neural and fuzzy is very popular in many applications so that Jang developed a very popular method, which is called ---- Adaptive Neuro Fuzzy Inference System (ANFIS) [116]. The reasoning mechanism of Sugeno model can be applied into a feed forward neural network with the capability of supervised learning, leading so called ANFIS architecture, which is given in Figure 3.7.
In terms of the topology point of view, the ANFIS is an implementation of a representative fuzzy inference system using back propagation neural network similar structure. The ANFIS architecture consists of five layers. Layers 1, 2, and 3 deal with the antecedent part of fuzzy system rules and layer 4 is responsible for the consequent part of it.

![Figure 3.7: Structure of the ANFIS](image)

\[ X = \{x_1, x_2, \ldots, x_p\} \]
The purpose of every layer is briefly described as follows: suppose $O_i^1$ refers the output of node $i$ in layer 1, and $x_i$ is the $i$th input of ANFIS, where $i = 1, 2, \ldots, p$. In layer 1, a node function $F$ related with each node:

$$O_i^1 = F_i (x_i)$$  \hspace{1cm} (3.5)

The use of the node functions $F_1, F_2, \ldots, F_t$, is equal to that of the membership functions $\mu (x)$ employed in the fuzzy systems, and $t$ is the number of nodes for every input. Here, Gaussian shape of membership functions is preferable choices. The adjustable parameters that define the places and shapes of these node functions are called premise parameters.

In layer 2, the output of each node is the product of all incoming signals. The output of every node presents the firing strength of the reasoning rule.

$$O_i^2 = F_i (x_i) \text{ AND } F_j (x_j)$$  \hspace{1cm} (3.6)

In layer 3, this layer is known as the rules layer. It calculates the ratio of $i$th rule firing strength to the sum of all rules firing strengths for generating the normalized firing strengths as:

$$O_i^3 = \frac{O_i^2}{\sum_l O_l^2}$$  \hspace{1cm} (3.7)

In layer 4, applies S-FIS, whereas a linear aggregation of the ANFIS input variables, $x_1, x_2, \ldots, x_p$, with the addition of a constant term $c_1, c_2, \ldots, c_p$, make the output of every IF-THEN rule. The node output is a weighted sum of these intermediate outputs.

$$O_i^4 = O_i^3 \sum_{j=1}^{p} (P_j x_j + c_j)$$  \hspace{1cm} (3.8)

The parameters $P_1, P_2, \ldots, P_p$, and $c_1, c_2, \ldots, c_p$, are denoted to as the consequent parameters. The overall output based on the sum of its inputs is calculated in layer 5.

$$O^5 = \sum_i O_i^4$$  \hspace{1cm} (3.9)

From the above discussion, it is concluded that ANFIS suggests a method for fuzzy modelling procedure in respect to learn information from a dataset to compute the membership function parameters that allow the related fuzzy inference system to track in the best way of given input/output data. The parameters related to membership functions will
change according to the learning process. ANFIS can use either back propagation or a combination of back propagation algorithm and least square estimation [116] for the estimation of membership function parameters. Moreover, the hybrid learning algorithm is more efficient for ANFIS [82], and ANFIS learning works as similar to the neural networks.

For the detail description of the ANFIS and its applications, readers may refer [82]. In this thesis, ANFIS is used as a binary classifier for developing a new intrusion detection system for MANETs.

3.2.4 Clustering Methods

Clustering is a mechanism to partition the dataset into the subsets (clusters) or more incisively, the classification of objects into the dissimilar groups so that the data of every group or subset carry out the same properties or features. Subsequently, clustering refers to the grouping of the same type of objects so that, there are two primary types of measures exist for determining whether two objects are same or different i.e., distance measures and similarity measures.

Most of the clustering methods apply the distance measures for finding the similarity or dissimilarity amongst any pair of objects. It is able to mention the distance between two objects \( o_i \) and \( o_j \) as: \( d(o_i, o_j) \). However, a similarity measure \( s(o_i, o_j) \) compares the two vectors \( o_i \) and \( o_j \) [117]. Clustering finds out the applications in many areas such as data mining, machine learning, image analysis, pattern recognition and system modelling. Furthermore, the clustering methods are divided into two main categories, i.e. hierarchical and partitioning methods [118].

Han et al. [119] categorise the clustering methods in three addition methods such as density based methods, model based clustering and grid based methods. Figure 3.8 presents the categorisation of all these methods. However, here we only discuss the fuzzy clustering algorithms (particularly “subtractive clustering”) that come under the category of model based clustering methods. For detail description of all the clustering methods, readers may refer [120].

In Sugeno type models, a polynomial function of the inputs can be presented as the consequent of a rule and the polynomial order specifies the order of the model. The optimum consequent parameters or coefficients of the polynomial function for a current set of clusters are determined by the Least Square Estimation method.
Subtractive clustering [120] is quick, one pass algorithm to calculate the number of clusters from a given set of data. This method is an extended version of the mountain clustering method that is proposed by Yager in [121]. The basic procedure of the subtractive clustering technique is given in the Figure 3.9.

![Clustering Methods Diagram]

**Figure 3.8: Categorisation of clustering methods**

Suppose a group of \( m \) data points \( \{x_1, \ldots, x_m\} \) in \( N \)-dimensional points. In subtractive clustering, first assumes all the data points are a potential cluster centre and then measure the potential for each data point of clusters depends on density of surrounding data points. The density of data point \( x_i \) can be measured as follows:

\[
D_j = \sum_{i=0}^{m} \exp\left( -\frac{(x_j - x_i)^2}{(r_a^2/2)^2} \right)
\]  

(3.10)

Where \( r_a \) is constant and defines the neighbourhood radius. For the first cluster centre, subtractive algorithm selects the higher density based data point and then destroys the nearest potential data points of first cluster centre. For the next cluster centre, the algorithm selects the remaining highest density based data points. The process of evolving a new cluster centre and destroying the nearest potential surrounding points of selected cluster centre, repeats this
procedure until the potential of all data points fall down a threshold. The influence range of the cluster centre in all data dimensions is known as the cluster radius. A small cluster radius is responsible to find the many clusters in data.

Figure 3.9: Shows the procedure of the subtractive clustering technique

According to the cluster information that is received from this algorithm, it is used to find out the initial number of rules and membership functions that are responsible to generate
the fuzzy inference system. In this work, the FIS structure of the proposed neuro-fuzzy classifier based IDS is obtained by the help of subtractive clustering to cover the whole features space.

For the generation of FIS, grid portioning can be used, but the generation of fuzzy rules in the grid portioning method depends on the membership functions of every input. Furthermore, if there are three membership functions selected for every input within two dimensional spaces, then the results in terms of fuzzy rules will be 9 rules, this grid portioning method requires only a small number of MFs for every input and it finds problems to moderate the large number of inputs [120] so that, this thesis has used the subtractive clustering to find out the initial number of fuzzy rules, membership functions and then ANFIS are used for further fine tuning of functions.

3.2.5 Related Work

Soft computing is a partnership of different methods, and the main purpose of soft computing is to construct an intelligent system. The inclusion of soft computing tools in the field of intrusion detection was first reported by Garcia and Copeland [122] in the year 2000. Many soft computing based IDSs are employed in wired networks because of their generalized properties that can help to detect known and unknown intrusions or even those attacks that have no prior described patterns [123].

Abraham et al. [124] illustrated the use of soft computing techniques for building the IDS and applied it to the KDD-99 data set [125]. Chet et al. [126] presented the detailed description of the proposed soft computing approaches based IDSs for the wired networks. Furthermore, one of the very commonly used hybrid approach of soft computing is “neuro-fuzzy” in the field of intrusion detection for the wired networks [124] [127] [128]. Neuro-fuzzy classifiers in the form of binary and multi-classifiers are applied on intrusion detection to classify the intrusive and normal activities in wired networks by Toosi et al. [129].

Recently, there are few proposed IDSs based on soft computing approaches in MANETs. First, support vector machine based approach applied in [130] by Deng et al. for intrusion detection in MANETs to emphasize on the security issues of network layer, but they have not provided any security mechanism for cluster heads. Some artificial neural network approaches used for MANETs in [65] [131] [132]. These proposed neural based approaches used limited features to detect all types of attacks.
Wahengbam et al. [16] suggested fuzzy logic based IDS for MANETs that is able to detect black hole and gray hole attacks according to the threshold values of each node, and some other fuzzy based IDSs have been reviewed in [133] for MANETs. In terms of hybrid approaches of soft computing, the proposed solutions in [134] [135] used ANFIS as neuro-fuzzy interface with limited features for the detection of specific attacks and these solutions were not used any clustering algorithm to automatically generate the initial fuzzy rules and antecedent membership functions without the need of human expert knowledge.

The output of these proposed IDSs [134] [135] is not clearly mentioned in the form of current data pattern which is either a normal pattern or an attack pattern. Sen et al. [136] proposed a grammatical evolutionary approach to intrusion detection on MANETs. They developed the detection programs using genetic programming and grammatical evolution for some specific attacks. This approach cannot detect new attacks.

To conclude, we develop a new intrusion detection system based on ANFIS as binary neuro-fuzzy classifier with subtracting clustering to automatically generate the initial fuzzy rules and membership functions for MANETs. The proposed system can also detect the new or unknown attacks in MANETs. In this subsection of the thesis, we have discussed the related work of soft computing techniques (particularly neuro-fuzzy systems) in the field of intrusion detection.

### 3.2.6 Why Soft Computing?

It is difficult to tell one intrusion detection technique is better than the others because all techniques of the intrusion detection have their advantages and disadvantages. For effective intrusion detection, different detection techniques are frequently applied in conventional networks. However, the architecture of intrusion detection in MANETs is also proposed with combining of different techniques. Even though, a combination of specification based and anomaly based techniques with misuse based approaches are also proposed, subsequently specification based techniques are unable to detect the DoS attacks.

In terms of misuse based detection technique, a little research has been done for finding the known attacks signatures in respect of MANETs. In this research, we address this issue by using the soft computing techniques with subtracting clustering method to find out the rules automatically for intrusion detection and train the system for facing the unknown and new attacks.
MANETs are a new kind of distributed network whose characteristics are very complex in nature. In the dynamic environment of MANETs, making the differentiation between normal and abnormal behaviour of activities is very hard. Furthermore, the resource constraint nodes need the different tradeoff to be developed between the detection programs and their usage of resources. In this research, soft computing techniques are proposed to identify automatically complex characteristics of MANETs. Although, many artificial intelligent techniques have been employed for intrusion detection, soft computing is one of the very promising techniques.

However, our selected soft computing techniques are very flexible. Since, the neural network has the good potentiality of learning and generalizing to the unobserved patterns and with the help of fuzzy variables or linguistic terms, intrusion detection features can be viewed easily and the decision of normal and abnormal activity in the network is based on its fuzzy nature that can be identified the degree of maliciousness of a node instead of yes or no conditions.

The capabilities of both these techniques are the main motivation to use the hybrid of neural network and fuzzy logic in this research for detecting the intrusions in MANETs. Moreover, our proposed solution is able to detect the known and unknown attacks in very effective (detect intrusions with high true positive rates and low false positive rates) and efficient (in respect of resource constraint solution) ways. These characteristics make soft computing very attractive and significant for developing an intrusion detection system for MANETs.

3.3 Evolving Soft Computing based Intrusion Detection System

In this section, we give the detail how to use the soft computing techniques for developing an IDS in MANETs.

3.3.1 Selection of Features

“Features” are the prominent attributes which are used as an input to our proposed system. Features are supplying the data set to any evolved system for evaluation of the results. The selection of attributes is most important to develop the system fundamentals. Our system has consideration on the degree of expensiveness of features.
Table 3.2 presents our selected features that are maintained at each node in MANETs through the AODV routing protocol. Basically, here the main emphasis is to detect all types of attacks so that it allows to concentrate a rich set of features to prove the efficiency of developed IDS. The features are collected based on two categories i.e. mobility based features and packet based features. Mobility based features such as added neighbours and remove neighbours give the information about the reflection of mobility for each node. These features directly reflect the mobility of a node.

The physical movements can simply give the information about local mobility rather than the network mobility so that the features based on the physical movements of nodes (direction, velocity speed) are not considered in this research. Packet based features include the information regarding the frequency of the AODV routing protocol control packets for sent (RREQ), received (RREP) and forwarded at each time interval.

There are some particular features in Table 3.2 that have presented the signatures of some particular attacks, for example detection of the route disruption attack is based on the feature “average hop count”. Increments in the number of data packets that are not forwarded by next node may be signs of packet dropping attack. The unusual decrements in the consumed battery power of each node may be considered as the sign of sleep deprivation attack that happens through malicious flooding in the network.

The data is gathered based on the selected features periodically by each node. During the data collection, there is no need of communication between the nodes because all selected features are local to each node [136]. Moreover, the event driven approach can be used for collecting the data. This approach collects the data when an event occurs in the network such as routing control packets is received and the like.

The event driven approach is not good choice for collecting the data in this type of dynamic environment because routing protocols purpose to meet the network requirements continually in the MANETs (particularly in case of on-demand routing protocols), they could result in a lot of control packets in the network. That’s why in this research, we used a periodic approach for collecting the data based on the selected features from each node in every second.
Table 3.2: The list of selected features

<table>
<thead>
<tr>
<th>Features Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg_num_hops</td>
<td>average number of the hop counts of active routes by this node</td>
</tr>
<tr>
<td>num_routes</td>
<td>no. of routes added to the route cache</td>
</tr>
<tr>
<td>num_req_initd</td>
<td>no. of RREQ packets initiates by this node</td>
</tr>
<tr>
<td>num_req_receivd</td>
<td>no. of RREQ packets received to this node</td>
</tr>
<tr>
<td>num_req_recvd_asDest</td>
<td>no. of RREQ packets received as a destination for this node</td>
</tr>
<tr>
<td>num_rep_initd_asDest</td>
<td>no. of RREP packets initiated from the destination by this node</td>
</tr>
<tr>
<td>num_rep_initd_asIntermde</td>
<td>no. of RREP packets initiated from the an intermediate node</td>
</tr>
<tr>
<td>num_rep_fwd</td>
<td>no. of RREP packets forwarded by intermediate nodes</td>
</tr>
<tr>
<td>num_rep_recvd</td>
<td>no. of RREP packets received by this node</td>
</tr>
<tr>
<td>num_rep_recvd_asSrce</td>
<td>no. of RREP packets received as source by this node</td>
</tr>
<tr>
<td>num_err_initd</td>
<td>no. of RERR packets initiated as this node detect the link break</td>
</tr>
<tr>
<td>num_err_fwd</td>
<td>no. of RERR packets forwarded by this node</td>
</tr>
<tr>
<td>num_err_recvd</td>
<td>no. of RERR packets received by this node</td>
</tr>
<tr>
<td>num_dataPkts_Initd</td>
<td>no. of data packets sent as source of the data by this node</td>
</tr>
<tr>
<td>num_dataPkts_fwd</td>
<td>no. of data packets forwarded by this node</td>
</tr>
<tr>
<td>num_dataPkts_recvd</td>
<td>no. of data packets sent as destination of the data by this node</td>
</tr>
<tr>
<td>num_brknLinks</td>
<td>total no. of broken links</td>
</tr>
<tr>
<td>consumed_battery</td>
<td>calculates the consumed battery to perform any operation by this node</td>
</tr>
<tr>
<td>dropped_datapkts</td>
<td>calculates not forwarded data packets by this next node</td>
</tr>
<tr>
<td>num_nbrs</td>
<td>no. of neighbours of node during simulation time</td>
</tr>
<tr>
<td>num_addNbrs</td>
<td>no. of added neighbours of node during simulation time</td>
</tr>
<tr>
<td>num_rmveNbrs</td>
<td>no. of removed neighbours of node during simulation time</td>
</tr>
</tbody>
</table>
3.3.2 Applications of M-FIS and S-FIS to Intrusion Detection in MANETs

Our main motivation towards the use of these Mamdani, and Sugeno fuzzy inference systems to provide a framework for using these systems in the field of intrusion detection. Precisely, how these inference systems can help in detecting the attacks in MANETs.

Fuzzy logic can deal with uncertainty and impreciseness that is derived from human reasoning or approximate reasoning. It can handle multi valued logic of fuzzy set theory within the range 0 to 1 and also the decisions are based in the form of degrees instead yes and no conditions [16]. IF-THEN based fuzzy rules are used to define all situations in the network for identifying the attacks or intrusions. The fuzzy rule based system is known as fuzzy inference system (FIS) that is responsible to take decisions. For this work, the description of both the fuzzy inference systems (i.e. M-FIS and S-FIS) is presented in the subsection 3.2.2. The common architecture of both the proposed M-FIS and S-FIS based intrusion detection systems is given in the Figure 3.10. It consists four modules such as: Extraction of fuzzy based parameters, Fuzzy inference module, Fuzzy decision module and response module. Both the proposed M-FIS and S-FIS based IDSs have the common architecture so that we define the all modules parallelly for both the M-FIS and S-FIS based IDSs. For analysing the performance, we applied the both M-FIS and S-FIS based IDSs for the detection of packet dropping attack (PDA) and sleep deprivation attack through malicious RREQ flooding (SDMF).

![Figure 3.10](image-url)

Figure 3.10: Presents the common architecture of the proposed IDSs using M-FIS and S-FIS
For the extraction of fuzzy based parameters, proposed system extracts the requisite parameters from the network traffic module and passes these parameters to the analysis in the fuzzy inference system (here, it can be either M-FIS or S-FIS). Various fuzzy rules and membership functions are applied on these parameters to calculate the verity level of the nodes in fuzzy inference module then for checking the behaviour of a node, the verity level of a node is compared to the threshold value (which is pre-defined). If the value of verity level is less than the threshold value, it shows that the particular node behaviour is malicious so that an alarm activates with this particular node IP address from the response module in the network. The detail description of all the modules is given below:

### 3.3.2.1 Extraction of Fuzzy Based Parameters

The fuzzy based input parameters for fuzzy inference system are extracted by listening of traffic to its direct neighbour nodes. For this reason node $j$ creates a neighbour table for its each neighbour node to contain the parameters list. All mobile nodes do work in promiscuous mode in MANETs i.e. each node can hear the traffic to its immediate neighbour nodes and collects the input parameters for fuzzy inference system. In this work for detection the PDA, node $j$ neighbour table considers the following parameters for its direct neighbour node $k$: Data Packet Forwarded Ratio, Average Data Packet Dropped rate and Verity Level. Subsequently, the detection of SDMF, node $j$ neighbour table considers the following parameters for its direct neighbour node $k$ such as Number of received route request packet, Average number of request packet received as destination and Verity Level.

**Data Packet Forwarded Ratio:** This parameter value calculates on the basis of how many numbers of data packets forwarded through intermediate nodes in a route that are observed by its direct neighbour nodes which work in promiscuous mode i.e. If node $k$ is an intermediate node in a route and $j$ is the direct neighbour of $k$ so that node $j$ can listen all the traffic which is passed through node $k$ and easily calculates the forwarded data packet ratio from its neighbour table.

**Average Data Packet Dropped Rate:** To evaluate the value of this parameter, each node calculates the difference between the number of data packets received and number of data packets forwarded through an intermediate node in a route which is determined from its direct neighbour nodes table. In case of the legitimate node, the calculated value could not be higher because the number of received packets must be forwarded to the other nodes.
according to the process of AODV routing protocol which is considered in this work. So if a malicious node present in the network, then the value of dropped rate should be high because malicious node (s) prevents to forward the data packets and finally drops these data packets. This research work calculates the average data packet dropped rate of each intermediate node in a route by its neighbour nodes table.

*Number of received route request packet:* This parameter value calculates on the basis of how many numbers of route request packets are received for the same destination address to a node during the discovery of a new route and that are observed by its direct neighbour nodes which work in promiscuous mode i.e. If node j is a direct neighbour of k then node j can listen all the route request packets of node k that are received for same destination address in the discovery phase of a new route and node j can easily calculate the received route request data packets from its neighbour table that are received for the same destination again and again.

*Average number of request packet received as destination:* To evaluate the value of this parameter, each node calculates the value of how many numbers of route request packets got the final destination address during the process of route discovery by its neighbour nodes. This paper considered the process of AODV routing protocol so that if a malicious node present in the network then the value of this parameter i.e. number of request packet received as destination should be low because during the flooding through the malicious node(s), very few number of route request packets can get the final destination address. This research work calculates the average number of request packet received as destination of each node in a new route discovery phase by its neighbour nodes table.

### 3.3.2.2 Fuzzy Inference System

In this research work, fuzzy inference system module uses either M-FIS or S-FIS interchangeably to analyze the behaviour of the system of a node because both the proposed M-FIS and S-FIS based IDSs have the common architecture (i.e. mentioned in Figure 3.10) so that we define the fuzzy inference module parallelly for both the M-FIS and S-FIS based IDSs.

For the detection of PDA, data packet forwarded ratio and average data packet dropped rate are used as an input parameters and verity level parameter is used as an output parameter and for the detection of SDMF, number of initiated route request packet ratio,
average number of request packet received as destination are used as an input parameters and has only one output parameter i.e. verity level.

Basically verity level parameter checks the behaviour of a node i.e. “normal or malicious” on the basis of input parameters. The MATLAB fuzzy logic toolbox is used for the implementations of M-FIS and S-FIS systems during the experiments. The rule bases of proposed systems are given in Tables 3.3 & 3.4, where Table 3.3 presents the fuzzy rule base for PDA and Table 3.4 shows the rule base for SDMF. For getting the optimum performance, membership functions are selected for the input and output parameters for PDA and SDMF attacks which are presented in Figures 3.11 & 3.12. In fuzzy inference systems (M-FIS and S-FIS), first rule of the rule base is translated as follows:

First rule for PDA: If Data Packet Forwarded Ratio is High and Average Data Packet Dropped Rate is Low then Verity level is High

First rule for SDMF: If No. of received route request packet ratio is High and Average number of request packet received as destination is Low then Verity level is Low.

Accordingly, the \( i^{th} \) rule of rule base interprets as either M-FIS or S-FIS. The verity level of each node is calculated to its direct neighbour nodes table on behalf of the input parameters membership functions. The value of verity level lies between 0 to10. As a result, with the low value of verity level shows the more malicious behaviour of a node so that the value of verity level 0 presents the particular node behaviour entirely malicious and 10 indicates the completely normal behaviour of a particular node.

3.3.2.3  Fuzzy Decision Module

In this module, a threshold of verity level is already set to compare with the calculated value of verity level for analyzing the behaviour of a node in MANETs. For this work, the threshold value set is 5.5. During the comparison, if the calculated value of verity level is greater than the threshold value, the node is not malicious otherwise it is.

3.3.2.4  Response Module

Based on the results of the fuzzy inference system, if any node is found malicious node in the network, then this module is responsible to initiate alarm with the IP address of this particular malicious node to isolate it from the network.
Table 3.3: Rule base for detecting PDA

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Data Packet Forwarded Ratio</th>
<th>Average Data Packet Dropped Rate</th>
<th>Verity Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.</td>
<td>High</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>II.</td>
<td>High</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>III.</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>IV.</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>V.</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>VI.</td>
<td>Medium</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>VII.</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>VIII.</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>IX.</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
</tr>
</tbody>
</table>

Table 3.4: Rule base for detecting SDMF

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Number of received route request packet ratio</th>
<th>Average number of request packet received as destination</th>
<th>Verity Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>II.</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>III.</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>IV.</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>V.</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>VI.</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>VII.</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>VIII.</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>IX.</td>
<td>Low</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

For the performance evaluation of intrusion detection system, some standard matrices were developed, i.e. true positive rate and false positive rate [38]. True positive rate is calculated as the ratio of the number of truly detected attacks and the total number of attacks presented in the network, although the false positive rate can be calculated as the ratio of the number of legitimate nodes detected as attacks and the total number of the legitimate nodes. The selected input parameters are very pragmatic for detecting a particular type of attack.
Our main motivation to develop the M-FIS and S-FIS based intrusion detection systems for just showing that how an inference system makes the decisions about the attack(s) in the network. The performance of these systems is analysed in the subsequent chapters. Furthermore, our main objective in this thesis is to develop a generalized IDS using neuro-fuzzy classifiers for detecting the known and unknown (new) attacks in MANETs so that the next subsection of this chapter introduces the neuro-fuzzy classifier to intrusion detection in MANETs.

Figure 3.11: Represents membership functions for detecting the PDA

Figure 3.12: Represents membership functions for detecting the SDMF
3.3.3 Evolution of Neuro-Fuzzy Classifier to Intrusion Detection in MANETs

The main motivation towards this proposed work to provide a framework for using hybrid of soft computing techniques i.e. neuro-fuzzy to build the binary classifier that can act better than the single soft computing technique. In this subsection, we give the detail how to apply the neuro-fuzzy classifier to derive an IDS for MANETs. A complete workflow of our proposed neuro-fuzzy classifier based IDS is depicted in Figure 3.13.

![Figure 3.13: Work flow for developing our proposed IDS](image-url)

- Selection of features for data collection
- Separation of local, and distributed and cooperative based features
- Extraction of data based on selected features by using QualNet simulator 6.1
- Format the data for rendering it into the MATLAB Toolbox or Format as per our detection scheme
- Apply the subtractive clustering technique on data set for deciding the membership functions
- Apply the neuro-fuzzy classifier for classifying the normal and abnormal activities in terms of 0 & 1
- Utilized the Mamdani FIS for identifying the type of attack
- Response based on malicious activity
This section explains the proposed architecture of binary neuro-fuzzy classifier based IDS for detecting the PDA and SDMF attacks in MANETs. The proposed architecture includes the three modules such as data source module, detection module and the response module. Figure 3.14 presents the proposed neuro-fuzzy classifier based architecture for MANETs.

![Block diagram of proposed system architecture](image-url)

**Figure 3.14: Block diagram of proposed system architecture**

In terms of brief description of each module, the data source module consists the dataset that is based on the selected features mentioned in Table 3.2, and it is also used as the input for our binary neuro-fuzzy classifier based IDS. Neuro-fuzzy classifier works as a detection module for detecting the intrusions or attacks and, if any kind of suspicious activity presents in the network then raises the alarm through the response module.

Here, a new IDS designed based on neuro-fuzzy classifier (which is in binary form) to distinguish the normal and abnormal activities in MANETs. For this purpose, ANFIS is used as a binary neuro-fuzzy classifier with subtracting clustering to automatically generate the initial fuzzy rules and membership functions. In this research work, the main motivation is to use ANFIS in binary form because ANFIS is generally more suitable as a binary classifier instead of multi-classifier [137].
Form the point of view binary classifier, patterns are labeled with 0 and 1 where 0 for normal data patterns and 1 for attack data patterns in MANETs. We have applied our developed neuro-fuzzy classifier based IDS for detecting the PDA and SDMF attacks, separately. Accordingly, we have trained our proposed neuro-fuzzy classifier based IDS in respect of each attack. For training and checking, we have used normal and abnormal data of PDA and SDMF attacks. For testing of the trained IDS (in respect of PDA), we have used normal and known attack data i.e. PDA. Similarly, for testing of the trained IDS (in respect of SDMF), we have applied normal and known attack data i.e. SDMF. This research used the subtractive clustering approach with neighbourhood radius $r_a = 0.5$ to partition the training dataset and constructs the automatic initial fuzzy rules to form the structure of FIS for the training of the ANFIS. The description of subtractive clustering has been explained in the subsection 3.2.4.

In case of PDA, seven fuzzy rules and seven membership functions (Gaussian type) were obtained for each input and in respect of SDMF attack, obtained the eight fuzzy rules and eight membership functions (Gaussian type) for each input. Further fine tuning and adaptation of membership functions are done by using ANFIS where PDA and SDMF training datasets are applied to train the ANFIS in respect of both the attacks separately and checking datasets are used to check the validation of the model because after a certain time in the training phase, the model starts over fitting with training dataset. If over fitting occurs, then the FIS may behave biased with other independent data sets. Figures 3.15 & 3.16 show the initial and final membership functions of some input features during ANFIS training phase in respect of PDA and SDMF attacks. The ANFIS model architecture used in this research has been explained in the subsection 3.2.3. The ANFIS related implementation is done by using MATLAB.

In respect of the PDA, the used ANFIS contains 272 nodes and total number of fitting parameters is 364, in which premise parameters are 238 and consequent parameters are 126 and in case of SDMF, the used ANFIS contains 308 nodes and total numbers of fitting parameters are 412, in which premise parameters are 268 and consequent parameters are 144. After learning of 50 epochs, the root mean square error (RMSE) of training and checking datasets are $0.34157$ and $0.38461$ for PDA and Figure 3.17 presents the RMSE error measures as a function based on epochs during the training phase for PDA.
In respect of SDMF attack, the RMSE for training is 0.3368 and checking is 0.3618 after 50 epochs of learning. ANFIS architecture gives only one output that has been previously mentioned in the subsection 3.2.3 so that in this research the output of ANFIS architecture is specified by the class number of the Table 3.2 input feature vector, class number 0 denotes the normal patterns and 1 denotes the attack patterns.

![Membership function graphs](image)

**Figure 3.15:** (a), (c), (e) Show the initial membership functions i.e. before training and (b), (d), (f) final membership functions i.e. after training for PDA
Figure 3.16: (a), (c) Show the initial membership functions i.e. before training and (b), (d) final membership functions i.e. after training for SDMF

Figure 3.17: Represents RMSE error and epoch numbers during training
The output of ANFIS is not necessary to give an output in the form of an integer value as the class number i.e. 0 or 1 so that it needs the approximate number for a class i.e. 0 or 1 by using rounding out the given number. Here the parameter \( \mu \) is responsible for rounding out, that gives the integer value to us. If it is given 1 then the specified pattern is attack otherwise it should be a normal pattern. We will show the effect of \( \mu \) on the basis of performance metrics in the subsequent chapters. We have considered the true positive rate and the false positive rate as the performance metrics for analysing the performance of our neuro-fuzzy classifier based IDS in respect of PDA and SDMF attacks.

Moreover, the performance of our developed neuro-fuzzy classifier based IDS is demonstrated in the subsequent chapters. A simulated network under low, medium, high mobility and traffic patterns is used to analyse the performance of our proposed solutions. Somehow, the proposed neuro-fuzzy classifier works as an anomaly detector for new attacks. New or unknown attacks mean which are not presented in the training dataset.

### 3.4 Conclusions

This chapter has described the soft computing techniques such as neural network, fuzzy logic and neuro-fuzzy for forming the background of our proposed approach. Moreover, the proposed intrusion detection systems based on the soft computing techniques in the literature for MANETs have been discussed. The applications of M-FIS and S-FIS in the field of intrusion detection have also been described. Furthermore, a new intrusion detection system has been designed based on binary neuro-fuzzy classifier to distinguish the normal and abnormal activities in MANETs. For this purpose, ANFIS is used as a binary neuro-fuzzy classifier with subtracting clustering to automatically generate the initial fuzzy rules and membership functions. The performance results of our proposed solutions are demonstrated in the subsequent chapters of this thesis.