

A Secure Framework for Flying Ad-hoc Networks

A THESIS

In the Partial Fulfillment of the Requirement for the Degree of

DOCTOR OF PHILOSOPHY

By

Kuldeep Singh

(Registration Number: 901403006)

Under the Supervision of

Dr. Anil Kumar Verma

Professor, Computer Science and Engineering Department,

Thapar Institute of Engineering and Technology, Patiala-147004, INDIA



THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

FEBRUARY 2018

“You are educated when you have the ability to listen to almost anything without losing your temper or self-confidence”

— Robert Frost

“Imagination is more important than knowledge. Knowledge is limited. Imagination encircles the world”

— Albert Einstein

“There is no silver bullet solution with cyber security; a layered defense is the only viable defense”

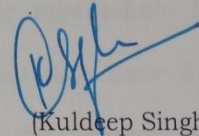
— James Scott

“Distrust and Caution are the parents of Security”

— Benjamin Franklin

CERTIFICATE

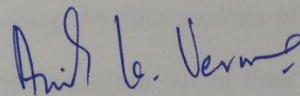
I, Kuldeep Singh, Regn. No. 901403006, hereby declare that the work which is being presented in this thesis entitled **“A Secure Framework for Flying Ad-hoc Networks”**, in partial fulfilment of the requirement for the award of degree of “Doctor of Philosophy” submitted in Computer Science and Engineering Department of Thapar Institute of Engineering and Technology, Patiala, India is an authentic record of my own work carried out under the supervision of Dr. Anil Kumar Verma. This report has not been submitted to any other institution for the award of any other degree.



(Kuldeep Singh)

Regn. No. 901403006

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.



(Dr. Anil Kumar Verma)

Professor, Computer Science & Engineering Department
Thapar Institute of Engineering and Technology
Patiala, Punjab, INDIA (147004)

ABSTRACT

The use of Unmanned Aerial Vehicles (UAVs) has significantly increased for forming an ad hoc network due to their astonishing working capabilities in the tactical as well as civilian areas such as armed attacks, border surveillance, disaster management, rescue operation and transportation. These UAVs aided ad hoc networks popularly known as Flying Ad hoc Networks (FANETs). A FANET consist a group of homogeneous or heterogeneous flying agents called Unmanned Aerial Vehicles (UAVs) connected by wireless links. UAVs are capable to communicate with each other in a group, and interact with their neighboring nodes to acquire valuable information. In FANETs, UAVs performs not only as a client, but also have the responsibility to perform as a router and server in the network. The large degree of freedom and self-organising capabilities makes FANETs totally different from any other network solution. FANETs can be rapidly deployed anytime in real-time for communication because they do not require any external resources. These unique features make FANETs a suitable solution for various applications. In FANETs, intermediate UAVs are responsible for forwarding the data packets of other network's UAVs and enable them to communicate out of their transmission range. In FANETs, UAVs are commonly constrained by limited energy and computation power. So, intermediate nodes can refuse to forward packets in order to preserve their resources, leads to disruption in network traffic. Nodes showing such behavior are known

as selfish nodes. Selfishness is generally a passive behavior of the node, and effect the overall performance of the network by degrading it. FANETs are more prone to attacks due to the self-organized, anonymous, distributive and independent behavior of nodes. Trust management can be one of the security solutions for FANETs because to accomplish the task/mission with cooperation and coordination, nodes need to have trust on each other. Trust management schemes/techniques allowed a node to assess trustworthiness of other network nodes. A trust management technique helps in the detection of malicious and selfish node behavior and also enhances the overall network performance by segregating them. Measurement parameters and computation approach to evaluate trust for secure communication among nodes is crucial for the development of trust management technique. Trust evaluation in FANETs environment involves several intricate aspects like - node behavior assessment in terms of reliability and performance, correct recommendation etc.

Motivated from the above, a fuzzy based trust model is proposed for FANETs. This model makes use of principles of fuzzy classification and optimization to manage trust relationships among nodes of FANETs. Trust is calculated based on the quality of service (QoS) (includes- reliability and performance parameters) and social parameter (recommendation). Social trust improves the trust evaluation process for those nodes which lacks in direct observation of their neighbor nodes due to some constraints such as hidden terminal problem, bad mounting attack, etc. There are various advantages of

trust management- malicious node can be identified and isolated, evading of data forwarding to selfish and malicious nodes and it helps to select a higher trust level node, resulting in improvement of overall performance of FANETs. In the proposed model, network nodes communicate only with its immediate neighbor nodes (hop count=1), based on the fact that only immediate neighbor node can give recommendation correctly. Decay function is also used to replicate the real-world behavior of the UAVs. The node keeps only its neighbor node's trust information which helps in reducing the processing time for trust evaluation, utilization of energy and memory.

*Dedicated with extreme affection and gratitude to
my beloved parents and family
for their endless love, support, encouragement and prayers*

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my supervisor, Dr. Anil Kumar Verma, for being a pillar of support and encouragement throughout my research work. His experience, strength, tenderness and willfulness, has taught me valuable lessons of life, which are going to be of immense help to me in taking decisions in going forward. My sincere thanks are due to Dr. Maninder Singh, Associate Professor and Head, Computer Science and Engineering Department (CSED), Thapar Institute of Engineering and Technology, Patiala for providing me the necessary administrative assistance in completion of the work. I am thankful to my Ph.D. committee members, Dr. Neeraj Kumar, Associate Professor, Computer Science and Engineering Department, Dr. Mahesh Kumar Sharma, Associate Professor, School of Mathematics for their constructive comments and regularly ensuring the progress of my research work. I am thankful to all the faculty and staff members of CSED for their support.

I offer my deepest gratitude to my loving parents, Mr. Rajbir Singh and Mrs. Jinesh Devi whose love and affectionate blessings have been a constant source of inspiration in making my vision a reality. I also acknowledge the cooperation and encouragement extended to me by my wife Ms. Palvi Aggarwal. I would also acknowledge my Father in Law and Mother in Law for their love, encouragement, motivation and confidence in me. I wish to thank all my friends and colleagues, Dr. Gaurav Sharma, JasminderKaur, and ShiwaniGarg. The financial

support from Ministry of Electronics and Information Technology, India (MEITY) is duly appreciated.

The chain of gratitude would be definitely incomplete without thanking the Almighty, the prime mover, for inspiring and guiding me (a humble being) to complete this task successfully.

Patiala

Kuldeep Singh

February, 2018

TABLE OF CONTENTS

CERTIFICATE.....	i
ABSTRACT	ii
ACKNOWLEDGEMENT	vi
TABLE OF CONTENTS.....	viii
LIST OF FIGURES.....	xii
LIST OF TABLES	xiv
LIST OF ALGORITHMS.....	xv
IMPORTANT ABBREVIATIONS.....	xvi
LIST OF PUBLICATIONS.....	xvii
1. INTRODUCTION.....	1
1.1 BACKGROUND	1
1.1.1 FANET Architecture.....	5
1.1.2 FANET Applications.....	8
1.2 FANET SECURITY.....	9
1.2.1 Security Challenges	10
1.2.2 Security Requirement	12
1.3 ORGANISATION OF THE THESIS.....	14
2. LITERATURE SURVEY	17
2.1 POTENTIAL ATTACKS	17
2.2 AD-HOC NETWORKS SECURITY.....	21
2.2.1 Cryptographic Solutions	21
2.2.2 Trust-Based Solutions	22
2.3 TRUST IN AD-HOC NETWORKS	23
2.3.1 Concept of Trust	24
2.3.2 Trust Properties	25
2.3.3 Trust Metrics	27
2.3.4 Trust Management Process.....	28
2.3.5 Stages of Trust Management	31
2.4 TRUST COMPUTATION METHODS.....	37
2.4.1 Fuzzy Model	37
2.4.2 Bayesian Model.....	38
2.4.3 Game Theory Model.....	39
2.4.4 Subjective Trust.....	39

2.4.5	Weighted Based.....	40
2.4.6	Markov Chain Model.....	41
2.4.7	Beta Distribution.....	41
2.5	PROBLEM STATEMENT.....	45
2.6	OBJECTIVES.....	46
	CHAPTER SUMMARY.....	46
3.	THREAT MODELLING AND RISK ANALYSIS	48
3.1	FANET THREATS	48
3.1.1	Threats to Physical Node	50
3.1.2	Threats to Network	51
3.1.3	Threats to Human Resource.....	53
3.1.4	Threats to Security Policy.....	55
3.1.5	Threats to Infrastructure	55
3.2	RISK ANALYSIS	58
3.2.1	Impact on Confidentiality	59
3.2.2	Impact on Integrity	60
3.2.3	Impact on Availability	60
3.2.4	Easiness	61
3.3	ATTACK RANKING.....	63
3.4	DEFENCE MECHANISM.....	66
3.4.1	Defence for Physical Node	67
3.4.2	Defence for Network	67
3.4.3	Defence for Human Resource.....	68
3.4.4	Defence for Policy	68
3.4.5	The Defence for Infrastructure.....	69
3.5	CHAPTER SUMMARY	69
4.	FCTM: FUZZY CLASSIFICATION TRUST MODEL.....	70
4.1	PROPOSED TRUST MODEL.....	71
4.1.1	Model Description	71
4.1.2	Fuzzy Classification.....	73
4.1.3	Objective Function.....	74
4.1.4	Trust Calculation.....	76
4.2	IMPLEMENTATION.....	78
4.2.1	Key Attribute Selection.....	78
4.2.2	Membership Functions.....	79
4.3	RESULTS	86
4.3.1	Classification Accuracy	87

4.3.2	Trust Evaluation based on QoS and Social Trust.....	89
4.3.3	Comparison of FCTM with Existing Models.....	92
4.4	CHAPTER SUMMARY	97
5.	WFCTM: WEIGHTED FUZZY CLASSIFICATION TRUST MODEL	98
5.1	THE PROPOSED TRUST MODEL	99
5.1.1	Fuzzy Classification.....	102
5.1.2	Trust Calculation.....	107
5.2	IMPLEMENTATION	111
5.2.1	Key Attribute Selection	111
5.2.2	Membership Functions.....	112
5.3	RESULTS	118
5.3.1	Classification Accuracy	119
5.3.2	Mean Square Error(MSE).....	120
5.3.3	Trust Levels	121
5.3.4	Trust Convergence	122
5.3.5	Energy Consumption	123
5.3.6	Comparison of the Proposed Model with Existing Trust Models.....	124
5.4	CHAPTER SUMMARY	128
6.	TBCS: TRUST BASED CLUSTERING SCHEME FOR FANETs.....	130
6.1	SYSTEM MODELLING	131
6.1.1	Network Model.....	131
6.1.2	Threat Model.....	132
6.1.3	Security Goals	133
6.2	THE PROPOSED TRUST MODEL	134
6.2.1	Fuzzy Classification and Optimization.....	135
6.2.2	Trust Calculation.....	137
6.2.3	Reward and Punishment Calculation	139
6.2.4	Cluster Head Selection	141
6.2.5	Node Join	142
6.3	IMPLEMENTATION AND RESULTS.....	143
6.3.1	Key Parameter Selection	144
6.3.2	Membership Functions.....	144
6.4	EXPERIMENTAL RESULTS.....	149
6.4.1	Classification Accuracy	151
6.4.2	Mean Square Error (MSE)	152
6.4.3	Convergence.....	152
6.4.4	Comparison of the Proposed Model with Existing Models	153

6.5	CHAPTER SUMMARY	159
7.	CONCLUSION AND FUTURE WORK	160
	ACKNOWLEDGMENTS.....	163
	REFERENCES.....	164

LIST OF FIGURES

Figure 1.1: <i>MANET and its Sub-Class Networks</i>	3
Figure 1.2: <i>FANET Architecture</i>	7
Figure 2.1: <i>Trust Properties</i>	25
Figure 2.2: <i>Trust Management Process</i>	29
Figure 2.3: <i>Stages in Trust Management</i>	36
Figure 3.1: <i>FANET Threats</i>	49
Figure 3.2: <i>Proposed Threat Model for FANETs</i>	59
Figure 3.3: <i>A 2-D Graph Representation of Difficulty Level versus Impact on Confidentiality of Different Attacks.</i>	64
Figure 3.4: <i>A 2-D Graph Representation of Difficulty Level versus Impact on Integrity of Different Attacks.</i>	65
Figure 3.5: <i>A 2-D Graph Representation of Difficulty Level versus Impact on Availability of Different Attacks.</i>	65
Figure 3.6: <i>Defence Mechanism against Various Attacks in FANETs.</i>	67
Figure 4.1: <i>Proposed Fuzzy Classification Trust Model (FCTM)</i>	72
Figure 4.2: <i>Classification Accuracy of FCTM</i>	87
Figure 4.3: <i>Trust Level with $\alpha = 0.5$ and $\beta=0.5$</i>	89
Figure 4.4: <i>Trust Level with $\alpha = 0.7$ and $\beta=0.3$</i>	90
Figure 4.5: <i>Trust Level with $\alpha = 0.3$ and $\beta=0.7$</i>	91
Figure 4.6: <i>Overtime Trust Evaluation for Bad Node</i>	92
Figure 4.7: <i>Overtime Trust Evaluation for Good Node</i>	92
Figure 4.8: <i>Packet Drop Ratio Vs. Node Velocity</i>	94
Figure 4.9: <i>Packet Drop Ratio Vs. Malicious Nodes</i>	94
Figure 4.10: <i>Throughput Vs. Malicious Nodes</i>	95
Figure 4.11: <i>Throughput Vs. Node Velocity</i>	96
Figure 5.1: <i>Proposed Weighted Fuzzy Classification Trust Model (WFCTM)</i>	100
Figure 5.2: <i>Flow Chart of Fuzzy Classification Trust Model</i>	110
Figure 5.3: <i>Fuzzy Membership Functions (a. Signal Strength, b. Packet Delivery Ratio, c. Node Energy, d. Transmission Delay)</i>	115
Figure 5.4: <i>Classification Accuracy of Proposed Model</i>	120
Figure 5.5: <i>Mean Square Error (MSE)</i>	121
Figure 5.6: <i>Trust Levels of Bad, Neutral and Good Node</i>	122
Figure 5.7: <i>Convergence of Node 1 and Node 2 to their Actual Behaviour</i>	123
Figure 5.8: <i>Average Energy Consumption vs Malicious Node's Percentage</i>	124
Figure 5.9: <i>Node Speed vs Transmission Delay</i>	126
Figure 5.10: <i>Number of Malicious Nodes vs Transmission Delay</i>	126

Figure 5.11: <i>Number of Malicious Nodes vs Packet Drop Ratio</i>	127
Figure 5.12: <i>Node Speed vs Packet Drop Ratio</i>	128
Figure 6.1: <i>FANET Network Model</i>	132
Figure 6.2: <i>Proposed Trust Model</i>	135
Figure 6.3: <i>Direct and Indirect Trust Calculation</i>	137
Figure 6.4: <i>Fuzzy Membership Functions</i>	149
Figure 6.5: <i>Classification Accuracy of Proposed Model</i>	151
Figure 6.6: <i>Mean Square Error</i>	152
Figure 6.7: <i>Convergence of Node 1 and Node 2 to their Actual Behaviour</i>	153
Figure 6.8: <i>Cluster Size vs Communication Overhead</i>	155
Figure 6.9: <i>Cluster Duration vs Packet Drop Ratio</i>	156
Figure 6.10: <i>Node Speed vs Packet Drop Ratio</i>	156
Figure 6.11: <i>Number of Malicious Nodes vs Packet Drop Ratio</i>	157
Figure 6.12: <i>Node speed vs. Transmission Delay</i>	158
Figure 6.13: <i>Number of Malicious Nodes vs Transmission Delay</i>	158
Figure 7.1: <i>Security Framework for FANETs</i>	161

LIST OF TABLES

Table 2.1: <i>Comparison of Various Trust Computation Methods</i>	42
Table 2.2: <i>Advantage and Disadvantages of Trust Computation Models</i>	44
Table 3.1: <i>Comparison of Various Threat Models</i>	58
Table 3.2: <i>Score based on the Difficulty Level to Perform an Attack</i>	62
Table 4.1: <i>Reward and Punishment</i>	76
Table 4.2: <i>Attribute Values of Network Nodes</i>	83
Table 4.3: <i>Rewards based on Node Classification</i>	85
Table 4.4: <i>Trust Calculation of a Node in FCTM</i>	85
Table 4.5: <i>Confusion Matrix for Network Nodes (N= 10)</i>	88
Table 4.6: <i>Confusion Matrix for Network Nodes (N= 25)</i>	88
Table 4.7: <i>Confusion Matrix for Network Nodes (N= 50)</i>	88
Table 4.8: <i>Weight Table for QoS and Social Trust</i>	89
Table 4.9: <i>Simulation Parameters</i>	93
Table 5.1: <i>Symbol Table of WFCTM</i>	101
Table 5.2: <i>Reward and Punishment</i>	108
Table 5.3: <i>Attribute Values of Network Nodes</i>	116
Table 5.4: <i>Rewards/Punishment based on Node Classification</i>	117
Table 5.5: <i>Parameters Weights</i>	118
Table 5.6: <i>Trust Calculation of a Node in WFCTM</i>	118
Table 5.7: <i>Simulation Parameters</i>	125
Table 6.1: <i>Symbol Table for TBCS</i>	134
Table 6.2: <i>Reward and Punishment</i>	140
Table 6.3: <i>Simulation Parameters</i>	154

LIST OF ALGORITHMS

Algorithm 4.1: <i>Fuzzy Classification for FCTM</i>	74
Algorithm 4.2: <i>Trust Calculation in FCTM</i>	77
Algorithm 5.1: <i>Weight Optimization and Node Clustering in WFCTM</i>	105
Algorithm 5.2: <i>Trust Calculation in WFCTM</i>	109
Algorithm 6.1: <i>Trust Calculation in TBCS: Cal_Trust(P)</i>	138
Algorithm 6.2: <i>Direct Trust Calculation in TBCS: Direct_Trust(P)</i>	138
Algorithm 6.3: <i>Indirect Trust Calculation in TBCS: Indirect_Trust(i)</i>	139
Algorithm 6.4: <i>Reward and Punishment in TBCS: Payoff (i, class_label(i))</i>	140
Algorithm 6.5: <i>Cluster Head Selection in TBCS: CH_selection(n, T^r, T)</i>	141
Algorithm 6.6: <i>New Node Joining in TBCS: Node_join()</i>	143

IMPORTANT ABBEREVATIONS

AAF	A erial A d-hoc F ormation
AGPS	A dvance G lobal P ositioning S ystem
AES	A dvanced E ncryption S tandard
AODV	A d-hoc O n-Demand D istance V ector
BLOS	B eyond L ine O f S ight
CH	C luster H ead
DOS	D enial O f S ervice
DOLSR	D irectional O ptimized L ink S tate R outing
DSDV	D estination S equenced D istance V ector R outing
DSR	D ynamic S ource R outing
ETX	E xpected T ransmission C ount
FANET	F lying A d-hoc N etwork
FCTM	F uzzy C lassification T rust M odel
GCS	G round C ontrol S tation
GPS	G lobal P ositioning S ystem
GT	G round T roops
IDS	I ntrusion D etection S ystem
IEEE	I nstitute of E lectrical and E lectronics E ngineers
LOS	L ine O f S ight
MANET	M obile A d-hoc N etwork
MAV	M icro A ir V ehicle
MIMA	M an I n the M iddle A ttack
MN	M obile N ode
MPR	M ulti P oint R elay
MSE	M ean S quare E rror
OLSR	O ptimized L ink S tate R outing
QoS	Q uality o f S ervice
TBCS	T rust B ased C lustering S cheme
UAV	U nmanned A ir V ehicle
VANET	V ehicular A dhoc N etwork
WSN	W ireless S ensor N etwork

LIST OF PUBLICATIONS

JOURNAL

1. Singh, K., & Verma, A. K., FCTM: A Novel Fuzzy Classification Trust Model for Enhancing Reliability in Flying Adhoc Networks (FANETs). *Ad Hoc and Wireless Sensor Networks* (**Accepted- SCI Indexed Journal**).
2. Singh, K., & Verma, A. K., A Fuzzy based Trust Model for Flying Ad hoc Networks (FANETs). *International Journal of Communication Systems* (**Accepted- SCI Indexed Journal**).
3. Singh, K., & Verma, A. K., TBCS: A Trust Based Clustering Scheme for Secure Communication in Flying Ad-Hoc Networks. *Wireless Networks* (**Under review- SCI Indexed Journal**).
4. Singh, K., & Verma, A. K., A Systematic Review of Trust Management in MANETs - Attack Analysis and Defense Mechanisms, *Computer Science Review* (**Under review-Scopus Indexed**).

CONFERENCES

5. Singh, K., & Verma, A. K., Threat Modeling for Multi-UAV Adhoc Networks, TENCON 2017, (**Accepted and Presented**).
6. Singh, K., & Verma, A. K. (2016, November). Adaptability of Various Mobility Models for Flying Ad hoc Networks-A Review. *International Conference on Recent Advancement in Computer, Communication and Computational Sciences (ICRACCCS-2016)* (**Accepted and presented**).

BOOK CHAPTER

7. Singh, K. and Verma, A. (2017). Flying Adhoc Networks: Concept and Challenges. In: M. Khosrow-Pour, ed., *Encyclopedia of Information Science and Technology, 4th ed. USA: IGI Global*.
8. Singh, K., Aggarwal P. and Verma, A. Analysis of Various Trust Computation Methods: A Step Towards Secure FANETs, *Computer and Cyber Security: Principles, Algorithm, Applications and Perspectives*, CRC Press (**Accepted- Scopus Indexed**).

CHAPTER 1

INTRODUCTION

The use of Unmanned Aerial Vehicles (UAVs) has significantly increased for forming an ad-hoc network because of their ability to perform in an exciting environment such as armed attacks, border surveillance, disaster management, and rescue operations. This type of ad-hoc networks is popularly known as Flying Ad-hoc Networks (FANETs). FANETs can be rapidly deployed anytime in real-time for communication because they do not require any external resources. These unique features make FANETs a suitable solution for various applications. In various applications, UAVs carry sensitive information, which needs to be protected from the adversary and the intruders. The nodes in FANETs are exposed to various internal and external threats during the critical missions. So there is a need for security mechanism which can protect the sensitive data.

1.1 BACKGROUND

A group of mobile nodes that are connected through a wireless communication channel without any support of established

infrastructure is termed as mobile ad-hoc networks (MANETs) (Akhbani, Kokmaz, & Raju, 2012; Saboji & Akki, 2009). In MANETs nodes are free to move in any direction to support the assigned task of the network. Due to which the network topology changes frequently. Any node can join or leave the network without any restriction. The large degree of freedom and self-organizing capabilities makes MANETs totally different from any other network solution. However, the self-organized and independent behaviour of nodes may lead to malicious activities in the network. Some of the nodes may misbehave and act selfishly to fulfill their interests (Moustafa, & Zhang, 2009). These problems do occur in vehicular ad-hoc networks (VANETs), which is promoted as a dedicated ad-hoc network comprising of the vehicles on the road, suitable for traffic management involving an intelligent transportation system (ITS) (Nzouonta, Rajgure, Wang, & Borcea, 2009). The application area of VANETs is expanded into road safety that includes traffic management, collision avoidance (Grey et al., 2015) and co-operative driving and other user applications like Electronic toll collection, Entertainment Applications, Internet Access and Locating fuel station. One misbehaving vehicle can cause disaster situations in the network which can lead to the destruction of property and life. Along with VANETs, the Flying Ad-hoc Networks (FANETs) are also a subclass of MANETs. FANET is a group of Unmanned Aerial Vehicles (UAVs), i.e., mobile network node(s) that create an ad-hoc network (Muller, 2012). These mobile nodes are

connected through the wireless communication channel without any support of established infrastructure (Sahingoz, 2014).

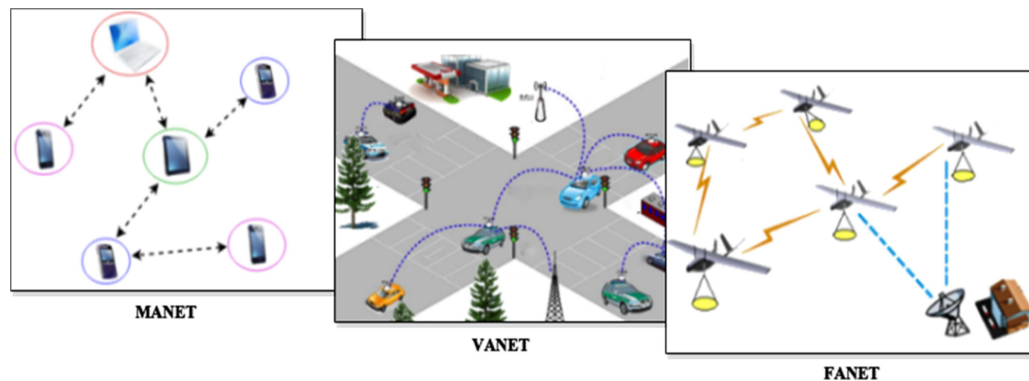


Figure 1.1: *MANET and its Sub-Class Networks*

In FANET, UAVs can fly autonomously based on pre-programmed flight plans or can be operated using complex dynamic automation systems and are versatile and flexible in implementation (Singh & Verma, 2014). When ordinary communication infrastructure is out of service or simply not available, a group of small flying robots can provide a rapidly deployable and self-managed ad-hoc Wi-Fi network to connect and coordinate rescue teams on the ground (Bekmezci, Sahingoz, & Temel, 2013). The communication standards proposed for ad-hoc networks are not only liable to grounded networks, but rather they also have wide application in unmanned aerial vehicle networking (UAV). Similar to MANETs, FANET nodes are free to move in any direction to complete the assigned task, due to which the network topology changes frequently. Nodes can join or leave the network as and when required. FANETs as a subclass of MANETs offer self-organized and independent behaviour of nodes which may lead to malicious and selfish behaviour of nodes in the network (Sentürk,

2016). Due to heterogeneous applications of FANETs, it is mandatory for nodes to follow the protocols and cooperative behaviour, highlighting the need for security of such networks. MANET and FANET have certain common features, thus inviting to explore the possible usage of the existing solution to FANETs as well. The common features of MANETs and FANETs are listed below (Bekmezci, Sahingoz, & Temel, 2013):

- i. Multi-hop Routing:** Both the networks follow multi-hop routing scheme in which they select multiple hops to transfer a message rather than choosing a single long path.
- ii. Infrastructure Less:** Both MANETs and FANETs are infrastructure less networks because there is no pre-established infrastructure available. They create a network as per requirement.
- iii. Dynamic Topology:** Both these networks do not follow one topology. Node leaves and joins the network so frequently. Hence topology remains dynamic.
- iv. Random Mobility Pattern:** Mobility patterns for nodes are not fixed. Nodes can randomly move in any direction.
- v. Limited Device Capability:** Both the networks have devices with limited computation power, load capacity, and energy.
- vi. Limited Physical Security:** In these networks, physical security is very less. This is because the nodes keep moving in different directions. So it is difficult to provide them physical security.

- vii. Node Density:** Both FANETs and MANETs have very low node density. Nodes are sparse in the network.
- viii. Message Propagation Model:** Both the networks follow centralized and de-centralized message propagation model.
- ix. Network Scalability:** MANETs and FANETs are easily scalable. Nodes can be added at any point in time as per requirement.
- x. Threats from External and Compromised Nodes:** In both MANETs and FANETs node are self-organized in nature. Nodes may get compromised easily due to no central control. Both the networks have threats from such nodes.

1.1.1 FANET Architecture

FANET comprises of three key components (Marshall, Barnhart, & Shappee, 2015): nodes (UAVs), Ground Control Station (GCS), and communication links. The network model of FANET with its components is shown in Figure 1.2. UAVs are the flying aircraft without human onboard. The UAVs consists of an infrared camera, GPS module, sensors, gyro-stabilized observation platform, a flight controller, and communication subsystem. UAVs come in different shapes and sizes, from micro UAVs (2cm) to large UAV (wingspan 120m), they are used according to the application scenarios (Marshall et al., 2015). To accomplish a mission, multiple UAVs form a cluster based on their geographical location, nature of the task, *etc.* Inside the cluster, UAVs periodically elect their cluster head for inter cluster and GCS communication. As shown in Figure 1.2, UAVs collect information

from their environment and transmit it to the cluster head for further transmission to the GCS. A GCS is the ground unit that provides control and monitoring facilities for UAVs during their operations. A computer is used to run the GCS software that is used to keep an eye on the progress of the mission. For the sensitive and tactical mission UAVs, GCS has more workstations (Marshall et al., 2015). GCS mainly communicates with the flight controller and transfers useful information. To transmit data and control information between UAVs and GCS, wireless communication links are used. Links are categorized into two types: Line-of-sight (LOS), where UAVs and GCS can transmit and receive control signals through direct radio waves; and beyond line-of-sight (BLOS), where UAVs are controlled via satellite (Marshall et al., 2015). The flight controller is the nerve center of a UAV. The flight controller is a small computer installed on the UAV airframe and capable of controlling propulsion system, data received from various sensors and autonomous flights. To achieve various tasks, a flight controller must be equipped with appropriate components such as firmware. Each UAV is equipped with multiple sensors to accomplish various tasks and smooth missions. Global Positioning System (GPS) is an important sensor available for UAVs which provides information about positions, navigation, and timing. GPS sensors receive signals from GPS satellites and use the received data to determine 3-D position. To control a UAV, a real-time communication network is used. The common and sophisticated

modes of communication are through Wi-Fi, radio control and Bluetooth.

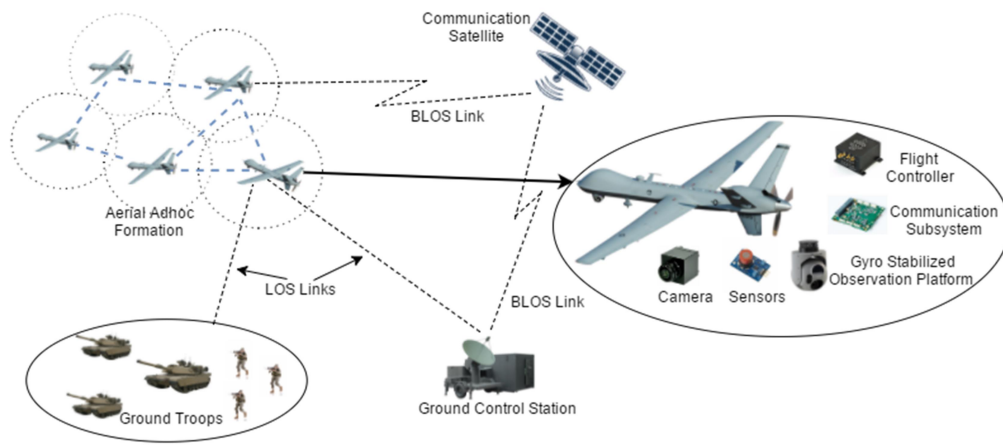


Figure 1.2: FANET Architecture

To communicate with other nodes, FANETs uses various routing protocols. OLSR is proactive routing protocol which is designed especially for ad-hoc networks (Jacquet, Clausen, & Viennot, 2001). In proactive protocol every node maintains one or more tables representing the entire topology of the network. Because of the proactive nature, this protocol has a benefit of having routes quickly available whenever required. To maintain the up-to-date routing information, all the nodes declared and broadcast their topology information in the whole network (Misra & Rajesh, 2011). OLSR provides following features:

- It reduces the size of control packets by declaring a node as a multi-point-relay (MPR) selector to every node of its neighbour.
- By using those MPRs which were selected, it reduces to scatter its messages to the whole network.

OLSR uses MPR through which it can reduce the whole network traffic and also reduces the flooding in the network which arises when every node transmits data to each other to send the message to the exact destination. Hence, this routing protocol is the best fit for large and dense ad-hoc networks (Kru'zelecki, Traynard, & Rosati, 2013; Clausen et al., 2004).

1.1.2 FANET Applications

The Flying ad-hoc network has various applications as follows:

- i. Calamity Administration:** FANET is useful when the existing communication system is damaged due to a natural disaster like floods, earthquake, firing, *etc.*
- ii. In Sensor Networks:** Different sensor devices can be used to collect data to do daily functions like weather forecasting, terrestrial movement tracking, *etc.*
- iii. Location Aware Services:** FANETs can be used in various location services (Bekmezci et al., 2013) such as forwarding calls to any location, a travel guide for passengers and to identify information regarding specific location.
- iv. Military Services:** FANET are very useful in military services. Setting up proper communication system is very difficult in military areas. So FANETs are used for information exchange among soldiers, military headquarters.

- v. Search and Rescue Operations:** FANET can be used provide a better way to do search and rescue operations such as rescue operation of hostages (Bekmezci et al., 2013).
- vi. Security Purpose:** FANET is capable of receiving information quickly. It can be used to collect information for the security purpose of a delegate visiting a place where no network infrastructure exists.

1.2 FANET SECURITY

As we have already discussed the FANET architecture and its properties. In FANET, high mobility nodes join and leave networks very frequently. This results in link breakage and packet losses. So, it requires some technique that can detect the behaviour of nodes in the network. Another issue with ad-hoc networks is the decentralization. Each node decides itself that to which neighbour node, it should communicate based on their reputation. So, there is a need to define a system that can provide reputation feedback to the nodes in the network. Information forwarding and sharing is another important aspect of such real-time ad-hoc networks. These networks become prone to various types of attacks due to malicious nodes in the network. These attacks aim to send false information, forged packets and change routing information. In such environments, classical methods of security, i.e. cryptographic techniques demand more resource consumption as compared to their availability (Pathan, & Al-

Sakib Khan, 2010). Hence, securing FANETs with limited resources becomes an essential need to research. Past research has been done for trust management in the area of ad-hoc networks, indicating that trust management techniques are helpful in securing ad-hoc networks from various kinds of attacks (Clausen et al., 2004; Olmedilla et al., 2006). Hence, to secure FANETs, the concept of trust management is viable (Pathan, & Al-Sakib Khan, 2010). Trust management systems can play an extremely important role in evaluating the reputation of network nodes. Trust can help to detect untrustworthy nodes in the network and keep the records of activities performed by nodes to calculate trust values.

1.2.1 Security Challenges

The basic idea behind FANET is same as MANET, but there are certain differences like node speed, topology change, mobility model, *etc.* Hence, FANET faces some additional challenges than MANET. Ensuring confidentiality, integrity, and availability of valuable information in FANET is the major issue faced by these networks (Poovendran, Sampigethaya, & Bushnell, 2008; Yap, Liu, Tan, & Goi, 2015). Based on the finding of literature review, the following are the security issues identified that requires significant research to be done:

- i. Physical Security:** FANETs faces an issue of node compromise due to lack of physical security. The risk to the network increases even more if the node is a centralized entity (Bekmezci et al., 2013).

ii. Secure Routing: To manage secure routing is another point of concern in FANET. Existing routing algorithms for ad-hoc networks are impotent against frequent network topology changes and malicious attacks in FANET. Node movement is relatively very high for flying ad-hoc networks. So the topology changes very frequently.

- One of the biggest challenges for researchers is to suggest an efficient routing algorithm that can work for high mobility nodes.
- The routing algorithm should be quick to update its routing table frequently as the topology changes.

iii. Trust Management: Due to high topology changes, trust Management among nodes is another challenge. Mobile nodes in FANETs join and leave the groups very frequently. Hence, it is very difficult to manage trust in such large and dynamic network.

iv. Scalability: Single UAV system can perform limited tasks. To perform a number of tasks, a collection of UAVs is required. This motivated the concept of the multi-UAV based system. This improves performance, reduces delay and optimizes the task. FANET algorithms should be so designed that they can accommodate any number of UAVs.

v. Reliable and Secure Data Transfer: FANET applications transfer sensitive information. So the reliability of the network should be very high. Reliability and security should be defined

as the criticality of data. So different level of security needs to be defined.

- vi. UAV Mobility and Placement:** The placement of UAVs at the appropriate location is one of the major research concerns in FANETs. UAVs of different capacity and capability are used for different purpose. Mini-UAVs are meant for carrying fewer payloads, like a thermal camera, single radar, camera, an image sensor, etc. So, this is an open challenge to optimize the UAV placement to diminish energy feeding when the retrieved information is taking more time.

1.2.2 Security Requirement

As we have observed, a variety of complex security attacks are increasing in FANETs day by day. Hence, there is a requirement of security framework to prevent the network from these attacks and minimize the negative impact of these attacks on the network functionality. FANET's security mechanisms can be designed based on its security requirements. These requirements depend on some primary security properties within intrinsic resource constraints of FANETs. An effective security mechanism should accomplish following properties (Zhou, & Haas, 1999; Yanli et al., 2012; Cherdantseva & Hilton, 2013):

- i. Confidentiality:** Confidentiality ensures the data disclosure to its intended user only. A message can only be understandable by the sender of the message and recipient of the message. No

adversary should be able to get the message. Confidentiality ensures about eavesdropping during the data transmission.

- ii. Integrity:** The data transferred and received by the mobile nodes in FANETs must not be maliciously tainted. Integrity protects data from unauthorized modification in transit, ensuring that the data received is the same as the data sent.
- iii. Authorization:** A node can perform only a set of operations on the network for which it has authority. Each device can only perform certain operations which were allocated to it by the administrator.
- iv. Authentication:** Authentication facilitates a receiver to validate about the receiving data that is sent by a claimed and legitimate sender. The receiver should be correctly identified and verify the sender information.
- v. Availability:** The network services should be available for its legitimate users whenever it is required for users. The UAVs, sensors, GCS, etc. should always be available to communicate with each other. Unavailability of these components due to malicious attacks can harm the assigned tasks and make FANETs useless.
- vi. Freshness:** A node should consume only those data packets which are timely produced, means it should not be very old data. A node should be able to discard the old messages and work based upon latest information to avoid message replay attack.

- vii. Self-organization:** UAV nodes should be flexible enough to handle problematic situations autonomously, and it should be self-managing and self-healing.
- viii. Non-repudiation:** It guarantees that a node cannot refuse having sent the packets which it has previously sent. This is helpful for recognition and separation of compromised nodes from the network.

1.3 ORGANISATION OF THE THESIS

Including the introductory chapter, this thesis is organized into seven chapters. The details of each chapter are as follows:

- i. Chapter 2: Literature Review:** This chapter presents a review of state of the art of trust schemes/models/frameworks in wireless networks, especially in ad-hoc networks. The primary focus of the chapter is to identify key factors for evaluating trust in a dynamic environment. Further, research gaps are also identified in existing trust-based security schemes. The identified gaps are taken as an opportunity for work in next chapters.
- ii. Chapter 3: Threat Modelling and Risk Analysis:** This chapter presents a threat model, which is designed to identify the threats to the FANETs. First, various threats to FANETs have been identified. After identifying the security threats, a novel threat model for FANETs is proposed. The proposed model

considers adversary capabilities, environment, targets, and vulnerable domains of FANETs. Furthermore, the risk analysis and classification of attacks are done on their influence on confidentiality, integrity, and availability.

iii. Chapter 4: FCTM: Fuzzy Classification Trust Model: In this chapter, a fuzzy classification trust model (FCTM) for FANETs has been proposed. FCTM primarily utilizes the fuzzy classification method to distinguish among network nodes considering its social parameters and quality of service parameters. The primary aim of this model is to classify the network nodes in various clusters: good, neutral and bad. For cluster initialization, the fuzzy membership function has been defined using the expert's domain knowledge. Further, to improve cluster centers fuzzy optimization was applied, which reallocate the cluster of network nodes and redefine the cluster centers for each cluster. Based on these optimized parameters, nodes are classified in different clusters (bad, neutral and good) and the node gets punished or rewarded accordingly. After that bad node are segregated from the network.

iv. Chapter 5: WFCTM: Weighted Fuzzy Classification Trust Model: In chapter 5, an enhanced version of FCTM has been proposed, which includes a weighted technique for node's trust calculation in FANETs. WFCTM model helped to handle the behavioural uncertainty of FANET nodes using the fuzzy optimization. In this model, we introduce weights for each

parameter during the fuzzy optimization to calibrate appropriate weightage of each parameter that contributed to trust evaluation. Optimized weights for parameters helped in the improvement of node classification accuracy which results in better network performance.

- v. **Chapter 6: TBCS: Trust-Based Clustering Scheme:** Chapter 6 presents a trust-based clustering scheme, i.e., TBCS for cluster-based FANETs has been proposed. The proposed scheme makes use of Takagi-Sugeno-Kang, fuzzy inference method. In TBCS, the network was divided into various clusters, and each node in the cluster computed trust values of its one-hop neighbours. A secure cluster head (CH) has been selected based on calculated trust values that were calculated using fuzzy optimization. The concept of reward and punishment with decay function have been introduced to transform node behaviour into a trust value, and to segregate malicious and misbehaving nodes in the FANET
- vi. **Chapter 7: Discussion and Conclusion:** Finally, chapter 7 summarizes the presented work. It also briefly highlights the contributions and future scope.

CHAPTER 2

LITERATURE SURVEY

Various trust-based security solutions have been proposed for ad-hoc networks. The recent and prominent proposals published in the literature are discussed in this chapter. Potential attacks on FANET environment have been discussed. The concepts of trust, trust properties, trust metrics, trust management process and stages of trust management have been explained. We have analyzed different trusts computation models such as fuzzy, Bayesian, game theory, subjective trust, weighted sum, Markov chain and beta distribution. The proposed trust computation models are compared and analyzed for their advantages and disadvantage.

2.1 POTENTIAL ATTACKS

To understand the importance of trust management scheme, first, we should understand the extent to which the FANETs are vulnerable. FANETs are prone to security attacks due to self-organized and independent behavior. Survey (Wu, Chen, Wu, & Cardei, 2007; Ahmed et al., 2015) listed all the possible attacks that occur in an ad-hoc

network and affect its performance. The explanation in this section considers all those attacks that affect the performance of ad hoc networks in one way or other.

- i. Black Hole Attack:** In mobile ad hoc networks one of the severe attacks observed is black hole attack which is also called packet drop attack. In this attack, an adversarial node advertises itself to the target node about the shortest route (Al-Shurman, Yoo, & Park, 2004; Jaisankar, Saravanan, & Swamy, 2010). Using this strategy, a forged route is created. All the packets for the destination node come the adversarial node. Now the adversarial node decides either to drop the packets or to manipulate the message before forwarding to the destination. A black hole is basically a denial of service and man in the middle attack. This attack on the network can harm the reliability of the system.
- ii. Gray-Hole Attack:** Gray-hole attack is a special case of black hole attack where malicious node selectively drops packets. It chooses selected packets or a part of the packet, random time in a day or specific destination to drop a packet (Cai, Yi, Chen, Wang, & Liu, 2010; Sen et al., 2007). Such type of malicious nodes is difficult to detect. The gray-hole attack is challenging attack on trust management systems.
- iii. Wormhole Attack:** In wormhole attack, the attacker creates a tunnel to that node which is at a longer distance than normal route. Attacker route the received packets through the tunnel to the node and communicate through that node. Malicious nodes

create a group that pretends to communicate through low latency link, known as wormhole link. This causes interruptions in normal traffic flow (Hu et al., 2006; Nait-Abdesselam, Bensaou, & Taleb, 2008).

iv. Denial of Service Attack: Denial of service attack is an attack on the availability of resources. This attack is also referred as Jelly Fish attack. The main focus of DoS attack is to reduce the goodput of the network by dropping packets or by consuming the bandwidth of the network. To achieve this target malicious nodes may use one of three strategies. The first strategy is reordering packets because TCP packets are vulnerable to packet reordering. The second strategy is to drop packets periodically which will cause retransmission of the dropped packet. Hence the throughput of the network will be reduced. The third strategy is to vary delay to perform the attack (Pelechrinis, Iliofotou, & V Krishnamurthy, 2011; Begum, Mohan, & Ranjitha, 2012).

v. Impersonation Attack: In ad hoc networks, there is no fixed underlying infrastructure. Nodes are mobile in nature, so network topology changes very frequently. In such scenario, node authentication becomes very difficult, and nodes are more prone to impersonation attack in such kind of networks (Glynos, Kotzanikolaou, & Douligieris, 2005). Detection of such nodes is a very difficult task.

- vi. Sybil Attack:** Sybil attack is an attack on identity manipulation of a node (Abbas, Merabti, Llewellyn-Jones, & Kifayat 2013). An attacker may create fake identities of nodes to other legitimate nodes. If there is no authentication scheme, the genuine node will not be able to detect the real identity of the node.
- vii. Rushing Attack:** Rushing attack is a kind of denial service attack that occurs against on-demand network routing protocols. This attack basically makes route discovery difficult for other nodes (Kim, Oliveira, Bhargava, & Song, 2013). Whenever any node advertises a route request, the attacker also gets that route request. Attacker quickly floods the whole network with a reply, before any other node can react to this message.
- viii. Sinkhole Attack:** In this attack, the attacker sends false route information to the network. Attacker node tries to pretend as a most important node in the network. By doing this, it receives all the traffic from the network (Gandhewar & Patel, 2012). Now the attacker can read secret information, modify information or it can drop packets to harm the network.
- ix. Replay Attack:** Replay attack is a kind of network attack where valid information is repeatedly transmitted by the attacker (Kannhavong et al. 2007). This is done to proof his identity to other nodes so that it can disrupt the normal functioning of the network. This attack is basically impersonation attack, where malicious nodes try to pretend authentic nodes.

- x. Flooding and Packet Drop Attack:** Flooding attack is a kind of denial of service attack (Balakrishnan, Varadharajan, & Tupakula, 2006). In this attack, malicious node floods the network with huge amount of request or data packets to consume the bandwidth. Thus the resources become unavailable to the legitimate nodes.
- xi. Byzantine Attack:** Byzantine attacks (Yu, Zhou, & Su, 2009) are attacks against routing protocols, where two or more routers get together to forge, drop, fabricate or misroute the packets to interrupt routing services.

2.2 AD-HOC NETWORKS SECURITY

Being in an early stage of development, FANETs lacks the effective security mechanisms. Security is one of the important issues to be addressed. To secure FANETs two techniques can be used: cryptographic schemes or trust mechanism. Security techniques used for securing of ad-hoc networks can be classified into two classifications namely: cryptographic techniques and trust management techniques.

2.2.1 Cryptographic Solutions

Under the banner of cryptographic techniques majorly three cryptographic techniques have been used: Symmetric key, Asymmetric key, and the Identity-based cryptography. In symmetric key cryptography, a common key is shared between the communicating

nodes. That key is used for encryption and decryption of the message shared by the nodes. In asymmetric key cryptography, a pair of keys: a private key and public key are used. Sender node uses public key of the receiver of the message to encrypt the message, and recipient of that message use its private key to decrypt the message. Adi Shamir presented the concept of identity-based cryptography (Shamir, 1984). The prime objective of the identity-based technique is to use of user identity attributes, such as phone numbers, and email, as an alternative to digital certificates, for encryption and signature verification.

2.2.2 Trust-Based Solutions

The major focused of cryptographic solutions were on the confidentiality of the message delivered to the neighbour nodes and less attention to the quality of information. The information exchange by the nodes may be bogus because nodes can behave selfishly and maliciously. So, the security mechanism should be designed such that the malicious and selfish node can mitigate from the network. During the tasks network nodes collaborate and cooperate. Therefore, trust among them plays an important role. Trust refers to the degree of reliability of other nodes in the network for performing network operations (Akbani, Korkmaz, & Raju, 2012). Trust between the nodes is calculated by observing and recording their past behaviour with other nodes in the network. Calculated trust value helps mobile nodes to deal with uncertain behaviour in future communication. Trust-

based techniques are very helpful to cope with selfish and misbehaving nodes in the network (Govindan, & Mohapatra, 2012). Traditional ways to secure network communication and authentication using cryptographic techniques are costly in terms of resource consumption (Jatoi, Memon, Chowdhry, Ullah, & Latif, 2015). So, securing FANET with limited resource consumption is another challenge. So, there is a dire need of mechanism(s), which can allow network nodes to deduce about the trustworthiness of other network nodes. The trust metric for a node is required not only when it has to deal with misbehaving node, but also when it exchanges information with other nodes (Jøsang, 2007). Cosmic research has been done for trust management in the area of ad-hoc networks, indicating that trust management techniques are helpful in securing networks from various kinds of attacks (Chae, DiPippo, & Sun, 2015). Hence, to secure FANETs, the concept of trust management is viable.

2.3 TRUST IN AD-HOC NETWORKS

Trust is the belief among two or more communicating parties. Trust is defined differently based on its utility in that domain. This section deals with the application of trust in ad-hoc networks. Here the concept of trust in ad-hoc networks, its process and the various stages involved in trust management systems are discussed.

2.3.1 Concept of Trust

Trust is the confidence between two communicating parties that they will behave expectedly during the whole conversation. This is widely accepted component for any kind of communication. A wide variety of literature exists that define trust according to different disciplines. According to Mui et al. (2002), “Trust is a subjective expectation an agent has about another’s future behaviour based on the history of their encounters.” Another definition from Grandison and Sloman (2000) refers to the “competence” to act, “Trust is the firm belief in the competence of an entity to act dependably, securely, and reliably within a specified context.” In general computer science also as defined earlier by Mui et al. (2002), trust is dependent on feedback based on past interactions. In online systems also trust is build based on past interactions and experiences. Trust is considered as direct trust and indirect trust. Direct trust is that in which the user himself experience through communication with the system. In case of an indirect trust, some other user shares his/her experience. This is also called recommendation trust. Trust plays a very important role in ad-hoc networks. Blaze et al. (1996) defined “Trust Management” as a different component of security in networks. Trust management is very important in the intrusion detection systems, access control systems, detection of selfish nodes, secure routing, and authentication.

2.3.2 Trust Properties

Ad-hoc networks have unique characteristics. Therefore, the trust properties should be carefully designed for such networks where nodes are mobile and topology changes very frequently. Various properties of trust have represented the Figure 2.1 below, which are classified by considering the characteristics of ad-hoc networks (Liu, Datta, & Lim, 2014; Cho et al. 2011).

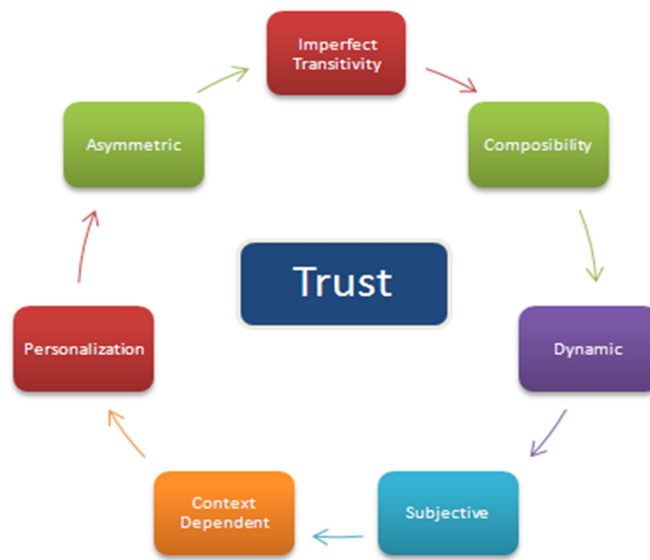


Figure 2.1: Trust Properties

- i. **Imperfect Transitivity:** One of the primary properties of trust is transitivity (Sun, Yu, Han, & Liu, 2006). Trust is not necessarily transitive. If Alice trust Bob and Bob trust Jack, then it is not necessary that Alice will also trust Jack. However, the Bob can recommend Alice to trust Jack. Here Alice may trust Bob but not his recommendation. So here two types of trust play a role: trust in person and trust in his

recommendations about another person. The concept of propagating trust along the chains is widely studied these days.

- ii. Asymmetric:** In ad-hoc networks, it should be considered that trust is asymmetric in the sense that it is not necessary that if Alice trusts Bob, Bob will also trust Alice (Yaniv & Kleinberger, 2000). It all depends on the experiences of both individuals about another partner. This is considered as a case of personalization. It means the trust will depend on person's expectation, beliefs, opinion, and perceptions.
- iii. Personalization:** Trust is completely personal opinion. Two persons can have different opinions about the same person. The trust is a belief that the person has on another person's action that it will lead to a positive outcome (Yaniv & Kleinberger, 2000).
- iv. Dynamic:** Trust is not a static property. It is dynamic in nature. The value of trust changes with available information which usually varies due to mobile nodes or node failure. (Adams et al., 2005) described in his paper that the trust should be considered as a continuous variable rather than a binary or discrete variable.
- v. Subjective:** Trust is subjective (Abdul-Rahman & Hailes, 1997). In ad-hoc networks, topology changes very frequently. For mobile and flying ad-hoc networks the trustee node changes the trust level based on experience. The same node gets different values of trust due to the highly dynamic behaviour of nodes.

- vi. Context-Dependent:** Trust is also context dependent (Staab et al., 2004). It means that one person may be expert in repairing mobile phones. He might not be expert in repairing laptops. Here trust value changes based on the context or application. Similarly, MANETs have huge application area. Trust may vary with the type of application.
- vii. Composability:** Another feature of trust is composability. Transitivity transfers trust from one node to another. Composability is a way to combine different trust values from different nodes. Richardson et al. (2003) used the of composition function in their work.

2.3.3 Trust Metrics

Trust has been evaluated differently by different researchers. The classification of trust evaluation matrices is given below:

- i. Trust Scale:** Trust value can be evaluated as continuous or discrete variables (Beth, Borcharding, & Klein, 1994). For discrete values trust is measured in interval $[-1, 1]$ and for continuous values interval is $[0, 1]$. Another technique is threshold-based approach; if the ratio of satisfactory interactions with respect to total interactions is greater than the threshold value, then the node will be called trustworthy.
- ii. Packet Drop Ratio:** Packet drop ratio is also a useful measure to evaluate trust (Sun, Yu, Han, & Liu, 2006). Due to malicious nodes or selfish nodes and various attacks packet drop in the

network becomes more. So evaluating trust management system on the basis of packet drop ratio is a good measure.

- iii. **Route Usage:** Route usage means a number of paths that are considered for secure routing. This can also be considered as one of the trust matrices for trust evaluation (Sun, Yu, Han, & Liu, 2006).
- iv. **Packet Overheads:** Another metric to measure the performance of trust management system is packet overheads (Sun, Yu, Han, & Liu, 2006). If the system can reduce the packet overheads, overall network performance will be enhanced.
- v. **Detection Accuracy:** Malicious node detection accuracy of trust management is another criterion to evaluate trust management system.

2.3.4 Trust Management Process

Trust management is about creating the trust, evaluating the trust values, updating trust and revocation. The detailed explanation of trust management process follows the steps shown in figure 2.2.

- i. **Trust Initialization:** The first step of trust management process is trust initialization. The Trust initialization process is also known as bootstrapping. Initially, when two nodes come into contact, then they do not have prior information about each other to calculate the trust value. So, they need some prior trust value to overcome this situation. According to (Nguyen & Camp, 2008; Theodorakopoulos, & Baras, 2006) nodes are categorized

in: trusted, non-trusted and un-defined. The trust value of nodes is updated based on recorded communications.

- ii. **Evidence Space:** The next step in trust management process is to collect evidence. The evidence can be collected through direct observation by the node itself and can be collected through others node observations or recommendations (Wang & Singh, 2007). Based on these direct and indirect methods, trust value of a node can be recalculated. A node is said to be trusted or non-trusted based on the final trust value.

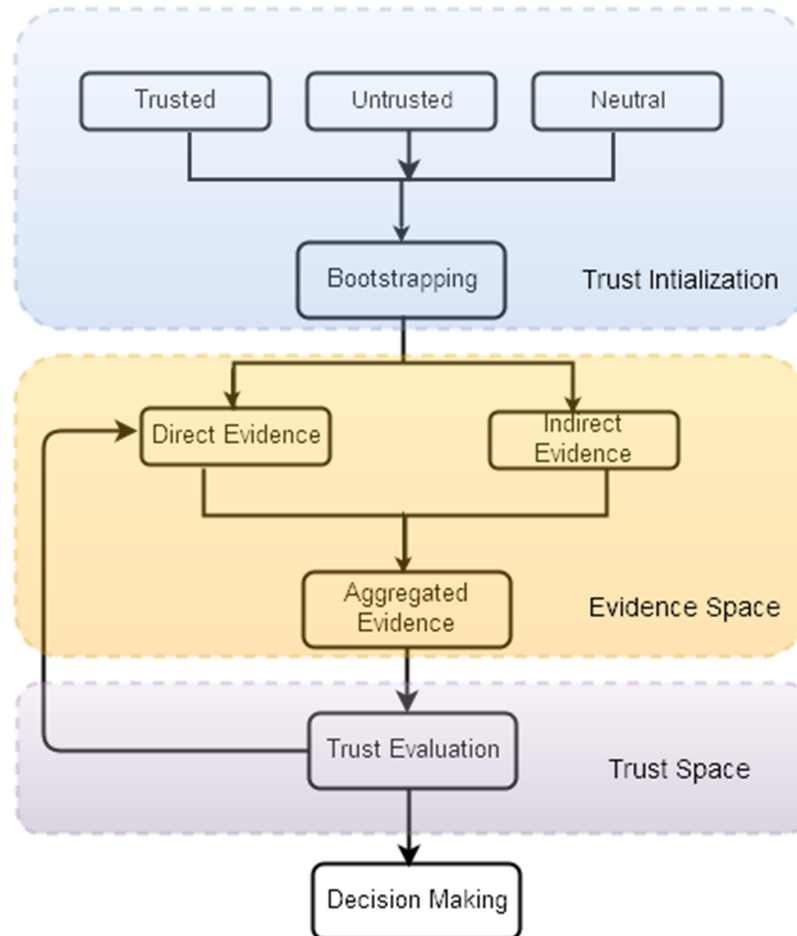


Figure 2.2: Trust Management Process

iii. Trust Space: As ad hoc networks are decentralized, so trustworthiness of network nodes is calculated by individual nodes. Trust can be represented in various formats such as <subject, agent, time, action, outcome> (Buehger & Le Boudec, 2002). This shows agent considering the particular subject, action done successfully or failed at time t. Another way is <subject, agent, action, k, n> where k is the number of successes, and n failures, that is a subject consider about an agent. Evidence space is mapped to the trust space based on the calculated trust value in the evidence space.

iv. Decision Making: In trust management process, decision making for a node that it is trustworthy or not is done on the results of the previous phase. Following are the methods which are used for decision making.

- **Threshold-based decisions:** If the trust value of a node in previous phases is less than the threshold value, then that node is not considered reliable for communication else it is reliable.
- **Ranking-Based Method:** In the ranking based method, the transformation function is used to calculate values, and they are represented in [0, 1]. Ranks are assigned to the nodes based on the trustworthiness values. A hybrid technique by using the ranking based method and threshold-based method is presented by (Buehger & Le Boudec, 2002; Gonzalez, Anwar, and Joshi, 2011).

- **Weight-based method:** Weight based method use aggregation technique to calculate trust value (Almenarez, Marin, Diaz, & Sanchez, 2006; Chen, Wu, Zhou, & Gao, 2007). The following equation is used to calculate total trust by the subject node:

$$T = \sum_{i=1}^n w_i t_i$$

Where, T represents the total trust, w_i represents node i weightage that provide trust value, and t_i represents reported trust value by node i .

2.3.5 Stages of Trust Management

In mobile ad-hoc networks, trust computation involves various stages. These stages can be classified into different sections based on their purpose. Based on the literature, we have classified trust computations stages in five dimensions: trust composition, trust formation, trust calculation, trust propagation and trust updation. Various stages involved in trust management are represented in figure 2.3.

i. Trust Composition: Trust composition refers to the components which are to be considered for trust computation, it includes quality of service and social trust.

- **Quality of Service Trust:** In mobile ad-hoc networks, quality of service (QoS) (Umuhoza, Agbinya, & Omlin, 2007) trust refers the performance of nodes in the networks in terms of node competence, cooperativeness among nodes, node reliability

and task performance (delay, PDR, throughput, and energy consumption) of a node.

- **Social Trust:** Social trust (Yan & Holtmanns, 2008) of nodes in the mobile ad-hoc network can be calculated on the base of their social relationship with other nodes. The social relationship of network nodes can be measured through their privacy, honesty, friendship, similarity, selfishness, and centrality.

ii. Trust Formation: Trust formation refers to the technique how trust value is identified for a node in the network. We have considered two factors (observation and property) on which trust formation complete in ad-hoc networks.

- **Based on Observation:** A node can calculate trust for another node in the network by three ways: direct observation, indirect observation and through recommendation (Nguyen, Camp, & Loiseau, 2007). In direct observation, a trustor node observes the behaviour of a node when having interaction with that node. In indirect observation, a trustor node observes the behaviour of trustee node when trustee node is having interaction with another node in the network. In the recommendation, trustor node gets information about trustee node from other nodes in the network.
- **The Based on Property:** Trust of a network node can be calculated based on the number of properties; we have used two classes single and multi-trust (Li, Parker & Joshi, 2012).

Single trust refers that only one trust property is being considered to calculate trust of a node, whereas multi trust depicts that multiple trust properties are used to calculate trust of a node in the ad-hoc network.

iii. Trust Computation: Trust computation refers to the mechanism through which total trust value of a network node is calculated. Two technique trust prediction and trust aggregation are used to calculate trust value of a node in the ad-hoc network.

- **Trust Prediction:** Trust prediction is a technique to predict trust among node by using the past and current behaviour of nodes in the network. In trust prediction method, a recommendation from other nodes is also considered to predict the trust value. In literature Markov chain model (Chatterjee, Sengupta & Ghosh, 2012), mathematical induction (O'Donovan, & Smyth, 2005) and fuzzy based (Dai, Jia, & Qin, 2009) prediction models are discussed.
- **Trust Aggregation:** Trust aggregation methods are used to calculate trust when trust of a network node is propagated through multiple paths. So aggregation technique combines these multiple values to obtain a single trust value of a node in the network. Literature consists of game theory, Bayesian approach, weighted sum and fuzzy as aggregation methods (Govindan, & Mohapatra, 2012; Cho, & Swami, 2009; Li, Li, & Kato, 2008).

iv. Trust Propagation: Trust propagation is a technique through the trust of networks nodes are propagated in the network. Trust propagation is used to avoid the recalculation of trust for a node again and again through which resource can be conserved. There are two schemes, distributed and centralized to propagate node trust in the network.

- **Distributed Trust:** In the distributed scheme, a node's trust can be shared either through trust chaining or through recommendation (Theodorakopoulos & Baras, 2006). In chaining concept, trust is forwarded from a trusted node to another node in a chain fashion. In the recommendation, a node trust value is shared by another node that has information about recommended node through direct or indirect methods.
- **Centralized Trust:** In centralized trust propagation scheme, a node in the network works as a central authority and work as a trust propagation entity in the network for every node in the network (Park, Lee, & Chung, 2008).

v. Trust Update: Trust updation refers to the answer to a question that when trust of a network node will be updated. There is two schemes time based and event based can be applied to update node trust in the network.

- **Time-Based:** In the time-based approach (Ferdous, Muthukkumarasamy, & Sithirasenan, 2011), trust based on (observation or property) is updated in two ways- periodic

updating and continuous updating. In periodic updating, trust value of a node updated after a certain time period, whereas continuous updation is to evaluate trust of a node continuously without any time interval.

- Event Based: In event-based trust updating technique, the trust of network nodes is updated when an event occurs (Kute, Patil, Pardakhe, & Kathole, 2012). Event-based update technique can be further divided into two ways- certain event and uncertain event. In certain event-based scheme, the trust will be updated only when a predefined or specific event will occur, whereas in uncertain scheme trust of network node will be updated when any event will occur.

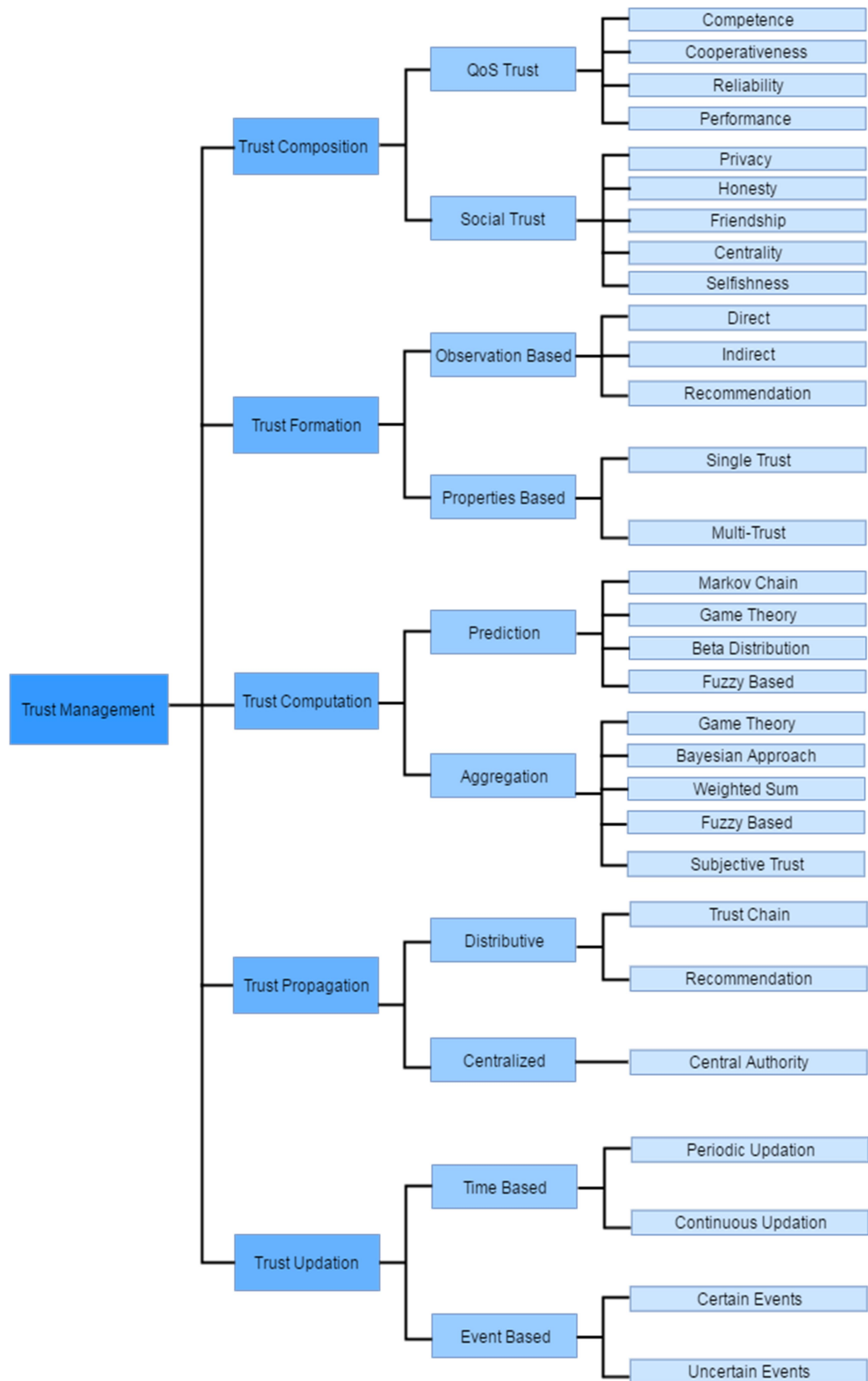


Figure 2.3: Stages in Trust Management

2.4 TRUST COMPUTATION METHODS

Trust management systems have wide applications in various fields like in E-Commerce systems, distributed systems, sensor networks, ad-hoc network and information technology. Trust and reputation have been calculated using various trust computation techniques, i.e., fuzzy model, Bayesian trust model, game theory model, *etc.* Here, we discussed popular trust computational methods that are already being implemented in MANETs and explain their applicability in FANET.

2.4.1 Fuzzy Model

Fuzzy model is one of the major techniques to estimate trust of mobile nodes in the network. RFSTrust (Luo, Liu, & Fan, 2009), FuzzyTrust (Song, Hwang, Zhou, & Kwok, 2005) and TSR (Xia, Jia, Li, Ju, & Sha, 2013) are some popular models that calculate trust scores using fuzzy logic. Trust value can be classified in two ways: discrete trust values and continuous trust values. The common range of discrete trust values could be $[-1, 0, 1]$, whereas, continuous trust values range in between $[0-1]$. Fuzzy logic deals with approximate values rather than exact values. Compare to binary set, fuzzy variables have values in range $[0, 1]$. Fuzzy logic is used to compute trust values, handle partial truth, their aggregation and dissemination (Mallapur, & Patil, 2015). In MANETs node trust is symbolized by a fuzzy variable with membership function explaining the degree of node trust, e.g., if a node has trust value in the range $[0, 0.2]$ then it may be considered as a less trusted node. If a node has trust value in range $[0.3, 0.6]$ then it

may be considered as a medium trusted node. If a node has trust value in range $[0.7, 1]$ then it may be considered as a highly trusted node. Following are the steps for reasoning with fuzzy rules.

1. First, define fuzzy sets and criteria of evaluation
2. Then, do variable initialization for fuzzy engine
3. Apply fuzzy rules to obtain output, and draw conclusions
4. Now, evaluate results and modify fuzzy rules according to requirement.

2.4.2 Bayesian Model

Probabilistic trust methods are used to develop trust management models based on probability theory. Applying probability theory provides the opportunity to explore a wide range of possible derivation methods (Jøsang, 2007). As trust is uncertain in nature, probabilistic models can offer better performance as compared to deterministic trust models to achieve the required security measures. The mechanism of trust calculation depended on the past behaviour of a node and updated according to the current behaviour in the network. The bayesian theory works on the same principal. Prior probability in Bayes theorem considers the past behaviour of nodes and posterior probabilities represents the updated trust value based on the current behaviour. Hence Bayesian trust model is a suitable method for calculation of trust. Few researchers explored the concept of Bayesian trust models in various domains like Dong et al. (2009), PTME for distributed e-commerce, Wang and Vassileva (2003) in peer-to-peer

networks, LogitTrust (Wang, Lu, Chen, Cho, Swami, & Lu, 2015), Feng et al. (2010) and Wei et al. (2014) for MANETs etc.

2.4.3 Game Theory Model

Game theory is the study of mathematical models where two decision makers either cooperate or conflict. This study helps to predict the future behaviour of an entity. Srinivasan et al. (2003), Felegyhazi et al. (2006) used game theoretic models for the development of a trust management system. Game theory captures the dynamic system behaviour where the success of one player depends on the actions of another player. Game theory is a good technique to detect the selfish behaviour of a node in the network and to design optimal strategies using Nash equilibrium for gaining more benefits. In ad-hoc networks, nodes behave cooperatively. Hence, any node playing in an uncooperative manner will be detected easily. Komathy, and Narayanasamy (2008) used this technique to detect misbehaving nodes on the network. Interaction among nodes can be modeled as games such as Prisoner's dilemma. Game theory acts as a behaviour analysis tool that can be used by policymakers of the network.

2.4.4 Subjective Trust

Due to network congestion or network unavailability, cooperative nodes are labeled as selfish nodes. This reduces the performance of the network. To solve such issues, Josang et al. (2006) introduced the concept of subjective logic for calculating trust among communicating nodes. Subjective logic works on subjective beliefs about the object

called an opinion. The opinion denoted by $w_{y:x}$ is a tuple defined below:

$$\omega_{yx} \equiv b_{yx}, d_{yx}, u_{yx}, a_{yx}$$

b: Belief , $b_{yx} \in [0.0, 1.0]$ represent node y's belief in node x.

d: Disbelief, $d_{yx} \in [0.0, 1.0]$ represent node y's disbelief in node x.

u: Uncertainty, $u_{yx} \in [0.0, 1.0]$ denotes y's uncertainty value on node x, and

a: Base rate, $a_{yx} \in [0.0, 1.0]$ is the relative atomicity. It is used in the absence of evidence.

The relation between b, d, and u is defined as:

$$b_{yx} + d_{yx} + u_{yx} = 1.0$$

2.4.5 Weighted Based

The weighted sum is a widespread technique to combine trust evidence. Many trust systems such as Xai et al. (2011), and TRIP (Mármol, & Pérez, 2012) are using a weighted sum for aggregating ratings and feedbacks. More trusted feedbacks are given higher weightage than less trusted. In Nitti et al. (2014) calculates trustworthiness which is obtained from QoS and social trust parameters. This trustworthiness is used as criteria to decide weightage of a rater for indirect trust aggregation. In Chen et al. (2015) also uses similarity, which is derived from the social trust as the criteria to decide weights for indirect trust aggregation. The weighted sum is also used for aggregating the direct trust and indirect trust values. Further research focuses on dynamic weight allocation.

2.4.6 Markov Chain Model

Andrey Markov discovered a Markov chain model which is also known as discrete-time Markov chain. It is a stochastic process that goes through transitions from one state to another state defined on a state space (Zhang, Jia, Xia, Li, & Edwin, 2012). The probability of moving to next state depends only on the current state of the model, previous events in the sequence does not play any role in deciding probability distribution values. Thus, Markov chain model possesses memory-less property. Markov model does not assume any fixed probability distribution about the behaviour of a node. Markov based trust model, are helpful to decide the node's current status. Because the Markov model predicts short-term trust values with the help of the current behaviour of the node. Moe et al. (2008), and Estahbanati et al. (2014) used Markov chain model to predict the trust values in a network.

2.4.7 Beta Distribution

The beta distribution is a probabilistic model based on Bayesian probability. Similar to Bayesian model, the beta distribution model has been used to develop various trust models like HTMF (Li, Li, Liu, & Kato, 2009), CONFIDANT (Buegger, & Le Boudec, 2002), OTMF (Li, Li, & Kato, 2008). This model gets the trust ratings as input and calculates the trust score using the probability density function of beta distribution. Li et al. (2008) proposed trust and reputation models using the beta distribution. They used beta probability density function to compute the trust score. The beta distribution is estimated

by α and β that represent the rating of positive and negative outcomes. The expectance of PDF of beta function is: $\frac{\alpha}{\alpha+\beta}$. After discussing the concept of most commonly used trust computation methods, we provide the summary of an analysis of all the methods. Table 2.1 summarizes the main advantages, examples of trust models, the context, i.e., what kind of security problem is handled and list the attacks considered by various methods.

Table 2.1: Comparison of Various Trust Computation Methods

Trust Computation Method	Advantages	Context	Attacks Considered	Trust Models
Fuzzy Model	- Fuzzy logic handles uncertainty, incomplete information and imprecision effectively - Fuzzy model incorporates the feedback and improve its performance - Fuzzy controller rules help to keep track of past information - Human expert's knowledge can be fed into the model	Secure Route, malicious and selfish node detection	Black, Grey hole, worm hole, and message modification	RFSTrust (Luo et al., 2009); FuzzyTrust (Song et al., 2005); TSR (Xia et al., 2011); AHF-Trust (Xia et al., 2013); COI-HiTrust (Chen et al., 2014)
Bayesian Model	- The Bayesian model considers past and present behaviour to calculate trust. - However, the adversary may form its trust by providing a more number of truthful ratings within the short span of time.	Misbehaving node, Secure Routing	Bad mouthing, DOS, Man in the middle attack	LogitTrust (Wang et al., 2015), Feng et al. (2010); Wei et al. (2014)
Game Theory	- Allow strategy based analysis - Helps to identify nodes, which are behaving rationally - Allows to observe and	Selfish node detection	Message forging, replication,	Srinivasan et al. (2003); Felegyhazi et al.

	update trust over the period of time using dynamic cyber security games		modification, Blackhole, a grey hole	(2006)
Subjective Logic	• Can be used in situations of uncertainty and incomplete knowledge • Partial ignorance and lack of information can be handled • Representation of belief is human understandable	Secure Route	Spoofing, Grey hole and black hole	Jøsang et al. (2006)
Weightage based	• Combination of Direct and indirect trust • Helpful to construct hybrid models • It cannot be used as a standalone model for complex applications.	Misbehaving Nodes	DOS, link spoofing, black hole and jellyfish, Selfish Node	Xai et al. (2011); TRIP (Mármol, & Pérez, 2012)
Markov Chain	• It can account for nodes whose behaviour is not static with time. • It can estimate the rate with which an agent's behaviour changes. • The Markov model considers time component so that a bad node cannot deceive by mimicking good for some time • Large number of parameters	Secure Routing	Selfish Nodes, packet dropping	Moe et al. (2008); Estahbana ti et al. (2014)
Beta Distribution	• Less resource intensive method due to only two parameters • Helps to evaluate trust accumulated over number of experiences • Addition property of beta function allows combination of direct experiences and recommendations • This model does not consider time as a component; it just relies on the number of rating for trust calculation.	Misbehaving Nodes	Node Misbehaviour, and Bad Mouthing	HTMF (Li et al., 2009); CONFIDANT (Buechegger et al., 2002)

After analyzing various trust computation methods, we analyzed their applicability in FANETs. Table 2.2 summarizes various trust computation methods and provide their applicability status in FANETs.

Table 2.2: *Advantage and Disadvantages of Trust Computation Models*

Trust Computation Method	Applicable in FANETs	Comments
Fuzzy	Yes	The ability to handle incomplete information, uncertainty in dynamic environments and inclusion of expert knowledge makes the fuzzy model applicable for FANETs. Fuzzy trust models in literature have been identified as scalable, dynamic and less complex with high security in nature.
Bayesian	No	Bayesian models require more continuous computation that increases the complexity and reduce dynamicity. The Bayesian models are prone to mimicking attacks and could give false alarms, which reduce their security level.
Game Theory	Yes	Models in the literature support that game theory models are successful for selfish node detection and strategy development with low complexity. High security, scalability, and dynamicity features make the game theory models suitable for FANETs.
Markov Chain	Yes	Trust evaluation through Markov chain helps to investigate the rate at which a node changes its behaviour. The Markov models for trust management are identified as low complexity, high dynamicity, and scalable models. Hence, Markov models are suitable for FANETs for belief propagation and evaluation.
Weightage based	No	These are very simple models, but they cannot handle the trust evaluation of complex and dynamic networks such as FANETs.
Subjective Logic	No	Subjective logic can handle incomplete information, and uncertainty. However, this has not been explored much in the domain of MANETs.
Beta Distribution	No	The beta distribution is prone to deceptive attacks which reduce its security. The models found in the literature are not very dynamic and scalable in nature. The computation complexity is also not suitable for FANETs.

2.5 PROBLEM STATEMENT

As we have already discussed, FANET is a subclass of MANET, in which high mobility nodes join and leave networks very frequently. This results in link breakage and packets losses. So it requires some technique that can detect the behaviour of nodes in the network. Another issue with ad-hoc networks is the decentralization. Each node decides independently about which node to trust. So, there is a need to define a system that can provide reputation feedback to the nodes in the network. Information forwarding and sharing is another important aspect of such real-time ad-hoc networks. These networks become prone to various types of attacks: Active and Passive attacks due to malicious nodes in the network. Active attacks aim to send false information, forged packets and change routing information. Whereas a Passive attack corresponds to silently analyze the available information that might be useful for launching active attacks. Existing routing protocols for ad-hoc networks experience the serious issues in the area of security. FANETs require routing algorithms that perform well with high mobility nodes and rapid topology changes. Different research communities have highlighted security needs of FANETs. This research aims to provide a Security Framework for FANETs that can calibrate with rapid topology changing high mobility nodes for secure communication and trust management. This framework will enhance the reliability of the critical applications area of FANETs.

2.6 OBJECTIVES

On the basis of literature and research gaps, the following objectives have been identified to be investigated under the research entitled on *“A Security Framework for Flying Ad-hoc Networks.”*

1. To study and analyze the existing secure routing protocols/ framework/ schemes for FANETs.
2. To design and develop a secure framework for FANETs.
3. To verify and validate the proposed secure framework.
4. To compare and evaluate it with other proposed secure protocols/ framework/ schemes.

CHAPTER SUMMARY

To prevent FANETs from various attacks due to node misbehaviour and selfishness, trust management in the networks can play a significant role. Trust management schemes can help to identify malicious and non-cooperative nodes in the network; can predict future behaviour of nodes, and discover trusted nodes. FANETs are highly dynamic in nature which involves more uncertainty and high chances of information loss. Hence, the use of trust management schemes in FANETs can be beneficial for successful mission accomplishment. In this chapter, we analyzed various trust management techniques that have already been implemented for traditional ad hoc network. It has been observed that Fuzzy logic,

Game theory and Markov model based trust schemes are the best suited for FANETs. These techniques provide the capability to handle uncertainty, non-cooperativeness, and frequent behavioural changes. The fuzzy technique has been used in the environments that involve uncertainty and incomplete information. FANETs are highly dynamic in nature which involves more uncertainty and high chances of information loss. Fuzzy logic is a suitable trust management model for such dynamic environments. The fuzzy model can include knowledge of human experts which helps to deal with the unknown situation that may occur in disaster situations handled by FANETs.

CHAPTER 3

THREAT MODELLING AND RISK ANALYSIS

FANETs face various security issues due to inherent infrastructure. So, it is important to design a threat model that can identify threats, quantify risks involved and address them. To understand the potential threats associated with FANETs, it is important to do a detailed study of various loopholes attached to its architecture and communicating network. Threat analysis helps to prioritize threats and design countermeasures. This work¹ intends to identify:

- i. Which are the vulnerable elements of the network?
- ii. Which attacks can exploit the vulnerability of these elements?
- iii. How risky these threats are and what are the impacts of these threats?
- iv. How can these attacks be mitigated and prevented?

3.1 FANET THREATS

In the context of computer security, a threat is a potential danger to the system that can exploit a vulnerability to break system security.

¹ The contents of this chapter have been reviewed and published in: Singh, K., & Verma, A. K., Threat Modeling for Multi-UAV Adhoc Networks, TENCON 2017.

The countermeasure is a process or action that can stop or reduce the effect of a threat to the system. Certain key factors are considered in defining a threat model (Spiewak, Engel, & Fusenig, 2006). In this paper, first different domains in FANET architecture have been identified which can be targeted by an adversary. Secondly, to know about the adversary capabilities and threat target is also essential to evaluate the threat. Furthermore, threats to each domain are identified, and impacts of these attacks on the targets are discussed. Figure 3.1 schematically represent various threats and their impact on different domains of FANETs.

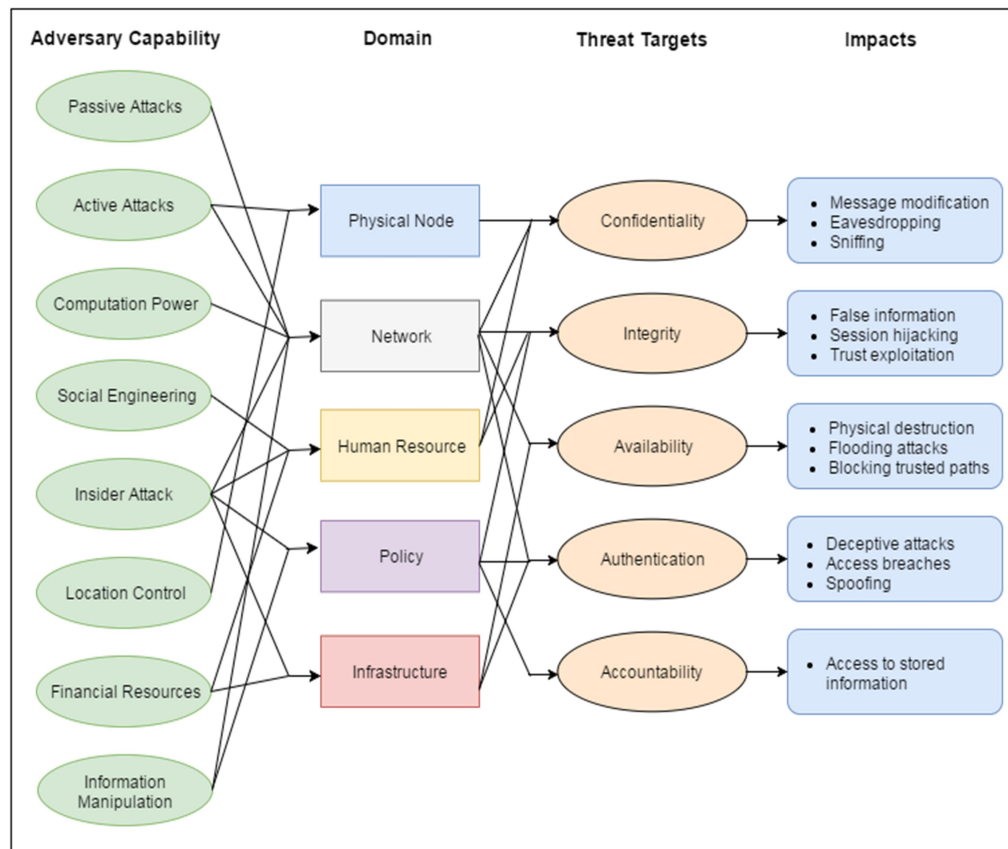


Figure 3.1: FANET Threats

3.1.1 Threats to Physical Node

A physical node in a FANET is represented by a single UAV that is prone to various kinds of threats which may become one of the reasons for the failure of the whole mission.

- i. Drone Collision:** When multiple UAVs fly to accomplish a mission, they usually move at the same time and same speed (Maddox, & Stuckenberg, (2015). This increase chance of collisions among UAVs. Another reason for drone collisions can be malfunctioning of hardware or software configuration, unauthorized access to UAVs or GCS module.
- ii. Theft and Vandalism:** UAVs that are flying at a short distance are more prone to theft and vandalism. This is being done by using various methods like simple dart guns or by anti-drone rifles (Altawy, & Youssef, 2016). Another method to disrupt UAVs is using a software virus such as Maldrone (Kozliuk, Bakhtiiarov, Lavrynenko, & Tretiak, 2016).
- iii. Environmental Threats:** Many environment-related threats like weather conditions, i.e., rainfall, thunderstorms do degrade the performance of UAVs. The performance degradation also depends on the size, shape, and energy of UAVs (Carr, 2014). Furthermore, civilian mission UAVs can collide with different disruptions like trees, electric infrastructure or buildings.
- iv. Unauthorized Access:** Another security threat to FANETs is unauthorized access to any of the UAV. This can lead to modification of its parameters, circuitry or even physical

damage of a UAV (Kozliuk *et al.*, 2016). Single misconfigured UAV can create unavoidable consequences for the mission.

These threats to physical nodes directly harm the confidentiality of FANETs which may lead to the disclosure of information and unavailability of resources during the mission.

3.1.2 Threats to Network

FANETs are used for critical missions. Therefore, the security of information and communication is an essential feature for such missions. However, FANETs are vulnerable to various threats listed below.

- i. Signal Jamming:** Jamming for UAV networks is defined as the interference of existing wireless communications by decreasing the signal-to-noise ratio at the receiver end through the transmission of interfering wireless signals. GPS signals are very prone to jamming attacks (Bekmezci, 2016). If GPS signals are jammed, UAVs may not get navigation directions. In this scenario, the aircraft could potentially lose the capability to monitor its route, calculate its location, altitude, and the direction in which it is traveling. Therefore, UAV collisions or crash may happen in the network. Another trick used by an adversary to get control of FANET is GCS control signal jamming (Sentürk, 2016). Once the UAV lose its control from GCS that UAV goes into lost link state where it navigates

according to GPS signals only. This may be a malicious move made by hackers to trap the UAVs.

ii. Node Compromise: Node compromise is one of the dangerous threats to FANETs. A node may possess multiple identities to pretend as multiple nodes in the networks. Eavesdropping and Sybil attacks are major threats to UAV nodes in the network (Javaid, Sun, Devabhaktuni, & Alam, 2012). Here, an untrustworthy or malicious UAV may take upon the identity of reliable UAV and participate in the communication. This can cause hazardous consequences for the mission, and it has been found difficult to detect as well.

iii. Signal Spoofing: A spoofing attack is a technique where the adversary masquerades as another by falsifying identity or data to gain access to illegitimate information. Signal spoofing falls under man in the middle attack umbrella. In FANETs, GCS control signals spoofing, UAV transmissions and GPS data spoofing may take place (Kim et al., 2012). During GCS control signals spoofing, the adversary disrupts the authentic communication between the UAV and the GCS and starts commanding the network herself. SkyJack software has been used to launch GCS spoofing attack. Similarly, spoofing the UAV transmissions and GPS data is another way opted by an adversary to control the UAVs. Spoofing the GPS information may mislead the location of UAV.

iv. False Data Injection: On a mission, UAVs collect data through sensors and a camera that is requested by the flight controller. Hacker has information about the commands followed by flight controller to collect data from sensors and cameras. Data captured by sensors can be easily manipulated (Kim *et al.*, 2012).

v. Routing Attacks: FANETs are used for very crucial missions as well as for civilian operations. The information shared during the communication is of high value. Routing in FANETs is prone to rushing attack, flooding, blackhole attack, wormhole attack, grey hole attack, *etc.* (Gupta, Jain R, & Vaszkun, 2015) to gain access to the network and its data. These attacks mainly either remove trust worthy nodes or misroute the information to malicious nodes. For example, in grey hole attack, hackers drop packets for a specific target node, after a certain time period or after a certain number of packets or randomly.

Threats to the network can harm most of the security requirements as shown in Figure 3.1. The impact of these threats is very severe and requires effective countermeasures.

3.1.3 Threats to Human Resource

Humans play a very crucial role in the completion of any typical mission. Human error may act as a major factor for various breaches. Misaddressed emails, confidential data leakage, sending information to insecure home systems, mistakes can be very costly. Hence, human

resources are one of the riskiest threats to FANETs (Strome, 2018).

Here we list various ways the human can create threats to FANETs.

- i. Social Engineering:** In the context of cybersecurity, Social engineering refers to psychological manipulation of people into performing actions or exposing confidential information. Phishing, Shoulder surfing, IVR phishing, baiting, pretexting, etc. are various techniques used by an adversary to information gathering, fraud, or system access (Saeed et al., 2016).
- ii. Inadequate Training:** Employees involved in the mission may not be aware of various cyber security aspects. Inadequate knowledge of various humans about various types of possible breaches, how they can be a reason of breach and how important their role in the security of mission may lead to unavoidable issues.
- iii. Poor Decision Making:** FANETs are being used in very critical missions where decision-making plays an important role. Due to lack of knowledge and awareness of various security threats, humans involved may take poor decisions. This can lead the mission to the very crucial stage.
- iv. Insider:** Insiders are the employees, partners, officials of the organization who disclose the valuable information about the company or mission. Insider threats are often launched by dissatisfied employees or ex-employees who either believes that the company has not been fair with them and feel justified in gaining revenge or got compromised by the greed of money.

It is said that humans are the weakest link in the security chain. Vulnerable human resources can breach confidentiality and integrity of the network.

3.1.4 Threats to Security Policy

A security policy is a document that how a company plans to protect the physical and network infrastructure. A security policy is "live document" that is never finished. It is updated as technological, communicational and employee requirements change (What is an IT Security Policy? - Palo Alto Networks. (2018). However, if this document is not properly reviewed and updated, it can harm authentication, integrity and accountability in FANETs.

- i. Improper Documentation:** Security policy should be designed and documented. Any ambiguity or opacity in the definition of the security policy may lead to costly consequences to FANETs.
- ii. Lack of Policy Implementation:** Another threat to security policy is that it is defined but never implemented by the mission operators.
- iii. Poor Update and Maintenance:** Security policy acts as a live document. This should be reviewed and updated as and when required during the mission. Poor updation and maintenance of policy is another threat to FANETs.

3.1.5 Threats to Infrastructure

Multi-UAV systems consist of mainly three main elements, which are the unmanned aircraft, the ground control station (GCS) and the

communication link. The aircraft contains an airframe, flight controller, a set of sensors, and actuators. Figure 1.2 provides a high-level architecture of a UAV system and its main elements. Here, we give a brief overview of the main threats to various components of FANETs that can be very harmful to availability and authenticity of the network.

- i. Ground Control Station:** Adversary use GCS control signal jamming to get control of FANET. Once the UAV lost its control from GCS, it transfers to lost link state where it navigates according to GPS signals only. This may be a malicious move made by hackers to trap the UAVs (Hartmann & Steup, 2013). During GCS control signals spoofing, the adversary disrupts the authentic communication between the UAV and the GCS and starts commanding the network herself. SkyJack software (Samy 2017) has been used to launch GCS spoofing attack.
- ii. Flight Controller:** Flight controller is a crucial component of UAV. Attack to the flight controller module may cause data disclosure, falsified data, and damage to infrastructure. An adversary can launch a denial-of-service attack to gain access to flight controller. Flight controllers are prone to sensor data manipulation, camera footage modification, malicious software, and hardware attacks (Roberts, 2016).
- iii. Communication Infrastructure:** Unsecure communication of FANET is a threat to the whole mission. Information exchanged between the UAV and GCS must be protected against

unauthorized disclosure in case of interception (Roberts, 2016). Another threat to communication infrastructure is its unavailability. An adversary can launch DOS or DDOS attacks against communication networks.

iv. Malicious Hardware/Software: Both the flight controller and ground control unit are vulnerable to hardware and software trojans. Such trojans can be either discretely designed in the system or transferred to it. SkyJack is an exploit for the Parrot AR Drone, and Maldrone is the payload (Samy 2017) Maldrone is installed as a backdoor. Virus-copter and SkyJack are malicious software that has been used to hack UAVs.

In table 3.1, comparison of some of the important proposed threat models for UAVs and ad-hoc networks is given. These threat models are compared by identification of threats under various domains and risk analysis and threat prioritization. Most of the models list threats under physical node and network domains. However, for better security mechanism it is also important to identify threats due to humans and security policy management. Even threats to network infrastructure are significant to consider for better security.

Table 3.1: Comparison of Various Threat Models

Threat Model Gaps	Cyber Security Threat Analysis and Modelling of an Unmanned Aerial Vehicle System (Javaid et al., 2012)	Threat Modelling for Ad Hoc and Wireless Sensor Networks (Mansfield et al., 2013)	Unmanned Aerial Vehicle Smart Device Ground Control Station Cyber Security Threat Model (Clark et al., 2007)	Enhanced C4I Security using Threat Modelling (Alghamdi, Hussain, and Khan, 2010)	Threat Model for Multi-UAV Adhoc Network [Proposed]
Identifies the threat to physical nodes	✓	✓	✓	✓	✓
Identifies the threat to network	✓	✓	✓	✓	✓
Identifies the threat to human	×	×	✓	×	✓
Identifies the threat to policy	×	×	×	×	✓
Identifies the threat to infrastructure	×	✓	×	×	✓
Risk Analysis and Threat Prioritization	✓	×	✓	✓	✓

3.2 RISK ANALYSIS

After designing a threat model for FANETs, it is essential to evaluate the overall effect of the attacks on the security requirements, i.e., confidentiality, integrity, and availability. To assess the cost of these attacks, we formulate a quantitative method to measure confidentiality, integrity, and availability. In Figure 3.2, a block diagram of the proposed threat model is shown.

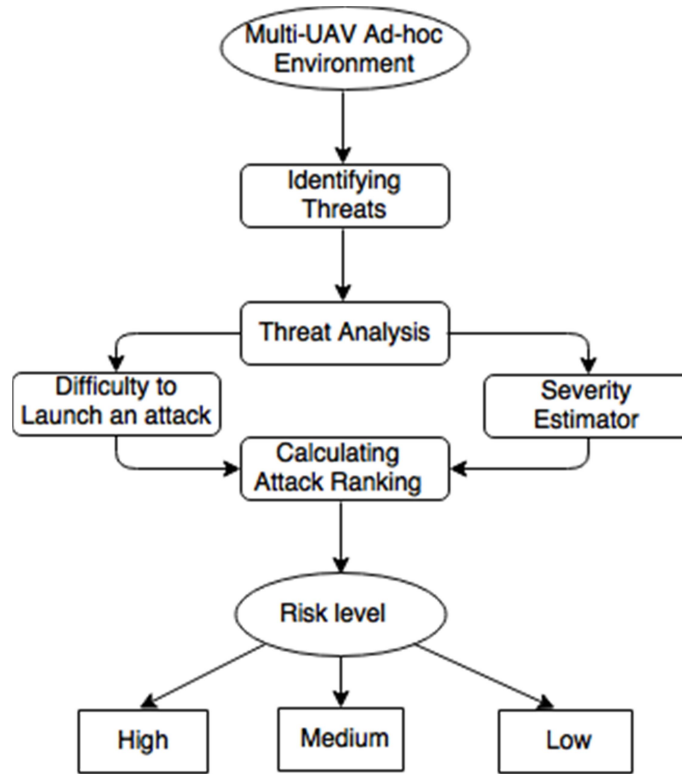


Figure 3.2: Proposed Threat Model for FANETs

Initially, various threats are identified from the FANETs environment. These threats are classified under various domains, i.e., physical node, network, infrastructure, human, and policies. Then, threat analysis is done based on the severity of the attack and the difficulty level to perform that attack. Further, the rank of attack is calculated by using the value of attack severity and difficulty. Finally, based on the rank, the attack is classified under one of the three risk level (high, medium and low).

3.2.1 Impact on Confidentiality

Confidentiality of data indicates the protection of data from unauthorized access. The impact of the attack on confidentiality of network is calculated using equation 3.1.

$$C(n, s) = \sum_i^l n_i s_i \quad \dots (3.1)$$

Here C represents the impact of the attack on the confidentiality of data in the network. The function has two parameters, i.e., the number of affected nodes (n) and their respective data sensitivity. Sensitivity of data is defined as three ($l=3$) levels: $L = \{\text{low, medium, high}\}$, which is application dependent (Information Classification, 2017).

3.2.2 Impact on Integrity

Integrity refers to protecting data from being modified by unauthorized parties. The impact of an attack on the integrity of the network is calculated using equation 3.2.

$$I(d, d') = \sum_i^N (d_i - d'_i)^2 \quad \dots (3.2)$$

Here, I represent the impact of the attack on the integrity of data in the network. Integrity is calculated using two parameters: d_i i.e., actual data node i at time t and d'_i i.e., the modified data attacker manages to transmit in the network. Here, N represents the total number of nodes in the network.

3.2.3 Impact on Availability

Availability of data refers to ensuring that authorized parties can access the data when needed. The impact of an attack on the availability of network is calculated using equation 3.3.

$$A(n, t) = \sum_i^T n_i t_i \quad \dots (3.3)$$

Here, A represents the impact of an attack on the availability of data in the network. The function has two parameters, the affected nodes (n) and their time (t) duration under seized. The time duration for which network nodes are unavailable is defined at three levels: T = 3 (short, average, long).

3.2.4 Easiness

Easiness refers to the efforts required to launch an attack on the network. Easiness “E” is the aggregation of all the parameters which represent the easiness to perform an attack on the network.

$$E = \frac{H+S+PA}{12} \quad \dots (3.4)$$

In equation 3.4, H represents the requirement of hardware resources other than PC/Laptop. It is divided into three cases: in the first case, there is no other hardware required other than PC or Laptop for launching an attack, in the second case, basic hardware other than PC, in the third case, the advanced hardware is required to attack the network. As various attacks can be launched on a network, but each attack requires different skills to be successfully launched. Skill is divided into three levels and represented by (S). In the first case, no expert knowledge is required. In the second case, tools for launching attacks are required which are easily available in the public domain, and in the third case, advanced skills are required to perform an attack. PA depicts the requirement of physical access to initiate an attack on the network. In table 3.2, each parameter is ranked for

different cases on the scale of 1-4 where 1 represents easiness, and 4 represents difficulty. The equation is normalized by dividing numerator by the maximum total score of easiness, i.e., 12 that can be assigned to attack.

Table 3.2: Score Based on the Difficulty Level to Perform an Attack.

Criteria	Case	Value
Extra Hardware Requirement (<i>H</i>)	No extra hardware required	1
	Basic hardware other than PC/Laptop	2
	Advance hardware requirement	4
Skills Required (<i>S</i>)	No expert knowledge required	1
	Tools are available in the public domain	2
	Advance skills needed	4
Physical Access Required (<i>PA</i>)	Physical access required	4
	Physical access not required	1

Rank of an attack defines the risk level of attack for the system. It also defines the level of complexity involved in a particular attack.

$$\text{Attack}_{\text{rank}} = [\{\alpha \times C(n, s) + \beta \times I(d, d') + \gamma \times A(n, t)\} + (1 - E)] \quad \dots (3.5)$$

Equation 3.5 calculates the rank of an attack by aggregating the impact of the attack on the security objectives, i.e., confidentiality, integrity, and availability. It also calculates the effort required to launch that attack. We have used (α, β, γ) as weights for confidentiality, integrity, and availability respectively. These weights are used to give priorities according to the requirement of the application. In our analysis, we have used the equal weights for calculating the rank of attacks. The lower value of $\text{Attack}_{\text{rank}}$ represents the high risk associated with an attack.

- **Example:** We have explained the same with an example of jamming attack, which is an attack on the resource availability. The impact can be calculated using equation $A(n, t) = \sum_i^T n_i t_i$. The number of network nodes affected is directly proportional to the area covered by the jamming device. Let jamming radius be r , and the network node density is $\rho = \frac{\text{number of nodes}}{\text{the deployment area}}$. So, the number of network nodes affected by jamming attack is $n = \pi r^2 \times \rho$. Assume that the attack is for t seconds which is divided into three ($T = 3$) intervals. Let some nodes (n_1) are not available for short duration (t_1) and some other nodes (n_2) are of average duration time (t_2) and remaining nodes (n_3) are of long duration (t_3), where $n_1 + n_2 + n_3 = n$ and $t_1 + t_2 + t_3 = t$. The number of packets transmits is P in one unit time. Then the total packet lost due to resource unavailability is calculated as $\sum_i^T (n_i t_i) \times P$. Now, the ease of launching the jamming attack can be calculated using table 3.2. Further ranking of attack can be done by aggregating the impact and difficulty to perform that attack.

3.3 ATTACK RANKING

In this section, the impact of various attacks on security metrics (i.e., confidentiality, integrity, and availability) has been calculated using equation from 3.1-3.3. Furthermore, the attack score is calculated by

using equation 3.5. The analysis of various threats assumes that there is no security mechanism in place. Threats to FANET have been identified and categorized under different security metrics: confidentiality, integrity, and availability given in table 2. The impact of an attack based on its difficulty level for each security requirement is represented by a 2-D diagram.

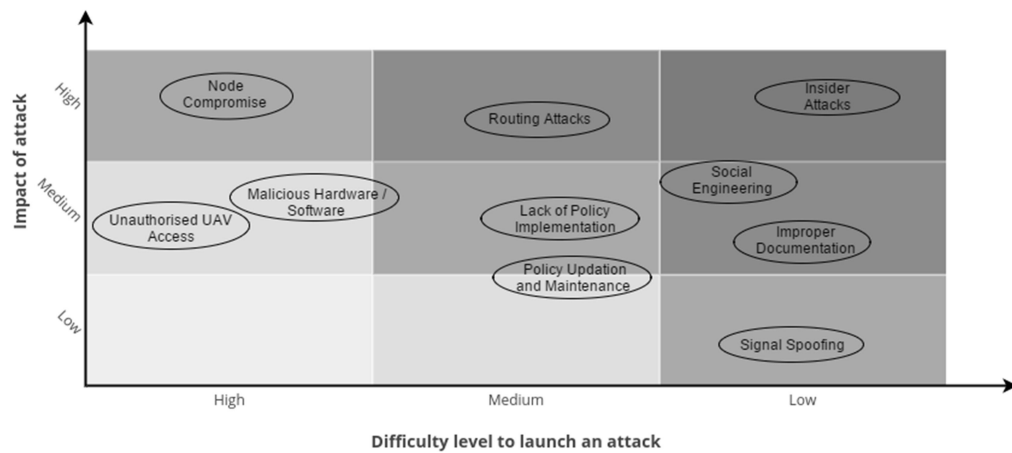


Figure 3.3: A 2-D Graph Representation of Difficulty Level versus Impact on Confidentiality of Different Attacks.

As seen in Figure 3.3, the whole area is divided into three difficulty levels and three impacts the insider attack is easy to carry out and can highly impact confidentiality. Detection of insider attack is also crucial hence; the remedy for such attacks will enhance the network performance and reliability. Policy implementation related threats are also significant threats to mitigate.

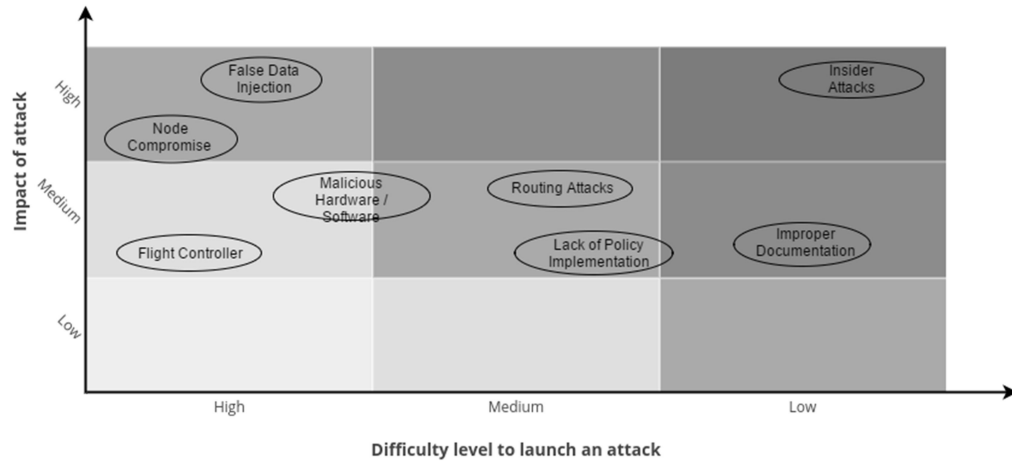


Figure 3.4: A 2-D Graph Representation of Difficulty Level versus Impact on Integrity of Different Attacks.

Figure 3.4 shows that although node compromise and false data injection are difficult to launch, but are the most critical attacks that impact the integrity of the network. Improper documentation of network policy can be easily launched with medium impact on integrity.

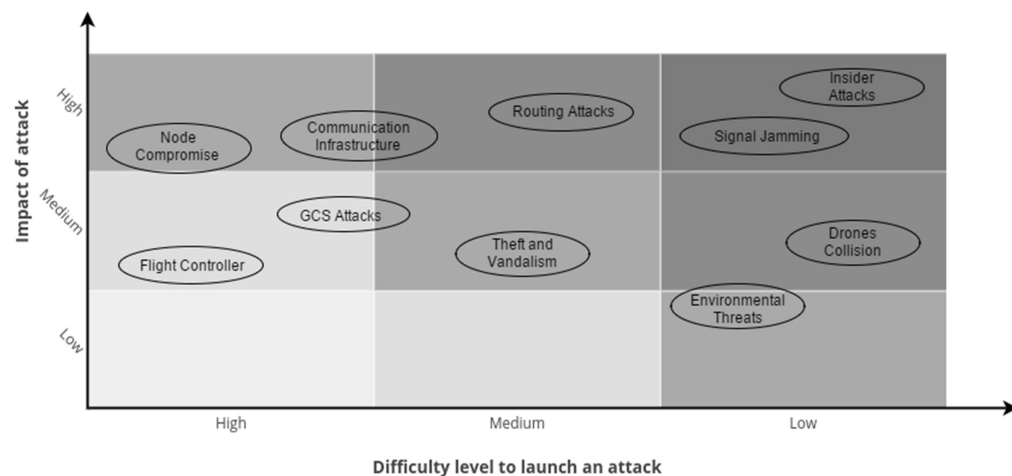


Figure 3.5: A 2-D Graph Representation of Difficulty Level versus Impact on Availability of Different Attacks.

Figure 3.5 shows that drone collision, signal jamming, and insider attacks can easily harm availability of the network. Attacks on GCS, node compromise, and communication infrastructure are difficult to launch and capable of high impact to network availability. It is evident from our findings that achieving confidentiality, integrity, and availability is an essential need for FANETs. By implementing basic countermeasures to ensure confidentiality, integrity, and availability, threats due to insider attacks, data injections, policy implementation, etc. can be mitigated.

3.4 DEFENCE MECHANISM

As we have discussed in previous sections, FANETs are prone to various security threats. Also, the most of the conventional security techniques were designed to attempt single threat and counter it individually. In other words, defending to a specific type of threat (e.g., physical node or network) may not inevitably be sufficient to prevent from other threats. Therefore, it is important to pay attention towards all possible threats to the multi-UAV ad-hoc network. For a successful and smooth accomplishment of FANET operations, detection and prevention mechanism should be identified and implemented properly. Figure 3.6, describes various possible threats and their defense mechanism (Angelov, 2012; Wu, Chen, Wu, & Cardei, 2007; Mansfield, Eveleigh, Holzer, & Sarkani, 2015) under different domains for FANET.

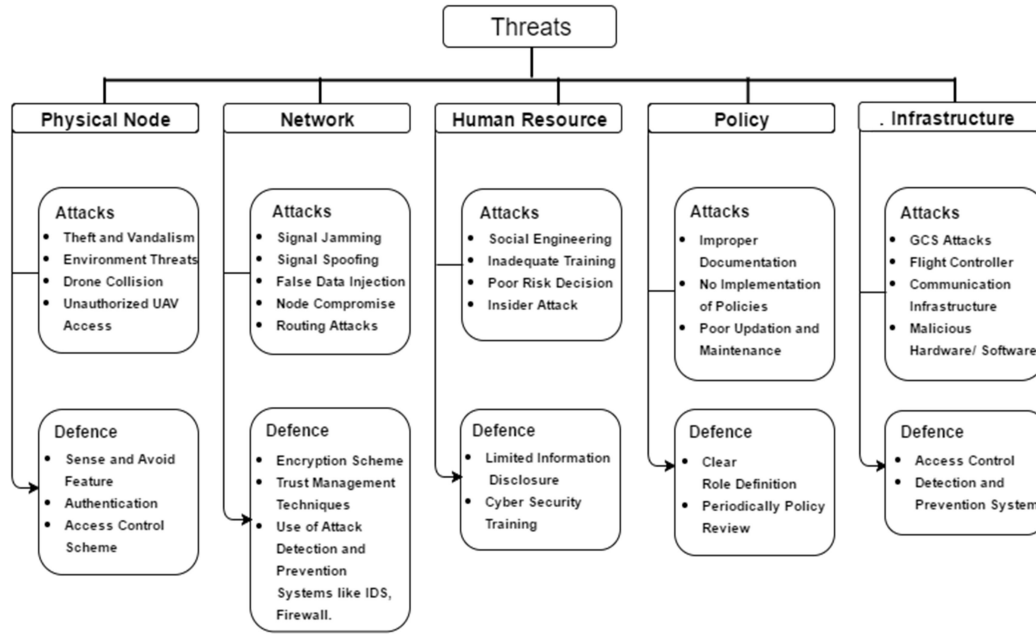


Figure 3.6: Defence Mechanism Against various Attacks in FANETs.

3.4.1 Defence for Physical Node

Physical nodes are prone to unauthorized access, drone collision, theft and environmental threats. To defend UAVs from the collision, collision avoidance algorithms such as sense-and-avoid algorithm (Angelov, 2012) should be implemented among the UAVs. However, such algorithms alone may not be sufficient for collision avoidance. An adversary may use such attacks to launch a denial-of-service attack by creating factious traffic in the routes of flying nodes. Hence, an additional inter-drone communication protocol or ADS-B systems are essential between UAV networks.

3.4.2 Defence for Network

To prevent UAVs from network related threats, the cryptographic techniques, trust management, and secure routing techniques are

essential. To secure the communication between the UAV and GCS, fast and efficient algorithms such as symmetric block or stream ciphers, hash functions, and MACs need to be implemented. Signcryption is another technique with no overhead on the communication bandwidth. Another key defense technique for detection and prevention of threats is intrusion detection systems which can be used for behaviour profiling. Trust management schemes could also be used to formulate a trustworthy network between UAVs.

3.4.3 Defence for Human Resource

Humans are the weakest link in the security chain. They are prone to mistakes, which help cybercriminals to launch attacks. These errors are harmful as the insiders have access to most sensitive information. Hence, appropriate access control, identification of insider intruders and train them against social engineering attacks will help to reduce threats from human resources.

3.4.4 Defence for Policy

An appropriate policy document that contains clear role definitions and their access to the infrastructure is very important to build up a secure network. Policies are loosely followed and rarely revised which leads to various threats. Clear assignment of roles and their responsibilities will help to protect the network from many attacks.

3.4.5 The Defence for Infrastructure

An efficient solution to protect infrastructure from spoofing attacks on GCS and flight controller module is to use of redundant sets of sensors, and the utilization of GPS receivers with multiple antennas. Access control algorithms, anti-malware software, firewalls, intrusion detection and prevention systems are software-based solutions that provide a secure layer to underlying hardware components.

3.5 CHAPTER SUMMARY

Security of FANET is a challenging and essential area of research. With the high demand and utility of such ad-hoc network, identification and mitigation of various vulnerabilities are one of the early milestones to be achieved. Aiming to achieve these milestones, this paper presented a complete threat analysis for FANETs. The proposed threat model identified the major target domains and discovered the potential threats to each domain of FANETs. The model takes into account hardware, software, and communication infrastructure to identify various threats. Further, risk analysis model was used to rate various threats to FANETs. Finally, suggestions were provided for possible security solutions against identified threats.

CHAPTER 4

FCTM: FUZZY CLASSIFICATION TRUST MODEL

An ad hoc network lacks pre-established infrastructure. As a result, the network node assumes the responsibility of router, client, and server at different times. Cooperation and coordination among network nodes are very crucial for the performance of this kind of networks (Perkins, 2001). It is assumed that nodes will cooperate and coordinate according to the underlying protocols and applications. However, this assumption does not suffice, because nodes may behave maliciously or selfishly resulting in decreased network performance, vulnerable to attack(s), and high resource consumption (Chang, Li, & Tang, 2016). Motivated by the above challenges, a fuzzy classification trust model (FCTM²) is proposed for FANETs. This model makes use of principles of fuzzy classification and optimization (Wang, Gao, & Ovaska, 2008) to manage trust relationships among nodes of FANETs. FANETs are highly dynamic in nature which involves more uncertainty and high chances of information loss. Fuzzy logic is a suitable trust management model for such dynamic environments

² The contents of this chapter have been reviewed and published in: Singh, K., & Verma, A. K., FCTM: A Novel Fuzzy Classification Trust Model for Enhancing Reliability in Flying Adhoc Networks (FANETs). Ad Hoc and Wireless Sensor Networks. (SCI Indexed Journal).

(Gnana, 2016). There are various advantages of trust management- malicious node can be identified and isolated, evading of data forwarding to selfish and malicious nodes and it helps to select a higher trust level node, resulting in improvement of overall performance of FANETs.

4.1 PROPOSED TRUST MODEL

The proposed trust model makes use of fuzzy classification and optimization, therefore is named as Fuzzy Classification Trust Model (FCTM). In FCTM, the nodes are classified using node's behaviour and performance parameter in the network. To segregate the selfish and malicious nodes, social parameters are used for evaluating the trust value of each node. A decay function is also considered to replicate the real-world behaviour of the UAVs. With the help of experiments, the best trust aggregation weights between QoS and social parameters are identified to classify the network nodes.

4.1.1 Model Description

FCTM primarily utilizes the fuzzy classification method to distinguish among network nodes considering its social parameters and quality of service parameters. Social parameters account the past behaviour of nodes, whereas the quality of service parameters considers the current performance of nodes in the network.

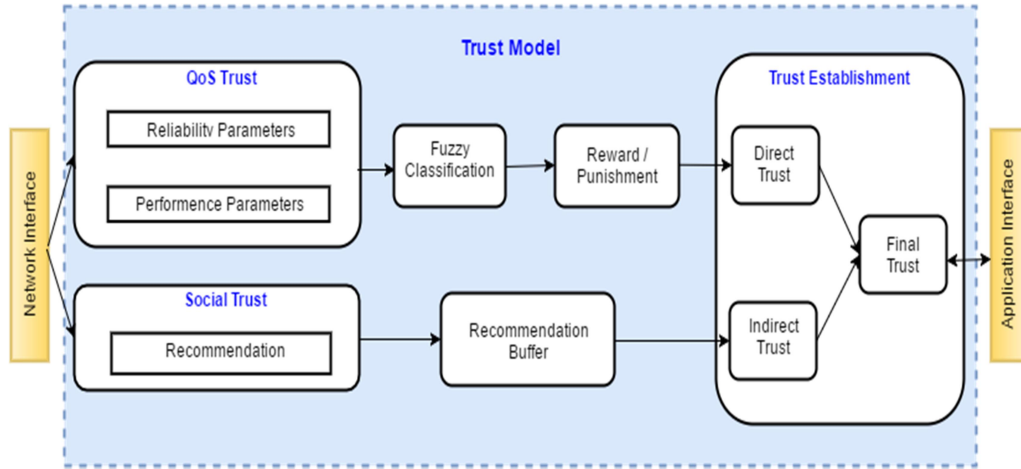


Figure 4.1: Proposed Fuzzy Classification Trust Model (FCTM)

The primary aim of this model is to classify the network nodes in various clusters: good, neutral and bad. Here, the fuzzy membership function has been defined using the expert's domain knowledge. These membership functions were used to assign the initial cluster of the network nodes. Further, to improve cluster center fuzzy optimization was applied, which reallocate the cluster of network nodes and redefine the cluster center for each cluster. Based on these optimized parameters, the network nodes are classified.

To calculate the trust values of network nodes, model introduced the concept of reward and punishment according to their cluster. At the start of simulations, the nodes are initialized with high trust value. A decay function is used based on the cluster allocated to the particular node. The recommendation is stored in recommendation buffer. Both the values are fed to trust the establishment system as shown in figure 4.1. Trust establishment system calculates direct trust and indirect trust. Final trust is calculated based on weights of direct and indirect trust values.

4.1.2 Fuzzy Classification

Let N be a set of total n nodes that create an ad hoc network is defined as $N = \{N_i | i = 1, 2, 3, \dots, n\}$ and P is set of m attributes (i.e. signal strength, packet delivery ratio, node's energy, and transmission delay) used to evaluate ad hoc network performance is defined as $P = \{P_i | i = 1, 2, 3, \dots, m\}$. Every node N_i evaluates the values for each parameter in set P from other nodes and computes a characteristic vector $c_i = \{c_{i1}, c_{i2}, c_{i3} \dots c_{im}\}$. Using the characteristic vectors of each node, characteristic matrix for whole network can be defined as $C = (c_{ij})_{n \times m}$. Characteristic matrix C is influenced by many factors. Before analysis of characteristics matrix the values should be normalized. Normalized matrix is defined as $R = (r_{ij})_{n \times m}$. Normalization of the values can be done by the equation 4.1.

$$r_{ij} = \frac{c_{ij} - c_{imin}}{c_{imax} - c_{imin}} \quad \dots (4.1)$$

Where c_{ij} = value of parameter j of i^{th} node, c_{imin} Represents the minimum attribute value of the i^{th} parameter, c_{imax} Represents the maximum attribute value of the i^{th} parameter. Let Cl be a vector of c clusters for node classification. Each node will be classified into c clusters defined as $Cl = \{\text{Bad, Neutral, and Good}\}$. First, we calculate the membership of each node in each cluster. The membership matrix U is defined as $U = (u_{ij})_{c \times n}$. Where, c represents the number of clusters and n represent the number of nodes. u_{ij} represent the membership of j^{th} node in i^{th} class. The constraint on u_{ij} is as shown in equation 4.2:

$$\sum_{i=1}^c u_{ij} = 1 \quad \dots (4.2)$$

The initial cluster centre is calculated by using the membership functions for each parameter. Membership of node j in i^{th} cluster is represented as u_{ij} and calculated using equation 4.3.

$$u_{ij} = \left[\sum_{j=1}^c \left[\frac{\sum_{k=1}^m (r_{ik} - cc_{ij})^2}{\sum_{k=1}^m (r_{ik} - cc_{jk})^2} \right] \right]^{-1} \quad \dots (4.3)$$

The membership of each node in a cluster is calculated, and the node is assigned to a cluster using equation 4.4.

$$clusterlabel = \operatorname{argmax}(u_{ij}) \quad \dots (4.4)$$

Cluster centre are stored in matrix computed as $CC = (cc_{ij})_{m \times c}$. Optimized membership matrix U is used to calculate optimized cluster centers using the following equation 4.5.

$$cc_{ij} = \frac{\sum_{j=1}^n u_{ij}^2 r_{ij}}{\sum_{j=1}^n u_{ij}^2} \quad \dots (4.5)$$

4.1.3 Objective Function

To get final membership matrix U and cluster centers CC , optimize the following function in equation 4.6.

$$J = \sum_{i=1}^n \sum_{j=1}^c \left\{ u_{ij}^f \sum_{k=1}^m (r_{ik} - cc_{jk})^2 \right\} \quad \dots (4.6)$$

Where f is fuzzy coefficient, here $f = 2$. Once we found all the optimized values, we will use a threshold value to stop the optimization process.

Algorithm 4.1: Fuzzy Classification in FCTM

Input: Characteristic matrix C , Number of clusters c , Number of nodes n .

Output: Cluster centers $CC = \{cc_1, cc_2, \dots, cc_n\}$, membership matrix $U = (u_{ij})_{c \times n}$

-
1. $N = \{N_i | i = 1, 2, 3, \dots, n\}$ // N is a set of total n nodes in the network
 2. $P = \{P_i | i = 1, 2, 3, \dots, m\}$ // P is a set of m attributes
 3. Computes a characteristic vector $c_i = \{c_{i1}, c_{i2}, c_{i3}, \dots, c_{im}\}$ and define C as $C = (c_{ij})_{n \times m}$.
 4. Normalize matrix C to Recognition matrix $R = (r_{ij})_{n \times m}$.
 5. Use membership functions in section 4.2 to calculate the initial membership matrix U
 6. **While** (iterations < n)
 7. Calculating cluster centers using membership matrix U
 8. **for** $i = 1: c$
 9. **for** $j = 1: n$
 10.
$$cc_{ij} = \frac{\sum_{j=1}^n u_{ij}^2 r_{ij}}{\sum_{j=1}^n u_{ij}^2}$$
 11. **end**
 12. **end**
 13. Calculating the value of the objective function and compare $|J(\text{iteration}) - J(\text{iteration}-1)|$ with a threshold value. Exit the while loop if the criteria function is less than threshold values.
 14. **for** $i = 1: c$
 15. **for** $j = 1: n$
 16.
$$J = \sum_{i=1}^n \sum_{j=1}^c \{u_{ij}^f \sum_{k=1}^m (r_{ik} - cc_{jk})^2\}$$
 17. **end**
 18. **end**
 19. **if** (iteration ~ 1) & ($|J(\text{iteration}) - J(\text{iteration}-1)| < \delta$)
 20. **break;**
 21. **end**
 22. Now, optimizing the membership matrix U using cluster center CC
 23. **for** $i = 1: c$
 24. **for** $j = 1: n$
-

```

25.          for  $k = 1: c$ 

26.               $u_{ij} = \left[ \sum_{j=1}^c \left[ \frac{\sum_{k=1}^m (r_{ik} - cc_{ij})^2}{\sum_{k=1}^m (r_{ik} - cc_{jk})^2} \right] \right]^{-1}$ 

27.          end

28.      end

29.  end

30.  Similarly, recompute cluster center  $CC$  as in step 10.

31.  Assign cluster to nodes using membership matrix  $U$ 

```

$$cluster_label = argmax(u_{ij})$$

4.1.4 Trust Calculation

Initially, all nodes are assigned same trust value. The direct trust is calculated by adjacent nodes through direct observation, and the indirect trust is the recommended value shared by neighbouring nodes. All nodes are classified in different clusters (bad, neutral and good) and the node gets punished or rewarded accordingly, which is shown in Table 4.1.

Table 4.1: Reward and Punishment

Cluster	Rewards	Punishments
Bad	0	0.5
Neutral	1.5	0
Good	2	0

First, trust value T_i^w of each node i is calculated by assigning equal weights to each parameter using equation 4.7:

$$T_i^w = \sum_{j=1}^m w_j r_{ij} \quad \dots (4.7)$$

Further, to calculate direct trust T_i^d values, we apply the exponential decay function (Leike, 2001) as follows:

$$T_i^d = a.b^x \quad \dots (4.8)$$

Here, a refers to previous trust value, b refers to reward based on cluster assignment, x refers to the trust evaluated using weights (T_i^w).

A final trust value for each node is calculated using equation 4.9:

$$T_i^f = \alpha T_i^d + \beta T_i^R \quad \dots (4.9)$$

Where α and β are the weights given to direct observations and recommendations. The sum of α and β should be equal to one. T_i^d refers to direct trust and T_i^R refers to the indirect trust value.

Algorithm 4.2: Trust Calculation in FTCM

Input: Weight vector W , Recognition matrix R , Number of nodes n , Classified node vector cluster label, Punishment and Reward values.

Output: Final trust vector for n nodes.

1. Calculating trust values T_i^w for each node i based of optimized weights and recognition matrix
 2. **for** $i = 1 : n$
 3. $T_i^w = \sum_{j=1}^m w_j r_{ij}$
 4. **end**
 5. Reward and punishment for the nodes based on cluster assignment
 6. **for** $i=1: n$
 7. **if** $\text{strcmp}(\text{cluster_label}\{i\}, 'Bad') = 1$
 8. $\text{Reward}(i)=0.5;$
 9. **end**
 10. **if** $\text{strcmp}(\text{cluster_label}\{i\}, 'Neutral') = 1$
 11. $\text{Reward}(i)=1.5;$
-

```

12.          end
13.          if strcmp (cluster_label{i},'Good') == 1
14.              Reward(i)=2;
15.          end
16.      end
17. Applying an exponential decay function for calculating final direct trust
   values  $T_i^d$ 
18.      for i = 1 : n
19.           $T_i^d = a.b^x$ 
20.      end
21. Calculating the final trust value for  $T_i^f$  every node
22.      for i = 1 : n
23.           $T_i^f = \alpha T_i^d + \beta T_i^R$ 
24.      End

```

4.2 IMPLEMENTATION

In this section, various parameters have been chosen that define the behavior of a node and their membership in node classification.

4.2.1 Key Attribute Selection

For the trust evaluation of a network node, key attributes are responsible, and they should be defined carefully. To make the trust evaluation process efficient, defined attributes should reflect the characteristics and behaviour of a mobile node in the ad hoc networks (Jadidoleslamy, 2015). In the trust evaluation of a node in FANETs, we should consider both node's behaviour properties and network

features. When evaluating node trust through its behaviour, it is not only direct observation which can be used, but a recommendation from other neighbour nodes in the network can be used to get the more accurate trust value of that node. Node mobility is also an important factor for the reliability of that node. The higher is the mobility; the higher is the packet drop rate. Network communication attributes signal strength, and packet delivery ratio plays an important role to evaluate trust of a node. We have taken three reliability attributes, and one performance attributes into account in the trust assessment of a node. Node's energy, signal strength, and the packet delivery ratio reflect the reliability of a node while transmission delay reflects the performance. So, key attributes set for calculating trust value of a FANET node are $A = \{\text{signal strength, packet delivery ratio, node's energy, and transmission delay}\}$.

4.2.2 Membership Functions

The four key attributes {signal strength, packet delivery ratio, node's energy, and transmission delay} proposed to be used for calculating the trust value in FANETs are discussed along with their membership function.

- i. **Signal Strength:** Free space path loss formula is used for calculating signal strength for different working distance in FCTM. The relation between signal attenuation and distance is calculated as $FSPL(dB) = 10 \log_{10} ((\frac{4\pi}{c} df)^2)$ (Sklar, 2001), where $FSPL(dB)$ stands for signal attenuation, measured in dB,

d represents the distance from the transmitter, c stands for light speed in vacuum (2.99792458×10^8 m/sec), and f stands for signal frequency. A Device with 802.11n runs at 2.4GHz frequency. By using above formula we can have signal attenuation for working distance as: for a distance of 170m, 320m and 500m are 84.65dBm, 90.14 and 94.02 respectively with network transmission power is 20dBm. A node is considered as a good node if it has signal strength more than -64.65dBm ($3.42 \times 10^{-7}mW$) and will be considered as a neutral node if signal strength is approximately -70.14dBm ($0.968 \times 10^{-7}mW$), and considered as a bad node if signal strength is around -74.02 ($0.396 \times 10^{-7}mW$). Signal strength value is normalized using ($x/3.42$).

For an element 'a' of the first row in the data matrix R, 'a' belongs to the "bad" with membership function:

$$F_{bad}(a) = \begin{cases} 1 & a \leq 0.11 \\ \frac{0.22 - a}{0.11} & .11 < a \leq 0.22 \\ 0 & a > 0.22 \end{cases}$$

Node 'a' belongs to neutral, with membership function:

$$F_{neutral}(a) = \begin{cases} 0 & a \leq 0.22 \\ \frac{a - 0.22}{0.12} & 0.22 < a \leq 0.34 \\ \frac{0.68 - a}{0.34} & 0.34 < a \leq 0.68 \\ 0 & a > 0.68 \end{cases}$$

Node 'a' belongs to good, with membership function:

$$F_{good}(a) = \begin{cases} 0 & a \leq 0.68 \\ \frac{a - 0.68}{0.27} & 0.68 < a < 1 \\ 1 & a \geq 1 \end{cases}$$

- ii. Packet Delivery Ratio (PDR):** Based on general network evaluating principle (Rappaport, 2002), a node having more than 95% of packet delivery ratio is said to be reliable and considered as a node with a good reputation. A node with 70% of delivery drop ratio is considered as a node with neutral reputation, and if a network node has 35% packet delivery ratio, then it will be a non-reliable node and considered as a node with a bad reputation.

For an element 'a' of the first row in the matrix X, 'a' belongs to the "bad" with membership function:

$$F_{bad}(a) = \begin{cases} 1 & a \leq 0.35 \\ \frac{0.55 - a}{0.20} & 0.35 < a \leq 0.55 \\ 0 & a > 0.55 \end{cases}$$

Node 'a' belongs to neutral, with membership function:

$$F_{neutral}(a) = \begin{cases} 0 & a \leq 0.45 \\ \frac{a - 0.45}{0.20} & 0.45 < a \leq 0.65 \\ \frac{0.85 - a}{0.20} & 0.65 < a \leq 0.85 \\ 0 & a > 0.85 \end{cases}$$

Node 'a' belongs to good, with membership function:

$$F_{good}(a) = \begin{cases} 0 & a \leq 0.75 \\ \frac{0.95 - a}{0.20} & 0.75 < a \leq 0.95 \\ 1 & a \geq 0.95 \end{cases}$$

- iii. Node's Energy:** The energy level of a mobile node determines, how much that node is capable of handling the network task and for how long time it can work properly. A network node is having energy more than 70% than it is said to be reliable and

considered as a node with a good reputation. A node with more than 40% of energy is considered as a node with a neutral reputation. A node with less than 40% of its total energy will be a weak node and considered as a node with a bad reputation.

For an element ‘a’ of the first row in the matrix X, ‘a’ belongs to the “bad” with membership function:

$$F_{bad}(a) = \begin{cases} 1 & a \leq 0.20 \\ \frac{0.40 - a}{0.20} & 0.20 < a \leq 0.40 \\ 0 & a > 0.40 \end{cases}$$

Node ‘a’ belongs to neutral, with membership function:

$$F_{neutral}(a) = \begin{cases} 0 & a \leq 0.40 \\ \frac{a - 0.40}{0.20} & 0.40 < a \leq 0.60 \\ \frac{0.80 - a}{0.20} & 0.60 < a \leq 0.80 \\ 0 & a > 0.80 \end{cases}$$

Node ‘a’ belongs to good, with membership function:

$$F_{good}(a) = \begin{cases} 0 & a \leq 0.70 \\ \frac{0.90 - a}{0.20} & 0.70 < a \leq 0.90 \\ 1 & a \geq 0.90 \end{cases}$$

iv. Transmission Delay: Transmission delay is one of the network performance parameters, and it can be used to evaluate the trust of a node. If a node is selfish or having malicious intent than it will transmit data packets with delay. Transmission delay(D_T) = L/R , where L stands for is the data size and, R represents the rate of transmission (Kurose, & Ross, 2007). So, with packet size 526Byte and 20Mbps transmission rate, transmission delay will be 0.61ms. Node having transmission

time near to $0.61ms$ will be considered as good node, node taking transmission time around $0.96ms$ is typically neutral node, whereas node with more than $1.47ms$ will be considered as bad node. We have assumed that in worst case scenario transmission delay will not be exceed $2.45ms$.

For an element ‘a’ of the first row in the matrix X, ‘a’ belongs to the “bad” with membership function:

$$F_{bad}(a) = \begin{cases} 1 & a \leq 0.53 \\ \frac{0.73 - a}{0.20} & 0.53 < a \leq 0.73 \\ 0 & a > 0.73 \end{cases}$$

Node ‘a’ belongs to neutral, with membership function:

$$F_{neutral}(a) = \begin{cases} 0 & a \leq 0.73 \\ \frac{0.84 - a}{0.11} & 0.73 < a \leq 0.84 \\ 0 & a > 0.84 \end{cases}$$

Node ‘a’ belongs to good, with membership function:

$$F_{good}(a) = \begin{cases} 0 & a \leq 0.84 \\ \frac{a - 0.84}{0.16} & 0.84 < a < 1 \\ 1 & a \geq 1 \end{cases}$$

Here, we summarize the procedure of trust evaluation in proposed FCTM. Six nodes are given with their attribute values for trust evaluation in table 4.2.

Table 4.2: Attribute Values of Network Nodes

Node	Signal 10^{-7} (mW)	Packet Delivery Ratio (%)	Node Energy (%)	Transmission Delay (ms)
N1	0.82	56	63	1.34
N2	2.01	67	78	0.71

N3	1.19	53	62	0.89
N4	0.71	39	42	1.21
N5	1.47	48	55	0.95
N6	3.18	81	95	0.78

Characteristics matrix is defined as:

$$C = \begin{bmatrix} 0.82 & 56 & 63 & 1.34 \\ 2.01 & 67 & 78 & 0.71 \\ 1.19 & 53 & 62 & 0.89 \\ 0.71 & 39 & 42 & 1.21 \\ 1.47 & 48 & 55 & 0.95 \\ 3.18 & 81 & 95 & 0.78 \end{bmatrix}$$

For the standard, the signal strength is changed by (signal / 3.42), and the second and third column to the percentage on a scale of 1. Transmission delay is normalized by using (2.45-transmission delay)/1.84)).

$$C = \begin{bmatrix} 0.24 & 0.56 & 0.63 & 0.60 \\ 0.59 & 0.67 & 0.78 & 0.94 \\ 0.34 & 0.53 & 0.62 & 0.84 \\ 0.21 & 0.39 & 0.42 & 0.67 \\ 0.43 & 0.48 & 0.55 & 0.81 \\ 0.93 & 0.81 & 0.95 & 0.90 \end{bmatrix}$$

To surmount the influence of different dimensions, the characteristic matrix is changed into the recognition matrix as R :

$$R = \begin{bmatrix} 0.041 & 0.527 & 0.180 & 0 & 0.305 & 1 \\ 0.404 & 0.666 & 0.333 & 0 & 0.214 & 1 \\ 0.396 & 0.679 & 0.377 & 0 & 0.245 & 1 \\ 0 & 1 & 0.705 & 0.205 & 0.718 & 0.882 \end{bmatrix}$$

By using membership function and weights, compute fuzzy recognition matrix U .

$$U = \begin{bmatrix} 0.821 & 0.093 & 0.023 & 0.811 & 0.051 & 0.011 \\ 0.143 & 0.310 & 0.967 & 0.162 & 0.934 & 0.021 \\ 0.035 & 0.590 & 0.008 & 0.026 & 0.013 & 0.967 \end{bmatrix}$$

Once we calculate membership matrix U , we assign labels to each node. Based on the cluster assignment, each node is rewarded or punished as shown in table 4.3:

Table 4.3: Rewards based on Node Classification

Node	Cluster Label	Reward/Punishment
N1	Bad	0.5
N2	Good	2.0
N3	Neutral	1.5
N4	Bad	0.5
N5	Neutral	1.5
N6	Good	2.0

Once we have all the clusters, we compute the direct trust values using equations (4.8) and collect indirect trust values, i.e., recommendations from other nodes displayed in table 4.4. Further, we calculate final trust value from direct and indirect trust using equation (4.9). It is evident from results that final trust of bad nodes is very less, medium for neutral and high for good nodes.

Table 4.4: Trust Calculation of a Node in FCTM

Node	Direct Trust	Indirect Trust	Final Trust	Actual
N1	0.351	0.325	0.342	Bad
N2	0.837	0.852	0.841	Good
N3	0.633	0.454	0.578	Neutral
N4	0.373	0.631	0.450	Bad
N5	0.629	0.765	0.668	Neutral
N6	0.931	0.903	0.922	Good

4.3 RESULTS

In this section, proposed FCTM is simulated, and experimental results are discussed. The network consists of good nodes, neutral nodes, and bad nodes. Good nodes are the nodes that follow network protocols; neutral nodes are those who sometimes behave maliciously and sometimes behave ideally whereas bad nodes are those who always behave maliciously. The accuracy of the model is evaluated by performing experiments by varying set of network nodes. The accuracy of the model is tested by the correct classification of each node into their respective clusters. FCTM has been evaluated by considering different weights of QoS and recommendation parameters to obtain trust values. FCTM has also been compared with two trust models: COI-HiTrust (Chen et al., 2014) and AFStrust (Xia et al., 2011). COI-HiTrust model has been selected because this model is built to achieve critical mission irrespective of a node failure, compromise or disconnection. Also, the COI-HiTrust model considers the QoS and social behaviour of nodes to evaluate the trust level of nodes. The second model compared with FCTM is AFStrust, which calculates the trust values using fuzzy logic. This model also considers direct and indirect (recommendations) as a part of the parameters for trust evaluations. Hence, these two models were found suitable for performance comparison. The experimental results show the effectiveness and accuracy of the proposed model.

4.3.1 Classification Accuracy

Classification accuracy is evaluated by experimenting with varying network size ($n=5, 10, 25, 40$ and 50). We tested the FCTM for these sets of nodes and obtained quite high accuracy as shown in Figure 4.2.

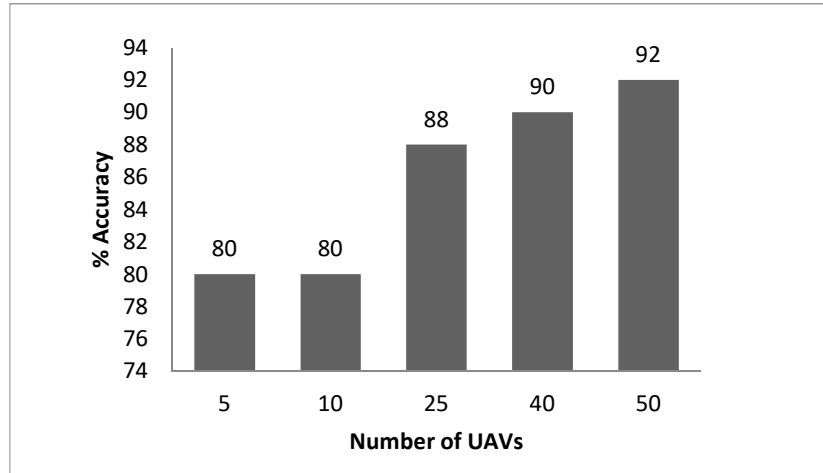


Figure 4.2: Classification Accuracy of FCTM

It is evident from Figure 4.2, accuracy for network size ($n=5$ and 10) is comparatively less than the network size ($n=25, 40$ and 50). FCTM could not learn the behaviour of nodes when the network was relatively small. Hence, the detection rate of the appropriate cluster is less. Whereas, when the network size increases, the proposed model works efficiently. Confusion matrix for different network size representing the error in the detection of node clusters. Table 4.5 represents the confusion matrix for small network, i.e. $N=10$. Here, bad nodes are accurately classified. However neutral nodes and good nodes were not appropriately classified into their respective classes.

Table 4.5: Confusion Matrix for Network Nodes ($N= 10$)

Actual Cluster\ Predicted Cluster	Bad Node	Neutral Node	Good Node
Bad Node	3	0	0
Neutral Node	1	2	0
Good Node	0	1	3

Similarly, table 4.6 represent the confusion matrix for medium network i.e. $N=25$. Here, bad nodes are accurately classified. However neutral nodes and good nodes were not appropriately classified into their respective classes.

Table 4.6: Confusion Matrix for Network Nodes ($N= 25$)

Actual Cluster\ Predicted Cluster	Bad Node	Neutral Node	Good Node
Bad Node	8	0	0
Neutral Node	1	6	1
Good Node	0	1	8

In table 4.7 confusion matrix for large network i.e. $N=50$ has been presented where few nodes are inaccurately classified. It has been observed that neutral nodes are confused with good nodes that cause classification inaccuracies.

Table 4.7: Confusion Matrix for Network Nodes ($N= 50$)

Actual Cluster\ Predicted Cluster	Bad Node	Neutral Node	Good Node
Bad Node	14	1	0
Neutral Node	0	13	2
Good Node	0	1	19

4.3.2 Trust Evaluation based on QoS and Social Trust

The FCTM calculate trust values of nodes by varying the weights of direct trust (QoS parameters) and indirect trust obtained from social parameters. Table 4.8 summarizes the different weight proportion of QoS and Social trust to obtain final trust value of nodes in the network.

Table 4.8: Weight Table for QoS and Social Trust

Test Cases	α	β
Equal QoS and Social Trust	0.5	0.5
More QoS Trust	0.7	0.3
More Social Trust	0.3	0.7

In the first experiment, trust values are evaluated by giving equal weight to both QoS and Social trust. It can be clearly observed from figure 4.3 that trusts level of bad nodes approaching the trust level of neutral nodes in the network for smaller network size.

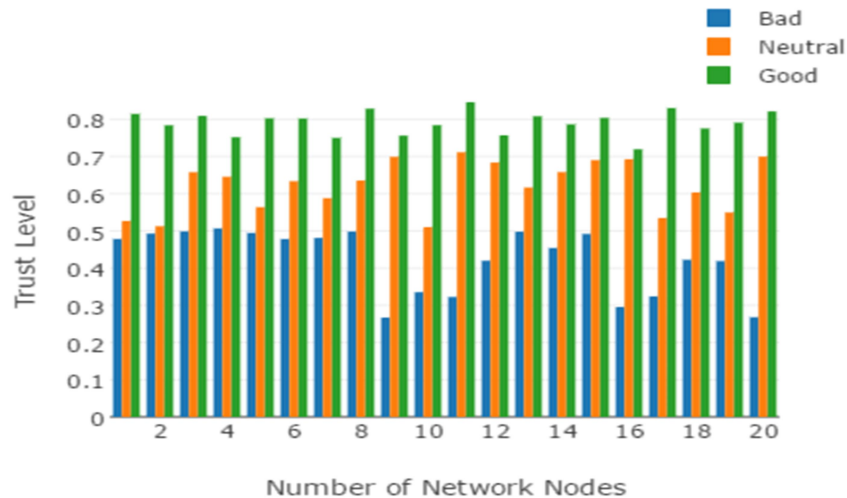


Figure 4.3: Trust Level with $\alpha = 0.5$ and $\beta = 0.5$

Next, we analyzed our model by giving more weight to trust obtained from QoS parameters rather than social parameters. Figure 4.4 depicts that trust values of all the three clusters are in non-overlapping fashion. Trust level obtained for bad nodes is less than 50%, for neutral nodes the trust obtained is 50% to 75% and for good nodes 75% to 100%. Hence, giving more weight to QoS can calculate trust values accurately.

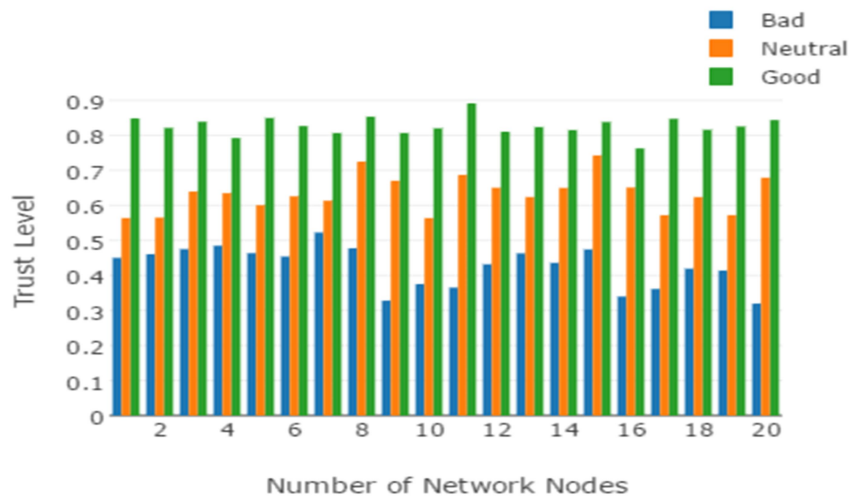


Figure 4.4: Trust Level with $\alpha = 0.7$ and $\beta=0.3$

Further, the proposed model is analysed by giving more weight to trust obtained from Social parameters rather than QoS parameters in Figure 4.5. We found overlapping values among bad and neutral nodes for smaller network size. For larger network size, trust levels of neutral and good nodes overlap. For some nodes, the trust level calculated for bad nodes is even more than the trust level of neutral nodes. Hence, for calculating trust, we cannot highly rely on social recommendations. The network consists of malicious and selfish

nodes termed as bad nodes. These bad nodes may not give a correct recommendation about other nodes.

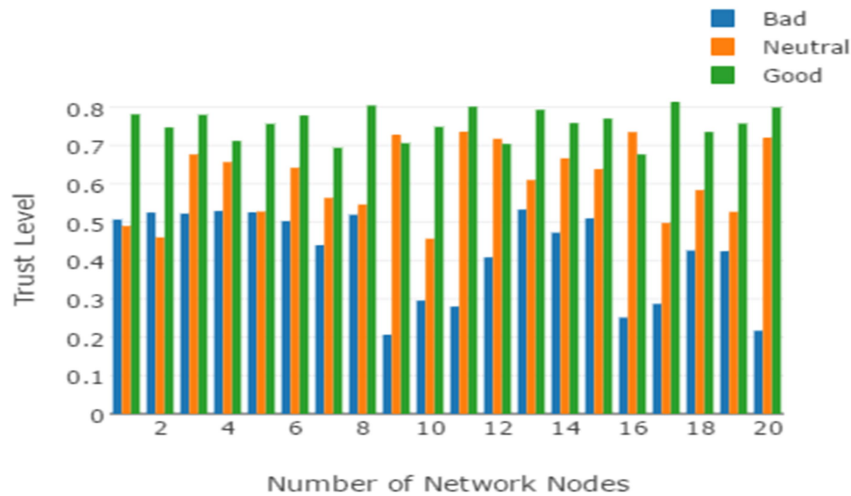


Figure 4.5: Trust Level with $\alpha = 0.3$ and $\beta=0.7$

The behaviour of the network node is shown by varying α in Figure 4.6 and Figure 4.7. The graphs represent the trust levels obtained over time by varying α value. In Figure 4.6, nature line represents the overtime behaviour of a bad node, whereas, in figure 4.7, nature line represents the overtime behaviour of the good node. We can observe from both the graphs that with $\alpha = 0.7$, trust level evaluated by FCTM is very close to the actual nature of node than $\alpha = 0.3$ and $\alpha = 0.5$. The even trust level is obtained more quickly with $\alpha = 0.7$. Initial trust assumed for every node was 0.9, which decay over time based on the behaviour of nodes in the network. Here $\alpha = 0.7$, represents that more weight is given to the quality of parameters (reliability and performance) as compared to social parameters. Hence, the proposed model can obtain the actual trust levels, timely and accurately by giving higher weights to QoS parameters.

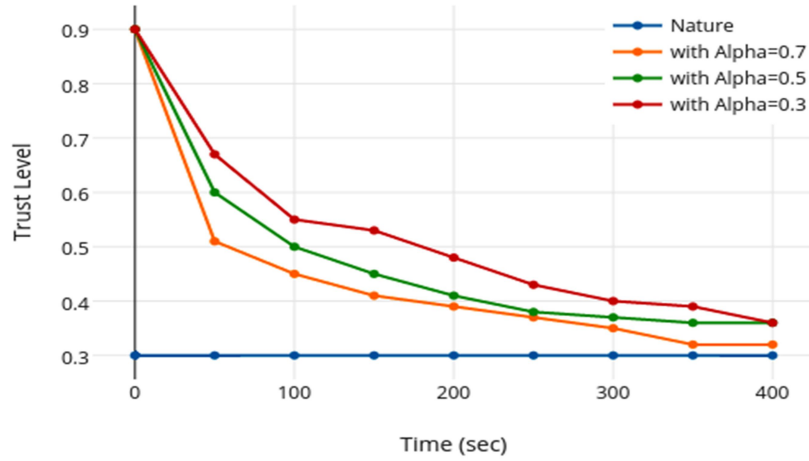


Figure 4.6: Overtime Trust Evaluation for Bad Node

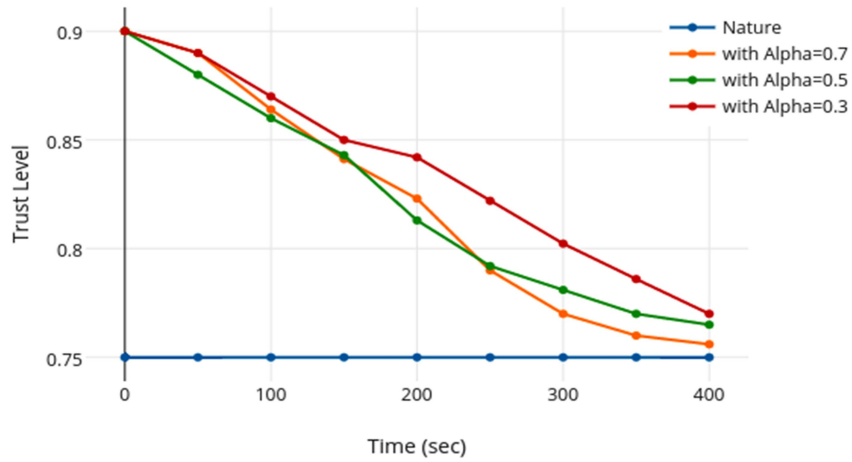


Figure 4.7: Overtime Trust Evaluation for Good Node

4.3.3 Comparison of FCTM with Existing Models

For evaluating and comparing the performance of FCTM with existing COI-HiTrust and AFStrust, simulation environment has been set up. UAVs are randomly placed in the simulation area. The network consists of two types of nodes: normal nodes and malicious nodes. Normal nodes are the nodes which behave ideally and follow the network protocols. Malicious nodes try to breach the integrity of the network and drop the packets. We simulated this environment in

Omnetpp. The proposed model has been evaluated for Packet Drop Ratio (PDR) and Throughput.

Table 4.9: *Simulation Parameters*

Parameter	Value
Field Size	(800x600x400) m^3
Number of UAVs	50
Traffic Type	CBR
Packet Size	512 bytes
Mobility Model	Random Waypoint
Node Placement	Random
Maximum Node Speed	10m/s
Default Trust Value	0.9
Trust Calculation Interval	10 Sec
Transmission Range	50-300m
Transport protocol	UDP
Mac Layer	802.11b
Propagation Radio Model	Two Ray
Routing Protocol	OLSRv2
Simulation time	400 Sec

Figure 4.8 shows a scenario, packet drop ratio is observed by the different velocity of network nodes. In this scenario, a total number of nodes considered are 50, and 15 nodes out of them are taken as malicious nodes.

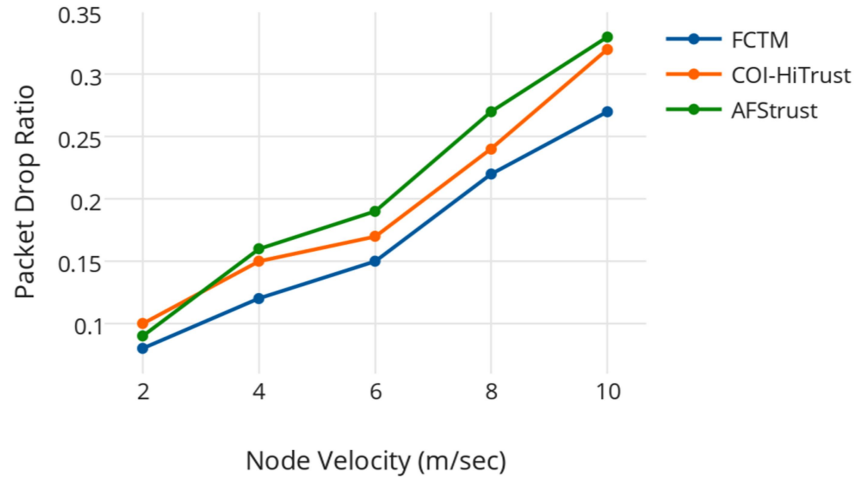


Figure 4.8: Packet Drop Ratio Vs. Node Velocity

As the velocity of network nodes increases, packet drop ratio starts increasing. We have accounted packet drop ratio for a velocity of 2, 4, 6, 8 and 10m/s. FCTM deliver packets at a higher rate as compared to the other existing models. The packet drop ratio increases with a velocity of the node because the nodes join and leaves the network very frequently. As the node speed increases, it is assumed that node will change its neighbouring nodes very frequently. The nodes will drop packets destined for the old nodes.

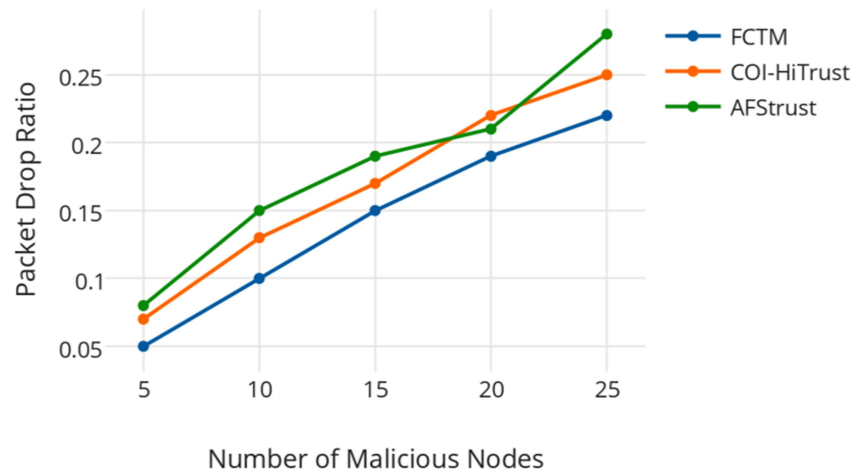


Figure 4.9: Packet Drop Ratio Vs. Malicious Nodes

In Figure 4.9, the packet drop ratio is calculated by introducing a varying number of malicious nodes in the network. The total numbers of nodes in this scenario are taken as 50 and node speed for each node in this scenario is taken as 6 m/s. The graph depicts that as the number of malicious nodes increases, it affects the packet drop rate. The packet drop ratio has been observed for 5, 10, 15, 20 and 25 malicious nodes in the network. The packet drop rate of FCTM is less than COI-HiTrust and AFStrust model. For 25 malicious nodes in the network, the packet drop ratio of FCTM is around 22%, but other models have more than 25% drop ratio.

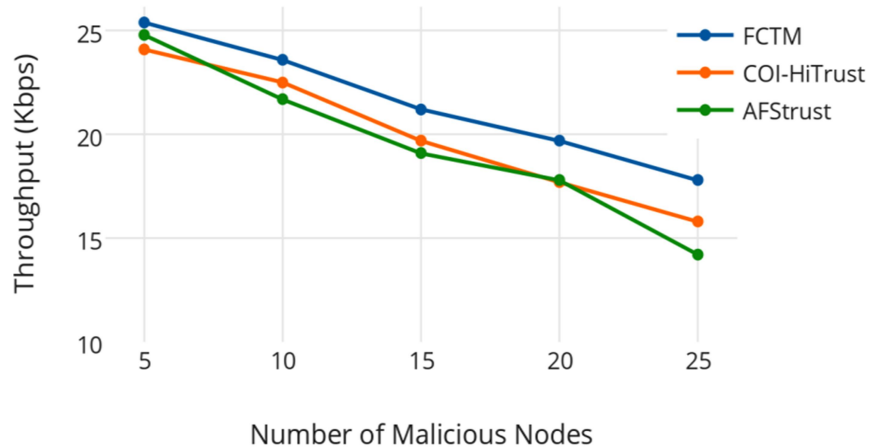


Figure 4.10: *Throughput Vs. Malicious Nodes*

In Figure 4.10, the throughput of the network is calculated. The graph depicts that as the number of malicious nodes increases, the throughput of the network decreases. The network throughput is recorded for 5, 10, 15, 20 and 25 malicious nodes in the network. The network throughput of FCTM is higher than COI-HiTrust and AFStrust model. For 10 malicious nodes in the network, proposed

model does not have significant differences in throughput, but for more number of malicious nodes in the network, the difference is significant. The throughput of COI-HiTrust model and AFStrust model are very close to each other when the numbers of malicious nodes are 20 or more.

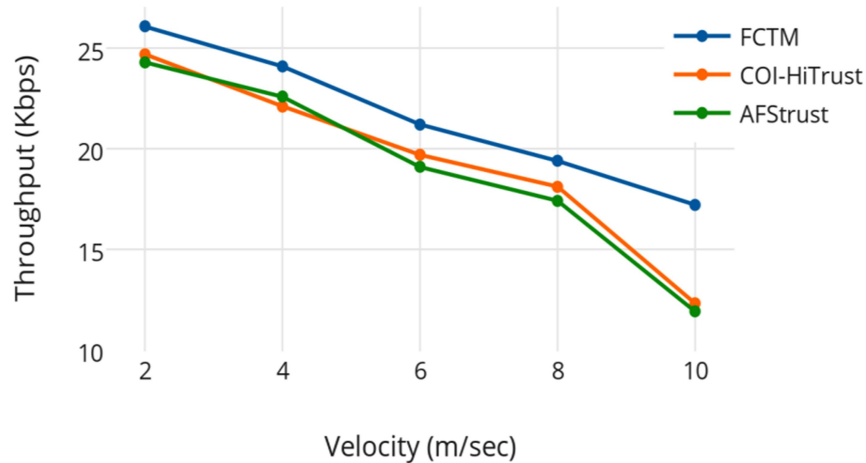


Figure 4.11: Throughput Vs. Node Velocity

Figure 4.11 shows a scenario of throughput against nodes velocity. As the velocity of the network nodes increases, network throughput starts decreasing. The network throughput is recorded for a node velocity of 2, 4, 6, 8 and 10m/s. There is no significant difference between COI-HiTrust and AFStrust model; they overlap each other. However, FCTM performs well and have a high throughput compared to the other existing models. At a higher speed of 10m/sec, FCTM have a throughput of more than 17kbps whereas COI-HiTrust and AFStrust have throughput close to 12kbps.

4.4 CHAPTER SUMMARY

The presence of non-cooperative and malicious nodes degrades the performance of FANETs. In this Chapter, a fuzzy classification trust model (FCTM) is proposed for detecting and segregating non-cooperative and malicious nodes in FANETs. The results vouch for the use of FCTM to identify non-cooperative and misbehaving nodes. This model is highly scalable because the detection accuracy of the model increases with an increase in the network size. Trust levels obtained from the model are in sync with the node behaviour. Further, it is also observed that high reliance on recommendation, the i.e. social trust may give false trust values. The proposed FCTM is evaluated and compared with COI-HiTrust and AFStrust models. FCTM model copes with the high-velocity nodes and out-performs the existing models even for a larger number of malicious nodes in the FANETs. For future research, FCTM can be extended for different weight of QoS parameters. Efficient estimation of weights of the individual parameter can also help to improve the accuracy of trust calculation model, especially in FANETs.

CHAPTER 5

WFCTM: WEIGHTED FUZZY CLASSIFICATION TRUST MODEL

The use of Unmanned Aerial Vehicles (UAVs) has significantly increased for forming an ad hoc network due to their ability to perform in an exciting environment such as armed attacks, border surveillance, disaster management, rescue operation and transportation. Such types of ad hoc networks are popularly known as Flying Ad-hoc Networks (FANETs). The FANET nodes have two prominent characteristics- collaboration and cooperation. Trust plays an important role in predicting the behaviour of such nodes. Researchers have proposed various methods (direct and indirect) for calculation of the trust value of a given node in ad hoc networks, especially in Mobile Ad hoc Networks (MANETs) and Vehicular Ad-hoc Networks (VANETs). The major characteristic that differentiates a FANET from other Ad-hoc networks is - the velocity of the node. As a result, there are frequent losses in connection and topology change. Therefore, the existing methods of trust calculation are not efficient

and effective. In this paper, a fuzzy-based novel trust model ³has been proposed to handle the behavioural uncertainty of FANET nodes. Nodes are classified using a multi-criteria fuzzy classification method based on node's behaviour and performance in the fuzzy and complex environment. Quality of Service (QoS) and Social parameter (recommendation) are considered for evaluating the trust value of each node to segregate the selfish and malicious nodes. Based on the node classification, FANET nodes are rewarded or punished to transform node behaviour into a trust value. Compared to the existing trust techniques, the simulation results show that the proposed model has better adaptability, accuracy, and performance in FANETs.

5.1 THE PROPOSED TRUST MODEL

The fuzzy based trust model basically utilizes the fuzzy classification method to distinguish among network nodes considering its QoS and social parameters. Social parameters account the past behaviour of nodes, whereas QoS parameters consider the current performance of nodes in the network. The basic aim of this model is to classify the network nodes in various clusters: good, neutral and bad. Here, the fuzzy membership functions have been defined using the expert's domain knowledge. These membership functions were used to assign the initial cluster of the network nodes. Further, to improve the

³ The contents of this chapter have been reviewed and published in: Singh, K., & Verma, A. K., A Fuzzy based Trust Model for Flying Ad hoc Networks (FANETs). International Journal of Communication Systems, Wiley (SCI Indexed Journal).

selection of cluster centres, fuzzy optimization is applied, resulting in the reallocation of cluster centre for each cluster. Based on these optimized parameters, the network nodes are classified. The detailed working model has been shown in figure 5.1.

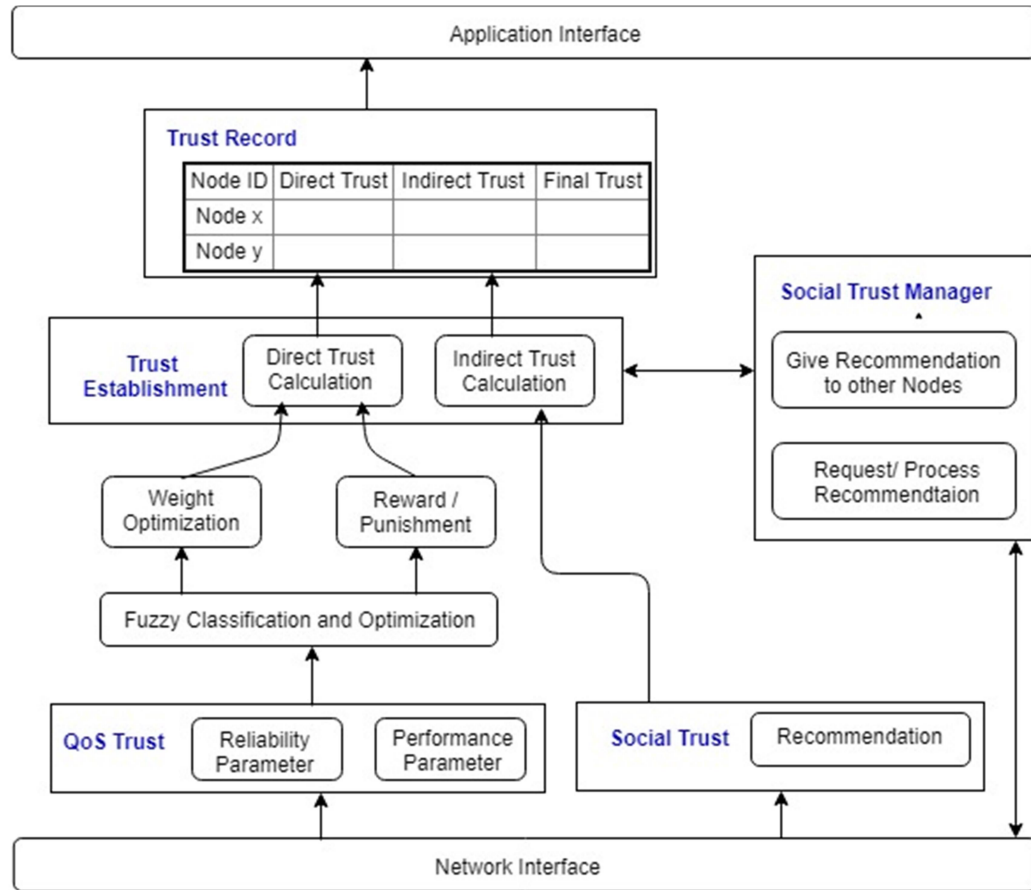


Figure 5.1: Proposed Weighted Fuzzy Classification Trust Model (WFCTM)

To calculate the trust values of network nodes, the model introduced the concept of reward and punishment according to their cluster. At the start of simulations, the nodes are initialized with high trust value. A decay function is used based on the cluster allocated to the particular node. The recommendation is stored in the recommendation buffer. Both the values are fed to the trust establishment system as shown in figure 5.1. Trust establishment

system calculates direct trust and indirect trust. Final trust is calculated based on weights of direct and indirect trust values. The different modules of the proposed model have been defined in following sub-sections. More frequently used symbols have been listed in table 5.1.

Table 5.1: Symbol Table of WFCTM

Symbol	Meaning
n	Number of nodes
M	Number of parameters
C	Number of classes
R	Recognition Matrix
W	Weight Vector
N	Vector of Nodes
P	Parameter Matrix
w_i	Weight of i^{th} parameter
c_i	Characteristic vector of i^{th} node
r_{ij}	Normalized value of j^{th} parameter of i^{th} node
cc_{ij}	Cluster center of j^{th} node for i^{th} class
u_{ij}	Membership of j^{th} node for i^{th} class
T_i^w	Weighted trust of node i
T_i^d	Direct trust of node i
T_i^f	Final Trust
T_i^R	Recommendation / Indirect trust
α	Weightage of direct trust
β	Weightage of indirect trust
A	Initial Trust Value

B	Rewards/ Punishment
NC	Node Cluster Vector
Λ	Lagrangian Multiplier

5.1.1 Fuzzy Classification

Let N be a set of total n nodes that create an ad-hoc network defined as $N = \{N_i | i = 1, 2, 3, \dots, n\}$ and P is set of m attributes (i.e. Signal Strength, Packet Delivery Ratio, Node's Energy, and Transmission Delay) used to evaluate ad hoc network performance defined as $P = \{P_i | i = 1, 2, 3, \dots, m\}$. Every node N_i evaluates the values for each parameter in set P from other nodes and computes a characteristic vector $c_i = \{c_{i1}, c_{i2}, c_{i3}, \dots, c_{im}\}$. Using the characteristic vectors of each node, a characteristic matrix for whole network can be defined as $C = (c_{ij})_{n \times m}$. The characteristic matrix C is influenced by many factors. Before the analysis of the characteristics matrix the values should be normalized. The normalized matrix is defined as $R = (r_{ij})_{n \times m}$. Normalization (Juszczak, Tax, & Duin, 2002) of the values can be done by the following method:

$$r_{ij} = \frac{c_{ij} - c_{imin}}{c_{imax} - c_{imin}} \quad \dots (5.1)$$

Where c_{ij} = value of parameter j of i^{th} node, c_{imin} represents the minimum attribute value of the i^{th} parameter, c_{imax} represents the maximum attribute value of the i^{th} parameter. Once the characteristic matrix is normalized, weightage for each parameter is decided by the weight vector W i.e. a set of m weights $W = \{w_i | i = 1, 2, 3, \dots, m\}$. Total

weight for all the parameters should be equal to 1 i.e. $\sum_{i=1}^m w_i = 1$. Once the weight vector is initialized, each node aggregates these parameters according to their weights. The aggregated trust values are stored in vector T i.e. $T_i^d = \{t_i | i = 1, 2, 3 \dots n\}$. T_i^d represents the direct trust value of i^{th} node. Once all the nodes have their trust values, now the fuzzy classification is used to classify nodes in c classes. Let Cl be a vector of c clusters for node classification. Each node will be classified into c clusters defined as $Cl = \{\text{Bad, Neutral, and Good}\}$. First, we calculate the membership of each node in each cluster. The membership matrix U is defined as $U = (u_{ij})_{c \times n}$. Where, c represents the number of clusters and n represent the number of nodes. u_{ij} represent the membership of j^{th} node in i^{th} cluster. The constraint on u_{ij} is $\sum_{i=1}^c u_{ij} = 1$. In order to find optimal weights W , membership values U and cluster centers CC , we need to optimize the following function (Bezdek, Ehrlich, & Full, 1984):

$$J = \sum_{i=1}^n \sum_{j=1}^c \{u_{ij}^f \sum_{k=1}^m w_k (r_{ik} - cc_{jk})^2\} \quad \dots (5.2)$$

Here f is a fuzzy coefficient with $f = 2$. Following are the steps to find the optimal values W , U , and CC .

- i. Finding the optimized membership matrix, given that we have optimized weights and cluster centers. Since $\sum_{i=1}^c u_{ij} = 1$ is a constraint on U , so we will subtract this constraint from objective function. Here λ represents langragian multiplier for constrained optimization.

$$L(u_{ij}, \lambda) = \sum_{i=1}^n \sum_{j=1}^c \{u_{ij}^2 \sum_{k=1}^m w_k (r_{ik} - cc_{jk})^2\} - \lambda (\sum_{i=1}^c u_{ij} - 1)$$

Now taking derivative of above equation:

$$\frac{\partial L(u_{ij}, \lambda)}{\partial u_{ij}} = 2 u_{ij} \sum_{k=1}^m w_k (r_{ik} - cc_{jk})^2 - \lambda = 0$$

$$2 u_{ij} \sum_{k=1}^m w_k (r_{ik} - cc_{jk})^2 = \lambda$$

Now taking derivative with request to λ :

$$\frac{\partial L(u_{ij}, \lambda)}{\partial \lambda} = \sum_{i=1}^c u_{ij} - 1$$

This gives new membership values, based on which nodes will change their clusters:

$$u_{ij} = \left[\sum_{k=1}^c \left[\frac{\sum_{l=1}^m w_l (r_{jl} - cc_{kj})^2}{\sum_{l=1}^m w_l (r_{jl} - cc_{kl})^2} \right] \right]^{-1} \quad \dots (5.3)$$

- ii. Now considering we have optimized weight vector W and recognition matrix R, we will calculate optimized cluster centers:

$$L(u_{ij}, \lambda) = \sum_{i=1}^n \sum_{j=1}^c \left\{ u_{ij}^2 \sum_{k=1}^m w_k (r_{ik} - cc_{jk})^2 \right\}$$

Taking derivative with respect to c_{ij} :

$$\frac{\partial L(u_{ij}, \lambda)}{\partial cc_{ij}} = 2 \sum_{j=1}^n u_{ij}^2 w_i^2 r_{ij} - 2 \sum_{j=1}^n u_{ij}^2 w_i^2 cc_{ij}$$

$$cc_{ij} = \frac{\partial L(u_{ij}, \lambda)}{\partial cc_{ij}}$$

$$cc_{ij} = \frac{\sum_{j=1}^n u_{ij}^2 r_{ij}}{\sum_{j=1}^n u_{ij}^2} \quad \dots (5.4)$$

- iii. Now we have both optimized cluster centers CC and membership values U; we will find optimized weights:

$$w_i = \left[\sum_{k=1}^m \left[\frac{\sum_{j=1}^n \sum_{h=1}^c [w_k (r_{ij} - c_{ih})^2]}{\sum_{j=1}^n \sum_{h=1}^c [w_k (r_{kj} - c_{kh})^2]} \right] \right]^{-1} \quad \dots (5.5)$$

iv. Once we found all the optimized values, we will use a threshold value to stop the optimization process.

if ($|T - T'| > \delta$) keep optimizing

else stop.

Algorithm 5.1 defines a step by step approach to classify various nodes in clusters by minimizing the objective function using fuzzy c means clustering technique. This algorithm optimizes the weights of each parameter that helps to calculate the best-representing trust values of each node. Algorithm 5.2 calculates the trust values of each node by incorporating direct observations and indirect observations.

Figure 5.2 represents the detailed flow chart of a working model.

Algorithm 5.1: Weight optimization and node clustering in WFCTM

Input: Characteristic matrix C , initial weight vector W , number of clusters c .

Output: Cluster centers $CC = \{cc_1, cc_2, \dots, cc_n\}$, membership matrix $U = (u_{ij})_{c \times n}$

1. Let N be a set of total n nodes that create ad-hoc network defined as $N = \{N_i | i = 1, 2, 3 \dots n\}$
2. P is a set of m attributes defined as $P = \{P_i | i = 1, 2, 3 \dots m\}$
3. W is set of m weights $W = \{w_i | i = 1, 2, 3 \dots m\}$. Total weight for all the parameters should be equal to 1 i.e. $\sum_{i=1}^m w_i = 1$.
4. Computes a characteristic vector $c_i = \{c_{i1}, c_{i2}, c_{i3} \dots c_{im}\}$ and define $C = (c_{ij})_{n \times m}$.
5. Normalize matrix C to Recognition matrix $R = (r_{ij})_{n \times m}$.
6. Use membership functions (in figure 5.3) to calculate initial membership matrix

U^0

```

7. while (iterations<n)
8.   Calculating cluster centers using membership matrix U
9.   for i = 1: c
10.    for j = 1: n
11.     
$$cc_{ij} = \frac{\sum_{j=1}^n u_{ij}^2 r_{ij}}{\sum_{j=1}^n u_{ij}^2}$$

12.    end
13.  end
14.  Calculating the value of the objective function and compare | J(iter)-J(iter-1)|
    with a threshold value. Exit the while loop if the criteria function is less than
    threshold values.
15.  for i = 1: c
16.    for j = 1: n
17.     
$$J = \sum_{i=1}^n \sum_{j=1}^c \{u_{ij}^f \sum_{k=1}^m w_k (r_{ik} - cc_{jk})^2\}$$

18.    end
19.  end
20.  if (iteration~=1) & (|J(iter)-J(iter-1)| < δ)
21.    break;
22.  end
23. Now, optimizing the membership matrix U using weight vector W and cluster
    center CC
24.  for i = 1: c
25.    for j = 1: n
26.      for k = 1: c
27.       
$$u_{ij} = \left[ \sum_{k=1}^c \left[ \frac{\sum_{l=1}^m w_k (r_{il} - cc_{jk})^2}{\sum_{l=1}^m w_k (r_{il} - cc_{jk})^2} \right] \right]^{-1}$$

28.      end
29.    end
30.  end
31. Now, optimizing the weight vector W using membership matrix U and cluster

```

```

center CC
32.   for  $i=1:m$ 
33.       for  $k=1:m$ 
34.           for  $j=1:n$ 
35.               for  $h=1:c$ 
36.                   
$$w_i = \left[ \sum_{k=1}^m \left[ \frac{\sum_{i=1}^n \sum_{h=1}^c [w_k(r_{ij}-cc_{ih})^2]}{\sum_{i=1}^n \sum_{h=1}^c [w_k(r_{kj}-cc_{kh})^2]} \right] \right]^{-1}$$

37.               end
38.           end
39.       end
40.   end
41. end while
42. Assign classes to nodes using membership matrix U
      
$$class\ label = argmax(u_{ij})$$

43. Calculate accuracy of classification using confusion matrix

```

5.1.2 Trust Calculation

Initially, all nodes are assigned the same trust value, i.e. 0.5. The direct trust is calculated by adjacent nodes through direct observation, and the indirect trust is the recommended value shared by neighbouring nodes. All nodes are classified in different clusters (bad, neutral and good) and the nodes get punished or rewarded accordingly as shown in table 5.2. These values help to reduce or increase the direct trust values based on their behaviour (Wang, 2008). Therefore, the behaviour of node converges faster to its actual behaviour using reward and punishment mechanism.

Table 5.2: *Reward and Punishment*

Cluster	Rewards	Punishments
Bad	0	0.5
Neutral	1.5	0
Good	2	0

First, we calculate trust values T_i^w for each node i based on the equal weights of each parameter:

$$T_i^w = \sum_{j=1}^m w_j r_{ij} \quad \dots (5.6)$$

Further, to calculate direct trust T_i^d values, we apply the exponential decay function (Luo, Liu, & Fan, 2009) as follows:

$$T_i^d = a \cdot b^x \quad \dots (5.7)$$

Here, a refers to previous trust value, b refers to reward/punishment based on cluster assignment, x refers to the trust evaluated using weights (T_i^w). A final trust value for each node is calculated as:

$$T_i^f = \alpha T_i^d + \beta T_i^R \quad \dots (5.8)$$

Where α and β are the weights given to direct observations and recommendations. The sum of α and β should be equal to one. T_i^d refers to direct trust and T_i^R refers to the indirect trust value.

Algorithm 5.2: Trust Calculation in WFCTM

Input: Optimized weight vector W , Recognition Matrix R , Classified node vector NC , Punishment and Reward values.

Output: Final trust vector for n nodes.

1. Calculating trust values T_i^w for each node i based of optimized weights and recognition matrix

```

2. for  $i = 1$  to  $n$ 
3.    $T_i^w = \sum_{j=1}^m w_j r_{ij}$ 
4. end
5. Rewarding the nodes based on cluster assignment
6. for  $i=1: n$ 
7.   if  $\text{strcmp}(NFC\{i\}, 'Bad') == 1$ 
8.      $Reward(i) = 0.5;$ 
9.   end
10.  if  $\text{strcmp}(NFC\{i\}, 'Neutral') == 1$ 
11.     $Reward(i) = 1.5;$ 
12.  end
13.  if  $\text{strcmp}(NFC\{i\}, 'Good') == 1$ 
14.     $Reward(i) = 2;$ 
15.  end
16. end
17. Applying exponential decay function for calculating final direct trust values  $T_i^d$ 
18. for  $i = 1$  to  $n$ 
19.    $T_i^d = a \cdot b^x$ 
20. end
21. Calculating the final trust value  $T_i^f$  for every node
22. for  $i = 1$  to  $n$ 
23.    $T_i^f = \alpha T_i^d + \beta T_i^R$ 
24. end

```

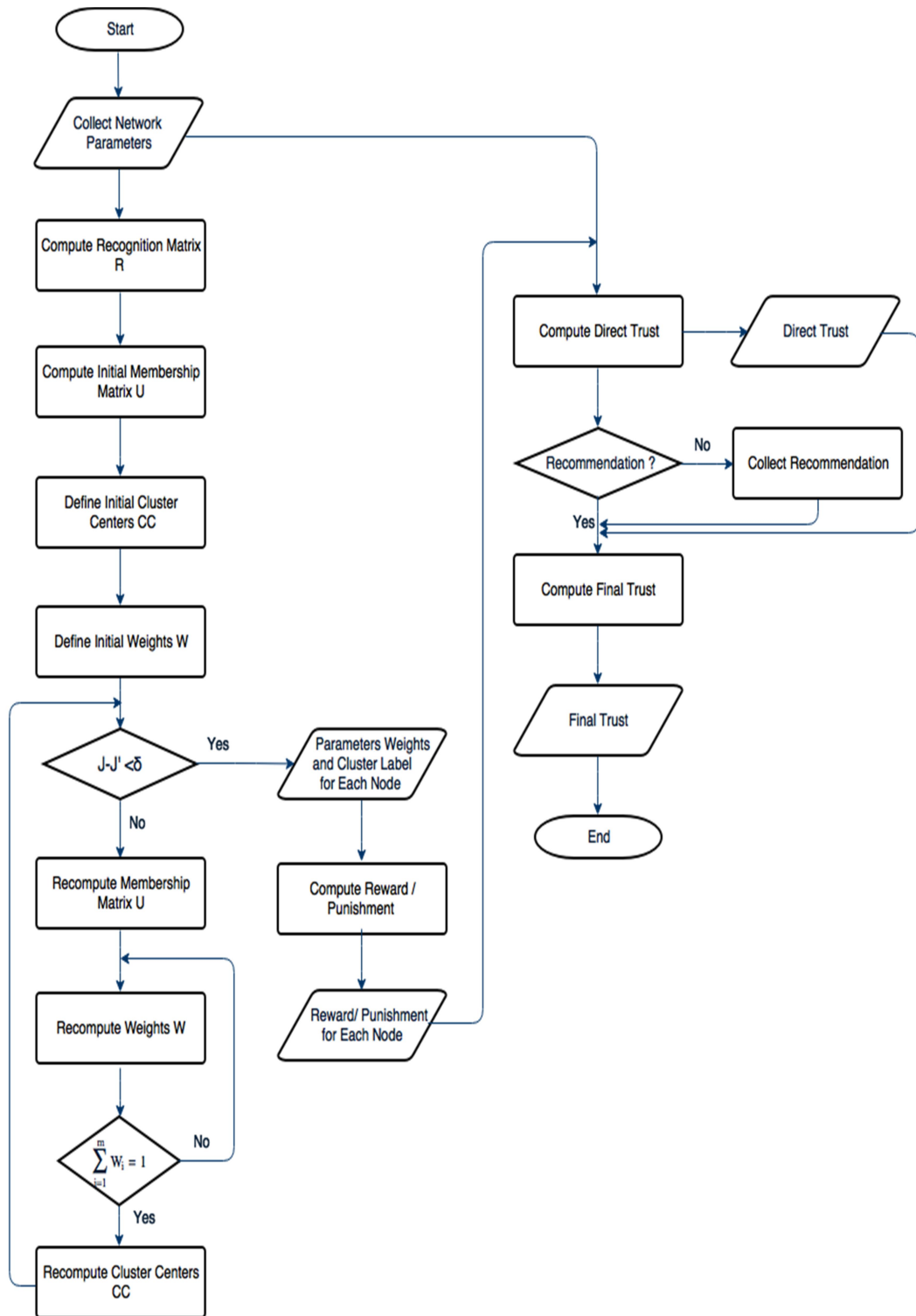


Figure 5.2: Flow Chart of Fuzzy Classification Trust Model

5.2 IMPLEMENTATION

In this section firstly, selection criteria have been defined for various parameters, and further various membership functions for classification have been explained. Further, we describe the model implementation using an example of six nodes. A step by step approach has been explained to classify nodes in various categories and to calculate the final trust of nodes.

5.2.1 Key Attribute Selection

For the trust evaluation of a network node, key attributes are responsible, and they should be defined carefully. To make the trust evaluation process efficient, defined attributes should reflect the characteristics and behaviour of a mobile node in the ad-hoc network (Jadidoleslamy, 2015). In the trust evaluation of a node in FANETs, we should consider both node's behaviour properties and network features. When evaluating node trust through its behaviour, it is not only direct observation which can be used, but a recommendation from other neighbour nodes in the network can be used to get the more accurate trust value of that node. Node mobility is also an important factor for the reliability of that node. The higher is the mobility; the higher is the packet drop rate. Network communication attributes signal strength, and packet delivery ratio plays an important role to evaluate the trust of a node. We have taken three reliability attributes and one performance attribute into account in the trust assessment of a node. Node's energy, signal strength, and the

packet delivery ratio reflect the reliability of a node while transmission delay reflects the performance. So, key attributes set for calculating trust value of a FANET node are $A = \{\text{signal strength, packet delivery ratio, node's energy, and transmission delay}\}$.

5.2.2 Membership Functions

The four key attributes {signal strength, packet delivery ratio, node's energy, Transmission delay} proposed to be used for calculating the trust value in FANETs are discussed along with their membership function.

- i. **Signal Strength:** Free space path loss formula is used for calculating signal strength for different working distances in the proposed fuzzy model. The relation between signal attenuation and distance is calculated as $FSPL(dB) = 10 \log_{10} \left(\left(\frac{4\pi}{c} df \right)^2 \right)$ (Sklar, B. 2001), where FSPL(dB) stands for signal attenuation, measured in dB, d represents the distance from the transmitter, c stands for light speed in vacuum (2.99792458×10^8 m/sec), and f stands for signal frequency. A Device with 802.11n runs at 2.4GHz frequency. By using the above formula, we can have signal attenuation for a working distance as: for a distance of 170m, 320m and 500m are 84.65dBm, 90.14 and 94.02 respectively with a network transmission power of 20dBm. A node is considered as a good node if it has signal strength more than -64.65dBm ($3.42 \times 10^{-7}mW$) and will be considered as a neutral node if signal strength is approximately -70.14dBm

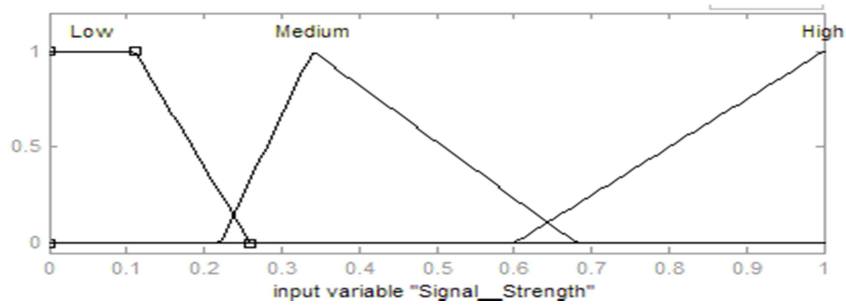
$(0.968 \times 10^{-7}mW)$, and considered as a bad node if signal strength is around $-74.02dBm$ ($0.396 \times 10^{-7}mW$). Signal strength value is normalized using $(x/3.42)$.

ii. Packet Delivery Ratio: Based on the network performance evaluation (Rappaport, 2002), a network of nodes was simulated, and the highest, lowest and average values of the packet delivery ratio were captured. A node having more than 90% of packet delivery ratio is said to be reliable and considered as a node with a good reputation. A node with more 75% of the delivery drop ratio is considered as a node with neutral reputation, and if a network node has less than 75% packet delivery ratio, then it will be a non-reliable node and considered as a node with a bad reputation.

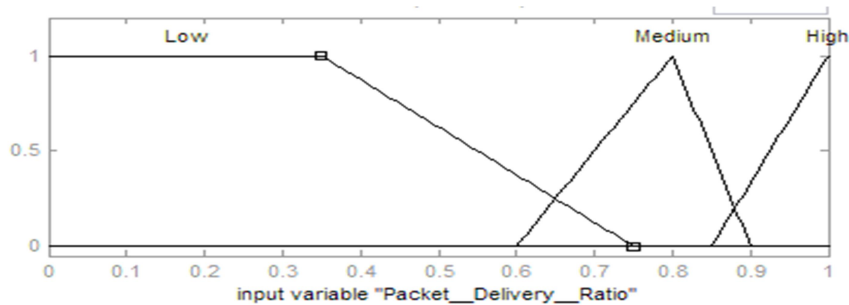
iii. Node's Energy: The energy level of a mobile node determines how much that node is capable of handling the network task and for how long it can work properly. A network node is having energy higher than 70% than it is said to be reliable and considered as a node with a good reputation. A node with more than 40% of energy is considered as a node with a neutral reputation. A node with less than 40% of its total energy will be a weak node and considered as a node with a bad reputation.

iv. Transmission Delay: Transmission delay is one of the network performance parameters, and it can be used to evaluate the trust of a node. If a node is selfish or having malicious intent, then it will transmit data packets with delay. Transmission

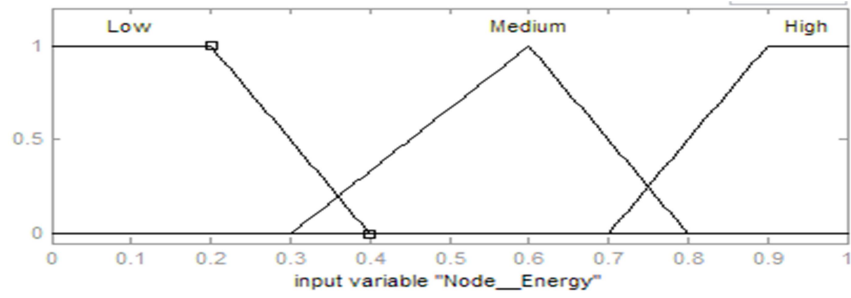
$\text{delay}(D_T) = L/R$, where L stands for is the data size and, R represents the rate of transmission (Kurose, and Ross, 2007) [36]. So, with packet size 1024 Byte and 11Mbps transmission rate, transmission delay will be $0.74ms$. A node with a transmission delay of $0.74ms$ will be considered as a good node, a node having a transmission rate of nearly 8 Mbps will be taking transmission delay around $1.1ms$ is considered as a neutral node, whereas a node with a transmission rate less than 6 Mbps will take more than $1.36ms$ transmission delay. Such node will be considered as a bad node. We have assumed that in worst case scenario transmission delay will not exceed $2.73ms$.



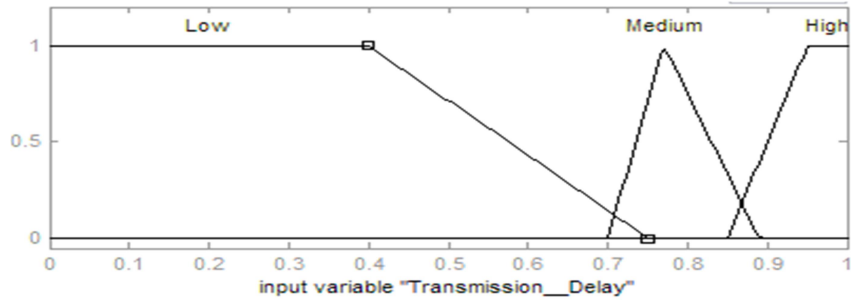
a. Signal Strength



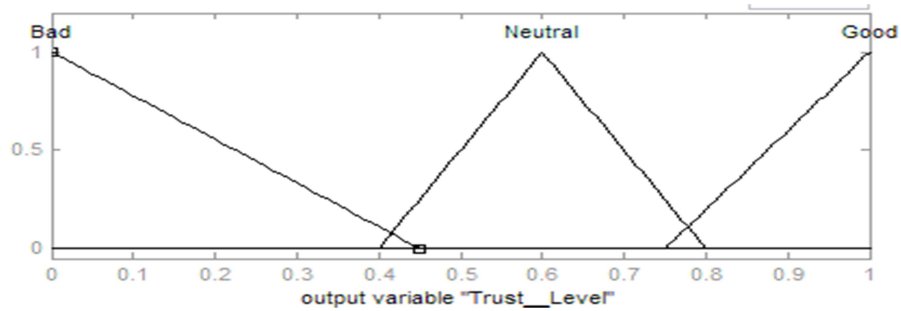
b. Packet Delivery Ratio



c. Node Energy



d. Transmission Delay



e. Output: Trust Level

Figure 5.3: Fuzzy Membership Functions (a. Signal Strength, b. Packet Delivery Ratio, c. Node Energy, d. Transmission Delay)

Here, we summarize the procedure of trust evaluation in the proposed model. Six nodes are given with their attribute values for trust evaluation in table 5.3.

Table 5.3: Attribute Values of Network Nodes

Node	Signal10^{-7} (mW)	Packet Delivery Ratio (%)	Node Energy (%)	Transmission Delay (ms)
N1	0.82	50	52	1.52
N2	2.01	91	82	0.86
N3	1.19	73	76	1.09
N4	0.71	62	39	1.31
N5	1.47	68	65	1.34
N6	3.18	88	92	0.78

Characteristics matrix is defined as:

$$C = \begin{bmatrix} 0.82 & 50 & 52 & 1.52 \\ 2.01 & 91 & 82 & 0.86 \\ 1.19 & 73 & 76 & 1.09 \\ 0.71 & 62 & 39 & 1.31 \\ 1.47 & 68 & 65 & 1.34 \\ 3.18 & 88 & 92 & 0.78 \end{bmatrix}$$

For the standard, the signal strength is changed by (signal / 3.42), and the second and third column to the percentage on a scale of 1. Transmission delay is normalized by using (2.45-transmission delay)/1.84)).

$$C = \begin{bmatrix} 0.24 & 0.50 & 0.52 & 0.50 \\ 0.59 & 0.91 & 0.82 & 0.86 \\ 0.34 & 0.73 & 0.76 & 0.74 \\ 0.21 & 0.62 & 0.39 & 0.62 \\ 0.43 & 0.68 & 0.65 & 0.60 \\ 0.93 & 0.88 & 0.92 & 0.91 \end{bmatrix}$$

To surmount the influence of different dimensions, the characteristic matrix is changed into the recognition matrix as R :

$$R = \begin{bmatrix} 0.044 & 0.527 & 0.194 & 0 & 0.307 & 1 \\ 0 & 1 & 0.56 & 0.292 & 0.439 & 0.926 \\ 0.245 & 0.811 & 0.698 & 0 & 0.490 & 1 \\ 0 & 0.89 & 0.581 & 0.283 & 0.243 & 1 \end{bmatrix}$$

By using membership functions and weights, compute fuzzy recognition matrix U .

$$U = \begin{bmatrix} 0.944 & 0.051 & 0.005 & 0.859 & 0.287 & 0.006 \\ 0.044 & 0.231 & 0.991 & 0.114 & 0.646 & 0.018 \\ 0.011 & 0.717 & 0.004 & 0.027 & 0.067 & 0.965 \end{bmatrix}$$

Once we calculate membership matrix U , we assign labels to each node. Based on the cluster assignment, each node is rewarded or punished as shown in table 5.4:

Table 5.4: Rewards/Punishment based on Node Classification

Node	Cluster Label	Reward/Punishment
N1	Bad	0.5
N2	Good	2.0
N3	Neutral	1.5
N4	Bad	0.5
N5	Neutral	1.5
N6	Good	2.0

Once we have all the clusters, we compute the direct trust values the using weights listed in table 5.5 and equations (5.7). Further, we collect indirect trust values, i.e., recommendations from other nodes displayed in table 5.6. Further, we calculate the final trust value from direct and indirect trust using equation (5.8). It is evident from the results that final trust of bad nodes is very low and for neutral is average, and for good nodes it is high.

Table 5.5: Parameters Weights

Signal Strength	Packet Delivery Ratio	Node Energy	Transmission Delay
0.183	0.304	0.279	0.234

Table 5.6: Trust Calculation of a Node in WFCTM

Node	Direct Trust	Indirect Trust	Final Trust	Cluster
N1	0.475	0.416	0.457657	Bad
N2	0.875	0.825	0.860012	Good
N3	0.614	0.523	0.587062	Neutral
N4	0.452	0.643	0.509627	Bad
N5	0.580	0.681	0.610973	Neutral
N6	0.987	0.911	0.96448	Good

5.3 RESULTS

In this section, the simulation results of the proposed trust model are discussed. A simulation environment has been established to evaluate the performance of the proposed trust scheme. The network consists of three types of nodes (good, neutral, bad nodes). Good nodes are the ones who follow network protocols and do not act maliciously; neutral nodes are those which most of the time behave ideally and sometimes behave maliciously whereas bad nodes are those who continually misbehave. Further, warm-up time of the 40s is taken before recording the readings, and each phase is simulated 10 times, and the average of these simulations is calculated and presented.

Experiments are done in two phases: In the first phase, classification accuracy and trust value for different clusters have been evaluated. In the second phase, a performance comparison of the proposed trust model is done with existing trust models in the area of ad hoc networks. The proposed model has also been compared with two trust models: COI-HiTrust and AFStrust. COI-HiTrust model is chosen because this model is built to achieve critical mission irrespective of a node failure, compromise or disconnection. Also, the COI-HiTrust model considers the QoS and social behaviour of nodes to evaluate the trust level of nodes. The second model compared with the proposed model is AFStrust, which calculates the trust values using fuzzy logic. This model also considers direct and indirect (recommendations) as a part of the parameters for trust evaluations. Hence, these two models were found suitable for performance comparison. The experimental results show the effectiveness and accuracy of the proposed model.

5.3.1 Classification Accuracy

The proposed model classifies nodes into good, neutral and bad based on their behaviour in the network. Correct classification of these nodes acts as a measure to evaluate the performance of any trust model. In this paper, classification accuracy is measured with two sets of parameters, a number of network nodes and malicious nodes percentage. Results are recorded with varying network size of network nodes (10, 20, 30, 40 and 50) and malicious node percentage as (10, 20, 40, 60 and 80).

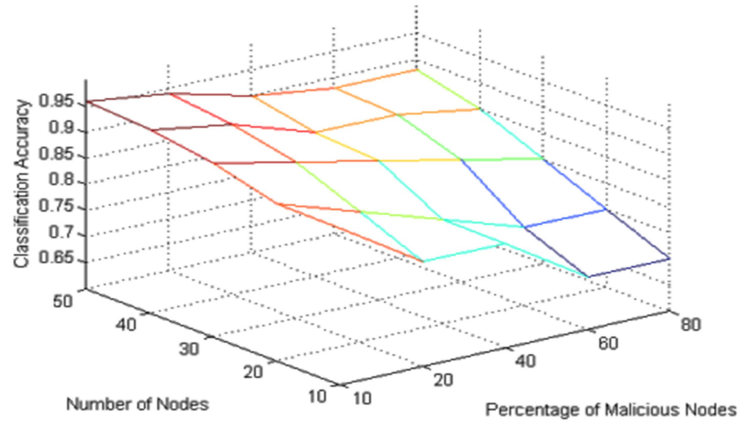


Figure 5.4: Classification Accuracy of Proposed Model

It is evident from figure 5.4 that classification accuracy of the model is dependent on both numbers of nodes as well as the percentage of malicious nodes. For network size ($n=10$ and 20), classification accuracy is comparatively less than the network size ($n=30, 40$ and 50), and as the percentage of malicious node increase, classification accuracy decrease. The proposed trust model can accurately classify above 90% of nodes if the network size is 30 or more and the malicious node percentage is around 20. Even in the situation where malicious node percentage is around 50, and network size is 20 or more, the proposed trust model performs well and accurately classify 80% of nodes.

5.3.2 Mean Square Error(MSE)

In this paper, the proposed model calculates trust using indirect and direct trust values. In order to decide the best proportion to represent the node's real behaviour, we calculate the MSE. We found the lowest MSE is for direct trust weightage = 0.7 and indirect trust weightage =

0.3. The MSE is higher when more weightage is given to indirect trust. This is because more weightage to indirect observation gives the opportunity to bad nodes to share false recommendations. Hence, MSE comes out to be minimum when direct observations are given more weightage.

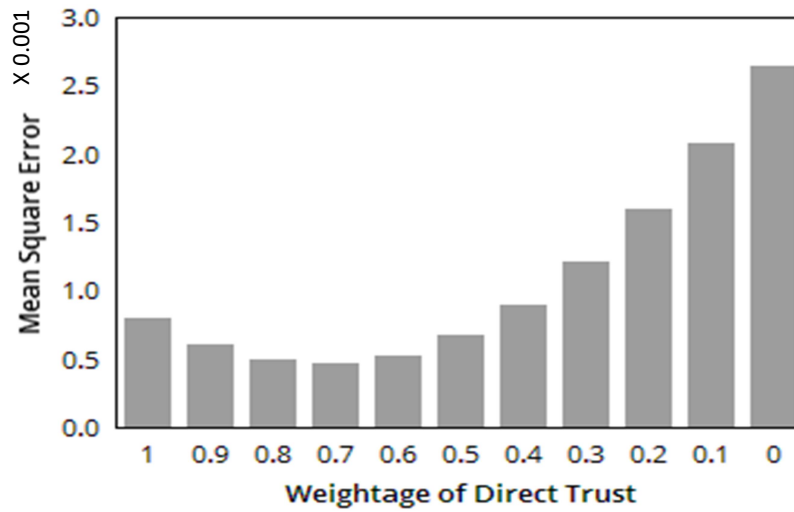


Figure 5.5: Mean Square Error (MSE)

5.3.3 Trust Levels

The proposed model calculates trust levels of bad, neutral and good nodes by using the best appropriate weight combination of direct and indirect trust observed in figure 5.5, i.e., 0.7 as direct trust and 0.3 as indirect trust. It can be clearly observed from figure 5.6 that the trust level of the bad nodes approaching to 0.4, the trust level of the neutral nodes is approximately 0.6 and the trust level for the good nodes is approaching 0.9 in the network. Trust levels define threshold levels to categorize the nodes as bad, neutral and good.

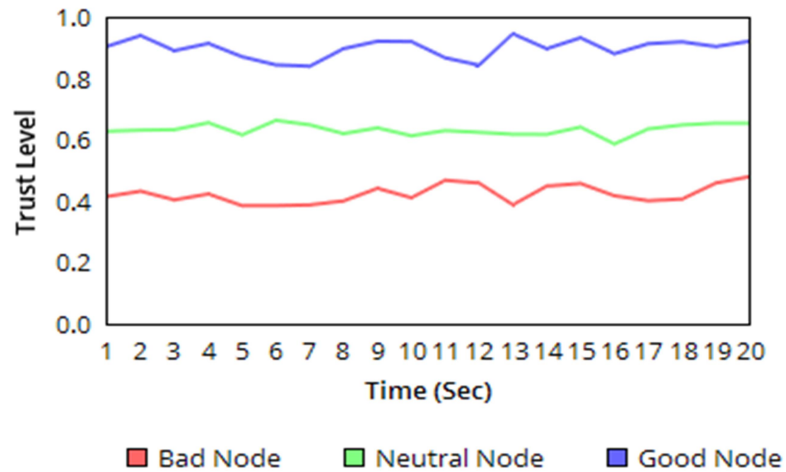


Figure 5.6: Trust Levels of Bad, Neutral and Good Node

5.3.4 Trust Convergence

Initially, all the nodes in the network are assigned a trust level of 0.5 i.e., all the nodes are considered in the neutral zone as defined by figure 5.7. However, nodes may behave maliciously or efficiently with time and acquire a different trust level. In this experiment, the network was simulated for 400 seconds where one of the nodes observes the behaviour of the other two nodes in the network over time. It has been observed that nodes converge towards their actual behaviour over the time. As shown in figure 5.7, the good node achieves approximately the trust level of 0.9, and the bad node converges towards the bad behaviour line, i.e., the trust level of 0.4. This helps nodes to segregate bad nodes in their further communication in the network.

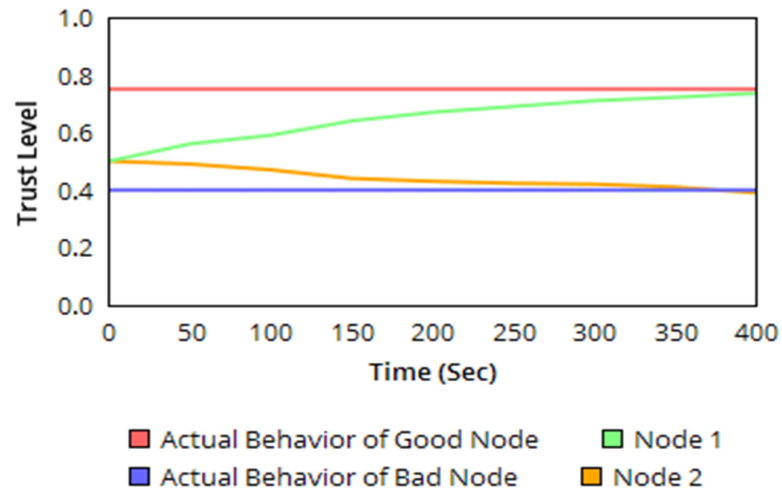


Figure 5.7: Convergence of Node 1 and Node 2 to Their Actual Behaviour

5.3.5 Energy Consumption

In FANET, UAVs have power constraint and its failure results in the inability of a UAV to generate and forward packets thus affecting the network performance and lifetime. There can be a case where the malicious node may reduce the energy level of good node by establishing more communication links. Here, we present the average energy consumption of a node, i.e. the ratio of drained energy to initial energy of the node. The drained energy is calculated as energy consumed in transmitting, receiving, and calculating trust values at the end of 400s, i.e. total simulation duration.

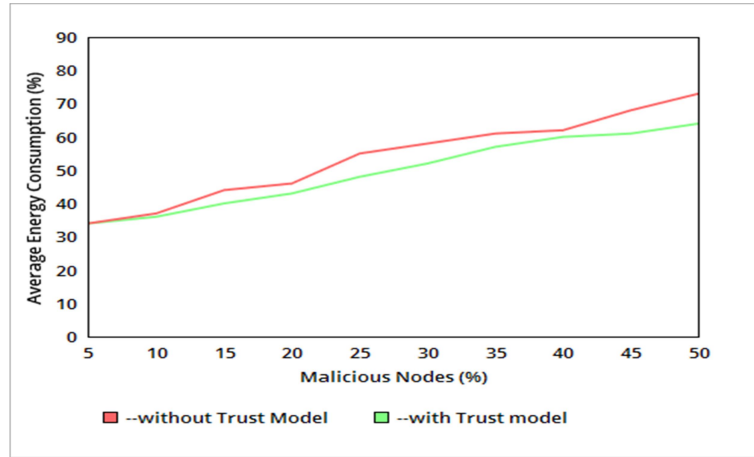


Figure 5.8: Average Energy Consumption vs. Malicious Node's Percentage

The percentage of average energy consumption of UAVs is measured against the malicious node's percentage in the network as shown in figure 5.8. The experimental results show that average energy consumption of UAVs is less while using the proposed trust model as compared to no trust model because of the limited communication with the malicious nodes.

5.3.6 Comparison of the Proposed Model with Existing Trust Models

For evaluating and comparing the performance of the proposed model with existing COI-HiTrust and AFSTrust, a simulation environment has been setup shown in table 5.7. UAVs are randomly placed in the simulation area. The network consists of two types of nodes: normal nodes and malicious nodes. Normal nodes are the nodes which behave ideally and follow the network protocols. Malicious nodes try to breach the integrity of the network and drop the packets. We simulated this environment in Omnetpp (Released, omnetpp-announcements &

Released 2018). The proposed model has been evaluated for Transmission delay and Packet Drop Ratio (PDR).

Table 5.7: *Simulation Parameters*

Parameter	Value
Field Size	(800x600) Sq. m
Number of UAVs	50
Traffic Type	CBR
Packet Size	1024 bytes
Mobility Model	Random Waypoint
Node Placement	Random
Maximum Node Speed	10m/s
Default Trust Value	0.5
Trust Calculation Interval	10 Sec
Transmission Range	50-300m
Transport protocol	UDP
Mac Layer	802.11b
Propagation Radio Model	Two Ray
Routing Protocol	OLSRv2
Simulation time	400 Sec

In this analysis, PDR and transmission delay has been observed by varying node speed and number of malicious nodes. A network of 50 nodes has been created where 15 nodes are kept as malicious referred to as scenario 1. In scenario 1, node speed has been varied between 2-20 m/s. Another network of 50 nodes has been created, and malicious nodes vary from 5 to 50 which is referred to as scenario 2. Here, node speed is kept at 8 m/s for all the nodes.

For scenario 1, we analyze the transmission delay against varying node speed. As figure 5.9 explains, with increasing node speed

transmission delay also increases. It has been observed that existing models cannot work efficiently with high-speed nodes. However, the transmission delay is less for the proposed model as compared to existing models. Hence the proposed model can better incorporate the high-speed nodes as available in FANETs.

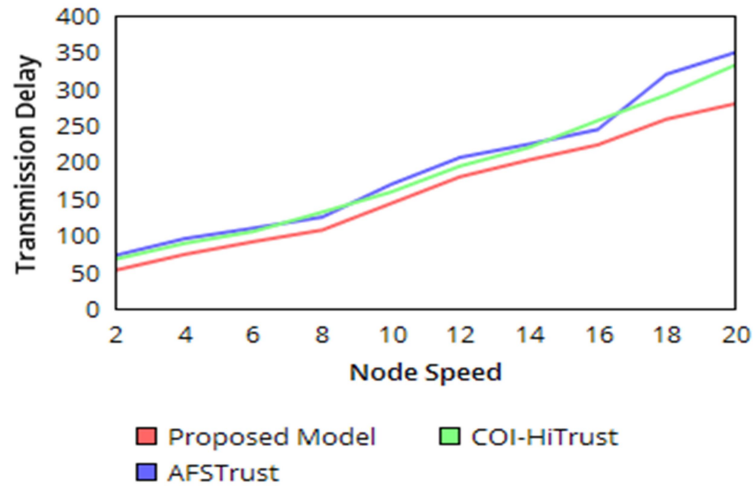


Figure 5.9: Node Speed vs Transmission Delay

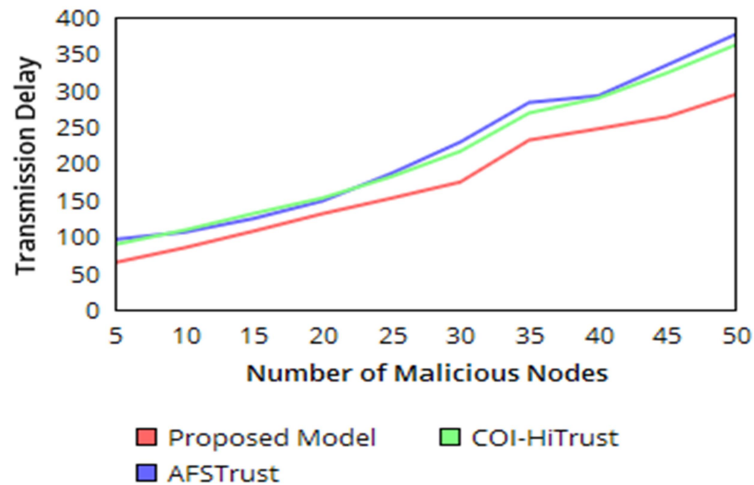


Figure 5.10: Number of Malicious Nodes vs. Transmission Delay

For scenario 2 figure 5.10 shows that as the number of malicious nodes increases, the transmission delay (Sec) also increases. The

proposed model delivers packets with low transmission delay as compared to the other existing models. The proposed model can achieve an accuracy of more than 95% for node classification. Once the bad nodes are removed from the network definitely the performance of the network will be enhanced. Next, the proposed model has been evaluated and compared with existing models by observing packet drop ratio of nodes. In scenario 2, it has been observed that packet drop ratio increases as the malicious nodes increase in the network. As shown in figure 5.11, implementing the proposed model helps to reduce the packet drop ratio in the network as compared to other techniques. Further, the proposed model is evaluated upon the packet drop ratio in scenario 1. As shown in figure 5.12, the proposed model outperforms COI-HiTrust and AFSTrust trust models regarding packet drop ratio. Malicious nodes or selfish nodes in the network drop the packets to save their energy in the network. Segregation of such nodes using the proposed trust model helps to enhance the network performance.

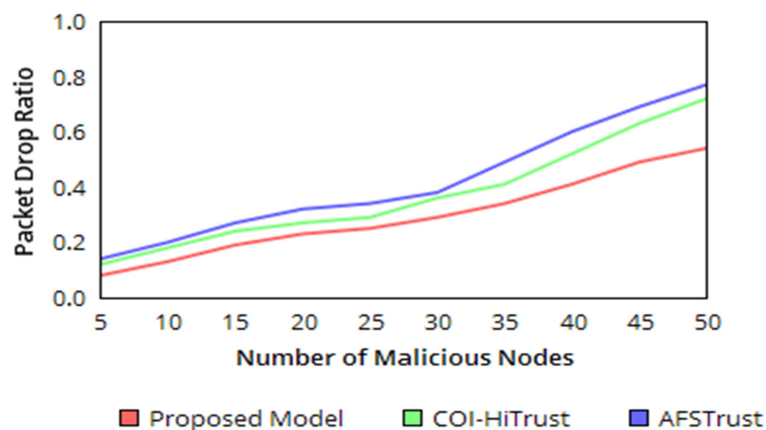


Figure 5.11: Number of Malicious Nodes vs. Packet Drop Ratio

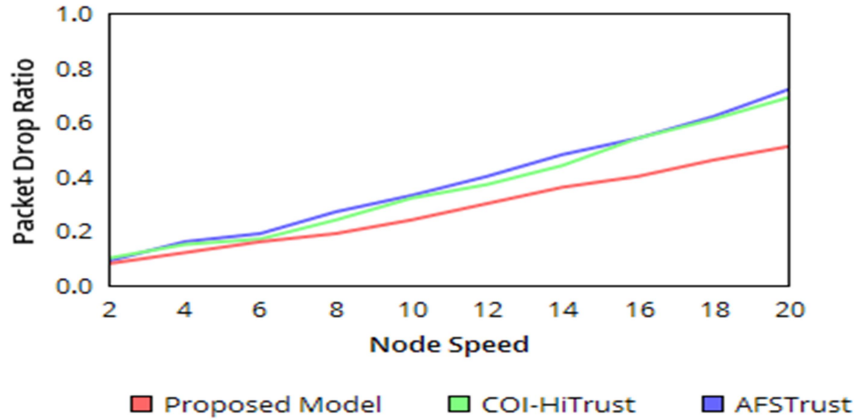


Figure 5.12: Node Speed vs. Packet Drop Ratio

5.4 CHAPTER SUMMARY

In this chapter, a fuzzy-based trust model incorporating both direct - indirect trust for detection and segregation of non-cooperative as well as malicious nodes in FANETs is proposed. The results vouch for the use of the proposed trust model to identify non-cooperative and malicious nodes. Further, the detection accuracy increase with the increase in network size indicating that the proposed method is highly scalable. The trust levels obtained from the model are in sync with node behaviour. Furthermore, it has also been observed that high reliance on recommendation, i.e., social trust gives false trust values. The proposed model saves node energy by reducing links with malicious nodes. The proposed trust model has also been evaluated and compared with well-known trust models in ad-hoc network, i.e., COI-HiTrust and AFSTrust models. The proposed model out-performs these models in a highly dynamic environment of FANETs. This is a

simulation work, and it would be interesting to note the behaviour of the proposed scheme on a dedicated test-bed for FANETs. For future research, the proposed model can be extended further to reduce the energy consumption as well as the prediction of node behaviour in FANETs.

CHAPTER 6

TBCS: TRUST BASED CLUSTERING SCHEME FOR FANETs

In FANETs, coordination and cooperation among nodes are important for efficient data transmission. Cooperation among the nodes hinges on the node behaviour. The behaviour of a node can be quantified using the concept of trust. Trust helps in segregation of malicious nodes in the network, thus increasing the reliability of information exchanged among nodes. In this chapter, a trust-based clustering scheme, i.e., TBCS⁴ has been proposed. In TBCS, the nodes are classified using a multi-criteria fuzzy classification method based on node's behaviour and performance in the fuzzy and complex environment. The proposed scheme makes use of Takagi-Sugeno-Kang (1993), fuzzy inference method. The concept of reward and punishment has been introduced to transform node behaviour into a trust value, and to segregate malicious and misbehaving nodes in the FANET. Further, a secure cluster head (CH) has been selected based on calculated trust values. TBCS is compared with existing trust

⁴ The contents of this chapter are under review in: Singh, K., & Verma, A. K., TBCS: A Trust Based Clustering Scheme for Secure Communication in Flying Ad-Hoc Networks. Wireless Networks (under review- SCI Indexed Journal).

schemes such as COI-HiTrust, AFStrust and the model proposed by Pushpita et al. (2014). The simulation results revealed that the proposed model has better adaptability, accuracy, and performance in FANETs.

6.1 SYSTEM MODELLING

6.1.1 Network Model

FANET comprises of these key components (Marshall, Barnhart, & Shappee, 2015): nodes (UAVs), cluster head, Ground Control Station (GCS), and communication links. The network model of FANET with its components is shown in Figure 6.1. UAVs are the flying aircraft without human onboard. UAVs come in different shapes and sizes, from micro UAVs (2cm) to large UAV (wingspan 120m), they are used according to the application scenarios (Marshall, Barnhart, & Shappee, 2015). UAVs are having various components like a gyroscope, communication subsystem, magnetic orientation sensor, GPS module, infrared camera, *etc.* To accomplish a mission, multiple UAVs form a cluster based on their geographical location, nature of the task, *etc.* Inside the cluster, UAVs periodically elect their cluster head for inter cluster and GCS communication. As shown in Figure 6.1, UAVs collect information from their environment and transmit it to the cluster head for further transmission to the GCS. A GCS is the ground unit that provides control and monitoring facilities for UAVs during their operations. A computer is used to run the GCS software

that is used to keep an eye on the progress of the mission. GCS mainly communicates with the flight controller and transfers useful information. To transmit data and control information between UAVs and GCS, wireless communication links are used. Links are categorized into two types: Line-of-sight (LOS), where UAVs and GCS can transmit and receive control signals through direct radio waves; and beyond line-of-sight (BLOS), where UAVs are controlled via satellite (Marshall, Barnhart, & Shappee, 2015).

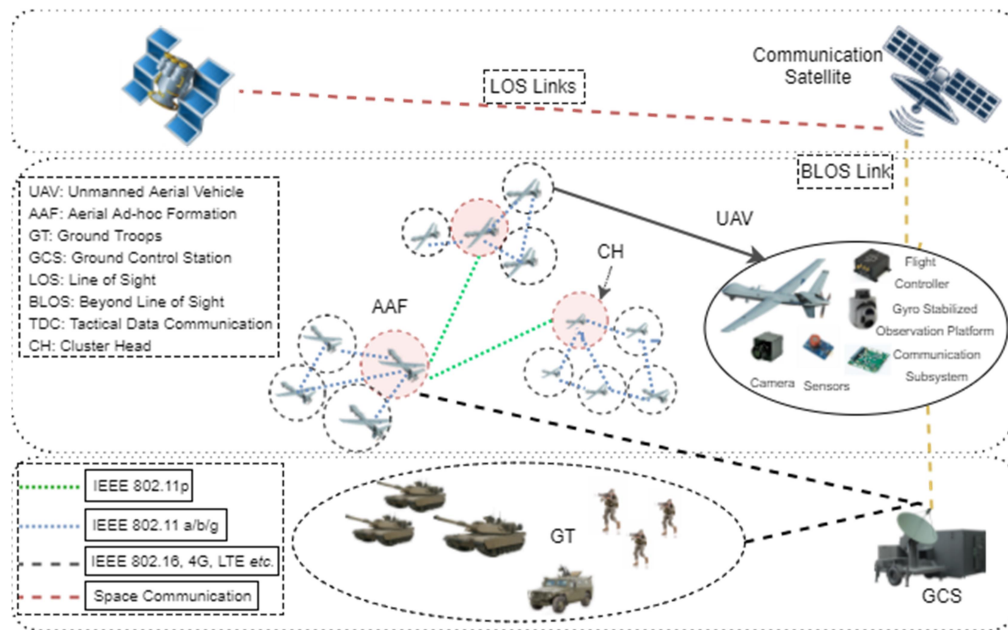


Figure 6.1: FANET Network Model

6.1.2 Threat Model

We have considered the following attacks that can be launched on FANETs:

- The adversary can compromise a benign node and convert it into a malicious node for stealing information and performing malicious activities in the network.

- A malicious or compromised node can counterfeit the routing information so that network data can be tunneled to attacker's location (wormhole, sinkhole attacks).
- Malicious nodes conspire to give negative feedback about benign nodes in the network (bad mouthing attack) to destroy their reputation.
- A malicious node can randomly drop the packets instead of forwarding them (Selective forwarding attack).
- A malicious node can misbehave in the network by showing non-cooperation in network communication.

6.1.3 Security Goals

Following security goals are considered in the proposed approach:

- To provide security against various types of attacks on routing such as sinkhole, Denial of Service, wormhole, and selective forwarding.
- To segregate malicious and misbehaving network nodes by maintaining the trustworthiness among nodes.
- To provide secure data transmission from the source node to the destination node with high throughput and packet delivery ratio.
- To provide data dissemination efficiently without additional overhead on the network.

6.2 THE PROPOSED TRUST MODEL

To maintain a trustworthy communication between nodes, we propose the fuzzy-based trust model. Fuzzy based trust model utilizes the fuzzy classification (Bezdek, Ehrlich, & Full, 1984) method to distinguish among network nodes considering their performance, reliability and social parameters. Social parameters account the past behaviour of nodes, whereas QoS parameters consider the current performance of nodes in the network. The basic aim of this model is to classify the network nodes in various classes: good, neutral and bad. Also helps in make a secure cluster of nodes and to select a trustworthy cluster head. Each node in the cluster will compute trust values of its one-hop neighbours, and algorithm 6.5 will be used to select a secure cluster head. Whenever a new node wishes to join the network, it will locate the cluster head and send node join request. The detailed working model has been shown in Figure 6.2.

Table 6.1: Symbol Table for TBCS

Symbol	Meaning	Symbol	Meaning
n	Number of nodes	u_{ij}	Membership of j^{th} node for
M	Number of parameters	T_i^w	Weighted trust of node i
C	Number of classes	T_i^d	Direct trust of node i
R	Recognition matrix	T_i^f	Final trust
W	Weight Vector	T_i^R	Recommendation / Indirect
N	Vector of nodes	α	Weightage of direct trust
P	Parameter matrix	β	Weightage of indirect trust
P_i	Parameter vector for node i	$I\bar{h}$	Trust threshold for new
w_i	Weight of i^{th} parameter	A	Initial trust value
c_i	Characteristic vector of i^{th} node	B	Rewards/ Punishment
r_{ij}	Normalized value of j^{th}	NC	Node cluster vector

cc_{ij}	Cluster center of j^{th} node for i^{th}	Λ	Lagrangian multiplier
CH	Cluster head	$\mathcal{T}$	Trust threshold for CH
V	Matrix containing nominated	W_t	Warning message waiting

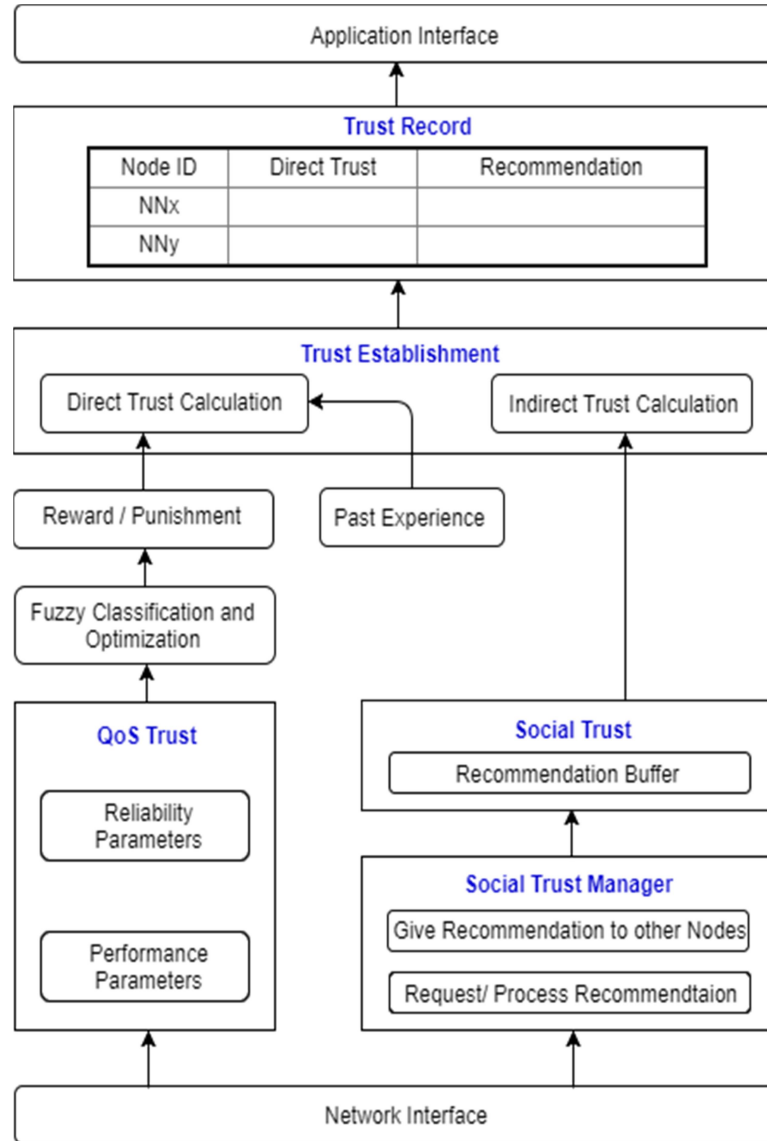


Figure 6.2: Proposed Trust Model

6.2.1 Fuzzy Classification and Optimization

Let N be a set of total n nodes that create a cluster in ad-hoc network and defined as $N = \{N_i | i = 1, 2, 3, \dots, n\}$. P is set of m attributes used to

evaluate ad hoc network performance and defined as $P = \{P_i | i = 1, 2, 3, \dots, m\}$. Every node N_i evaluates the values for each parameter in set P of other nodes and computes a characteristic vector $c_i = \{c_{i1}, c_{i2}, c_{i3} \dots c_{im}\}$. Using the characteristic vectors of each node, a characteristic matrix for the whole network can be defined as $C = (c_{ij})_{n \times m}$ where, c_{ij} = value of parameter j of i^{th} node. Before analysis of the characteristics matrix, the values should be normalized. The normalized matrix is defined as $R = (r_{ij})_{n \times m}$. Normalization (Wang, Y., 2008) of the values can be done by algorithm 2 in line 5-6. Here, Once the characteristic matrix is normalized, initial weights for each parameter is decided by the weight vector W , i.e., a set of m weights $W = \{w_i | i = 1, 2, 3 \dots m\}$. Total weight for all the parameters should be equal to 1, i.e., $\sum_{i=1}^m w_i = 1$. Similarly, the initial membership matrix U is defined as $U^0 = (u_{ij})_{c \times n}$. Where c represents the number of classes and n represent the number of nodes. u_{ij} represent the membership of j^{th} node in i^{th} cluster. The weights W , membership U , and cluster centers CC are optimized using fuzzy optimization technique in algorithm 2, line 15-18. Once the weight vector is initialized, each node aggregates these parameters according to their weights. The aggregate weighted trust values are stored in vector T , i.e., $T_i^w = \{t_i | i = 1, 2, 3 \dots n\}$. T_i^w represents the weighted trust value of i^{th} node calculated using algorithm 6.2 in line 20-22. Once all the nodes have trust values, then the fuzzy classification is used to classify nodes in c classes. Let Cl be a vector of c clusters for node

classification. Each node will be classified into c classes defined as $Cl = \{\text{Bad, Neutral, and Good}\}$ using membership matrix U as shown in algorithm 6.2 line 19. Initially, all the nodes were considered as neutral and had been assigned a trust value 0.5. Further, the rewards and punishments are calculated in line 23, and these values are used for direct trust evaluation using an exponential decay function (Bao, and Chen, 2012) as shown in line 24-27.

6.2.2 Trust Calculation

To calculate trust values, each cluster node calls $Cal_Trust(P)$ algorithm with a set of parameters (P). The final trust calculation consists of direct trust and indirect trust, i.e., recommendations. Each node calculates direct trust value of its one-hop neighbours using $Direct_Trust(P)$. The node request for the recommendations to its neighbour node using $Indirect_Trust(i)$. α and β are the weights given to the direct trust and indirect trust. The sum of α and β should be equal to one.

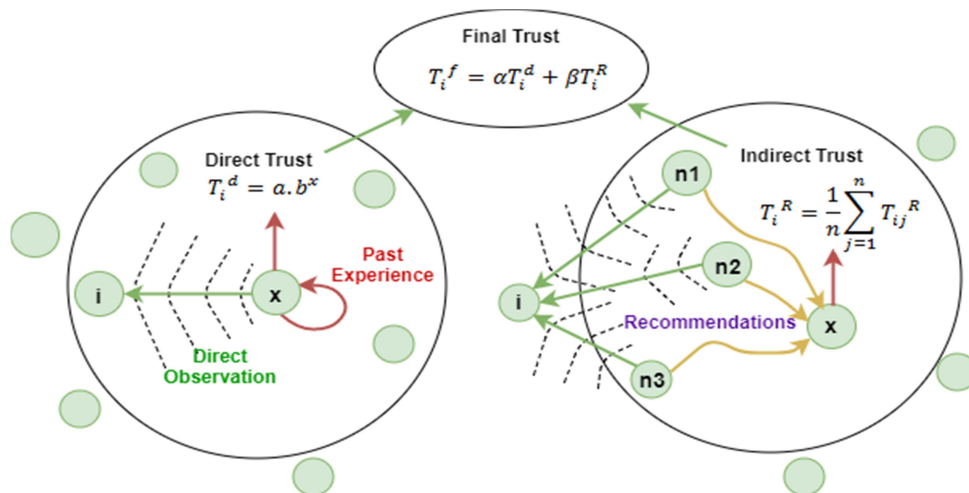


Figure 6.3: Direct and Indirect Trust Calculation

Algorithm 6.1: Trust Calculation TBCS: Cal_Trust(P)**Input:** *Weights for direct and indirect trust α and β* **Output:** *Final trust value T_i^f*

1. $T_i^d \leftarrow \text{DirectTrust}(P_i)$
2. $T_i^R \leftarrow \text{IndirectTrust}(i)$
3. $T_i^f = \alpha T_i^d + \beta T_i^R$

return T_i^f **Algorithm 6.2: Direct Trust Calculation in TBCS: Direct_Trust(P)****Input:** *Parameter matrix P , initial weight vector W , number of clusters c , previous trust a* **Output:** *Direct trust value T_i^d*

32. $N = \{N_i | i = 1, 2, 3 \dots n\}$ // N is a set of total n nodes in the network
33. $P = \{P_i | i = 1, 2, 3 \dots m\}$ // P is a set of m attributes
34. $W = \{w_i | i = 1, 2, 3 \dots m\}$, where // W is set of m weights
35. $\sum_{i=1}^m w_i = 1$
36. Computes a characteristic vector $c_i = \{c_{i1}, c_{i2}, c_{i3} \dots c_{im}\}$ and define $C = (c_{ij})_{n \times m}$.
37. Normalize matrix C to Recognition matrix $R = (r_{ij})_{n \times m}$.
38.
$$r_{ij} = \frac{c_{ij} - c_{imin}}{c_{imax} - c_{imin}}$$
39. Use membership functions to calculate initial membership matrix U^0
40. $cc_{ij} = \frac{\sum_{j=1}^n u_{ij}^2 r_{ij}}{\sum_{j=1}^n u_{ij}^2}$ // for calculating initial cluster center
41. **while** ($iter < n$)
42. $J = \sum_{i=1}^n \sum_{j=1}^c \{u_{ij}^f \sum_{k=1}^m w_k (r_{ik} - cc_{jk})^2\}$
43. **if** ($iter \sim 1$) & ($|J(iter) - J(iter-1)| < \delta$)
44. **break**;
45. **end**
46. $u_{ij} = \left[\sum_{k=1}^c \left[\frac{\sum_{l=1}^m w_l (r_{il} - cc_{kj})^2}{\sum_{l=1}^m w_l (r_{il} - cc_{lj})^2} \right] \right]^{-1}$ // optimizing membership matrix U
47. $cc_{ij} = \frac{\sum_{j=1}^n u_{ij}^2 r_{ij}}{\sum_{j=1}^n u_{ij}^2}$ // optimizing cluster centers CC
48. $w_i = \left[\sum_{k=1}^m \left[\frac{\sum_{j=1}^n \sum_{h=1}^c [w_k (r_{ij} - cc_{ih})^2]}{\sum_{j=1}^n \sum_{h=1}^c [w_k (r_{kj} - cc_{kh})^2]} \right] \right]^{-1}$ // optimizing weights

```

49. end while
50. Assign classes to nodes using membership matrix  $U$ 
    
$$\text{class\_label} = \text{argmax}(u_{ij})$$

51. for  $i = 1$  to  $n$ 
52.    $T_i^w = \sum_{j=1}^m w_j r_{ij}$  // Calculating weighted trust ( $T_i^w$ )
53. end for
54.  $b \leftarrow \text{Payoff}(i, \text{class\_label}(i))$ 
55. for  $i = 1$  to  $n$ 
56.    $T_i^d = a \cdot b^{T_i^w}$  // Calculating direct trust ( $T_i^d$ ) using decay function
57. the end of
58. return  $T_i^d$ 

```

Algorithm 6.3: Indirect Trust Calculation in TBCS: Indirect_Trust(i)

```

1.  $T_i^R \leftarrow \text{request\_Trust}(i)$ 
2. Wait for time  $t$  to receive feedback from other nodes
3.  $T_i^R = \frac{1}{n} \sum_{j=1}^n T_{ij}^R$  // Recommendation of node  $i$  from other network nodes
4. return  $T_i^R$ 

```

6.2.3 Reward and Punishment Calculation

All nodes are classified in different classes (bad, neutral and good) and they get reward or punishment accordingly. These values help to reduce or increase the direct trust values based on node's behaviour. Therefore, the behaviour of node converges faster to its actual behaviour using reward and punishment mechanism (Wang, Y., 2008). Table 6.2 represents the reward and punishment values based on different classes.

Table 6.2: *Reward and punishment*

Class	Rewards	Punishments
Bad	0	0.5
Neutral	1.5	0
Good	2	0

Algorithm 6.4 explains the assignment of reward and punishment based on the class. For example, let a node i belongs to bad class. Node i will be punished and assigned punishment value 0.5 as shown in line 2-4 of algorithm 6.4. Similarly, if a node belongs to a good class. Node i will be rewarded and assigned reward value 2.

Algorithm 6.4: Reward and Punishment in TBCS: Payoff (i , $class_label(i)$)

Input: Node and its assigned classes

Output: Reward / Punishment based on the class label

```
25. for  $i=1$  to  $n$ 
26.   if  $strcmp(class\_label\{i\}, 'Bad') == 1$ 
27.      $Punishment(i) = 0.5;$ 
28.   end if
29.   if  $strcmp(class\_label\{i\}, 'Neutral') == 1$ 
30.      $Reward(i) = 1.5;$ 
31.   end if
32.   if  $strcmp(class\_label\{i\}, 'Good') == 1$ 
33.      $Reward(i) = 2;$ 
34.   end if
35. end for
36. return Reward / Punishment
```

6.2.4 Cluster Head Selection

Initially, all the cluster nodes calculate trust values of their neighbouring nodes and select the cluster head with highest trust value. After a certain time period, CH starts election process for the new CH and broadcast this information shown in algorithm 6.5 (line 2). CH broadcasts the information about the election and wait for t seconds for other nodes to nominate their names. CH lookup trust table for trust values of candidate nodes using $lookup(FT, X_{id})$. All the nominated node's trust value is compared with threshold (T) value. List of all eligible nodes to be a new CH is prepared and broadcasted to cluster nodes. The cluster nodes vote for the eligible nodes and information regarding this is stored in a matrix V . The nodes which receive maximum votes declared as a leader. Leader id (lid) and the *node-ids* of those nodes which voted for it are broadcast. If any node sense false voting information, it can send a warning message. CH waits for warning messages for time period W . If no warning message received than leader node is declared as a new CH.

Algorithm 6.5: Cluster Head Selection in TBCS: CH_selection(n, T^f, T)

Input: Number of nodes in clusters (n), Final trust table (T^f), Threshold (T)

Output: Cluster heads (CH)

1. $N = \{N_i \mid i = 1, 2, 3, \dots, n\}$
2. Broadcast start_election() // CH starts new CH election
3. Wait(t) // wait for time period t
4. $CH \leftarrow interested_election(X_{id})$ // Node X sends request to become a new

```

CH
5.  $Trust\_val \leftarrow lookup(FT, X_{id})$  // CH lookup trust table for trust values of
   candidate nodes
6. if ( $Trust\_val(X) > T$ )
7.      $Eligible\_list \leftarrow Node\ X_{id}$  // List of all eligible nodes to be a CH
8. end
9. Broadcast  $eligible\_list$ 
10.  $V \leftarrow respond(Own_{id}, Nom_{id})$  // V is a matrix containing id of nominated nodes
   and voter
11.  $[lid, node-ids] \leftarrow Max(V)$  // Max returns node with highest votes and its
   voter ids
12. Broadcast  $leader(lid, node-ids)$ 
13.  $Warning\_msg = Wait(W_t)$  // wait for warning messages for time period
    $W_t$ 
14. If ( $Warning\_msg \neq 0$ )
15.     Discard that lid and repeat from line 10
16. else
17.     Declare node with lid as new CH
18. end if

return CH

```

6.2.5 Node Join

In the FANET, whenever a new node wishes to join the network, it sends a join request $\{Send_join(id, P)\}$ with its id and parameter set (P). CH calculates the trust value of that node using $Cal_Trust(P)$. If it qualifies a threshold value (Th), then only CH adds this node as a temporary node in the network. If its fail to qualify then discard that

node. CH observe the new joining node for a certain time period (t) and recalculates the new trust value (nT_{val}) before making it permanent node. The detailed procedure of a new node registration is shown in algorithm 6.6.

Algorithm 6.6: New Node joining in TBCS: Node_join()

```

1.  $CH \leftarrow Send\_join(id,P)$            // A new send join the CH and send its ids
   and parametrs
2.  $T_{val} = Cal\_Trust(P)$ 
3. if ( $T_{val} > I_h$ )
4.   Add node id to cluster as a Temporary node
5. else
6.   Discard that node
7. end if
8. if( $Wait\_time < t$ )
9.   Keep observing
10. else
11.    $nT_{val} = Cal\_Trust(P)$            // recalculate trust
12. end if
13. if ( $nT_{val} > I_h$ )
14.   Add node to cluster as a permanent node
15. end if

```

6.3 IMPLEMENTATION AND RESULTS

In this section, various parameters have been chosen that define the behaviour of a node and their membership in node classification.

Furthermore, we describe the simulation setup and discuss various results based on the proposed trust model.

6.3.1 Key Parameter Selection

To evaluate trust of a network node, parameter selection plays an important role. The selected parameters should be able to reflect the real behaviour and characteristics of a node in the network. For trust evaluation in FANET, we plan to select those features which represent both behavioural properties and network properties. The FANETs node moves with very high velocity and dynamic mobility pattern, which affects the packet delivery ratio. Higher the mobility of a node leads to more packet drop. Another parameter, i.e., node's energy plays an important role in effective communication. The nodes with lower energy may behave selfishly in the network. Network communication attribute, i.e., signal strength is also an important parameter to evaluate the trust of a node. We have taken four reliability parameters: node's energy, signal strength, the packet delivery ratio and past interactions. Along with one performance parameter, transmission delay. Therefore, the key parameter set is $A = \{\text{signal strength, packet delivery ratio, node's energy, transmission delay and past interactions}\}$.

6.3.2 Membership Functions

The five key attributes {signal strength, packet delivery ratio, node's energy, transmission delay, and past interactions} have been proposed to be used for calculating the trust in FANETs. These parameters are

discussed along with their membership functions; they are developed using Takagi-Sugeno-Kang fuzzy inference method (Sugeno, 1993). The objective of the fuzzy membership function is to include prior knowledge about the crisp value in the input space. We have taken triangular and trapezoidal membership fuzzy function to define our parameters because their properties are more appropriate to model the trust problem.

- i. **Signal Strength:** Free space path loss formula is used for calculating signal strength for different working distances in the proposed fuzzy model. The relation between signal attenuation and distance is calculated as $FSPL(dB) = 10 \log_{10} \left(\left(\frac{4\pi}{c} df \right)^2 \right)$ (Sklar, 2001), where FSPL(dB) stands for signal attenuation, measured in dB, d represents the distance from the transmitter, c stands for light speed in vacuum (2.99792458×10^8 m/sec), and f stands for signal frequency. A device with 802.11n runs at 2.4GHz frequency. By using the above formula, we can have signal attenuation for a working distance as: for a distance of 170m, 320m and 500m are 84.65dBm, 90.14 and 94.02 respectively with a network transmission power of 20dBm. A node is considered as a good node if it has signal strength more than -64.65dBm ($3.42 \times 10^{-7}mW$) and will be considered as a neutral node if signal strength is approximately -70.14dBm ($0.968 \times 10^{-7}mW$), and considered as a bad node if signal

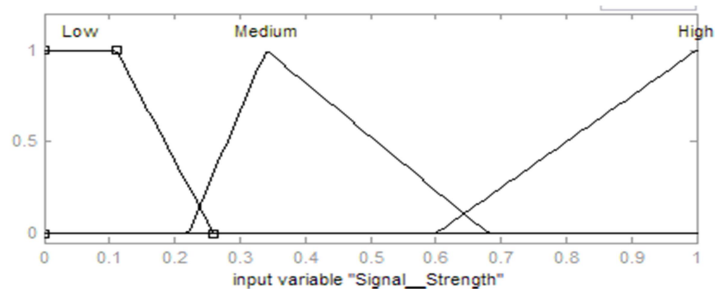
strength is around -74.02dBm ($0.396 \times 10^{-7}\text{mW}$). Signal strength value is normalized using $(x/3.42)$.

- ii. Packet Delivery Ratio:** Based on the network performance evaluation (Rappaport, 2002), a network of nodes was simulated, and the highest, lowest and average values of the packet delivery ratio were captured. A node having more than 90% of packet delivery ratio is said to be reliable and considered as a node with a good reputation. A node with more than 75% of the delivery ratio is considered as a node with neutral reputation, and if a network node has less than 75% packet delivery ratio, then it will be a non-reliable node and considered as a node with a bad reputation.
- iii. Node's Energy:** The energy level of a node determines how much that node is capable of handling the network task and for how long it can work properly. A network node having energy higher than 70% is said to be reliable and considered as a node with a good reputation. A node with more than 40% of energy, is considered as a node with a neutral reputation. A node with less than 40% of its total energy, will be a weak node and considered as a node with a bad reputation.
- iv. Transmission Delay:** Network performance can be evaluated by measuring the transmission delay between source and destination. This criterion can also be used to assess the trust of a node. A selfish or malicious node may transmit data with delay. Transmission delay can be defined as a ratio of data size

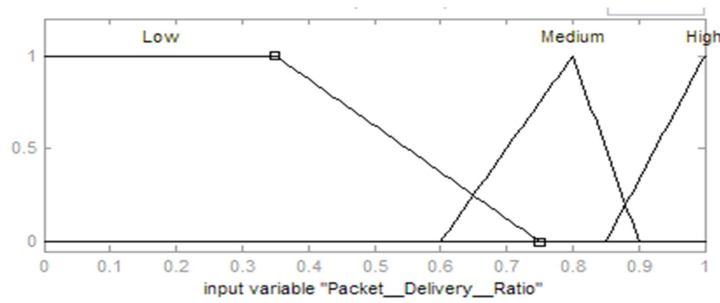
and rate of transmission using equation ($D_T = L/R$). Here, L stands for data size and R represents the rate of transmission (Rappaport, 2002). For example, with packet size 1024 Byte and 11Mbps transmission rate, transmission delay will be 0.74ms. A node with a transmission delay of 0.74ms was considered as a good node, a node having a transmission rate of nearly 8 Mbps taking the transmission delay around 1.1ms was considered as a neutral node, whereas a node with a transmission rate less than 6 Mbps will take more than 1.36ms transmission delay. Such node was considered as a bad node. We have assumed that in worst case scenario transmission delay will not exceed 2.73ms.

- v. **Past Interactions:** Building trust is an overtime phenomenon, e.g., humans build opinions based on their past experiences with others and calculate trust. Similarly, we consider that the network nodes will also keep track of their past experience with their neighbouring node. Past experiences include all the good and bad activities performed by neighbour nodes. Past interactions (PI) are measured by the ratio of interactions of node i with j, i.e., $I_i(j)$ to the total interactions of node i, i.e., I_i during the time period (CT – tw) where CT represents the current time and tw is the time window. The equation used to calculate PI is as follow: $PI(CT - tw) = \frac{I_i(j)}{I_i}$. A network node having PI more than 50% is said to be reliable and considered

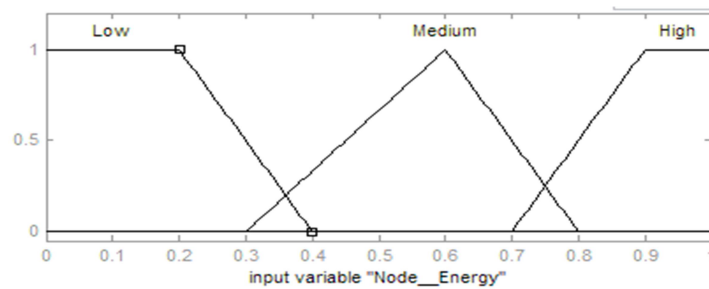
as a node with a good reputation. A node with PI between 25% and 55% is considered as a neutral node. A node with less than 30% PI is a weak node and considered as a node with a bad reputation.



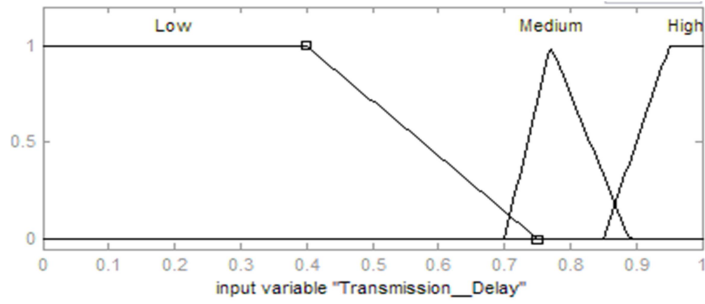
a) Signal Strength



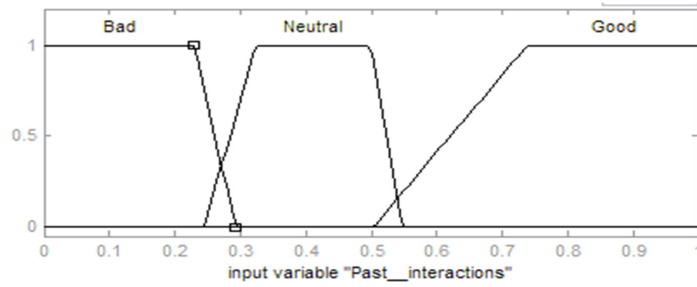
b) Packet Delivery Ratio



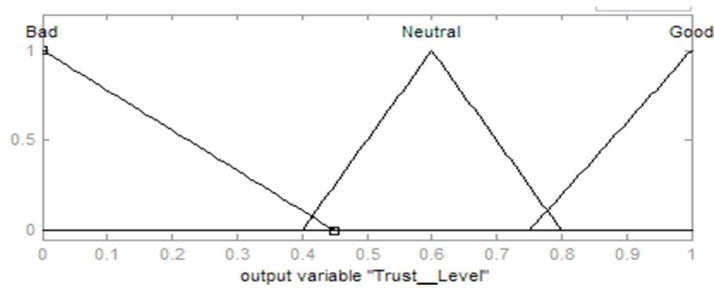
c) Node Energy



d) Transmission Delay



e) Past Interactions



Output: Trust level

Figure 6.4: Fuzzy Membership Functions

6.4 EXPERIMENTAL RESULTS

In this section, the proposed model is evaluated by setting up a simulation environment. The simulation environment creates an ad-hoc network, which consists of benign and malicious nodes. There are three types of nodes (good, neutral and bad nodes). Good nodes are

the ones who follow network protocols and do not act maliciously. The neutral nodes are those which most of the time behave ideally and sometimes act maliciously whereas bad nodes are those who continually misbehave. Further, warm-up time of 40 sec is taken before recording the readings, and each phase is simulated ten times, and the average of these simulations is calculated and presented.

The proposed model is evaluated based on classification accuracy, trust values and its performance in the network. The proposed model has been compared with other trust models: COI-HiTrust, AFStrust and the model proposed by Pushpita *et al.* COI-HiTrust model is chosen because this model is built to achieve critical mission irrespective of a node failure, compromise or disconnection. Also, the COI-HiTrust model considers the QoS and social behaviour of nodes to evaluate the trust level of nodes. The second model compared with the proposed model is AFStrust, which calculates the trust values using fuzzy logic. This model also considers direct and indirect trust (recommendations) as a part of the parameters for trust evaluations. The model proposed by Pushpita *et al.* is a cluster based trust model that computes the trust of a node using self and recommendation evidence of its one-hop neighbours. Hence, these models were found suitable for performance comparison. The experimental results show the effectiveness and accuracy of the proposed model.

6.4.1 Classification Accuracy

Using fuzzy classification method, the proposed model categorizes nodes into good, neutral and bad based on their behaviour in the network. Correct classification of these nodes acts as a measure to evaluate the performance of any trust model. In this paper, fuzzy c-means classification algorithm has been evaluated at different network sizes. The number of nodes in the network was varied between 10-50 and percentage of malicious nodes was varied between 10-80.

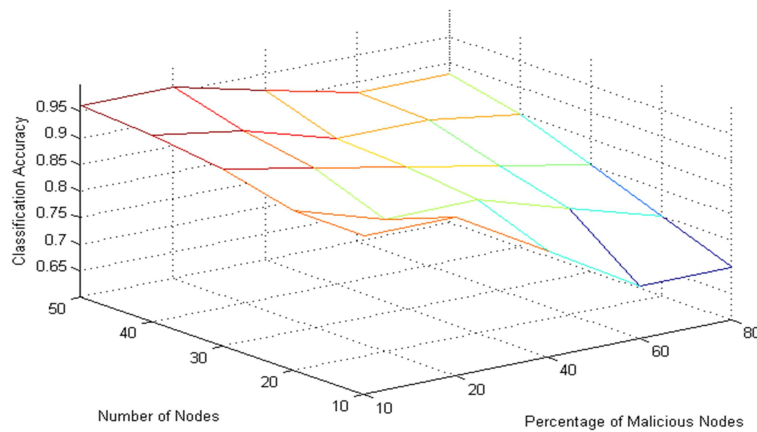


Figure 6.5: Classification Accuracy of Proposed Model

Figure 6.5 shows that the classification accuracy is dependent on network size and percentage of malicious nodes. The classification accuracy improves with larger network size and lesser malicious nodes. The proposed trust model can predict the behaviour of nodes with more than 90% accuracy when the network size is 40 or more and percentage of malicious nodes is 20. However, the accuracy reduces when large network having more malicious nodes.

6.4.2 Mean Square Error (MSE)

In this paper, the proposed model calculates trust using indirect and direct trust values. To decide the best proportion to represent the node's real behaviour, we calculate the MSE as shown in Figure 6.6. We found the lowest MSE is for direct trust weightage =0.7 and indirect trust weightage =0.3. The MSE is higher when more weightage is given to indirect trust. This is because more weightage to indirect observation gives the opportunity to bad nodes to share false recommendations. Hence, the MSE comes out to be the minimum when direct observations are given more weightage.

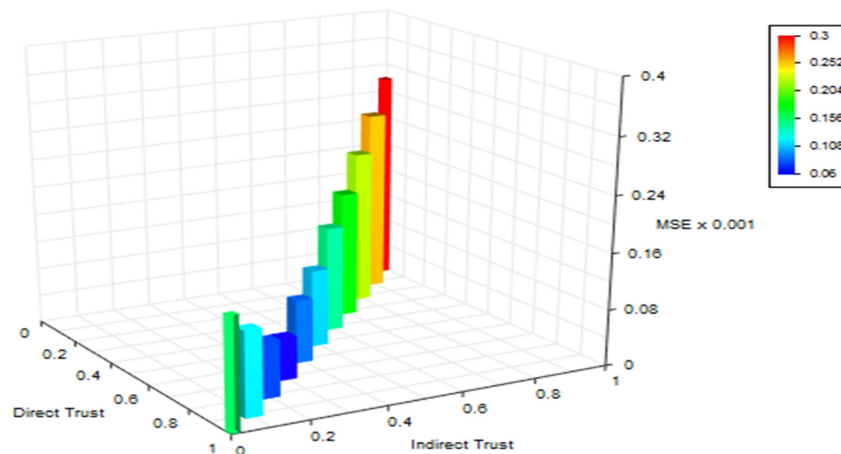


Figure 6.6: Mean Square Error

6.4.3 Convergence

The simulation environment assumes that all the nodes are neutral with a trust value 0.5 as shown in Figure 6.7. However, nodes may change their behaviour over a period of time and obtain different trust values. Here, two network nodes were observed for 400 seconds in a

simulated environment. The simulation results revealed that both the nodes started converging towards their actual behaviour. As shown in Figure 6.7, the good node achieves approximately the trust level of 0.75, and the bad node converges towards the bad behaviour line, i.e., the trust level of 0.4. This helps nodes to segregate bad nodes in their further communication in the network.

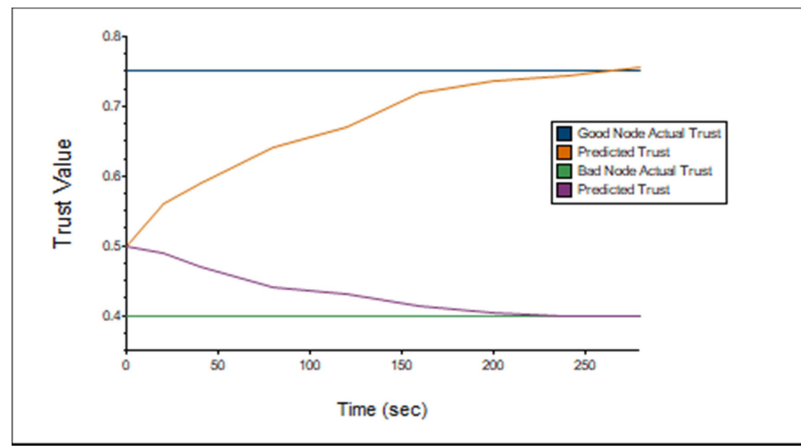


Figure 6.7: Convergence of Node 1 and Node 2 to their Actual Behaviour

6.4.4 Comparison of the Proposed Model with Existing Models

For evaluating and comparing the performance of the proposed model with COI-HiTrust, AFSTrust, and Pushpita *et al.*, a simulation environment has been setup. UAVs were randomly distributed in the simulation area. The network consists of three types of nodes: Good, Neutral and Bad. Normal nodes are the nodes which behave ideally and follow the network protocols. Malicious nodes try to breach the integrity of the network and drop the packets. We simulated this environment in Omnetpp. Using the Omnetpp, two network scenarios

were created. In the first scenario, an ad-hoc network of 50 nodes was created where a number of malicious nodes were fixed (15 nodes) but the node speed was varied (2-20 m/s). In a second network scenario, an ad hoc network of 100 nodes was created where node speed was fixed (8 m/s) but a number of malicious nodes were varied (5% to 50% nodes). The proposed model has been evaluated for Packet Drop Ratio (PDR) and Transmission delay.

Table 6.3: *Simulation Parameters*

Parameter	Value
Field Size	(800x800x400) m^3
Number of UAVs	100
Traffic Type	CBR
Packet Size	1024 bytes
Mobility Model	Random Waypoint
Node Placement	Random
Maximum Node Speed	20m/s
Default Trust Value	0.5
Trust Calculation Interval	10 Sec
Transmission Range	50-300m
Transport protocol	UDP
Mac Layer	802.11b
Propagation Radio Model	Two Ray
Routing Protocol	OLSRv2
Simulation time	400 Sec

- i. **Communication Overhead:** Next, the network has been evaluated by calculating communication overhead per node for different cluster sizes. The cluster size has been varied by

keeping a different number of nodes per cluster. Figure 6.8 shows, the effect of cluster size on communication overhead per node in the cluster. It has been observed that with the increase in cluster size, communication overhead also increases for both the cluster based methods. The communication overhead increases because some message exchanges during cluster formation is more for large sized clusters.

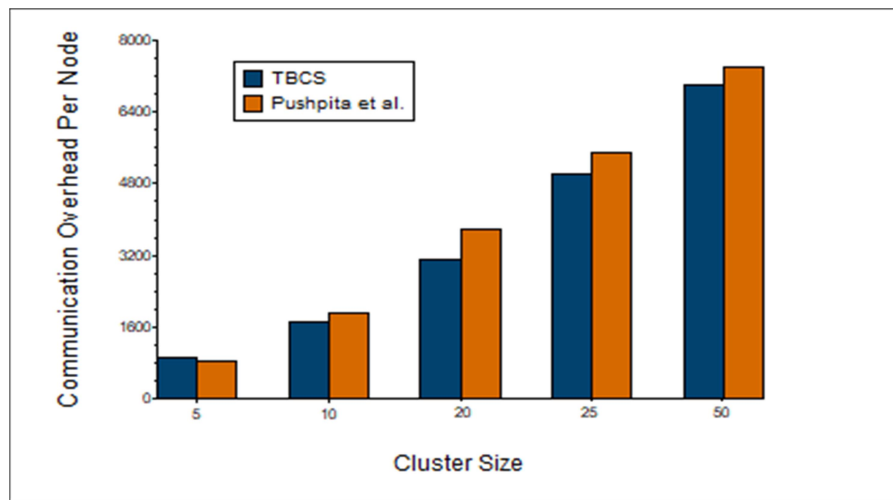


Figure 6.8: Cluster Size vs. Communication Overhead

- ii. **Packet Drop Ratio:** The value of packet drop ratio (PDR) decreases with the increment in the cluster duration. Because of the time required to deliver the messages decreases when the channel has less traffic (Figure 6.9). PDR decreases from 16 to 3% as the cluster duration increases from 5 to 35 seconds.

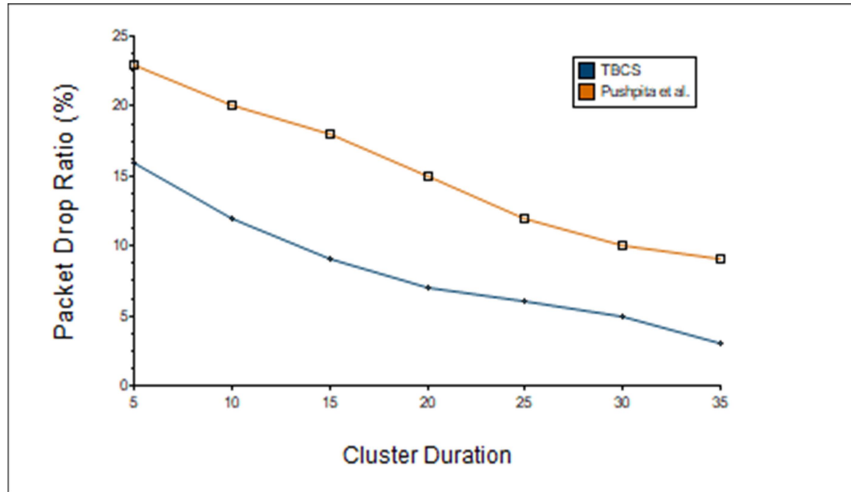


Figure 6.9: Cluster Duration vs. Packet Drop Ratio

The performance of both scenarios was evaluated using packet drop ratio (PDR). In scenario 1, the packet drop ratio increases as the node speeds are increased because of frequent connection breakage. As shown in Figure 6.10, the cluster-based models i.e., the proposed model and Pushpita *et al.*'s model outperform COI-HiTrust and AFSTrust trust models in terms of packet drop ratio. Among the two cluster-based models, the proposed model performs best.

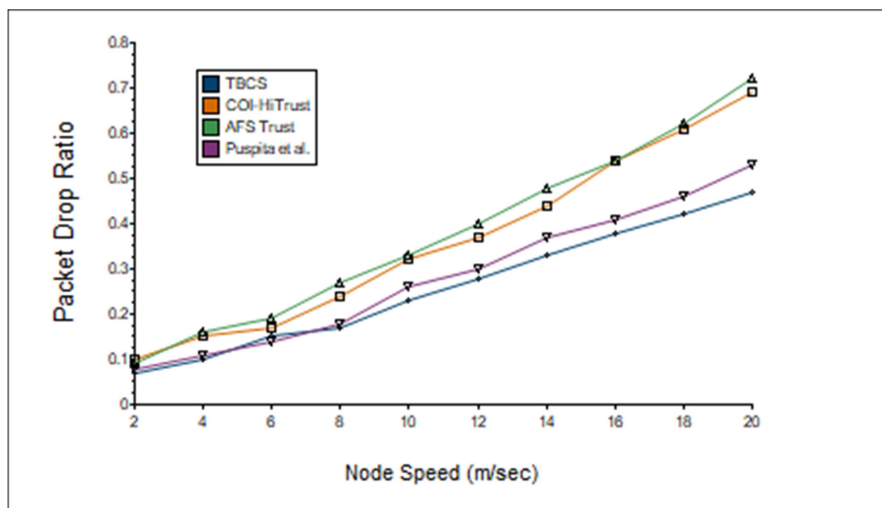


Figure 6.10: Node Speed vs. Packet Drop Ratio

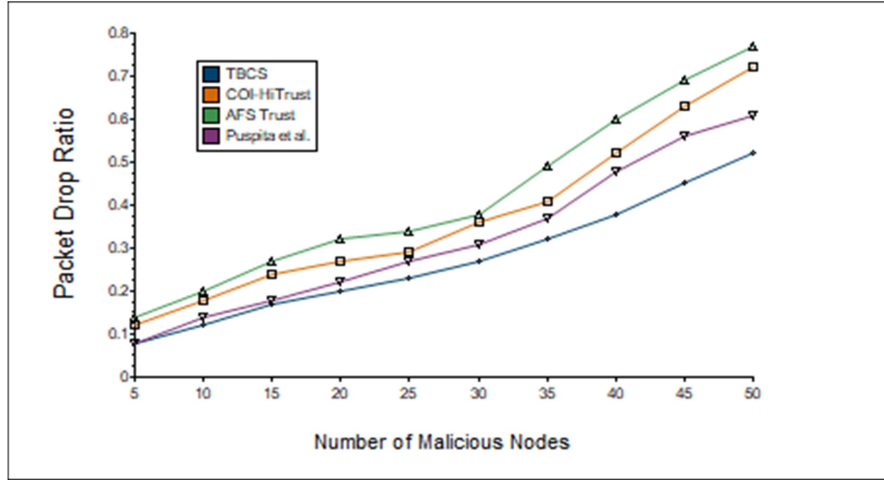


Figure 6.11: Number of Malicious Nodes vs. Packet Drop Ratio

In scenario 2, it has been observed that packet drop ratio increases as the malicious nodes increase in the network. As shown in Figure 6.11, implementing the proposed model helps to reduce the packet drop ratio in the network as compared to other techniques. Malicious nodes usually drop the packets, either to destroy communication or to save their energy for sending malicious packets in the network. The proposed trust model will help to segregate such nodes from network to improve the network performance.

iii. Transmission Delay: Next, we analyze all the algorithms in terms of transmission delay using both the scenarios. For scenario 1, Figure 6.12 explains that with increasing node speed transmission delay also increases. It has been observed that the performance of the proposed model is better compared to existing models with high-speed nodes. Hence the proposed model can better incorporate the high-speed nodes as available in FANETs.

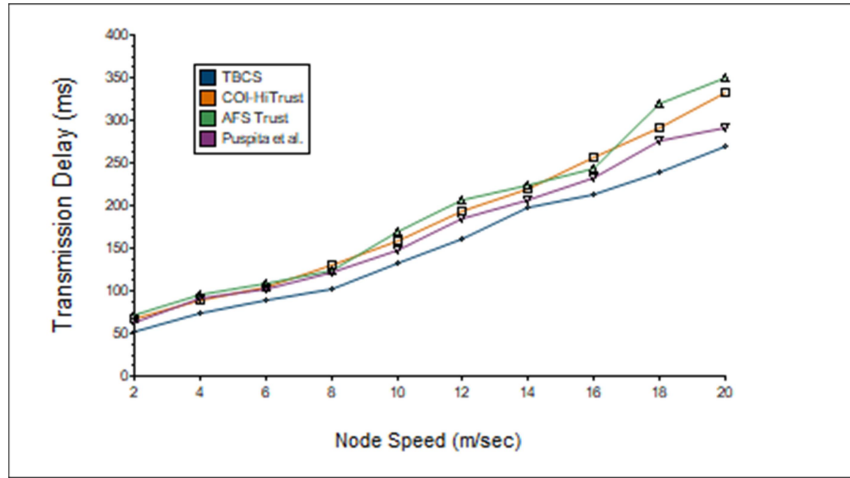


Figure 6.12: Node Speed vs. Transmission Delay

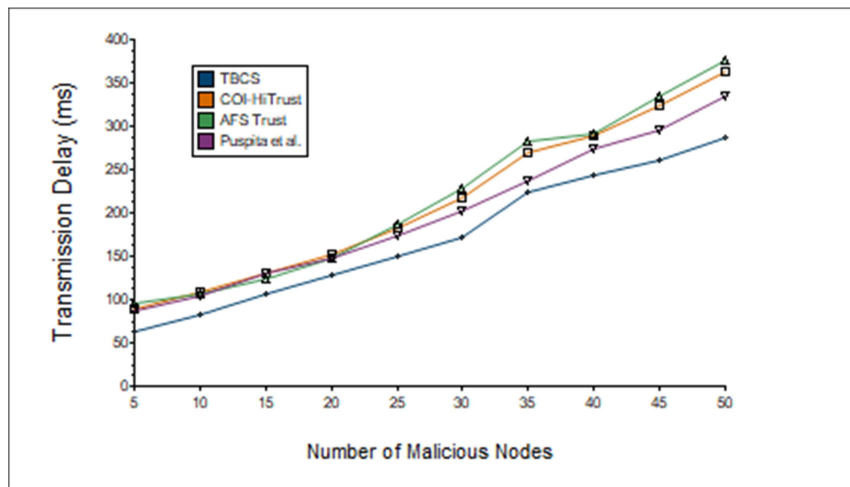


Figure 6.13: Number of Malicious Nodes vs. Transmission Delay

Figure 6.13 shows that, with the increasing number of malicious nodes, the transmission delay (ms) also increases in scenario 2. The transmission delay is low for the proposed model as compared to the other models. The proposed model can achieve an accuracy of more than 95% for node classification. As soon as the bad nodes are removed from the network, the performance of the network will be enhanced.

6.5 CHAPTER SUMMARY

In this chapter, a trust-based secure clustering scheme for FANET has been proposed. The proposed scheme uses fuzzy classification and optimization to evaluate trust. The scheme incorporates both direct and recommendations for detection and segregation of malicious nodes in FANETs. The secure CH is elected by giving priority to the node with highest trust value. The results vouch for the use of the proposed trust model to identify non-cooperative and malicious nodes. As the network size increase, detection accuracy increase, it indicates that the proposed method is highly scalable. The trust levels obtained from the model are in sync with node behaviour in the network. The proposed trust model has also been evaluated and compared with well-known trust models in ad-hoc network, i.e., COI-HiTrust, AFStrust model, and Pushpita *et al.*'s trust scheme. The proposed scheme out-performs these schemes in a highly dynamic environment of FANETs. For future research, the proposed model can be extended further to reduce the energy consumption.

CHAPTER 7

CONCLUSION AND FUTURE WORK

To prevent FANETs from various attacks due to node misbehaviour and selfishness, trust among network nodes can play a significant role. Trust management schemes can help to identify malicious and non-cooperative nodes in the network; can predict future behavior of nodes, and discover trusted nodes. FANETs are highly dynamic in nature which involves more uncertainty and high chances of information loss. In this thesis, we analyzed various trust management techniques that have already been implemented for ad hoc networks. It has been observed that fuzzy logic based trust schemes provide the capability to handle uncertainty, non-cooperativeness, and frequent behavioural changes of network nodes. So, Fuzzy logic based trust schemes are the best candidate for securing FANETs.

In this thesis, a trust-based secure framework for FANETs has been proposed that is capable of calculating trust among nodes and segregating malicious nodes from the network. Figure 7.1 provides a security framework for FANETs that has been developed in this thesis.

The aim of this thesis was to achieve confidentiality, integrity and availability. To achieve this aim, firstly a detailed threat analysis has been done to identify various threats in FANETs. Furthermore, risk analysis has been carried to measure criticality levels of identified threats. Next, this thesis studied various security solutions and analysed trust management solution in detail.

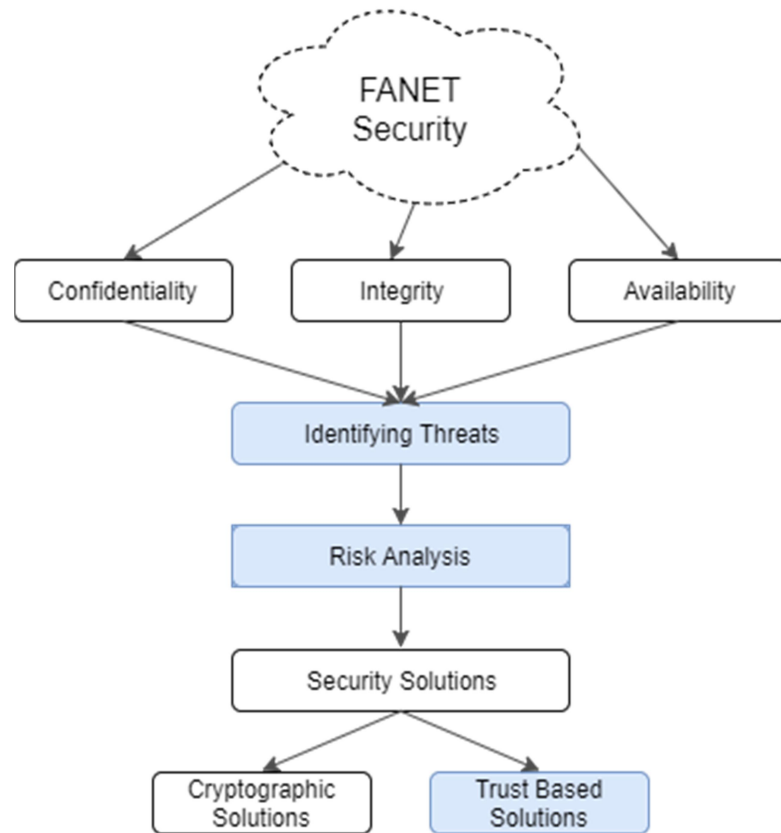


Figure 7.1: *Security Framework for FANETs*

This thesis developed three variants of trust models to secure FANETs: Fuzzy Classification Trust Model (FCTM), Weighted Fuzzy Classification Trust Model (WFCTM) and Trust Based Clustering Scheme (TBCS). FCTM is proposed for FANETs. Nodes are classified using a multi-criteria fuzzy classification method based on node's

behaviour and performance in the fuzzy and complex environment. Quality of Service (QoS) and Social parameter (recommendation) are considered for evaluating the trust value of each node to segregate the selfish and malicious nodes. Based on the node classification, FANET nodes are rewarded or punished to transform node behaviour into a trust value. Compared to the existing trust techniques, the simulation results show that the proposed model has better adaptability, accuracy, and performance in FANETs.

Furthermore, FCTM has been extended to WFCTM by introducing weight for different parameters. The concept of weights has been helpful to analyze the importance of various parameters. Furthermore, the concept of fuzzy classification has been used to develop clustering scheme, i.e., TBCS. The proposed scheme makes use of Takagi-Sugeno-Kang, fuzzy inference method. In TBCS, a secure cluster head (CH) has been selected based on calculated trust values.

The proposed trust models have also been evaluated and compared with well-known trust models in an ad-hoc network, i.e., COI-HiTrust, AFStrust models and the model proposed by Pushpita et al. The simulation results revealed that the proposed model has better adaptability, accuracy, and performance in FANETs. Among three proposed models, TBCS out-performs these models in a highly dynamic environment of FANETs.

Trust management in ad-hoc networks is an inspiring area of research. Its adaptation and applications in various situations or task-specific ad-hoc networks make it more challenging and open new vistas for the research community. This thesis highlights various trust computation methods that will help the research community to implement trust-based security solutions for FANETs. In this thesis, the proposed models have been tested in a simulated environment. The validation of simulation results in real time environment will be a significant step for future research.

ACKNOWLEDGMENTS

This research work is supported by Ministry of Electronics and Information Technology (MEITY), Government of INDIA as a part of the VISVESVERYA Ph.D. Scheme for Electronics and IT under Grant Ref. No. PHD-MLA/4(33)/2014-15.

REFERENCES

- Abbas, S., Merabti, M., Llewellyn-Jones, D., & Kifayat, K. (2013). Lightweight sybil attack detection in manets. *IEEE systems journal*, 7(2), 236-248.
- Abdul-Rahman, A., & Hailes, S. (1997, November). Using recommendations for managing trust in distributed systems. In *Proceedings IEEE Malaysia International Conference on Communication* (Vol. 97).
- Adams, W. J., Hadjichristofi, G. C., & Davis, N. J. (2005, April). Calculating a node's reputation in a mobile ad hoc network. In *Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International* (pp. 303-307). IEEE.
- Ahmed, A., Bakar, K. A., Channa, M. I., Haseeb, K., & Khan, A. W. (2015). A survey on trust based detection and isolation of malicious nodes in ad-hoc and sensor networks. *Frontiers of Computer Science*, 9(2), 280-296.
- Akbani, R., Korkmaz, T., & Raju, G. V. S. (2012). Mobile ad-hoc networks security. In *Recent Advances in Computer Science and Information Engineering* (pp. 659-666). Springer, Berlin, Heidelberg.
- Alghamdi, A. S., Hussain, T., & Khan, G. F. (2010, March). Enhancing c4i security using threat modeling. In *Computer Modelling and Simulation (UKSim), 2010 12th International Conference on* (pp. 131-136). IEEE.
- Almenarez, F., Marin, A., Díaz, D., & Sanchez, J. (2006, March). Developing a model for trust management in pervasive devices. In *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on* (pp. 5-pp). IEEE.
- Al-Shurman, M., Yoo, S. M., & Park, S. (2004, April). Black hole attack in mobile ad hoc networks. In *Proceedings of the 42nd annual Southeast regional conference* (pp. 96-97). ACM.
- Altawy, R., & Youssef, A. M. (2017). Security, privacy, and safety aspects of civilian drones: A survey. *ACM Transactions on Cyber-Physical Systems*, 1(2), 7.
- Angelov, P. (Ed.). (2012). *Sense and avoid in UAS: research and applications*. John Wiley & Sons.
- Balakrishnan, V., Varadharajan, V., & Tupakula, U. K. (2006, April). Fellowship: defense against flooding and packet drop attacks in MANET. In *Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP* (pp. 1-4). IEEE.
- Begum, S. A., Mohan, L., & Ranjitha, B. (2012). Techniques for resilience of denial of service attacks in mobile ad hoc networks. *Proceedings published by International Journal of Electronics Communication and Computer Engineering*3, 1.
- Bekmezci, I., Sahingoz, O. K., & Temel, Ş. (2013). Flying ad-hoc networks (FANETs): A survey. *Ad Hoc Networks*, 11(3), 1254-1270.
- Bekmezci, İ., Şentürk, E., & Türker, T. (2016). Security Issues in Flying Ad-Hoc Networks (FANETs). *Journal of Aeronautics and Space Technologies*, 9(2), 13-21.

- Beth, T., Borchherding, M., & Klein, B. (1994, November). Valuation of trust in open networks. In *European Symposium on Research in Computer Security* (pp. 1-18). Springer, Berlin, Heidelberg.
- Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2-3), 191-203.
- Blaze, M., Feigenbaum, J., & Lacy, J. (1996, May). Decentralized trust management. In *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on* (pp. 164-173). IEEE.
- Buchegger, S., & Le Boudec, J. Y. (2002). Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. In *Parallel, Distributed and Network-based Processing, 2002. Proceedings. 10th Euromicro Workshop on* (pp. 403-410). IEEE.
- Buchegger, S., & Le Boudec, J. Y. (2002, June). Performance analysis of the CONFIDANT protocol. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing* (pp. 226-236). ACM.
- Cai, J., Yi, P., Chen, J., Wang, Z., & Liu, N. (2010, April). An adaptive approach to detecting black and gray hole attacks in ad hoc network. In *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on* (pp. 775-780). IEEE.
- Carr, E. B. (2013). Unmanned aerial vehicles: Examining the safety, security, privacy and regulatory issues of integration into US airspace. *National Centre for Policy Analysis (NCPA)*. Retrieved on September, 23, 2014.
- Chae, Y., DiPippo, L. C., & Sun, Y. L. (2015). Trust management for defending on-off attacks. *IEEE Transactions on Parallel and Distributed Systems*, 26(4), 1178-1191.
- Chang W. J., Li H Y. and Tang J. J. (2016). A Flexible Cooperation Scheme for Mitigating the Effect of Selfish Nodes over MANETs, Adhoc & Sensor Wireless Networks, 33.
- Chatterjee, P., Sengupta, I., & Ghosh, S. K. (2012). STACRP: a secure trusted auction oriented clustering based routing protocol for MANET. *Cluster Computing*, 15(3), 303-320.
- Chatterjee, P., Ghosh, U., Sengupta, I., & Ghosh, S. K. (2014). A trust enhanced secure clustering framework for wireless ad hoc networks. *Wireless networks*, 20(7), 1669-1684.
- Chen, H., Wu, H., Zhou, X., & Gao, C. (2007, July). Agent-based trust model in wireless sensor networks. In *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on* (Vol. 3, pp. 119-124). IEEE.
- Chen, R., & Guo, J. (2014, May). Dynamic hierarchical trust management of mobile groups and its application to misbehaving node detection. In *Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on* (pp. 49-56). IEEE.

- Chen, R., Guo, J., & Bao, F. (2016). Trust management for SOA-based IoT and its application to service composition. *IEEE Transactions on Services Computing*, 9(3), 482-495.
- Cherdantseva, Y., & Hilton, J. (2013). Information Security and Information Assurance: Discussion about the Meaning. *Organizational, Legal, and Technological Dimensions of Information System Administration*, 167.
- Cho, J. H., & Swami, A. (2009). *Towards trust-based cognitive networks: A survey of trust management for mobile ad hoc networks*. ARMY RESEARCH LAB ADELPHI MD COMPUTATIONAL AND INFORMATION SCIENCES DIRECTORATE.
- Cho, J. H., Swami, A., & Chen, R. (2011). A survey on trust management for mobile ad hoc networks. *IEEE Communications Surveys & Tutorials*, 13(4), 562-583.
- Clark, J. A., Murdoch, J., McDermid, J. A., Sen, S., Chivers, H., Worthington, O., & Rohatgi, P. (2007, September). Threat modelling for mobile ad hoc and sensor networks. In *Annual Conference of ITA* (pp. 25-27).
- Clausen, T., Jacquet, P., Adjih, C., Laouiti, A., Minet, P., Muhlethaler, P., ... & Viennot, L. (2004). The optimised routing protocol for mobile ad-hoc networks: protocol specification. *INRIA*.
- Dai, H., Jia, Z., & Qin, Z. (2009). Trust Evaluation and Dynamic Routing Decision Based on Fuzzy Theory for MANETs. *JSW*, 4(10), 1091-1101.
- Dong, P., Wang, H., & Zhang, H. (2009, November). Probability-based trust management model for distributed e-commerce. In *Network Infrastructure and Digital Content, 2009. IC-NIDC 2009. IEEE International Conference on* (pp. 419-423). IEEE.
- Estahbanati, M. M., Rasti, M., & Hamami, S. M. S. (2014, September). A mobile ad hoc network routing based on energy and Markov chain trust. In *Telecommunications (IST), 2014 7th International Symposium on* (pp. 596-601). IEEE.
- Felegyhazi, M., Hubaux, J. P., & Buttyan, L. (2006). Nash equilibria of packet forwarding strategies in wireless ad hoc networks. *IEEE Transactions on Mobile computing*, 5(5), 463-476.
- Ferdous, R., Muthukkumarasamy, V., & Sithirasenan, E. (2011, November). Trust-based cluster head selection algorithm for mobile ad hoc networks. In *Trust, security and privacy in computing and communications (TrustCom), 2011 IEEE 10th international conference on* (pp. 589-596). IEEE.
- Gandhewar, N., & Patel, R. (2012, November). Detection and Prevention of sinkhole attack on AODV Protocol in Mobile Adhoc Network. In *Computational Intelligence and Communication Networks (CICN), 2012 Fourth International Conference on* (pp. 714-718). IEEE.
- Grey J., Weed J., Piana M., Bhandari S., Tang F., et al., "Collision Avoidance System for Unmanned Aerial Vehicles", in *Proceedings of the AUVSI Unmanned Systems*, 2015.

Glynos, D., Kotzanikolaou, P., & Douligeris, C. (2005, April). Preventing impersonation attacks in MANET with multi-factor authentication. In *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2005. WIOPT 2005. Third International Symposium on* (pp. 59-64). IEEE.

GnanaPrakasi, O. S. (2016). EFG-AOMDV: Evolving Fuzzy based Graph-AOMDV Protocol for Reliable and Stable Routing in Mobile Ad Hoc Networks. *Adhoc & Sensor Wireless Networks*, 33.

Gonzalez, J. M., Anwar, M., & Joshi, J. B. (2011, July). A trust-based approach against IP-spoofing attacks. In *Privacy, Security and Trust (PST), 2011 Ninth Annual International Conference on* (pp. 63-70). IEEE.

Govindan, K., & Mohapatra, P. (2012). Trust computations and trust dynamics in mobile adhoc networks: A survey. *IEEE Communications Surveys & Tutorials*, 14(2), 279-298.

Grandison, T., & Sloman, M. (2000). A survey of trust in internet applications. *IEEE Communications Surveys & Tutorials*, 3(4), 2-16.

Gupta, L., Jain, R., & Vaszkun, G. (2016). Survey of important issues in UAV communication networks. *IEEE Communications Surveys & Tutorials*, 18(2), 1123-1152.

Hartmann, K., & Steup, C. (2013, June). The vulnerability of UAVs to cyber attacks- An approach to the risk assessment. In *Cyber Conflict (CyCon), 2013 5th International Conference on*(pp. 1-23). IEEE.

Hu, Y. C., Perrig, A., & Johnson, D. B. (2006). Wormhole attacks in wireless networks. *IEEE journal on selected areas in communications*, 24(2), 370-380.

Information Classification – Who, Why And How, <https://www.sans.org/reading-room/whitepapers/auditing/information-classification-who-846> [Accessed 10.06.17].

Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., & Viennot, L. (2001). Optimized link state routing protocol for ad hoc networks. In *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International* (pp. 62-68). IEEE.

Jadidoleslamy, H. (2015). TMS-HCW: a trust management system in hierarchical clustered wireless sensor networks. *Security and Communication Networks*, 8(18), 4110-4122.

Jaisankar, N., Saravanan, R., & Swamy, K. D. (2010). A novel security approach for detecting black hole attack in MANET. In *Information processing and management* (pp. 217-223). Springer, Berlin, Heidelberg.

Jatoi, P. A., Memon, A. A., Chowdhry, B. S., Ullah, M. G., & Latif, S. (2015). An Efficient Hybrid Cryptographic Algorithm, Consuming Less Time for Exchanging Information in Wireless Sensor Networks. *Wireless Personal Communications*, 85(2), 449-462.

Javaid, A. Y., Sun, W., Devabhaktuni, V. K., & Alam, M. (2012, November). Cyber security threat analysis and modeling of an unmanned aerial vehicle system.

In *Homeland Security (HST), 2012 IEEE Conference on Technologies for* (pp. 585-590). IEEE.

Jøsang, A., Hayward, R., & Pope, S. (2006, January). Trust network analysis with subjective logic. In *Proceedings of the 29th Australasian Computer Science Conference-Volume 48*(pp. 85-94). Australian Computer Society, Inc.

Jøsang, A. (2007). Trust and reputation systems. In *Foundations of security analysis and design IV* (pp. 209-245). Springer, Berlin, Heidelberg.

Juszczak, P., Tax, D., & Duin, R. P. (2002, May). Feature scaling in support vector data description. In *Proc. ASCI* (pp. 95-102). Citeseer.

Kannhavong, B., Nakayama, H., Nemoto, Y., Kato, N., & Jamalipour, A. (2007). A survey of routing attacks in mobile ad hoc networks. *IEEE Wireless communications*, 14(5).

Kim, A., Wampler, B., Goppert, J., Hwang, I., & Aldridge, H. (2012). Cyber attack vulnerabilities analysis for unmanned aerial vehicles. In *Infotech@ Aerospace 2012* (p. 2438).

Kim, H., de Oliveira, R., Bhargava, B., & Song, J. (2013). A novel robust routing scheme against rushing attacks in wireless ad hoc networks. *Wireless personal communications*, 70(4), 1339-1351.

Komathy, K., & Narayanasamy, P. (2008). Trust-based evolutionary game model assisting AODV routing against selfishness. *Journal of Network and Computer Applications*, 31(4), 446-471.

Kozliuk, I. O., Bakhtiiarov, D. I., Lavrynenko, O. Y., & Tretiak, I. V. (2016). Problems of unauthorized interference to the work of uav and methods of its solving. *Наукоемні технології*, 30(2), 206-211.

Kurose, J. F., & Ross, K. W. (2009). *Computer networking: a top-down approach* (Vol. 4). Boston, USA: Addison Wesley.

Kute, D. S., Patil, A. S., Pardakhe, N. V., & Kathole, A. B. (2012). A Review: Manet Routing Protocols and Different types of attacks in Manet. *International Journal of Wireless Communication*, 2(1), 26.

Leike, A. (2001). Demonstration of the exponential decay law using beer froth. *European Journal of Physics*, 23(1), 21.

Li, F., & Wu, J. (2010). Uncertainty modeling and reduction in MANETs. *IEEE transactions on mobile computing*, 9(7), 1035-1048.

Li, J., Li, R., & Kato, J. (2008). Future trust management framework for mobile ad hoc networks. *IEEE Communications Magazine*, 46(4).

Li, R., Li, J., Liu, P., & Kato, J. (2009, June). A novel hybrid trust management framework for MANETs. In *Distributed Computing Systems Workshops, 2009. ICDCS Workshops' 09. 29th IEEE International Conference on* (pp. 251-256). IEEE.

Li, W., Parker, J., & Joshi, A. (2012). Security through collaboration and trust in manets. *Mobile Networks and Applications*, 17(3), 342-352.

- Liu, X., Datta, A., & Lim, E. P. (Eds.). (2014). *Computational trust models and machine learning*. CRC Press.
- Luo, J., Liu, X., & Fan, M. (2009). A trust model based on fuzzy recommendation for mobile ad-hoc networks. *Computer Networks*, 53(14), 2396-2407.
- Maddox, S., & Stuckenberg, D. (2015). Drones in the us national airspace system: A safety and security assessment. *Harvard Law School National Security Journal*.
- Mallapur, S. V., & Patil, S. R. (2015). Fuzzy logic based trusted candidate selection for stable multipath routing. *IJ Information Technology and Computer Science*, 6, 12-21.
- Mansfield, K., Eveleigh, T., Holzer, T. H., & Sarkani, S. (2013, November). Unmanned aerial vehicle smart device ground control station cyber security threat model. In *Technologies for Homeland Security (HST), 2013 IEEE International Conference on* (pp. 722-728). IEEE.
- Mansfield, K. M., Eveleigh, T. J., Holzer, T. H., & Sarkani, S. (2015). *DoD comprehensive military unmanned aerial vehicle smart device ground control station threat model*. DEFENSE ACQUISITION UNIV FT BELVOIR VA.
- Mármol, F. G., & Pérez, G. M. (2012). TRIP, a trust and reputation infrastructure-based proposal for vehicular ad hoc networks. *Journal of Network and Computer Applications*, 35(3), 934-941.
- Marshall, D. M., Barnhart, R. K., Shappee, E., & Most, M. T. (Eds.). (2015). *Introduction to unmanned aircraft systems*. Crc Press.
- Misra, S., & Rajesh, G. (2011). Bird flight-inspired routing protocol for mobile ad hoc networks. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 6(4), 25.
- Moe, M. E., Helvik, B. E., & Knapskog, S. J. (2008, October). TSR: Trust-based secure MANET routing using HMMs. In *Proceedings of the 4th ACM symposium on QoS and security for wireless and mobile networks* (pp. 83-90). ACM.
- Moustafa, H., & Zhang, Y. (2009). *Vehicular networks: techniques, standards, and applications*. Auerbach publications.
- Mui, L., Mohtashemi, M., & Halberstadt, A. (2002, January). A computational model of trust and reputation. In *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on* (pp. 2431-2439). IEEE.
- Müller, M. (2012). Flying ad-hoc networks. *Institute of Media Informatics Ulm University*, 53.
- Nait-Abdesselam, F., Bensaou, B., & Taleb, T. (2008). Detecting and avoiding wormhole attacks in wireless ad hoc networks. *IEEE Communications Magazine*, 46(4), 127-133.
- Nguyen, C. T., Camp, O., & Loiseau, S. (2007, March). A Bayesian network based trust model for improving collaboration in mobile ad hoc networks. In *Research, Innovation and Vision for the Future, 2007 IEEE International Conference on* (pp. 144-151). IEEE.

- Nguyen, C. T., & Camp, O. (2008, October). Using context information to improve computation of trust in ad hoc networks. In *Networking and Communications, 2008. WIMOB'08. IEEE International Conference on Wireless and Mobile Computing*, (pp. 619-624). IEEE.
- Nitti, M., Girau, R., & Atzori, L. (2014). Trustworthiness management in the social internet of things. *IEEE Transactions on knowledge and data engineering*, 26(5), 1253-1266.
- Nzouonta, J., Rajgure, N., Wang, G., & Borcea, C. (2009). VANET routing on city roads using real-time vehicular traffic information. *IEEE Transactions on Vehicular technology*, 58(7), 3609-3626.
- O'Donovan, J., & Smyth, B. (2005, July). Trust no one: Evaluating trust-based filtering for recommenders. In *IJCAI*(Vol. 5, pp. 1663-1665).
- Olmedilla, D., Rana, O. F., Matthews, B., & Nejd, W. (2006). Security and trust issues in semantic grids. In *Dagstuhl Seminar Proceedings. Schloss Dagstuhl-Leibniz-Zentrum für Informatik*.
- Park, S. S., Lee, J. H., & Chung, T. M. (2008, November). Cluster-based trust model against attacks in ad-hoc networks. In *Convergence and Hybrid Information Technology, 2008. ICCIT'08. Third International Conference on* (Vol. 1, pp. 526-532). IEEE.
- Pathan, A. S. K. (Ed.). (2016). *Security of self-organizing networks: MANET, WSN, WMN, VANET*. CRC press.
- Pelechrinis, K., Iliofotou, M., & Krishnamurthy, S. V. (2011). Denial of service attacks in wireless networks: The case of jammers. *IEEE Communications surveys & tutorials*, 13(2), 245-257.
- Perkins, C. E. (2001). *Ad hoc networking* (Vol. 1). Reading: Addison-wesley.
- Poovendran, R., Sampigethaya, K., & Bushnell, L. (2008). Security of future enabled aircraft AD HOC networks. In *The 26th Congress of ICAS and 8th AIAA ATIO* (p. 8894).
- Rappaport, T. S. (2002). Wireless Communications--Principles and Practice, (The Book End). *Microwave Journal*, 45(12), 128-129.
- Released, O., omnetpp-announcements, N., & Released, I. (2018). OMNeT++ Discrete Event Simulator - Home. Omnetpp.org. Retrieved 2 February 2018, from <https://www.omnetpp.org/>
- Richardson, M., Agrawal, R., & Domingos, P. (2003, October). Trust management for the semantic web. In *International semantic Web conference* (pp. 351-368). Springer, Berlin, Heidelberg.
- Roberts, G. A. (2016). *Open-source unmanned aerial vehicles: Vulnerabilities, exploits, and risk* (Doctoral dissertation, Utica College).
- Rosati, S., Kruzelecki, K., Traynard, L., & Rimoldi, B. (2013, December). Speed-aware routing for UAV ad-hoc networks. In *Globecom Workshops (GC Wkshps), 2013 IEEE* (pp. 1367-1373). IEEE.

- Saboji, S. V., & Akki, C. B. (2009). A survey of fourth generation (4G) wireless network models. *Journal of Telecommunications Management*, 2(1).
- Sahingoz, O. K. (2014). Networking models in flying ad-hoc networks (FANETs): Concepts and challenges. *Journal of Intelligent & Robotic Systems*, 74(1-2), 513-527.
- Samy Kamkar - SkyJack: autonomous drone hacking. (2017). Samy.pl. Retrieved 10 December 2017, from <http://www.samy.pl/skyjack/>
- Saeed, S., & Umar, M. S. (2016, October). PassNeighbor: A shoulder surfing resistant scheme. In *Next Generation Computing Technologies (NGCT), 2016 2nd International Conference on* (pp. 797-802). IEEE
- Sen, J., Chandra, M. G., Harihara, S. G., Reddy, H., & Balamuralidhar, P. (2007, December). A mechanism for detection of gray hole attack in mobile Ad Hoc networks. In *Information, Communications & Signal Processing, 2007 6th International Conference on* (pp. 1-5). IEEE.
- Shamir, A. (1984, August). Identity-based cryptosystems and signature schemes. In *Workshop on the theory and application of cryptographic techniques* (pp. 47-53). Springer, Berlin, Heidelberg.
- Singh, K., & Verma, A. K. (2014, May). Applying OLSR routing in FANETs. In *Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on* (pp. 1212-1215). IEEE.
- Singh, K., & Verma, A. K. (2015, March). Experimental analysis of AODV, DSDV and OLSR routing protocol for flying adhoc networks (FANETs). In *Electrical, Computer and Communication Technologies (ICECCT), 2015 IEEE International Conference on* (pp. 1-4). IEEE.
- Sklar, B. (2001). *Digital communications* (Vol. 2). Upper Saddle River: Prentice Hall.
- Song, S., Hwang, K., Zhou, R., & Kwok, Y. K. (2005). Trusted P2P transactions with fuzzy reputation aggregation. *IEEE Internet computing*, 9(6), 24-34.
- Spiewak, D., Engel, T., & Fusenig, V. (2006, November). Towards a threat model for mobile ad-hoc networks. In *ISP'06: Proceedings of the 5th WSEAS International Conference on Information Security and Privacy* (pp. 35-40).
- Srinivasan, V., Nuggehalli, P., Chiasserini, C. F., & Rao, R. R. (2003). Cooperation in wireless ad hoc networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*(Vol. 2, pp. 808-817). IEEE.
- Staab, S., Bhargava, B., Leszek, L., Rosenthal, A., Winslett, M., Sloman, M., ... & Olmedilla, D. (2004). The pudding of trust. *IEEE Intelligent Systems*, 19(5), 74-88.
- Storm, D. (2018). Drones infected with malware can drop from the sky or be hijacked for surveillance. Computerworld. Retrieved 13 February 2018, from <http://www.computerworld.com/article/2876912/drones-infected-with-malware-can-drop-from-the-sky-or-be-hijacked-for-surveillance.html>

Sun, Y. L., Yu, W., Han, Z., & Liu, K. R. (2006). Information theoretic framework of trust modeling and evaluation for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(2), 305-317.

Takagi, T., & Sugeno, M. (1993). Fuzzy identification of systems and its applications to modeling and control. In *Readings in Fuzzy Sets for Intelligent Systems* (pp. 387-403).

Theodorakopoulos, G., & Baras, J. S. (2006). On trust models and trust evaluation metrics for ad hoc networks. *IEEE Journal on selected areas in Communications*, 24(2), 318-328.

Umuhoza, D., Agbinya, J. I., & Omlin, C. W. (2007, August). Estimation of trust metrics for MANET using QoS parameter and source routing algorithms. In *Wireless Broadband and Ultra Wideband Communications, 2007. AusWireless 2007. The 2nd International Conference on* (pp. 80-80). IEEE.

Wang, X., Gao, X. Z., & Ovaska, S. J. (2008, September). A hybrid optimization method for fuzzy classification systems. In *Hybrid Intelligent Systems, 2008. HIS'08. Eighth International Conference on* (pp. 264-271). IEEE.

Wang, Y., & Vassileva, J. (2003, September). Trust and reputation model in peer-to-peer networks. In *Peer-to-Peer computing, 2003.(P2P 2003). Proceedings. Third international conference on* (pp. 150-157). IEEE.

Wang, Y., & Singh, M. P. (2007, January). Formal Trust Model for Multiagent Systems. In *IJCAI* (Vol. 7, pp. 1551-1556).

Wang, Y. (2008). Enhancing node cooperation in mobile wireless ad hoc networks with selfish nodes.

Wang, Y., Lu, Y. C., Chen, I. R., Cho, J. H., Swami, A., & Lu, C. T. (2014, December). Logittrust: A logit regression-based trust model for mobile ad hoc networks. In *6th ASE International Conference on Privacy, Security, Risk and Trust, Boston, MA* (pp. 1-10).

Wei, Z., Tang, H., Yu, F. R., Wang, M., & Mason, P. (2014). Security enhancements for mobile ad hoc networks with trust management using uncertain reasoning. *IEEE Transactions on Vehicular Technology*, 63(9), 4647-4658.

What is an IT Security Policy? - Palo Alto Networks. (2018). Paloaltonetworks.com. Retrieved: 11 January 2018, from <https://www.paloaltonetworks.com/cyberpedia/what-is-an-it-security-policy>

Wu, B., Chen, J., Wu, J., & Cardei, M. (2007). A survey of attacks and countermeasures in mobile ad hoc networks. In *Wireless network security* (pp. 103-135). Springer, Boston, MA.

Xia, H., Jia, Z., Ju, L., Li, X., & Zhu, Y. (2011, August). A subjective trust management model with multiple decision factors for MANET based on AHP and fuzzy logic rules. In *Green Computing and Communications (GreenCom), 2011 IEEE/ACM International Conference on* (pp. 124-130). IEEE.

Xia, H., Jia, Z., Ju, L., & Zhu, Y. (2011, November). Multicast trusted routing with QoS multi-constraints in wireless ad hoc networks. In *Trust, Security and Privacy in*

Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on (pp. 1277-1282). IEEE.

Xia, H., Jia, Z., Li, X., Ju, L., & Sha, E. H. M. (2013). Trust prediction and trust-based source routing in mobile ad hoc networks. *Ad Hoc Networks*, 11(7), 2096-2114.

Yaniv, I., & Kleinberger, E. (2000). Advice taking in decision making: Egocentric discounting and reputation formation. *Organizational behavior and human decision processes*, 83(2), 260-281.

Yan, Z., & Holtmanns, S. (2008). Trust modeling and management: from social trust to digital trust. *IGI Global*, 290-323.

Yap, W. S., Liu, J. K., Tan, S. Y., & Goi, B. M. (2015). On the security of a lightweight authentication and encryption scheme for mobile ad hoc network. *Security and Communication Networks*, 8(17), 3094-3098.

Yu, M., Zhou, M., & Su, W. (2009). A secure routing protocol against byzantine attacks for MANETs in adversarial environments. *IEEE transactions on vehicular technology*, 58(1), 449-460.

Yu, Y., Li, K., Zhou, W., & Li, P. (2012). Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures. *Journal of Network and computer Applications*, 35(3), 867-880.

Zhang, F., Jia, Z. P., Xia, H., Li, X., & Edwin, H. M. S. (2012). Node trust evaluation in mobile ad hoc networks based on multi-dimensional fuzzy and Markov SCGM (1, 1) model. *Computer Communications*, 35(5), 589-596.

Zhou, L., & Haas, Z. J. (1999). Securing ad hoc networks. *IEEE network*, 13(6), 24-30.

Thesis

by Kuldeep Singh

Submission date: 18-Aug-2018 12:12PM (UTC+0530)

Submission ID: 990910231

File name: Thesis_KuldeepSingh.pdf (13.46M)

Word count: 37332

Character count: 207208

Thesis

ORIGINALITY REPORT

6%

SIMILARITY INDEX

4%

INTERNET SOURCES

3%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

1

link.springer.com

Internet Source

<1%

2

users.encs.concordia.ca

Internet Source

<1%

3

Guo, Jia, Ing-Ray Chen, and Jeffrey J.P. Tsai. "A Survey of Trust Computation Models for Service Management in Internet of Things Systems", Computer Communications, 2016.

Publication

<1%

4

"Intelligent Computing, Networking, and Informatics", Springer Nature America, Inc, 2014

Publication

<1%

5

www.academypublisher.com

Internet Source

<1%

6

project.esolweb.com

Internet Source

<1%

7

Qureshi, Basit I.(Kouvatsos, Demetres and Min, Geyong). "Trust Management for P2P

<1%