

Department of Electrical and Computer Engineering

Reinforcement Learning based Trust framework for MANET Environment

Prabath Lakmal Rupasinghe

The Thesis is presented for the Degree of

Doctor of Philosophy

Of

Curtin University

August 2018

DECLARATION

To the best of my knowledge and belief, this thesis contains no material previously published by any other person except where due acknowledgement has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

A handwritten signature in blue ink, appearing to be 'P. J.', with a stylized flourish extending to the right.

Signature:

Date: 2018/08/20

LIST OF PUBLICATIONS

Refereed Conference Proceedings

1. Evaluation of Trustworthiness for Online Social Networks Using Advanced Machine Learning, Hansi Mayadunne, **Rupasinghe P.L**, conference on International Conference on Advanced in Computing Technology, 2018, Sri Lanka
2. A reinforcement learning approach to enhance the trust level of MANETs, Gihani Jinarajadasa, Wayomi Jayantha, **Rupasinghe P.L**, Iain Murray, conference on Smart Computing and Systems Engineering, 2018, Sri Lanka
3. Improving Trusted Routing by Identifying Malicious Nodes in a MANET Using Reinforcement Learning, Shanen Leen De Silva, Hansi Mayadunna, Iesha Wedage, Sasanka Pabasara, **Rupasinghe P.L**, Chethana Liyanapathirana, Krishnadeva Kesavan, Chamira Nawarathna, Kalpa Kalhara Sampath, conference on International Conference on Advances in ICT for Emerging Regions, 2017, Sri Lanka
4. Enhancing the Security of OLSR Protocol Using Reinforcement Learning, **Rupasinghe P.L**, Chamira Nawarathna, Kalpa Kalhara Sampath, conference on National Information Technology Conference, 2017, Sri Lanka
5. A Model to Represent Recommendation Based Trust for MANET Using Reinforcement Learning, **Rupasinghe P.L**, Chamira Nawarathna, Kalpa Kalhara Sampath, conference on National Information Technology Conference, 2017, Sri Lanka
6. Androsafe: Online malware analysis with statistic and dynamic methods Krishnadeva Kesavan , Chethana Liyanapathirana , Gayani Sadamali , S.A.W.S Sampath, Y.M. Sureni Koshila , Chamod Premarathne , **Rupasinghe P.L**, Chamira Nawarathna, IET Conference, 2016, Sri Lanka
7. A Light Weight Provenance Aware Trust Negotiation Algorithm for Smart Objects in IoT J.A.D.C.Anuradha Jayakody, **Rupasinghe P.L**, N.T Mapa, T.S Disanayaka, D.S.A.Kandawala, K.D.Dinusha Chathurangi, Krishnadeva Kesavan, IET Conference,

2016, Sri Lanka

8. Predictive Analytics with online data for WSO2 Machine Learner with the support of Ensemble method, **Rupasinghe P.L**, Heshani Herath, Ishani Pathinayake, Ashani Diaz, Indujayani Karthigesu, Krishnadeva Kesavan , Chethana Liyanapathirana, Sripa Vimuthi, IET Conference, 2016, Sri Lanka
9. NExT-Plugin for MetaTrader 5 based on Recurrent Neural Networks, H.A.C.J Hettiarachchi, L.I Hettiarachchi, M.M.N.M.Mullegama, R.K.T.DRanaweera, **Rupasinghe P.L**, D. Kasthurirathne, IET Conference, 2016, Sri Lanka
10. ClusterMal: Automated Malware Analysis with clustering, anomaly detection, and classification of existing and new behavioral analysis, Krishnadeva Kesavan, Sripa Vimukthi Bannakkotuwa, V.V.YY. Wickramanayake, M.P.D.H De Silva, J.M.D. Fernando, K.K.K.K. Sampath, **Rupasinghe P.L**, IET Conference, 2016, Sri Lanka
11. SDN Based Security Solution for Legislative Email Communications, **Rupasinghe P.L**, Murray I, ICCCA, 2016, India
12. Trustworthy Provenance Framework for Document Workflow Provenance, **Rupasinghe P.L**, Murray I. ICCTICT 2016, India
13. Global Trust Identification in Infrastructure and Ad Hoc Network, Karunaratne K.H.M.C.D, Karunaratne K.H.M.C.T, Peiris M.H.D, Peiris P.A.S, **Rupasinghe P.L**, Senarathne A, UBIMEDIA, 2015, Sri Lanka
14. Developing a QoS Based Routing Algorithm for VANET, Madushanka, P.L.M, Wijethilake, K.K.G.N.T, **Rupasinghe P. L**, NCTM 2014, Sri Lanka
15. A Trust Model for On-Demand Routing in Mobile Ad-Hoc Networks, Lokuliyana S, **Rupasinghe P.L**, NITC, 2014, Sri Lanka
16. Trust-Based Framework for Handling Communication Using Social Networks as Applied to Mobile Sensor Based Indoor Navigation System, **Rupasinghe P.L**, Murray I, IPIN, 2014, Korea

17. Accepted Abstract titled: Efficient Approach to Model Trust in MANET's using Hierarchical Probabilistic Graphical Models as applied to Indoor Collaborative Map Construction, **Rupasinghe P.L**, Murray I, IPIN, 2013, France
18. Accepted Abstract titled: "Security Aware, Low Overhead MANET Using Position-Based Routing as applied to Indoor Collaborative Map Construction," **Rupasinghe P.L**, Murray I, IPIN 2013, France
19. Reliable Power saved Communication Environment for MANET, Shanthi V, **Rupasinghe P.L**, National Conference on Technology and Management, NCTM, 2013, Sri Lanka
20. Authentication Algorithm to MANETs through Challenge-Response Based architecture, **Rupasinghe P.L**, Tennekoon R, Anushka B, Visagan S, Hettiarachchi B, Basnayake P, National Conference on Technology and Management, 2013, Sri Lanka
21. Decentralized Peer to Peer Web Caching for Mobile Ad Hoc Networks (iCache) , Jayasooriya, I.U, Nallaperuma, T.A, Herath, U.K, Ranasinghe, S.R, Liyanage, M, Tennekoon, R.L, **Rupasinghe P.L**, 8th International Conference on Computer Science & Education ICCSE, 2013, IEEE, Sri Lanka
22. Efficient, Authentication and Access control Implementation in Mobile Ad hoc Networks (MANET) as applied to Indoor Navigation Guidance System for Vision Impaired People, **Rupasinghe P.L**, Murray I, published under Peer-Reviewed Section, IPIN, 2012, Australia

Statement of Contribution by Others

The following research papers listed below are based on data generated in this thesis. The candidates major research contribution related to the conceptual design, experiments, algorithms and implementation and is presented in this thesis. Co-authors of these papers helped in analysis and provided advised and assistance in the interpretation of the results. A further explanation of which part of the thesis the research papers were based on is outlined below.

Paper Title: “Evaluation of Trustworthiness for Online Social Networks Using Advanced Machine Learning”

The work presented in this paper was based on an on the section in chapter 4 of the thesis. Research work was co-authored by an undergraduate student. Co-authors contributed to 50% of the work and “Rupasinghe” contributed to 50% of the work.

Paper Title: “A reinforcement learning approach to enhance the trust level of MANETs”

The work presented in this paper was based on an on the section in chapter 4 of the thesis. Research work was co-authored by an undergraduate student. Co-authors contributed to 50% of the work and “Rupasinghe” contributed to 50% of the work.

Paper Title: “Improving Trusted Routing by Identifying Malicious Nodes in a MANET Using Reinforcement Learning”

The work presented in this paper was based on an on the section in chapter 4 of the thesis. Research work was co-authored by an undergraduate student. Co-authors contributed to 50% of the work and “Rupasinghe” contributed to 50% of the work.

Paper Title: “Enhancing the Security of OLSR Protocol Using Reinforcement Learning”.

The major work presented in this paper was based on the work explained in in chapter 3 of the thesis. In the thesis it presents a framework of using reinforcement learning in managing

routing in an adhoc network. The same logic is being used in the paper applying to OLSR protocol. Co-authors contributed to 20% of the work and “Rupasinghe” contributed to 80% of the work.

Paper Title: “A Model to Represent Recommendation Based Trust for MANET Using Reinforcement Learning”.

The work presented in the paper was based on the algorithms which was developed in chapter 3 of the thesis. Algorithms and evaluations which used in the paper was based on the work conducted in chapter 4. Co-authors contributed to 20% of the work and “Rupasinghe” contributed to 80% of the work.

Paper Title: “Predictive Analytics with online data for WSO2 Machine Learner with the support of Ensemble method”.

The concept of the paper was originated from the work which is incorporated in the thesis in chapter 3. The usage of online social network data was tested using WSO2 machine learner model and the results was published. Co-authors contributed to 20% of the work and “Rupasinghe” contributed to 80% of the work.

Paper Title: “SDN Based Security Solution for Legislative Email Communications”.

The concept of the paper was originated from the work which is incorporated in the thesis in chapter 3. The usage of online social network data was tested using SDN based email communication. Co-authors contributed to 20% of the work and “Rupasinghe” contributed to 80% of the work.

Paper Title: “Trustworthy Provenance Framework for Document Workflow Provenance” .

The concept of the paper was originated from the work which is incorporated in the thesis in chapter 3. The usage of online social network data was tested using Document workflow provenance. Co-authors contributed to 20% of the work and “Rupasinghe” contributed to 80%

of the work.

Paper Title: “Trust-Based Framework for Handling Communication Using Social Networks as Applied to Mobile Sensor Based Indoor Navigation System”.

The model proposed in the chapter 3 and 4, which is related to social network trust prediction is presented in this paper. Co-authors contributed to 20% of the work and “Rupasinghe” contributed to 80% of the work.

Paper Title: “Authentication Algorithm to MANETs through Challenge-Response Based architecture”

The model proposed in the chapter 3 and 4, which is related to MANET trust prediction is presented in this paper. Co-authors contributed to 20% of the work and “Rupasinghe” contributed to 80% of the work.

Paper Title: “Efficient, Authentication and Access control Implementation in Mobile Ad hoc Networks (MANET) as applied to Indoor Navigation Guidance System for Vision Impaired People”.

This paper was written based on the novel principle which presented in the introduction of the thesis. Co-authors contributed to 20% of the work and “Rupasinghe” contributed to 80% of the work.



(Signature of Candidate)



(Signature of the Supervisor)

To my loving parents, wife, son, and daughter who made all the sacrifices for almost five years until this day is today.

ABSTRACT

According to world health organization records in 2017, 253 million people are visually impaired worldwide of which nearly 36 million are sightless or blind, and 217 million have low vision. Finding the correct direction is one of the furthestmost effortful activities that the visually weakened people encounter. The proposed research will be the part of the overall research project, which focuses on methods for real-time local navigation based on edge detection techniques in image processing. In this research work, the challenges and requirements involved in finding a secure environment in the infrastructure that supports the mentioned project. Current research work and implementation schemes use MANET (Mobile Ad-hoc Networks) as the infrastructure expertise. On the other hand, the attributes of MANET fundamentally offer higher difficulties in areas like security, reliability, and performance.

Ad-hoc networks are designed and implemented without the need for any infrastructure support. In MANET Authentication and access-control trust associations accomplished through, obtainable online proofs may be temporary as well as basically P2P (peer-to-peer), where the associates possibly will not certainly have an appropriate network that can be positioned into an identifiable confidence hierarchy. Trust associations concerning an obtained node necessity to be discredited, and new trust proof wants to be composed and assessed to sustain node association in the ad-hoc network.

This proposal demonstrates three different security procedures which were established in order to meet the aforementioned MANET security requirements. The first procedure is a decentralized social network analysis framework and a protocol for any socially acceptable (API-Application Program Interface enabled) web system. The research used the Facebook API for simulation and verification of this protocol within the thesis. The next protocol is a decentralized trust building protocol which works in parallel with a reactive MANET routing protocol called AODV. The before-mentioned environment will be working with entropy-based trust development framework, which utilizes a newly formed “Spiral” model to detect and prevent malicious and collaborative malicious nodes in a MANET network. Next, a trust prediction scheme was developed using deep reinforcement learning. Here the deep reinforcement learning model will predict a Q value, which permits the AODV routing protocol to choose the next secure and reliable hop.

ACKNOWLEDGEMENT

I would like to acknowledge the supervision of Prof. Iain Murray AM.

It is glad to acknowledge people who have supported this thesis to successful. To my parents who have all the time been there with their prayers and support. To my Wife and my children, I owe my sincere love and appreciation. They have stood by me all the way through the Ph.D., continuously contributing their backing in numerous manners. I honestly show gratitude them for all their inspiration, which has assisted me throughout this difficult expedition. To every member of my family, it is an honour for me to dedicate this thesis.

Moreover, I am enormously thankful to my primary supervisor, Prof. Iain Murray, a fanatical and enthusiastic researcher, and a motivating advisor. Prof. Murray at all the time succeeds to find time to review the concerns associated with my investigation, and it has been a really comfort to have him as my supervisor. I will every time be appreciative for his inspiration and admirable supervision. I comprehensively appreciated this academic expedition and acquired precious abilities empowering me to be a great researcher.

Afterward, I would like to give my gratitude to my co-supervisor Dr. Hannes Herrmann. His supervision granted me with the way to complete my thesis with appropriate excellence and accuracy.

To conclude, I would like to acknowledge every colleagues at the SLIIT and mainly at the ISE division for given that cooperative and energetic conversation atmospheres.

TABLE OF CONTENTS

DECLARATION	1
LIST OF PUBLICATIONS	2
ABSTRACT	9
ACKNOWLEDGEMENT	10
TABLE OF CONTENTS	11
LIST OF FIGURES	16
LIST OF TABLES	19
LIST OF ABBREVIATIONS	20
Chapter 1 – Introduction	22
1.1 Background	22
1.2 Motivation	23
1.3 Problem Statement	25
1.4 Thesis Contribution	26
1.5. Outline of the thesis	28
Chapter 2 - Literature Review	29
2.1 Mobile Ad-hoc Networks	29
2.2 MANET Routing Protocols	31
2.2.1 Ad-hoc On-demand Distance Vector (AODV) routing protocol	32
2.2.2 Dynamic Source Routing (DSR)	32
2.2.3 Better Approach to Mobile Ad-hoc Networking (BATMAN)	33
2.2.4 TORA	33
2.3 Routing in MANET	34
2.4 Routing Issues in MANET	35
2.4.1 Scalability	35

2.4.2 Reliability	37
2.4.3 Quality of Service (QoS)	37
2.4.4 Security	39
2.5 MANET features and their impact on security	41
2.6 Social Network Trust	47
2.7 Single Agent-Based Trust (Reinforcement Learning Techniques)	48
2.7.1 Dynamic Programming	50
2.7.2 Temporal Difference Learning (TD Learning)	50
2.7.3 Monte Carlo Learning	50
2.7.4 Deep Q Learning	51
2.8 Trust	52
2.8.1 Trust in General	52
2.8.2 Trust Challenges	53
2.8.2.1 Network Transfer Entropy	54
2.8.2.2 Information Entropy	54
2.8.2.3 Trust Entropy	55
Chapter 3 - Methodology and Experimental Design	58
3.1 Social Network Trust	58
3.1.1 Social networks	58
3.1.2 Social Network Trust and Trust Calculating Methodologies	59
3.1.3 Proposed Trust Calculation Model	60
3.1.4 Social Network Trust	61
3.1.5 Bayesian Belief Networks (BBNs)	61
3.1.6 Traditional Social Network Analysis	63
3.1.7 Parameter Extraction	66

3.1.8 Literature Based Parameters Extracted	68
3.1.9 Implementation of the model	69
3.1.10 Network Structure	71
3.2 Entropy-based Spiral Trust AODV (ESTAODV)	77
3.2.1 MANET Trust	77
3.2.2. Proposed ESTAODV model	78
3.2.3 Trust Model Classification	79
3.2.3.1 Basic classification	79
3.2.3.2 Structure-based classification	81
3.2.4 Trust MANET Security	85
3.2.5 Proposed Trust Model	86
3.2.5.1 Trust Model for Direct Trust	86
3.2.5.2 Trust Model for Indirect Trust	91
3.2.6 Trust Architecture for MANETS	104
3.2.7 Trust Calculation	118
3.2.7.1 Direct Trust Calculation	118
3.2.7.2 Indirect Trust Calculation	120
3.2.8 Mathematical Analysis of the Model	122
3.3 Deep Reinforcement Learning Approach	128
3.3.1 Deep Reinforcement Learning Oriented Trust Protocol	129
3.3.2 Architecture and Implementation	130
3.3.3 System Analysis	131
3.3.4 System Architecture & the Process Summary	134
3.3.5 Reinforcement Learning and Q-Learning	138
3.3.5.1 Machine Learning	138

3.3.5.2 Reinforcement Learning	139
3.3.5.3 Deep Reinforcement Learning	139
3.3.5.4 MDP	140
3.3.5.5 Exploitation and exploration	141
3.3.5.6 Greedy Policy	141
3.3.5.7 Reward Function	142
3.3.5.8 Recurrent Neural Networks	143
3.3.5.9 Long Short-Term Memory (LSTM)	144
3.3.5.10 Implementation of the Recurrent Neural Network	145
3.3.5.11 Activation Function	147
3.3.5.12 Q-Value Prediction	148
3.3.6 Comparison with other RL approaches	151
3.3.6.1 Monte Carlo Approach	152
3.3.6.2 Temporal difference (TD) learning and SARSA learning	152
3.3.6.3 SARSA learning Vs. Q-learning	153
3.3.7 Reinforcement Learning Framework	156
3.3.8 ACT (Actor-Critic Trust) Approach	157
Chapter 4 - Simulation results and Discussions	159
4.1 Experimental Results and evaluations in Social Network Trust	159
4.1.1 Constructing the Probability Tables	159
4.1.2 Compile the Bayesian Belief Network	160
4.1.3 Threshold Values of Trust	166
4.1.4 Trusted Network Filter Plugin	168
4.2 Simulation Results of ESTAODV and DRL	176
4.2.1 Simulation Setup	176

4.2.2 Simulation Results of Reinforcement Learning Algorithm	178
4.2.3 Simulation Matrices	192
4.2.4 Traffic Load	192
4.2.5 Node Density	195
4.2.6 Mobility	196
4.2.7 Link Quality	197
4.2.8 Varying number of malicious nodes	199
4.2.9 Impact of the framework on the detection of malicious nodes	201
4.2.10 Impact of the framework on detection of Collaborative malicious nodes	203
Chapter 5 - Conclusion and Future Work	206
5.1 Conclusions	206
5.2 Anticipated Benefits and Contribution to the Body of Knowledge	208
5.3 Research Constraints	209
5.4 Future Directions	210
Appendices	211
REFERENCES	243

LIST OF FIGURES

Figure 1.1: High-Level structure of the Indoor Navigation System (<i>Source: Adapted from [4]</i>)	24
Figure 1.2: diagram showing the functions of the Way-finding system for visually impaired people (<i>Source: Adapted from [4]</i>)	24
Figure 2.1: Infrastructure mode vs. Ad-hoc mode	30
Figure 2.2: Overview of Mobile Ad-hoc Networks.....	31
Figure 2.3: Classification of routing protocols in MANETs	31
Figure 2.4: Passive attack	42
Figure 2.5: Wormhole situation with (a) Single adversary R making a fake connect among the actual nodes A and B, (b) Collaborating opponents A1 and A2 making a connection amongst their nodes utilizing an out of band channel.	44
Figure 2.6: Sinkhole attack in the wireless network.....	47
Figure 2.7. Reinforcement Learning.....	49
Figure 2.8: Deep reinforcement learning overview	51
Figure 3.1: Building Trust within a Social Network.....	59
Figure 3.2: Trust Definitions and Measurements for Social Networks	60
Figure 3.3: Representation of the Developed System Architecture.....	64
Figure 3.4: Flow Chart of the Trust Calculation Process	65
Figure 3.5: Analyzed social network using Gephi.....	67
Figure 3.6: Trust relationship between neighbours.....	71
Figure 3.7: Hypothetical graph	72
Figure 3.8: Bayesian Belief Network Structure	77
Figure 3.9: Trust Computing Methods Classifications	81
Figure 3.10: Pictorial demonstration of the discrete calculating strategy.....	85
Figure 3.11: Direct Trust Calculation Process	87
Figure 3.12: Example: node A obtains recommendations regarding node D	94
Figure 3.13: Node A requesting recommendations about D from B and C.....	95
Figure 3.14: Concatenation trust propagation.....	102
Figure 3.15: Combining trust recommendations	102

Figure 3.16: One entity provides multiple recommendations.....	103
Figure 3.17: A1 requesting a recommendation about C1 from B ₁ and D1	104
Figure 3.18: Basic Trust Level Identification Process.....	107
Figure 3.19: Spiral Model.....	112
Figure 3.20: After Transmission Process.....	116
Figure 3.21: Example Network.....	122
Figure 3.22: Cluster-based Deep Reinforcement Learning architecture	131
Figure 3.23: Relationship between learning agents and nodes	132
Figure 3.24: System Architecture	134
Figure 3.25: Flow of the process of generating output through RNN	136
Figure 3.26: Reinforcement learning module	139
Figure 3.27: MDP State Model.....	140
Figure 3.28: Highest Q Values.....	142
Figure 3.29: Recurrent Neural Network	144
Figure 3.30: Basic formation of an LSTM block.....	145
Figure 3.31: Structure of the developed Recurrent Neural Network	146
Figure 3.32: Comparison of Activation Functions	148
Figure 3.33: Cliff world example.....	154
Figure 3.34: Reward episode	155
Figure 3.35: RLTM Framework class diagram.....	156
Figure 3.36: ACT model for identifying Trustworthy Paths	158
Figure 4.1: Sample Evidence I.....	161
Figure 4.2: Sample Evidence II	162
Figure 4.3: Gephi Architecture with new Trusted Network Plugin.....	169
Figure 4.4: Test Results – MAE – Trusted Network	173
Figure 4.5: Test Results – RMSE – Trust Network.....	174
Figure 4.6: Test Results – MAE – Most Untrustworthy Nodes.....	174
Figure 4.7: Test Results – RMSE – Most Untrustworthy Nodes.....	175
Figure 4.8: System accuracy	175
Figure 4.9: Comparison of accuracy and loss of the model.....	179
Figure 4.10: Loss and accuracy of the model when the number of nodes = 16	181

Figure 4.11: Loss and accuracy of the model when the number of nodes = 25	182
Figure 4.12: Loss and accuracy of the model when the number of nodes = 50	182
Figure 4.13: Visualization of the data transactions of the developed network.....	183
Figure 4.14: Node A, C, and D trade facts in between and modernize Q value.....	185
Figure 4.15: Results of trained 100 epochs.....	185
Figure 4.16: final simulation output.....	186
Figure 4.17: Routing table	186
Figure 4.18: Data accuracy with epochs	187
Figure 4.19: Model loss with epochs	187
Figure 4.20: Routing table visualization with updated Q-values.....	189
Figure 4.21: Q-value evaluation of the nodes.....	189
Figure 4.22: Accuracy of the model	190
Figure 4.23: Loss of the model	190
Figure 4.24: Model accuracy and loss for 500 epochs.....	191
Figure 4.25: Model accuracy and loss for 1000 epoch	191
Figure 4.26: Delay	193
Figure 4.27: Packet Delivery Ratio.....	193
Figure 4.28: Average Number of Hops per flow	194
Figure 4.29: QoS Performance under Different Node Densities	196
Figure 4.30: QoS Performance under Different Motilities	197
Figure 4.31: QoS performance Under Different Ricean K Factors	198
Figure 4.32: Packet Delivery Ratio.....	200
Figure 4.33: Average Latency.....	200
Figure 4.34: Routing Packet Overhead.....	201
Figure 4.35: Malicious Node Detection for different Test nodes	202
Figure 4.36: True Positive Rate for Different Malicious Nodes.....	202
Figure 4.37: False Positive Rate for Different Malicious Nodes.....	203
Figure 4.38: Malicious node classification for different test nodes.....	203
Figure 4.39: True Positive Rate for different Collaborative Malicious nodes.....	204
Figure 4.40: False Positive Rate for different Collaborative Malicious nodes.....	205

LIST OF TABLES

Table 2.1: Performance of protocol in scalability scenario	35
Table 2.2: Comparison of simulation results of the scalability scenario	36
Table 3.1: Comparison of diverse trust calculating approaches with respect to numerous attack prototype.....	85
Table 3.2: Trust Table.....	88
Table 3.3: Recommendation Table	89
Table 3.4: Backup Table	89
Table 3.5: Threshold Table	105
Table 4.1: Initial Probabilities of Evidence Variable.....	159
Table 4.2: Probability Tables of Intermediate and Hypothesis Variables	160
Table 4.3: TrV Changes.....	164
Table 4.4: Properties of trained datasets	165
Table 4.5: Threshold Values for Trust.....	167
Table 4.6: Highly Trust / confidential combinations	167
Table 4.7: Medially Trust / confidential combinations.....	168
Table 4.8: Trusted Network Filter Plugin results.....	172
Table 4.9: Most Untrusted Nodes Filter Plugin results.....	173
Table 4.10: Simulation Parameters	177
Table 4.11: Weight Tuning of Reward Function	181
Table 4.12: Q Table	184

LIST OF ABBREVIATIONS

AODV	Ad-hoc On-demand Distance Vector routing protocol
BATMAN	Better Approach To Mobile Ad-hoc Networking routing protocol
CPS	Cyber-physical Systems
D2D	Device to Device
DKF	Distributed Kalman Filter
DOS	Denial of Service
DSR	Dynamic Source Routing
FHSS	Frequency Hopping Spread Spectrum
FSM	Finite State Machine
MAC	Message Authentication Code
MANET	Mobile Ad-hoc Networks
ND	Neighbourhood Discovery
NS3	Network Simulator v3
PKI	Public Key Infrastructure
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
DSDV	Destination-Sequenced Distance-Vector
LAR	Location-Aided Routing
ZRP	Zone Routing Protocol
MAODV	Multicast Ad hoc On-Demand Distance Vector
TTM	Transmission Time based Mechanism

RLTM	Reinforcement Learning Trust Manager
RREQ	Route Request
RPLY	Route Reply
DT	Direct Trust
IDT	Indirect Trust
GT	Global Trust
RLT	Relative Layer Trust
ReLU	Rectified Linear Unit
TrV	Trust Value

Chapter 1 – Introduction

The introductory section contains mainly five sub sections, respectively Background, Motivation, Problem Statement, Thesis Contribution, and Outline of the thesis where it describes the background of the addressed research area with major issues identified related to MANET security and contribution of this thesis to address some of those major issues. In the final section, it details the outline of the whole thesis and a summary of each chapter.

1.1 Background

Network Security has become a subject of critical considerations within the last few years when considering both academia and industry. Since the data network turns out to more distributive and it winds up in scale, network interruption and assault have turned out to be more serious threats to clients of the network, which are particularly valid for evolving wireless data networks. Contrasted with wired networks, wireless networks are inclined to attacks of security extending from passive eavesdropping to active interfering. Since it is significantly difficult to protect systems against the intruders in a wireless domain, intermittent incidents in a substantial-scale mobile network is nearly inevitable over a long period. These risks prevailing in wireless networks are more prominent in the Mobile Ad-hoc Network (MANET) domain [1].

In a MANET, mobile clients reach the network within a certain radio frequency range and establish the network topology for correspondents. Nodes inside the Ad-hoc network convey through wireless links or multi-hop routing without any infrastructure setup [2]. In terms of security, these networks are at stake which ultimately makes the client-vulnerable. Area of consideration of this thesis is the Trust-Based model of Non-Cryptographic MANET Security.

Accomplishing security within an ad-hoc system administration is ambitious because of the reasoning facts stated in [3]. As the first factor of consideration, the wireless network is more predisposed to attacks varied from passive eavesdropping to active interfering [3]. Secondly, the deficit of online Certificate Authority (CA) or Trusted Third Party including obstructions in the positioning of security mechanisms. Additionally, mobile devices incline to have limited battery life and capabilities of calculations makes it more susceptible to Denial of Service attacks and incapable to execute algorithms with large computations such as public key

algorithms [3]. Further, in MANET, there is a possibility for a node of trust to be compromised and being used by an intruder to set up attacks on networks, thus it is essential to be attentive about both insider and outsider attacks of MANETs, where insider attacks are harder to be detected. As the final factor, the mobile nature of nodes in an Ad-hoc network creates greater opportunities for attacks as the prevailing network reconfiguration is constrained, for example, it is hard to differentiate among old or expired routing data and fraud routing data.

The proposed mechanisms are to mitigate the problems that occur in MANET due to malicious or insecure nodes within the network. The mechanisms are listed in the order of importance

- a. Social trust implementation;
- b. Entropy-based Spiral Trust Protocol
- c. Deep reinforcement learning based mechanism to encourage trustworthiness among nodes in a trusted MANET environment.

1.2 Motivation

In the class of several motivational factors, the first factor is that this project is part of a larger project done at Curtin University, Western Australia. The project is to develop a way-finding algorithm for vision impaired using smart devices. A major section of the said research is cover through this research, which is to allow secure message passing between the mobile devices which ultimately allows the way-finding algorithm to gather data and process them securely.

Most of the conventional systems of path-finding systems use large computers such as mainframe computers which seems amateurish for visually impaired people [4]. Most of the current research does not reflect an indoor way-finding system that can be utilized in a practical manner for the visually impaired people.

The research objective is to develop a path-finding system for visually defective individuals that will escort them through a fresh location. The scheme is intended to accomplish low cost, portability, and convenience of use. Figure 1.1 demonstrates the whole arrangement which is involving of a consolidated map generation scheme, two implanted devices: first one for image and audio alteration and the second device for human posture investigation, and a smartphone in order to maintain Human-Computer interaction. The functional block illustration of the scheme is presented in Figure 1.2.

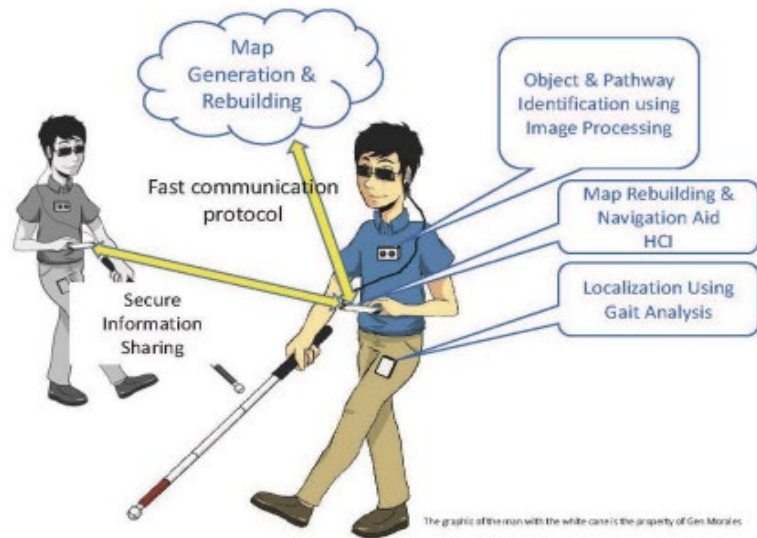


Figure 1.1: High-Level structure of the Indoor Navigation System (Source: Adapted from [4])

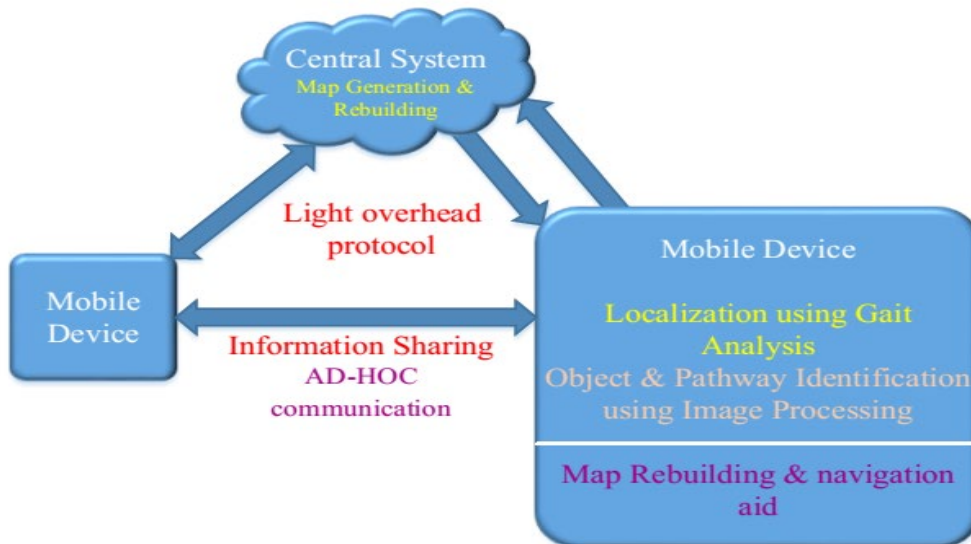


Figure 1.2: diagram showing the functions of the Way-finding system for visually impaired people (Source: Adapted from [4])

As the next motivational factor, this research will provide greater value towards the ever-growing mobile wireless landscape. Namely, most of the IOT based systems use MANET for communication etc. But these IOT based systems and Ad-hoc networks are still unable to provide proper security within the system. The outcome of this research will provide novel implementation architecture which can be used by mobile ad-hoc architectures including IOT systems.

1.3 Problem Statement

Nodes within the MANET convey through wireless links or multi-hop routing without any infrastructure setup. A practical situation like an indoor navigation system for visually impaired individuals is very reliant on MANETs, which is the scope of this research.

Current research work on MANET Security falls in Quality of Service (QoS), secure routing and encryption. Most of the said properties may work in independence but not in a collaborative mode. Further, they are based on several impractical assumptions [5]–[7].

The collaborative routing protocols in MANETs are vulnerable, mainly they are working as relay devices [8]. Generated routing information from these intermediary devices can attack a complete network. An adversary can insert deceitful and unreliable data into routing data or introduce denial of service type attacks by replaying decayed or expired data stored. A captured device by an opponent can transmit malevolent data to neighbour nodes and it may cause a severe impairment to the system. When a node gets vulnerable and compromised, it might result a serious damage to the system. Research work analyzed, provide dynamic management of routing changes, but more susceptible for security issues [9]. With all this reasons implementation of secure routing protocol is regarding be one of the most important research outcomes.

Insufficient attention has been taken on the problem of secure authentication and providing access controls in MANETs. As in Current solutions, the group-based approach has also been taken in CARAVAN [10] and AMOEBA [11]. One group is consisting of a set of nodes, which are moving together.

Most of the research works are towards solving the problem of selfish nodes in a MANET environment [12]. Marti et al, use a watchdog mechanism to detect selfish nodes and path rating

mechanism to avoid them [13]. Their work has managed to increase network throughput, but those mechanisms cannot cope with the selfish nodes effectively as they do not penalize any misbehaving nodes. Proposed research particularly looked at how to solve the issue of selfish nodes. Further, it penalizes malicious and collaborative malicious nodes.

The research work is targeted to review the definitions and measures of trust by focusing on other additional layers which are present within a MANET. First social network layer considered for defining a trust-based scheme. The research goal is to evaluate the trust on social networks and that can be done by means of graph based approach, interaction based approach or a hybrid approach. This work has used the graph-based evaluation model and the parameters are extracted initially by direct and literature-based parameter extraction methods. A TrV (Trust Value) for every social network node is evaluated by using the Bayes' theorem within the Bayesian Belief Network (BBN).

Next problem is solved at the routing layer. Trust framework for MANET was developed, with a routing protocol which named as Entropy Based Spiral Trust AODV (ESTAODV).

Finally, a trust prediction scheme was developed using deep reinforcement learning. The deep reinforcement learning model will anticipate a Q value, which permits the AODV routing protocol to choose the next secure and reliable hop.

1.4 Thesis Contribution

The purpose of this proposed research is to design and develop security algorithms which are efficient in a MANET environment. The proposed framework uses the concept of trust which provides a zero need for authentication and access control. In order to achieve this, the following objectives will be observed.

In this research, three security protocols were established by considering the aforementioned MANET security desires. The first protocol is a decentralized social network analysis framework and a protocol for any socially acceptable, Application Programming Interface (API) enabled web system. Research had used Facebook API for simulation and verification of this protocol within the thesis [14]. Next protocol is a decentralized trust building protocol which works parallel with a reactive MANET routing protocol called AODV. This protocol works with entropy-based trust development framework, which utilizes a newly introduced

“Spiral” model to detect and prevent selfish, malicious and collaborative malicious nodes in a MANET network. Afterward, the contribution is trust prediction scheme develop using deep reinforcement learning.

When considering the problems in security, confirmation and access control are the most complex and imperative issue in MANET, because it has become the objective of the entire security system. The research has been observed different approaches used in literature and we have developed a solution which is effective in handling security in a MANET.

The research project discusses three sections, which are social network trust, ESTAODV protocol, and deep reinforcement learning based trust routing protocol. Below mentioned points are explained about the contribution related to three sections.

1. Social network trust (References - section 3.1)
 - Reviewing the definitions and measures of trust by focusing on social networks to improve security within any kind of network.
 - Evaluation of social network trust is very important since there is an increasing attention on social network security.
2. ESTAODV (Entropy-based Spiral Trust AODV) (References - section 3.2)
 - Developing a trust framework for MANET using Information entropy and “Spiral” Trust model.
3. Deep reinforcement learning based trust routing protocol (References - section 3.3)
 - Implementation is used to set up a TrV using ESTAODV protocol.
 - Simulation of MANET is used to extract the parameters.
 - The development of deep reinforcement learning agent is used to calculate the TrVs.

This research is of significance for addressing the following challenges that have to be addressed in secure indoor navigation in a MANET.

- MANET environments lack infrastructure (E.g.: Access Points, Switches, Servers) because of the mobility needs. The network topology can be reformed quickly and arbitrarily, and it make difficulties the routing task. As for the security implementations in the applied world, they are based on fixed infrastructure. The proposed approach

will be implemented using infrastructure-less environments.

- In general, MANET security research uses cryptographic control. But, cryptographic controls are having many challenges in real time implementations in MANETS. The research follows mainly a non-cryptographic method of implementing trust.
- MANETs are inherently unreliable. Efficient security implementation is needed to make it reliable. The suggested research especially looks at a computationally efficient, trust-based implementation.
- Most of the existing research on trust implementations are focused towards general networks [15], [16]. The developed approach is specifically focused on trust in MANETs. Further, most of the research work studies only the node trust. This research will handle both node and route trusts.

1.5. Outline of the thesis

This thesis is organized into five chapters as follows.

- Chapter 2 (Literature Review) details MANET and existing technology and work on trust implementation of the security, techniques used and technology currently available.
- Chapter 3 (Methodology and Experimental Design) details Three main components of the research and proposed trust model design for each and every component.
- Chapter 4 (Testing and Simulation Results) details the Trust implementation framework using social network analysis and second major contribution of this research, namely, the Entropy-based spiral trust development framework. This model is implemented using reactive AODV MANET protocol. Finally discusses 3rd and final major contribution of using Deep Reinforcement learning as an intelligent trust prediction model using the above mentioned two techniques. The first section explains the monitoring mechanism of how this was used to identify malicious nodes.
- Chapter 5 (Future work and Conclusions) explains the conclusion of the research work and show some future directions for the research.

Chapter 2 - Literature Review

This chapter details MANET, existing technology and work on trust implementation of the security, techniques used and technology currently available. Throughout this chapter discusses Mobile Ad-hoc Networks, MANET routing protocols such as AODV, DSR, BATMAN, TORA and how those protocols works along with routing issues in MANET by means of scalability, reliability, QoS, security. Other than that, it details MANET features and their impact on security since we are mainly focus on the security of MANET and some well-known threats in MANET (Wormhole attack, Black hole attack, Sinkhole attack, Byzantine attack) then social network trust and finally, RL techniques which described as single agent based trust through dynamic programming, TD learning, Monte Carlo learning, deep Q learning. Later part of this chapter details definition of the trust consistent with this context and some major trust challenges.

2.1 Mobile Ad-hoc Networks

As represented in figure 2.1 wireless networks are primarily divided into “infrastructure-based networks” and “infrastructure-free networks” which are known as “Ad-hoc Networks”. A central access point is required by infrastructure mode to connect all the devices. Ad hoc mode basically is a “peer-to-peer” mode, and it does not require a centralized access point [17]. A central access point is required by infrastructure mode to connect all the devices.

Mobile ad-hoc networks entail wireless mobile nodes which move randomly. MANETs can be defined as self-forming, decentralized, self-organizing and infrastructure-less networks as shown in figure 2.2, since the mobile ad-hoc network nodes could join or leave the network arbitrarily, and new associations among nodes can appear or disappear accordingly [17]. For that reason, the network topology can be reformed quickly and arbitrarily, and it make difficulties the routing task. Furthermore, since the wireless medium is naturally unstable, it can be effortlessly congested due to the low bandwidth. In addition, since mobile nodes are operated on small batteries, it may fail at any time due to the limited battery power [17]. Therefore, in MANETs packet forwarding is enabled through paths which capture intermediate nodes which combine starting point and end point nodes.

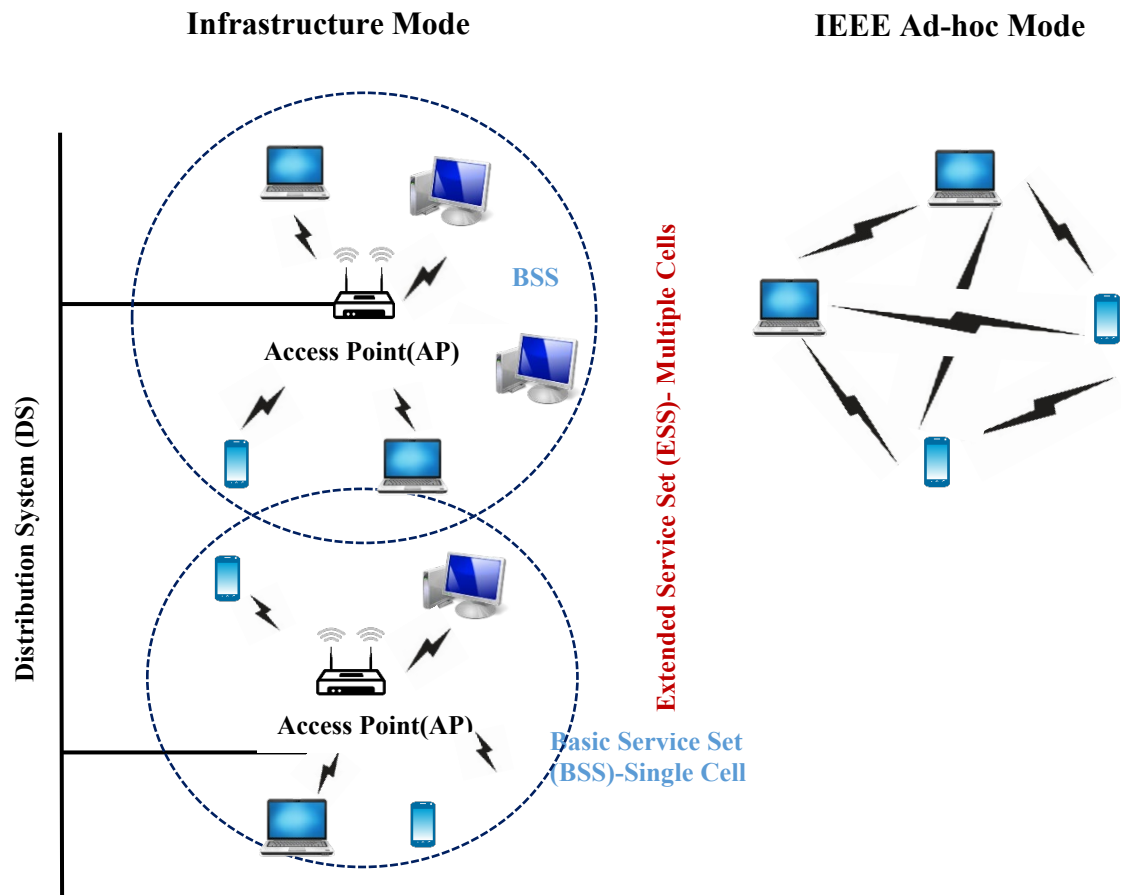


Figure 2.1: Infrastructure mode vs. Ad-hoc mode

The iMANETs are the mobile ad-hoc networks which are used to link mobile nodes in order to establish routes between distributed and fixed internet-gateway nodes. Vehicular Ad-Hoc Networks are a special kind of mobile Ad-Hoc networks where wireless-equipped (road) vehicles form a network with no additional infrastructure. While many communication scenarios exist for these networks, InVANET project focuses only on the application to improve vehicular safety by taking into account the physiological and ecological based context-aware / sensitive parameters as intelligence hence increasing driver convenience.

Basically in routing protocols there are three core forms of Mobile ad-hoc network, which are divided based on characteristics. These core forms can be classified as reactive routing protocols, proactive routing protocols, and hybrid routing protocols as shown in figure 2.3. Routes are normally resolved by using on-demand (Reactive) routing protocols, such as the Dynamic Source Routing (DSR) or the Ad hoc On-Demand Distance Vector Routing (AODV), which produce information about routing only when source node initiates a transmission

[18],[19]. Nodes do not have an important awareness of the topology of the MANET. The nodes need to figure it out dynamically. Hence, nodes in the network will use a routing protocol to share and update information about the other nodes within the network. Many routing protocols are trying to reduce the overhead by adding more control packets to the network.

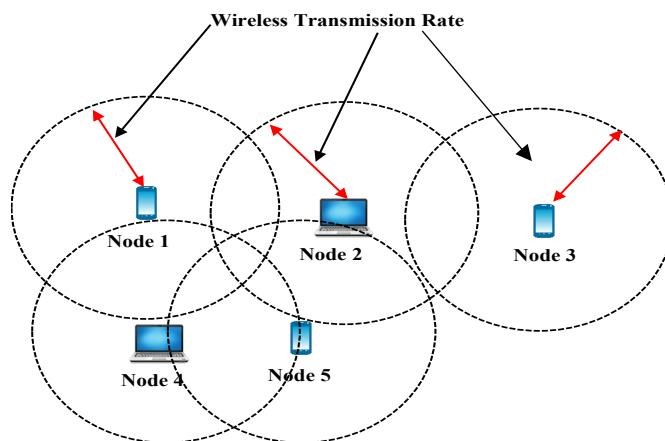


Figure 2.2: Overview of Mobile Ad-hoc Networks

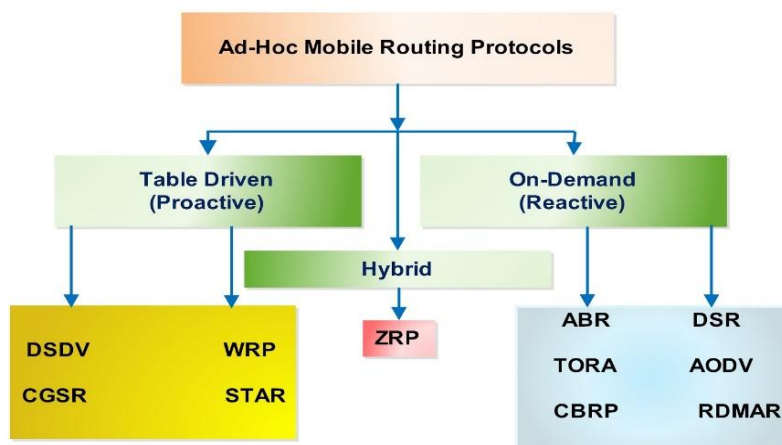


Figure 2.3: Classification of routing protocols in MANETs

2.2 MANET Routing Protocols

A brief description of some ad-hoc network routing schemes is described in this section. In MANET, there are numerous forms of routing procedures that can be applied according to the network periphery. Some of the simple routing procedures are presented next.

2.2.1 Ad-hoc On-demand Distance Vector (AODV) routing protocol

AODV operate as a reactive protocol. Consider a source S having data to transmit to the end point D with unknown direction. AODV works by sending a RouteRequest message to the destination [20]. The RRQ (Route Request message) is relayed by middle nodes until it close to the end point or a node that have a stand direction to the end node. The validity of the route is determined constructed on the end node sequence number requested by the starting node or the source. The particular node with the appropriate path (or the destination node) replies to the RRQ with a RouteReply message. This message is unicast by the nodes along the reverse path established via RRQ propagation.

During the forward propagation of the RouteRequest message, each node raises a reverse pointer to the node that relayed the message first. Any subsequent duplications of the packets received by the node are silently discarded [21]. In a realistic network, based on the congestion, a packet may have to be retransmitted several times prior to successful reception by a destination. Thus, an alternate view may be that the algorithm results in the least congested path to the destination.

In this research, a new entropy-based trust model and that was developed concentrating the AODV reactive routing model. Further, a novel model [“Spiral”] was developed to penalize selfish, malicious and collaborative malicious nodes. This was implemented with NS3 discrete event simulator, and results are explained in chapter 4.

2.2.2 Dynamic Source Routing (DSR)

Dynamic Source Routing is similar to AODV. In fact, since the development of DSR precedes AODV, AODV considered being a combination of DSR and DSDV. Similar to the structure of the AODV protocol, to determine a route to the end point D, a source S transmits a RouteRequest message that is relayed by the intermediate nodes. DSR, however, is a source routing protocol. Unlike AODV, rather than maintaining path pointers to the source, the intermediary nodes attach their ID to the RRQ message. Once the RRQ arrives the end point or a node which consist of a route to the destination, the sequence of nodes embedded in the message is relayed to backward to the source via the RouteReply message [20].

To conserve bandwidth and prevent loops, DSR uses a duplicate packet rejection mechanism like AODV. An intermediate node upon receiving the RouteRequest message checks whether that packet has already been relayed by checking the source sequence number. Duplicate packets are silently discarded by the intermediate nodes. Thus, similar to AODV, the fastest packet to reach an intermediate node or one that travels the least congested route, determines the view of the source. This feature enables application of our scheme.

2.2.3 Better Approach to Mobile Ad-hoc Networking (BATMAN)

Unlike AODV and DSR, BATMAN is a proactive routing protocol. It is intended as a replacement to the OLSR protocol, widely used in mesh and ad-hoc networks. Similar to AODV (and other distance vector protocols), nodes only maintain the directions (next hop) to forward the data, rather than tracking the entire path.

Every single node once in a while transmit a broadcast message (OGM) to notify the neighbouring nodes of its existence. This broadcast message is relayed by the neighbours to other nodes. This practice remains till every single node in the network has received the broadcast message. Every single node records the link over which it acknowledged the OGM from a precise end point. The selection of the forwarding neighbour may depend on several criteria such as the fastest received OGM (to denote the fastest link), or maximum number of OGMs received (to denote the most reliable link) [22]. The selection of the forwarding neighbour may depend on several criteria such as the fastest received OGM (to denote the fastest link), or maximum number of OGMs received (to denote the most reliable link)

The OGM messages are designed to be small (50 bytes) to ensure a low overhead. Similar to AODV and TORA (Temporally Ordered Routing Algorithm), for application of our scheme, the fastest route may be viewed as the least congested route in the network.

2.2.4 TORA

TORA is a link-reversal type algorithm for reactive discovery of routing paths in an ad-hoc setting. Starting with an undirected graph view of the network, it constructs a Directed Acyclic Graph (DAG) routed at the destination. Each node maintains a metric known as the 'height,' relative to each destination. The flow of packets occurs downstream, from 'higher' nodes

towards ‘lower’ nodes. Each node maintains and updates its height based on a set of rules enumerated in [23].

However, unlike the previously considered algorithms, each node maintains heights of all the neighbours, not just the most optimal one. This provides redundancy for routing in case of link failures. For the application of our scheme to TORA, we consider ‘height’ to be a link layer parameter that can be manipulated based on trust. Though this metric is not as intuitive as the ‘congestion’ metric used for the other schemes, it serves as an interesting application of our scheme.

2.3 Routing in MANET

The key objective of this investigation is to distinguish ‘critical parameters’ of the network stack that straightforwardly persuade the choosing of the directions and update them as a function of the TrVs. This research focuses not only link layer parameters but upper layer values which having a relationship towards trust creation. Even though the aim is to secure routing using a trust-based approach, there are multiple challenges to overcome within the network. The following section explains the major concerns to overcome and the next section follows a deeper look into the routing issues.

First, using only one route to send all the packets through will lead to congestion along the way. This can be addressed by considering the neighbouring nodes’ buffer occupancy in the path selection. Second, sending through a node that is moving away, and whose signal has low received signal strength at the source node, will lead to packet loss and further delay as the communication channel conditions degrade further with time.

Third, a path with fewer hops has larger inter-node distance, which can result in worse signal quality and greater latency. Anticipating battery power loss and switching to a new sending path will avoid losing the communication than having to perform a new route discovery. Finally, sending to nodes which have high connectivity will maximize the chances of successful transmission, since there will be more alternative routes along the way and consequently potentially less congestion, and more chances of having better neighbours with respect to the parameters mentioned previously. By anticipating the changing environmental conditions and reacting before route failure actually materializes, the better end-to-end delay

can be achieved.

2.4 Routing Issues in MANET

In general, the mobile devices are requiring less configuration of RAM, Hard Disk, Processor and bandwidth abilities. Executing the Routing for these mobile devices has become challenging because of having fewer resources. Many routing protocols have been anticipated in the area of MANET for routing and data transmitting [18].

2.4.1 Scalability

Scalability is a key feature to consider in the process of routing protocol designing since network size can grow and scale numerous placement situations of MANETs. Ultimately mobile routing network can raise from ten to thousand in a very short time period.

Traffic Type	Network Size	Suitability			Description
		Reliability	Energy Sensitive	Delay Sensitive	
TCP	Small	DSR, AODV	DSR	DSR, AODV	In TCP Traffic, the ZRP hybrid protocol performs poor in all metrics. The DSDV shows almost better performance even in high-density scenario due to the updating ability of recent paths
	Medium	DSDV, LAR, AODV	DSDV	LAR	
	Large	DSDV	DSDV	DSDV	
UDP	Small	LAR, AODV, DSR	DSDV, AODV	LAR, AODV	In UDP, the LAR and AODV expose their best almost in all network sizes. The restricted area allocation ability makes the LAR provide better performance
	Medium	LAR, AODV	DSDV	DSDV	
	Large	LAR, AODV	DSDV	DSDV, LAR	

Table 2.1: Performance of protocol in scalability scenario

The above table 2.1 displays the protocols performances, which expose the better performance under diverse dimensions of the network and constraints. In the process of protocol evaluation, different situations such as mobility and scalability are measured. It supports the professionals of the network to resolve a routing protocol which is suitable for a specific application prerequisite by assessing crucial routing protocol performance metrics. Since TCP and UDP traffic creates the major effect on the evaluation on the protocol, routing protocols are simulated using both the TCP and UDP traffic and this way the consideration of diverse network circumstances affords the effective and precise performance evaluation for each protocol. NS2 simulator has been applied broadly to model the various and distinct network simulation atmospheres for the purpose of measuring the efficiency of different protocols [20].

Protocol investigation based on TCP and UDP capably assist in finding routing protocols which are behaving better within MANET, over many significant scenarios including mobility and scalability. Ultimately the exact performance of routing protocols is determined by the simulation outcomes which are identical to the realistic environment. The comparison of performances of protocols in diverse metrics is demonstrated by the below table 2.2.

Network Scenario	Metrics	Traffic Type	AODV	DSR	LAR	DSDV	OLSR	FSR	ZRP
scalability	PDR	TCP	High	High	High	Medium	High	High	Low
	Throughput		High	Medium	High	High	Medium	Medium	Low
	Overhead		Low	Medium	Low	Low	Medium	Medium	High
	Delay		Low	Medium	Low	Medium	Low	Low	High
	PDR	UDP	High	Medium	High	Low	Medium	Medium	Low
	Throughput		High	Medium	High	Medium	Medium	Medium	Low
	Overhead		Low	Medium	Low	Low	Medium	Medium	High
	Delay		Medium	High	Medium	Low	Medium	Medium	High

Table 2.2: Comparison of simulation results of the scalability scenario

2.4.2 Reliability

MANET mobile agents are used to determine the MANET services, where the agents are able to travel all over the network assembling and occasionally dispersion the vigorously altering network infrastructure. However, it is crucial to inspect how reliable the agents of a given environment are trustworthiness concerns of MANET and how they highly affected by its changing nature [24].

2.4.3 Quality of Service (QoS)

To measure the quality of service features like jitter, delay, bandwidth, packet loss possibility is vital to building up a secure MANET routing protocol. Because of the ad-hoc networks connectivity duration, the quality of the connection remains to change and also these constraints on quality are much more difficult to sustain. Moreover, the behaviour of parameters concerned above on diverse routing protocol is unique. Mobile ad-hoc networks quality of service desire assimilation of vertical-layer or cross-layer. So that the means to discover and troubleshoot the artefacts of parameters listed above require being optimized with the purpose of approve the QoS to end consumers.

Multicasting plays a major role in hold upping applications of group orientation in networks. Multicasting decreases the cost of communication of applications that forward duplicate data packets to many recipients. In MANET environment, nodes can perform random moves, and they will organize in an arbitrary manner themselves. Because of that, group members will to 'leave' or 'join' the multicast term repetitively and resolve this concern it is needed to have an influential multicast transmitting procedure which is very efficient. To overcome this problem, during the period of multicasting need to choose the stable routing path using mobility prediction. In real time circumstances applications like video conferencing, MANET QoS is much more important. In the case of transferring a path from starting node to end point QoS means a set of services need to be addressed by the network. Parameters of the QoS differentiate for several real-world schemes. In the situation of real-time applications QoS, satisfaction is concurrently required. The main goal of MANETs QoS is to satisfy specific application requirements while optimizing the resource utilization of the network. The route generation must be done with consumption of bandwidth and minimum overhead [25].

In the period of Quality of Service (QoS) parameters, for example, effective transmission capacity, usage, least delay, least packet deficit, higher throughput can be utilized for the ongoing and multimedia applications backing by the MANET. Providing help for the quality of service was an issue because of insufficiency of unified foundation based framework, confined data transmission accessibility and constancy developments of nodes, dispute of channel gets to and the inconceivably powerful topology in the remote system. Quality of service requirements like low packet loss, higher throughput, and less delay is extremely basic in MANETs plan and improvement. To offer a stable MANET set up that holds fast to certain QoS parameters, it is required to ensure that an ideal route is set up between source and destination, yet because of dynamic nature of MANETs, the directing troublesome is significantly more problematical when contrasted with wired networks. In the meantime, the characteristic idea of MANETs is described by regular link damages and node faults, and it is approaching to have a framework upheld by productive directing which brings about enhanced QoS parameters [26]

Researchers in proposing a MANET Q-routing protocol seeing transmission capacity efficiency (bandwidth), link security, and power measurements [27]. The enhanced version of route deciding strategy prompts a massive change in QoS. It applies this learning calculation in the current MAODV (Multicast Ad hoc On-demand Vector) procedure which will furthermore expand the QoS of MAODV. The outcomes of model evaluations that the suggested framework accomplishes superior to the modern framework as far as enhancing QoS. It viable routes information packets to assemble individuals even if there should arise a rate of high mobility and successive association disappointments. The 'q-learning' calculation is developed for the path detection and path maintenance practice [25].

Direction-finding way fulfils diverse quality of service restrictions as per the QoS request. They exhibit a versatile routing protocol for MANETs in view of Q-learning out how to improve QoS conveyed to applications.

Nodes in the routing algorithm retrieve data from the environment and adapt to take actions based on the retrieved information. These actions are picked according to the feedback in Acknowledgements. It introduces an RL mapping onto the routing model to study a routing strategy that keeps best QoS.

This routing way fulfils numerous QoS limitations as indicated by the QoS request. They introduce a versatile routing protocol for MANETs in view of Q-learning out how to improve QoS conveyed to applications. Nodes in Routing algorithm fetch calculation recover information from the status and adjust to take activities in view of recovered data. These activities are picked by the criticism in Acknowledgments. It presents an RL mapping onto the r model to examine a directing methodology that keeps best QoS [27].

2.4.4 Security

Mobile ad-hoc networks encounter a radio situation that is not devoted, hence it is not secure to the network steadiness. Strong efforts for node-to-node/end-to-end security arrangement should be explored. Security of MANET is one of the significant facts as to help a sheltered and solid correspondence which means a safe communication among conveying nodes in an unfriendly environment. Because of this sensitive infrastructure, MANET can be specifically assaulted by third party people. By disregarding network privacy, eavesdroppers can access sensitive data of the other nodes in the network [28].

The inhibiting components are basically carried out by secure ad hoc protocols that prevent the vindictive from embedding the wrong circumstance at various hops. These routing protocols rely upon DSR ('Dynamic Source Routing'), AODV ('Ad hoc On-request Distance Vector'), DSDV('Destination-Sequenced Distance Vector'), and use miscellaneous cryptographic primitives (e.g., hash chains, digital signatures etc.) to approve the routing direction correspondence. The proceeding with attacks is detected by the detection component, by distinguishing the abnormal conduct of vindictive nodes. Such sort of misconduct is followed either in an end-to-end way or by adjacent hops by catching the medium lastly coming to a common agreement. Once a misbehaving node is discovered, the response node makes sequences of action in sending and routing procedure that incorporates avoidance of node or node in route defining in order to avoid the misbehaving hop altogether from particular network [16], [26].

MANETs' security is problematic to achieve, rather on account of the vulnerability of remote connections, the restricted physical assurance of nodes, the progressively evolving topology, the nonappearance of an affirmation expert, and the absence of an incorporated observing or

administration point [28]. Prior examinations on ‘mobile ad hoc networks’ (MANETs) have essentially considered on advising conventions for some basic issues, for example, detecting the routing paths, and endeavoured to adapt to the difficulties forced by the new conditions of new environments. They are subsequently powerless against attacks and misbehaviour [29].

Security requirements

Availability is a critical quality of system security, and it guarantees that the accessibility of wanted system administrations offered by the nodes, to its clients at whatever point they are normal, disregarding the nearness of attacks. Frameworks that guarantee accessibility in MANETs look to battle denial of administration and vitality starvation attacks, and also misbehaviour of nodes, for example, node selfishness of packet forwarding [26].

Integrity ensures the recognizing confirmation of packet when it is transmitted, or it signifies the genuineness of information sent starting with one node then onto the next and guarantees that packets are not altered amid transmission. That is, it guarantees that a data packet sent from A to B node was not fabricated by whichever malevolent node C amid its communication. On the off chance that a vigorous privacy component is utilized, guaranteeing information uprightness might be as straightforward as including one-way hashes before scrambling messages [26], [30].

Authentication ensures that the imparting nodes and the data source are genuine and approved which means that it guarantees that a malicious node can't take on the appearance of a trusted node of the network. An assailant can increase unlawful access to confidential data and assets and most likely meddle with the operation of different gatherings. Approval is typically used to enable consents to various individuals [26].

Confidentiality implies that some genuine messages are just agreeable to those hops that have been permitted to get to it and a given message cannot be comprehended by anybody other than its (their) coveted recipient(s). This ensures the assurance of confidential information and data. Information secrecy is normally empowered by applying symmetric or asymmetric information encryption. In delicate conditions, for example, military condition, the introduction of confidential data can have damaging results.

Non-Repudiation portrays the way that if a node in MANET communicates something specific, then it cannot decline the accomplished action. Non-denial is the capacity to guarantee that single client cannot preclude the directing from securing a data packet that it in progress, in computer networks. To guarantee this Digital signature may utilize. For the purpose of discovering selfish nodes, this is very useful. For instance, if a single client receives an erroneous facts from the receiving end, it can utilize that erroneous facts as an indication to report diverse members that a precise one in the network has been compromised [26].

Resilience to Attacks is the flexibility to attacks where it can figure out how to keep up the network functionality when a specific range of the network is destroyed by being compromised. Anonymity keeps the confidentiality of the data and helps to maintain the privacy.

The scheme of rate adaption can be partitioned into two gatherings in view of their help of contrast of loss. The plans without contrast of the loss change their information rate as indicated by loss from the frame or strength of the signal. For instance, it presents a plan which preferences a higher transmittal degree after various useful transmittals at a provided frequency and changes back to a poorer transmission frequency once a few continuous disappointments encountered in [16]. The Adaptive ARF (AARF) plot expert represented a comparable thought however progressively picks the threshold value for rate exchanging. An initial rate adaptation scheme, which reconciles the transmittal rate by considering the channel acquired through degree control messages is presented by Receiver Based Auto Rate (RBAR) scheme.

2.5 MANET features and their impact on security

MANETs have achieved more accessibility than traditional networks because of the features such as infrastructure less, a remote connection utilizing, multi-hop, node movement autonomy, power limitation, etc.

Infrastructure less: In this feature central servers, specific equipment, and fixed routers are essentially missing. The absence of such foundation blocks the formation of incorporated host connections. Rather, nodes maintain host connections, that is, any security arrangement ought to depend on a distributed agreeable scenario rather than a centralized scenario.

Multi-hop: In the case of absence of gateways and central routers, hosts become themselves

routers. Consequently, bundles of information packets take after multi-hop paths and go through various portable nodes previously landing at their ending destination. Because of the conceivable trustworthiness of such nodes, this component introduces a genuine vulnerability [29].

Threats that can influence ad hoc network security and it can be separated into two classes as, attacks and misbehaviour [27].

A passively attacking node may act selfishly to get the transmitted data. They are hard to distinguish as they do not aggravate the typicality of the network. Encryption is regularly used to combat against this kind of attacks [26].

It mentions that a node, which assaults unreceptively may act as selfish nodes to acquire the conveyed information. Passive assailants are tough to distinguish as they do not aggravate the comportment of the network. To battle against passive attackers' encryption is used normally.

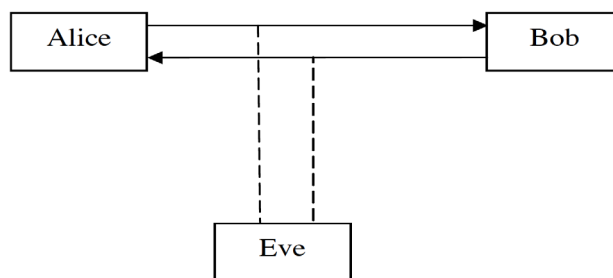


Figure 2.4: Passive attack

Active attacks: Incorporates every other attack propelled by currently connecting with victims, for example, lack of sleep torment, which focuses on the batteries; hijacking, in which the attackers take control of a correspondence between two devices and takes on the appearance of one of them; jamming, which causes channel inaccessibility by abusing it, attacks against routing protocols that we will find in the following area, and so on. A large portion of these assaults consequence in a denial of service (DoS), which is a corruption or an entire stop in correspondence in the middle of end users. Active attacks make jumps in the communication stream among nodes. Attackers infuse the erroneous information to the network. These assaults can happen at any protocol layer such as network, transport, application etc. They can

categorize into two different varieties ‘internal’ and ‘external’. External assaults are accomplished through illegitimate authority. Internal attacks are becoming more common and more damaging which are accomplished by narrow, selfish nodes. These types of attacks make unwarranted get to assemble that facilitate the network to make a confident alteration in the network. Essentially, in active outbreaks, there are four major classes [26].

Modification Attacks: In a message **modification attack**, a stalker modifies packet header addresses to direct a message to a diverse endpoint or **alter** the information on an objective machine. These assaults aggravate the common correspondence among nodes by fine-tuning the information packets. Compromised nodes advertise itself in such a way that it gives a concise and minimal path to the unambiguous receiver. Thusly, malicious nodes at that point determine routing data and utilize it for more attacks.

Dropping Attacks: All the nodes in the MANET atmosphere should transmit packets towards the endpoint node. Selfish nodes do not transmit bundles of data packets to any node in the network; rather dispose of that information to irritate the procedure of the network in this kind of attacks and if the dropping hop is at the critical edge, endwise correspondence among nodes stays away from selfish nodes.

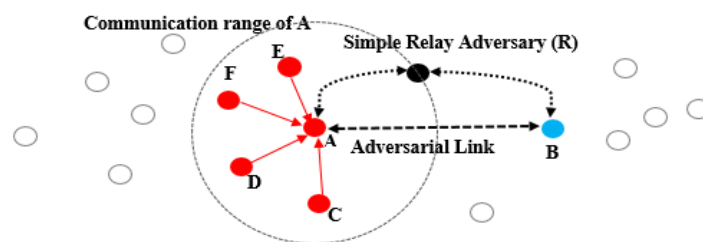
Timing Attacks: In this form of assault, it permits an attacker to determine vulnerabilities in the security of a network by reviewing how long it takes the system to answer back to dissimilar inputs and intruders advance itself such that it is closer to the last objective node, having an ultimate route, to attract in divergent nodes. This category is used by two forms of assaults namely ‘hello flood attacks’ and ‘rushing attacks’.

Fabrication Attacks: In this form of assault a fabricated message is inserted into the network by an illegal manipulator as if it is a legal user and without getting any related to message the misbehaving node transmit propagated data to its acquaintance nodes. Because of related legitimate path seek for a message, the assailants can equally propagate deceitful bundles of packets [26].

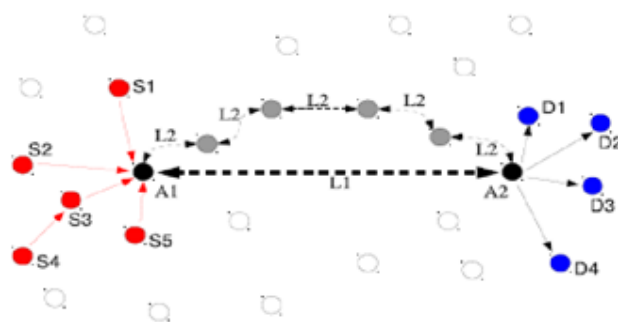
Wormhole attack

Wormhole attack is a challenging security threat to wireless sensor networks which results in

disrupting most of the routing protocols as this attack can be triggered in different modes. In this sort of assaults, the misbehaving node acquires information headed for one sector of the network and pushes it in the direction of one more misbehaving node. The wormhole is indicated as a route that occurs between two misbehaving nodes. This is more destructive to the network. Wormholes are convey opponents with the objective of representation scheme movement using low latency (or cost) directions. Wormholes are utilized by the intruders in the mesh and these nodes are capable to endorse that they have the reliable path through them. At the point when an assailant exploits this attack in routing procedures for instance ‘DSR’ and ‘AODV’, the intrusion attempts to forbid the encounter of any path instead of through these nodes. The existing routing protocols are not really good at discovering legitimate paths since safety tactic is not connected in network routing protocols [28].



(a)



(b)

Figure 2.5: Wormhole situation with (a) Single adversary R making a fake connect among the actual nodes A and B, (b) Collaborating opponents A1 and A2 making a connection amongst their nodes utilizing an out of band channel.

Predictable advanced layer authentication can securely contribute the nature of the communication maker. These qualifications can, nevertheless, be transmitted deprived of take no notice of the cryptographic primitives, execution such tactics insignificant compared to wormholes [28].

A conventional wormhole scenario is shown in Figure 2.5.

Wormholes might be arranged in an unexpected way, in view of the adversarial conduct. The review gives an excellent classification of various wormholes. Here shortly list the applicable highlights and conduct behaviour required for presentation. A wormhole might be covered up or noticeable, contingent upon whether the opposing nodes report themselves to the network. A concealed wormhole shows itself just through its activities. In the event that Figure 2.5 speaks to a concealed wormhole, nodes A_1 and A_2 will be imperceptible to the network. Along these lines, S_1 and D_1 will show up as one-hop neighbours.

Ordinarily, the adversaries making the wormhole are thought to be basic simple relays, equipped for catching the messages however not changing it. It is clear that this way which makes wormhole difficult to be distinguished by cryptographic strategies.

An all the more capable enemy might be one where the node can specifically change the messages before re-broadcasting. Nonetheless, for whatever remains of discussion, we do not concentrate on such sort of adversaries. There are a few upper layer procedures to save the trustworthiness of the transmitted messages, which might be utilized to counter such dangers. Moreover, such enemies would need to buffer bundles of data packets before rolling out any improvements. This would cause noteworthy planning overhead which can be distinguished. A decent investigation of the impact of speed on adversarial conduct is displayed in [28].

Countermeasures for wormhole attacks

Quite a few techniques have been suggested for revealing of ‘wormholes’ such as leashes provides a great survey about the wormhole attacks and the countermeasures of wormhole attack. Most of the scenarios be influenced by on severe timing restrictions or distinctive hardware.

Scheduling centered structures, for example, a bundle of packet leash requires tight

synchronization and specialized hardware. Other planning-based schemes, for example, TTM (transmission time based mechanism) utilize measurements, for example, the ordinary 'round trip time'. Moreover, researchers formally demonstrated the disappointment of scheduling centered structures in contrast to fast opponents. Location- centered structures, which are much protected, have the momentous liability of necessitating functional hardware. Arithmetical and graph theoretic prototypes suggested for wormhole recognition do not experience from special hardware prerequisites. In any case, these methods as shown necessitate fundamental conclusion constructing or have a great computational difficulty.

One common theme in the existing solution has been to circumvent the adversarial behaviour. We deviate from these approaches in the following sense. Firstly, we attempt to characterize the adversarial behaviour as more than a binary constraint of secure vs. insecure. Secondly, we rely on fundamental properties, i.e., physical layer characteristics of the channel, to establish trust. Such an approach makes our scheme agnostic to higher layers of the protocol stack and can be used in conjunction with other schemes proposed previously. Application at a lower layer also makes our scheme significantly more robust and decreases the power overhead.

Black hole attack

This sort of assaults practices the feature of 'AODV' that it recognizes the endpoint sequence number to conclude an restructured route to the endpoint i.e. the route, where it has the uppermost sequence number is nominated. In the black hole intrusion, the unapproved node attempts to intrude on the correspondence among nodes by announcing that it has an ultimate direction to the destination node. Once the node checks out how to position itself among cooperating nodes, with the packets propagating in the network, it can do anything [26].

Sinkhole attack

Here in this form of assaults, the goal is to attraction nearly every association from a precise region by using a negotiated node, creation a sinkhole with the opponent at the midpoint. Meanwhile nodes in the network on, or nearby to, the manner that set of packets proceeds after having numerous chances to mess with application information, a sinkhole can empower numerous different attacks. Sinkhole ordinarily works through construction a negotiated node

look mostly attractive to surrounding neighbours concerning the transmitting procedure. A few protocols may really endeavour to confirm the excellence of route with Endwise acknowledgement is enclosing latency data and consistency. The following figure 2.6 shows a sinkhole attack [31].

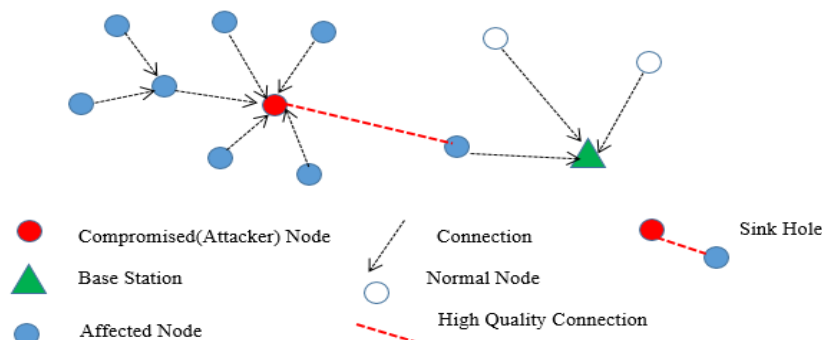


Figure 2.6: Sinkhole attack in the wireless network

Byzantine attack

In Byzantine attack, a traded off intermediate node or an arrangement of intermediate nodes work in plot and completes attacks, for example, making routing loops, sending packets on non-ideal routes and specifically dropping packets which bring about interruption or corruption of the routing services. It is difficult to distinguish byzantine disappointments. The system would appear to work regularly from the perspective of the nodes. However, it might really be indicating Byzantine conduct [32].

2.6 Social Network Trust

Lately, the growth of online social webs for instance 'Facebook', 'Twitter', and 'LinkedIn' have extended the awareness of the informal online social network on their sizes. The public approachability of online social networks utilizing cell phone makes such platforms popular. Node to node communication in a network gives a discussion to their entities to associate with diverse entities in the structures and exchange leisure activities, sentiments, and educational encounters, together with regular capabilities. A significant number of these online social networks are worked with the point of associating however to connect many people [33]. This research is focusing on implementing trust in the device layer. Information within the Social

networks can be used to get additional TrV for the devices. Hence trust can be calculated at the upper layers to be used at the device level.

Thereby, research has developed a social trust framework to allow MANET environment to move cross-layer to find trust-related information which can be used at the device level for decision making. The captured social network behaviour will provide an indication of how trustworthy the same device by capturing upper layer information. This can be an optional parameter to use in the next two frameworks, which were introduced. Usage would provide an upgrade in the trust levels.

A 'social network' is a cluster of individuals or association or additional individuals that associated by a social affiliation together with companionship, facts transaction or collaborative operational. Social network exploration or the analysis is the practice of mapping and determining associations, collaborations and flows among individuals, clusters, associations or additional social objects [33]. As a general rule, social network belief can be designated as a quantity of assurance that an individual or individuals perform in an anticipated way. The research work is reviewing the definitions and measures of trust by focusing on social networks where it can be utilized within further achievements such as improving security within any kind of network.

2.7 Single Agent-Based Trust (Reinforcement Learning Techniques)

Reinforcement learning (RL) is a form of 'machine learning' [34],[35]. It evaluates the performance of a learning agent with respect to a given set of goals. Positive rewards are given for actions that tend to bring about a desirable outcome, the better the higher. Negative rewards can also be used for actions with unwanted consequences. In RL, the mediator or the agent is not on condition that, by way of the accurate responses by a teacher at each step. Instead, each of the rewards indicates how favourable the outcome of an action is. This feedback is provided by the environment itself. Figure 2.7 summarizes this form of learning: the agent performs activities which have an influence on the surroundings, and the surroundings be responsible for the agent with opinion on how associated with the target the outcome is, as well as the new circumstance of the structure. The dilemma of this practice of knowledge can be expressed as this: no range of rewards is provided, therefore even if a reward is deemed satisfactory, other

unexplored actions could lead to even greater rewards. Initially, the agent performs exploration by trying different actions randomly to test their performance. As learning progresses, the agent encounters a conflict between exploration and exploitation. Should it perform an action that has a high payoff based on current knowledge? Or should it instead take an action that enables it to find out more about the environment and potentially discover a better future action course, at the risk of losing rewards or switching to an undesirable state?

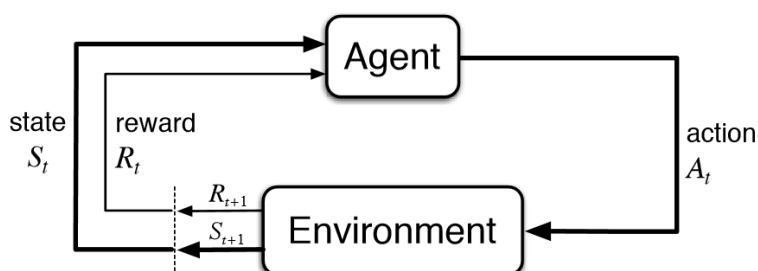


Figure 2.7. Reinforcement Learning

An accepted exploitation of reinforcement learning is in a ‘Markov Decision Process’ (MDP), which be responsible for a scientific structure for launching liable in an arbitrary domain [36]. MDP could be deliberated as a 4-tuple: $(S, A, Pa(s, s'), Ra(s, s'))$, where S and A are the sets of states and actions, $Pa(s, s')$ is the transition probability that state $s \in S$ changes to state $s' \in S$ when action $a \in A$ is accomplished, with an instant reward $Ra(s, s')$. It is not required that $s \neq s'$. The learning agent in the MDP prototype is adopted to discover a most ideal strategy that enhancements the composed reward.

If someone is aware of both the possibilities of transition $Pa(s, s')$ and rewards $Ra(s, s')$, resolving it could be done by the Dynamic Programming (DP) technique [36]. Nevertheless, if it is unaware of, $Pa(s, s')$ or $Ra(s, s')$, a reinforcement learning problem occurs. To predict the best action for a precise state the value of Q is introduced to the reinforcement learning. Hence, the value of Q can be preoccupied as a utility that designs a set of ‘state-action’ combinations as a real number \mathfrak{R} , which can be indicated as

$$Q: S \times A \rightarrow \mathfrak{R} \quad (2.1)$$

The value of Q can revision by deliberating the value of the reward and convinced state corresponding to the fresh action, which can be preserved as the usual estimate of the overall

value for the reward. For a specific state s and action a , value of Q can be appraised as

$$Q(s, a) = \sum P a(s, s') (R a(s, s') + \gamma Z(s')) s' \quad (2.2)$$

Where $Z(s')$ is the awaited reward of state s' , and γ is the discount factor that establishes how much weight is afforded for the forthcoming esteem. As well if we are unaware in the event that the subsequent state s' then the learning must be simply done in view of the state-action pairs (s, a) , which is essentially the Q-learning strategy. In Q learning, iteratively refreshing the value of Q is done from the longstanding values:

$$Q'(s, a) = Q(s, a) + \alpha \cdot (R a(s) + \gamma (\max_a, \max_a (s', a)) - Q(s, a)) \quad (2.3)$$

2.7.1 Dynamic Programming

Dynamic Programming uses the ‘Markov decision process’ (MDP) to describe an ideal strategy in the setting [36].

Hence, although both share the same working principles (either using tabular Reinforcement Learning/Dynamic Programming or approximated RL/DP), the key difference between classic DP and classic RL is that the first assume the model is known. This basically means knowing the transition likelihoods and the anticipated instant reward function.

On the contrary, RL methods only require to have access to a set of samples, either collected online or offline (depending on the algorithm). Hence, there are hybrid methods that can be placed between RL and DP. In this research we used the RL approach over the classic DP due to the unavailability of transition probabilities and immediate rewards.

2.7.2 Temporal Difference Learning (TD Learning)

This model is based on the “temporal credit assignment” problem. Every time this problem is solved iteratively, it results in a better solution through a maximal reward [37].

2.7.3 Monte Carlo Learning

When the environment route replies with a negative reward for an action by selecting a route with a misbehaving neighbour to transfer the data packets to the prearranged endpoint, then the

agent will recognize the set of actions as a negative response. This process is named as bootstrapping. Hence, in such situation, a non-bootstrapping methods used in Monte Carlo mechanism would slow down the learning process of the agent. So in RLTM (Reinforcement Learning Trust Manager) when enhancing the trust in MANETs needed to be highly efficient, therefore Monte Carlo approach based reinforcement learning algorithms are not suitable in our research work [34].

2.7.4 Deep Q Learning

Deep learning is a universally useful structure for portrayal learning, for a given target learn portrayal that is required to accomplish objective straightforwardly from crude sources of info utilizing negligible area information. In actualizing we look for a solitary specialist which can understand any human-level assignment, Reinforcement learning (RL) characterizes the goal, Deep learning (DL) gives the instrument, and the association of these two parts is the general insight. Following Figure 2.8 demonstrates the profound support learning review.

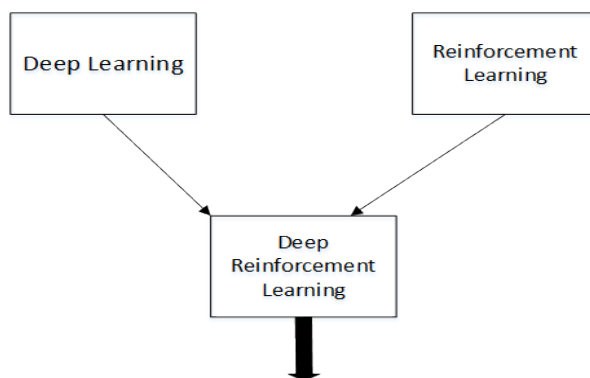


Figure 2.8: Deep reinforcement learning overview

In deep reinforcement learning, there are hidden layers between the input and output layers. By the side of every single layer without input layer, we calculate the impact to every single component, by means of the weighted summation of components from the previous layer; at that point, it naturally consume nonlinear alteration or ‘actuation function’. There are numerous activation functions for instance ‘sigmoid’, ‘tanh’, ‘logistic’ and ‘RELU’, to employ to the influence of a component, to acquire an additional demonstration of the influence from previous layer. Here it has layer wise weights on associates among components. Afterwards estimations stream transmit to output layer, by the side of every single hidden layer and output

layer, can identify mistake subordinates in reverse, and back-propagate inclinations towards the information layer, with the objective that weights can be restructured to enhance certain loss function [35].

DRL can be used in deep networks to represent value function, policy, and model and to optimize these value function, policy and model end-to-end using stochastic gradient descent. DRL also can be used in the field of developing games such as Atari, poker, Go, etc. to explore 3D worlds, to control physical systems like swim, walk, and also to interact with users.

2.8 Trust

2.8.1 Trust in General

Trust is an inherently difficult problem to grasp. There is a lot of ambiguity with this sort of decision-making. Trust is always a subjective process. Once someone needs to find an objective process for managing the trust, it is brought down several interesting types of questions. These questions should be answered in fully to generalize trust. Trust generalization is important for the internet-based/distributed system implementation.

Given a definition of trust,

"Trust is a connection between a trustor, whom we call Alice, and a trustee, who we call Bob. Trust speaks to the trustor's readiness to be defenceless against states of hazard and relationship. Trust is a connection between the trustor and the trustee that can be symmetric or awry. For the most part, the trustor confides in the trustee. The estimation of the trust is either a quantitative or a subjective measure that is utilized to check whether the trust relationship exists or to analyze trust for various decisions [38]."

Current Definitions

Researchers and specialists have thought trust from different perspectives and groundbreaking work has been done in the territory of humanism, psychology, and philosophy [30]–[32], [39], [40]. Trust is characterized as: certainty, solid conviction, in the decency, quality, and dependability of something or some person [41].

2.8.2 Trust Challenges

Obtaining the dynamic data of a structure of a system which has several segments can be done based on estimations about the degree of contribution of every single component for the creation of data and the data transmission rate among each component.

A common set of information has been utilized by most of the authors to evaluate the overlay of the data substance of two (sub) schemes [59]. Unluckily, common data neither contains dynamical nor directional data.

Measuring the information trading between two systems, individually for both directions, and if wanted, restrictive to general input signals can be done based on a minimal set of presumptions about the system dynamics and the coupling nature of them. When it comes to trust the same way, difficult to predict the TrV based on just simple facts like mutual information or communication history. Trust does not only depend on communication history/links, transferred information, how frequently they communicate with each other, a number of communication nodes or communication rate, frequently join or not and specially based on reputation and recommendation.

For example, if we consider two nodes called A and B in the same network, and communication link/direction in-between these nodes can be $A \rightarrow B$ and $B \rightarrow A$. Although we have mutual information related to these nodes we cannot say that the TrV or the trustworthiness of these two are same for both directions and it can be either same value or a different value.

In MANETs, a set of nodes will coordinate for the process of information sending and data packet transmission among source and destination. The routing function will be performed as a compulsory task by each and every node of the MANET. The traffic sent by other nodes should be forwarded by all the nodes. Nodes are allowed to move randomly; where it may occur a situation that the network topology may change arbitrarily and quickly at eccentric circumstances in a system. Thus, the mobility became a major issue in the network, due to dynamically changing topology. According to the dynamic behaviour of the network and the nodes, it should update the TrV time to time.

As the solution for the above-addressed problems, the research proposes the Trust Entropy model. There are some solution models for specific network related problems, which are based

on the Shannon entropy model, where it explains the mathematical theory of communication. Most eminent entropy-based models among all proposed models are Network Transfer Entropy and Information Entropy.

2.8.2.1 Network Transfer Entropy

A quantifier is determined to evaluate the data exchange between two vertices in a weighted network, over a route of a predetermined highest distance. Schreiber presented a mechanism called Transfer Entropy to measure the directed amount of data exchanged between two mutual subordinate time arrangements [43].

Developing a data theoretic measure, network information transfer entropy, evaluating the coordinated quantity of data exchanged between two vertices in a weighted network, with a negligible set of presumptions and general relevance, is the purpose of the approach discussed in [42]. A model which is conceivable to express whether two-time arrangements affected each other, is required by the meaning of information transfer entropy.

2.8.2.2 Information Entropy

A data theoretic measure is inferred that evaluates the factual consistency between frameworks developing over time. The standard time-postponed common realities neglect to recognize data that is really traded from shared data because of regular history and information signals. The time advancement of a framework might be called unpredictable on the off chance that it creates data at a nonzero rate. For stochastic or deterministically noisy frameworks, this is measured by the entropy. For a framework comprising of more than one segment, imperative data on its structure can be acquired by estimating to which degree the individual segments add to data generation and at what rate they trade data among each other.

Let us quickly review the most fundamental ideas of data theory [44]. The normal number of bits expected to ideally encode autonomous draws of the discrete variable i following a likelihood circulation $p(i)$ is specified by the Shannon entropy,

$$H1 = - \sum_i p(i) \log_2 p(i) \quad (2.4)$$

The base of the logarithm decides just the units utilized for estimating data and will be dropped

from this time forward.

The new transfer entropy can distinguish the coordinated trade of data between two frameworks [45]. Dissimilar to shared data, it is intended to disregard static connections because of the normal history or basic information signals. Most obvious applications include multivariate analysis of time arrangement and the investigation of spatially broadened frameworks.

2.8.2.3 Trust Entropy

It is critical to assess whether the neighbour nodes can be trusted or not instead of having centralized authorities, to upgrade the security level in ad-hoc networks.

This is a facts hypothetical framework to measurably estimate trust and determine trust proliferation in MANETs. Trust is considered as a quantifier to measure vulnerability with its esteem demonstrated by entropy. A scientific approach which includes five axioms is created by this research, which addresses the essential comprehension of trust and the guidelines for trust distribution. In view of these things trust model called entropy-based model is proposed by this research, which fulfills all the axioms. For the acquirement, maintenance, and refreshing the trust records related to the nodes' behaviour of packet forwarding and the behaviour of making recommendations about the neighbour nodes. Completed model will be discussed under the proposed trust models section.

Another major issue in an ad hoc network is anytime one node can leave or join the network without acknowledging other nodes in the same network. While transmitting a data packet if this happens it can cause a conflict of the transmission. The same way one node can leave the network, and after some time period, it will try to connect to the same network. Here we have two questions regarding the TrV, 'will it consider as an initial node?' and 'can we take the previous TrV as its new TrV?' Problem is how to calculate the trust for the nodes in those mentioned situations. There should be a mechanism to calculate the TrV which covers all the constraints. In addition, changes in network topology happen, because of versatility or battery limitations, which cause to occur troublesome situation to maintain data for each and every node [45].

In the proposed model nodes cooperate just with their neighbour nodes. Accordingly, nodes do not keep confidential data about each and every node in the network. Noteworthy lower energy

utilization, less preparing for computation of the value of trustworthiness, and less memory space is suggested by neighbourhood data custody. Another outcome is that recommendations are just traded among neighbours, that is, recommendations are not forwarded. Additionally, this mechanism provide a limitation or a restriction with the intention of decreasing the probability of fake recommendations by means of the maturity of the recommending node, and there is not an intermediate node to enhance liability of data. The reduction in the capacity of facts directed not only improves the traffic of the network, nevertheless decreases the degree of consuming power.

The proposed model does not necessitate propagating the data of trustworthiness throughout the whole system. Relatively, every single node in the network it is necessary to preserve and trade confidence evidence regarding the rest of the nodes within the radio range, and through the helpful behaviour of the nodes in the network time to time, it will refresh the direct trust and the recommendation trust as per to the dynamic conduct of the neighbour s. For the most part, a node cannot keep track of the entire behaviour of a neighbour node given, after some time period. Testimonials given by neighbour nodes are helpful in this situation for an exact trust level task.

Basically, the network will divide into small clusters, and each one of the clusters will have an agent who can communicate among the other clusters through the agents of those clusters. Agents of the network have to cooperatively collaborate with each other and enhance the consistency (reliability), security and the network performance by monitoring the dynamic behaviour of the neighbour s and updating the TrV. When it comes to forwarding a packet, there are two things to be done. First one is it has to broadcast using RREQ (Router Request) and discover the route and meantime it has to update the TrV with the recent records. Secondly, after receiving the RPLY (Router Reply) by considering the updated TrVs, it has to select the most secure and reliable path to forward the packet. Protocol specification will contain the communication mechanism in a multi-agent environment.

Giving a trust metric to every node is helpful when nodes act in a malicious way, as well as when nodes transmit data. As per the strategy of autonomic networks, a node is ought to self-configuring, self-managing, and self-learning by methods of gathering local data and trading among neighbour nodes [46]. Hence, it is imperative to just interact with trustworthy

neighbours, since creating interactions with nodes which act in a malicious way, can trade-off the autonomy of ad-hoc networks.

While forwarding a packet, there can be two possible ways of not receiving the packet at the destination end. A drop of the packet can occur due to the intermediate node/nodes selfish behaviour while parsing the packet one node may leave the network or else due to some other network related issue.

According to the trust model specification decided, the research has already defined 5 trust levels after the analysis of literature and the findings.

1. Trustworthy node
2. Partially trustworthy node
3. Selfish node
4. Pure Malicious node
5. Collaborative malicious node

Initially, when a node joins the network we do not have any transmission record or trust record in order to calculate or predict the trust. So initially before the transmission, it will assign with a default value of 0. Time to time-based on the activities that the particular node performs with the neighbouring node it will calculate a TrV and update the initial value by replacing the older value or initial value with the new TrV. Based on TrV, every single node can be categorized according to the five levels mentioned above. It is necessary to categorize all the neighbours according to given levels of the trust, and after identifying the trustworthy and partially trustworthy nodes, we have to carefully identify the selfish and the malicious nodes accordingly. Then the next challenge is to identify pure malicious and cooperative malicious nodes since both of the nodes are behaving more alike and overcome this challenge we can come with the spiral model where we have a recommendation-based trust model. The spiral model will be discussed in '3.2.6 Trust Architecture for MANETS' comprehensively.

Chapter 3 - Methodology and Experimental Design

This chapter details three main components of the research and proposed trust model design for each and every component. First trust calculation model is to ensure the social network trust by using a graph-based evaluation model and parameter extraction process is done using the specific two data analytics tools which are Netvizz and Gephi [49],[50]. The extracted parameters are filtered and categorized using a specific mapping method, which is called as the Bayesian Belief Network (BBN).

Second component contains entropy based spiral trust AODV (ESTAODV) model which calculate the trust by considering both direct and indirect trust and in the direct trust model, research proposed a trust and Q-learning based security model to detect the misbehaving nodes over Ad Hoc On-Demand Distance-Vector (AODV) routing protocol while indirect trust model proposed a recommendation based approach based on five axioms.

Final component is a deep reinforcement based approach and the proposed reinforcement learning approach is consisting of major two tasks, which are, the MANET simulation for parameter extraction and the development of deep reinforcement learning agent in order to calculate the TrVs.

3.1 Social Network Trust

3.1.1 Social networks

A social network is a group of entities, which are connected by a social relationship including friendship, information exchange or corporative working. ‘Social network analysis’ is the procedure of mapping and determining associations, collaborations and streams in the middle of persons, clusters, societies or additional community objects. Trust is a concept with many facets and dimensions. Social network concept is initially introduced by J. A. Barnes in 1954, where they can be described as a set of connected graphs which consist of nodes and edges that represents the entities and their interdependencies respectively [47]. In general, social network trust can be described as a portion of certainty that an object or objects perform in a projected way. The research work is reviewing the definitions and measures of trust by focusing on social networks where it can be utilized within further achievements such as improving security within the target MANET environment.

3.1.2 Social Network Trust and Trust Calculating Methodologies

Over the recent decades, social networks have gained the most popularity and they have become the most frequently visited entities in the internet infrastructure. Although there is an increasing attention to social network security standards and practices, revealing a large amount of information which is linked up with participants. Hence, social networking takes place in a context of trust and security.

The three main aspects of the social trust are

1. Trust information collection
2. Trust evaluation
3. Trust dissemination.

Social trust can be measured considering different disciplines such as psychology, sociology and computer science as well. Hence, when defining social trust in the environment of social network belief there are modern mechanisms concentrate on the above stated three aspects.

Trust evaluation can be Graph-based, Interaction based or hybrid.

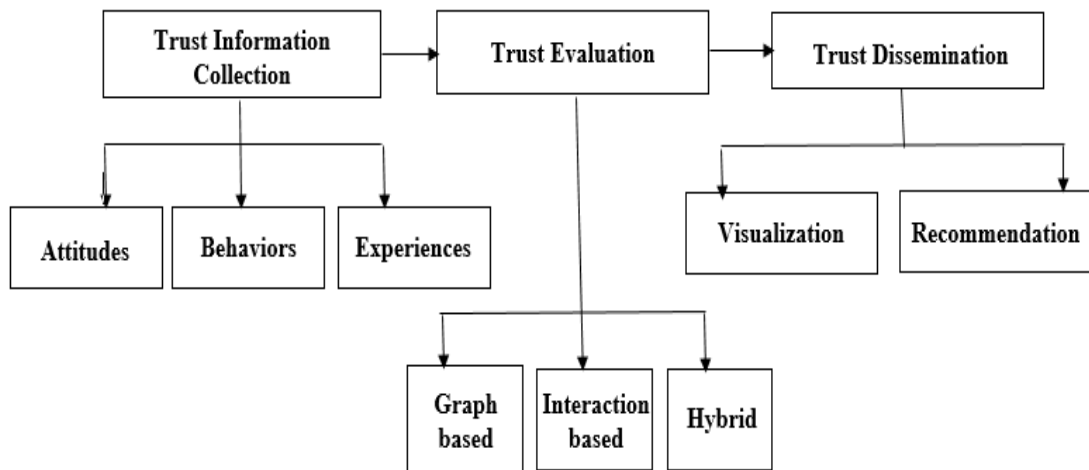


Figure 3.1: Building Trust within a Social Network

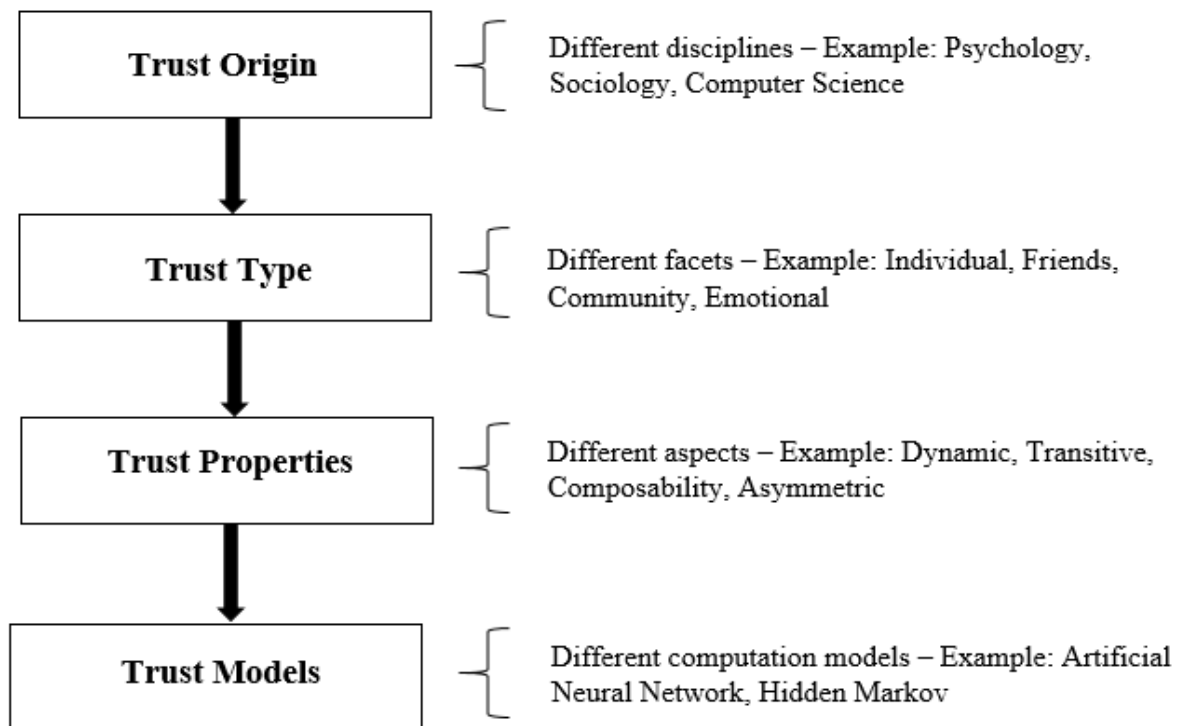


Figure 3.2: Trust Definitions and Measurements for Social Networks

As it is depicted in Figure 3.1 and 3.2, recent studies had given various definitions for trust within and beyond the social networks. According to the above categories in the Figure 3.2 trust can take different shapes. Trust origin comes from different disciplines such as psychology, sociology and computer science. Trust types can be different faces and Trust properties can be from different aspects. Finally, Trust models can be of different computational models [48]. This research defined machine learning based trust computational model, handling dynamic asymmetric trust in a collaborative environment.

3.1.3 Proposed Trust Calculation Model

For calculating the social network trust, a graph-based evaluation model is used. Initially, the related parameters are extracted based on two criteria which are direct parameter extraction and literature-based parameter extraction. Parameter extraction process is done using the specific two data analytics tools which are Netvizz and Gephi [49],[50]. The extracted parameters are filtered and categorized using a specific mapping method, which is called as the Bayesian Belief Network (BBN). Bayesian Belief Network which is also called as probabilistic

directed acyclic graphical prototype denotes a variable clusters and their interdependencies with a focused acyclic graph.

As the result of the process of extracting the related parameters, BBN is developed. When constructing the trust-based system the degree of trust level on various nodes have been mainly considered. Given a network N , the goal is to declare a TrV for any given node u within the network. Given any information regarding a specific node, BBN is making aware of what conclusions can be taken legitimately regarding a certain problem based on the optimal mathematical and statistical knowledge to date, meanwhile updating the hypothesis variables. Ultimately a TrV for every social network node is evaluated by utilizing the Bayes' theorem within BBN.

3.1.4 Social Network Trust

This section of the chapter applies design and validation principles of trust management for managing social network nodes. The research demonstrates the effectiveness of the composite social and routing trust management protocol for MANETs will be an effective solution. Research elaborate an innovative prototype-based tactic to categorize the greatest procedure situation under which P2P (peer-to-peer) particular trust as a consequence of accomplishing social trust management practice. Furthermore, the outcome of this social network model can be used for cross-layer implementation of a routing algorithm. In this research, it is not mandatory to use the social network TrV in the routing trust creation process. But it demonstrates and provides a methodology to calculate social network trust which can be used in routing implementation. This model uses Bayesian Belief Networks for implementing the trust in the social network.

3.1.5 Bayesian Belief Networks (BBNs)

BBN, which is referred as Bayesian Belief Network is a graphical interpretation of a probabilistic dependency model where it also called as a probabilistic directed acyclic graphical model. A set of stochastic variables and their conditional contingencies can be demonstrated by utilizing a Directed Acyclic Graph (DAG). A BBN comprises of arcs and nodes, aggregated with a related set of probability tables. Variables are represented by the nodes whereas arcs

symbolized by essential significant interactions between the nodes. Each variable or node contains a number of conceivable states and corresponding values [14]. A comprehensive probabilistic prototype of associations and variable conditions in a specific realm can be denoted obviously by a BBN.

Hence it is containing the data entirely, essential to reply whichever probabilistic investigation on the subject of whichever variable connections in the specified realm. Evidence may be applied to the network through evidence nodes. Once an evidence is applied, the belief in the state of the evidence changes and hence the belief of nodes in the entire network get changed. Given any information having about a specific node, the BBN enlightening what can be legitimately concluded for a certain problem, in view of a best mathematical and statistical knowledge to date, while updating the hypothesis variables [14]. As well as, the BBN is changeable and almost certainly theories, and their likelihoods, are undoubtedly going to be reformed. By utilizing Bayesian calculus, it determines the state probabilities of each variable from the predetermined conditional and prior probabilities.

Bayes' Theorem

Bayes' Theorem, which is also called as the Inverse Probability Law, distributed after the death in the eighteenth century by Reverend Thomas Bayes, says that one can utilize conditional probability to make predictions in reverse.

$$P(A|B) = \frac{(P(B|A)P(A))}{P(B)} \quad (3.1)$$

P (A|B): Posterior Probability

P (B|A): Likelihood

P (A): Prior

P (B): Evidence

Bayes' Theorem can be used for the purpose of finding the conditional probability of event A, where the conditional probability of event B and the unconditional probabilities of events A and B is given [51].

Advantages of BBN

Modelling and reasoning about vulnerability or the uncertainty are necessarily imposed by BBN. Furthermore, a total probabilistic model of the whole set of variables in a specific domain is represented by this, while affirming to response whichever probabilistic investigation regarding whichever variable in that realm. BBNs are a method for portraying complex probabilistic reasoning, and it can be used within forwarding inference and as well as with the backward inference [52]. A number of various advantages can be accomplished by applying BBNs to Social Network Analysis. Extra capacities are accommodated by finding new connections and recognizing specific nodes in the network [53].

3.1.6 Traditional Social Network Analysis

For the purpose of computing properties of nodes in a particular network, graph-theoretic algorithms are utilized within the traditional Social Network Analysis (SNA). But SNAs are suffering from a problem of assumptions an amount of completeness of social network facts, which means that a well-formed social network is deemed by SNA. And also, it assumes credibility of the social network information which cannot be guaranteed always. Completeness of the resulting social network and the fact that it contains the essential data may not ensure by Real-world methods of data collection. Further, traditional SNA primarily focusses on the occurrence of an association among every single neighbouring nodes in the network, on the other hand not on attributes of that association or the neighbouring nodes in the association. In addition, the vulnerability or the indecision of properties about nodes or relationships is not expressly considered by SNA. Hence traditional SNA relies upon social networks which are made with some degree of conviction, though it is not practical in real-world situations [53].

The foremost ambition of this exploration task is to construct a trust layer on top of a social environment, to achieve the advantages of trustworthy connections. A network structure has been developed to accomplish that achievement. Preceding that, information of Facebook networks (Anonymized dataset) has been extracted and analyzed. Important parameters were extracted from the literature study. These parameters were used to calculate security based TrV within the social network. While examining the separated information from social networks,

legitimate security-related parameters were selected with their conceivable states and values. Eventually, the Bayesian Belief Network (BBN) has been developed, where it calculates the TrV of a given social network node. Since BBN is a tool for improving social network analysis, it can compute highly accurate hypothesis for the TrV if it is correctly given with evidence for its parameters. Thereafter, two threshold values of trust have been declared for highly trusted data and medially trusted data. At last, the trust-based mechanism was developed where it can filter both highly and medially trusted nodes for a given social network. Figure 3.3 and Figure 3.4 gives the representation of the developed system structure and the flowchart of the process respectively.

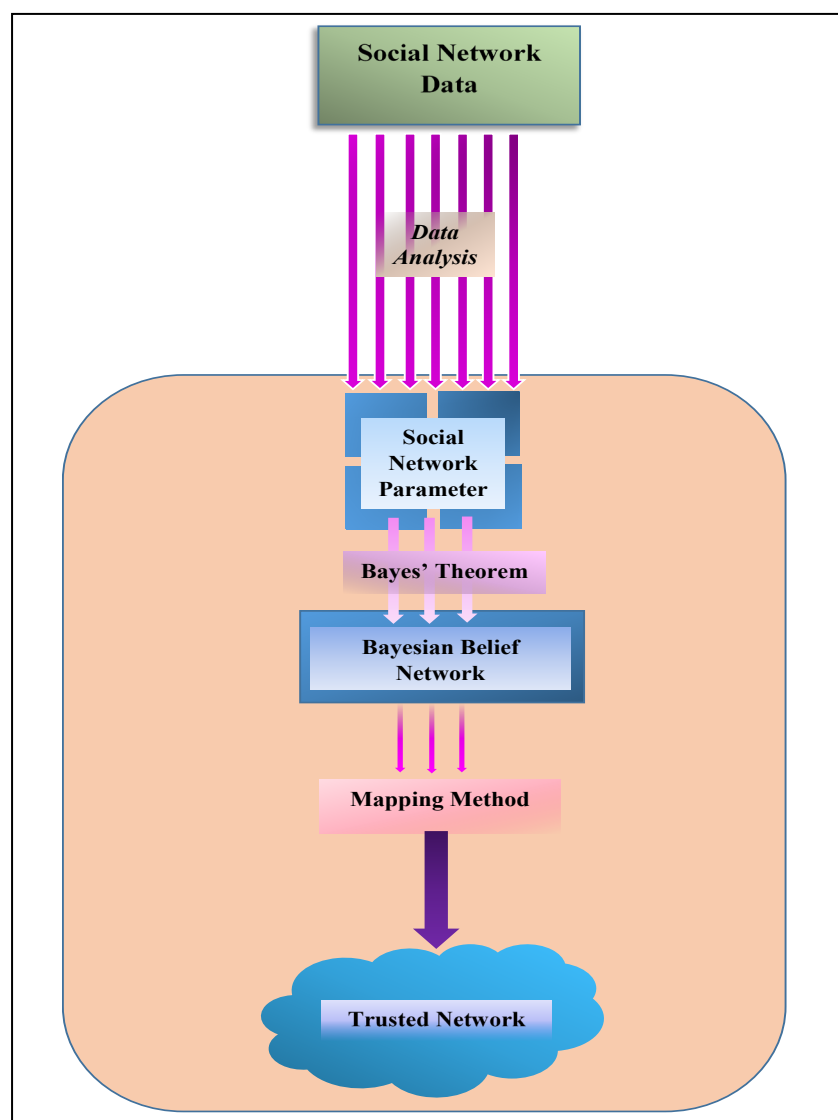


Figure 3.3: Representation of the Developed System Architecture

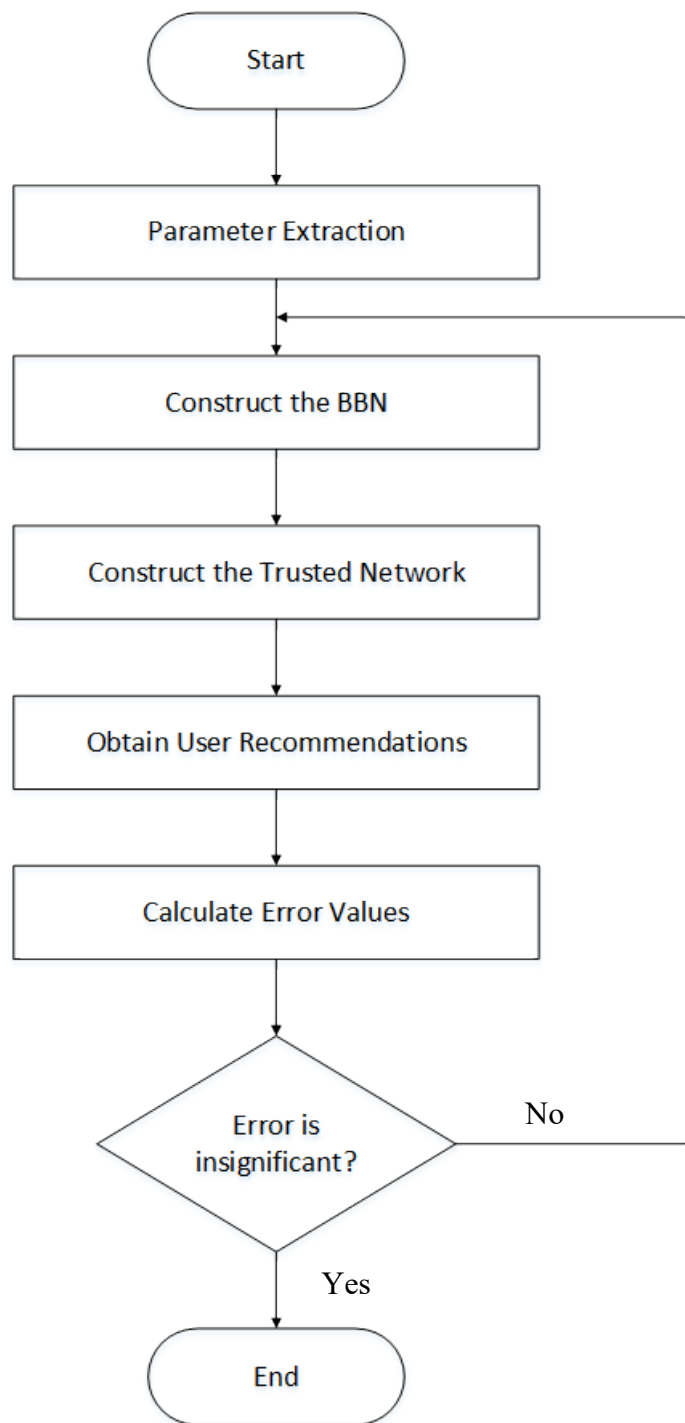


Figure 3.4: Flow Chart of the Trust Calculation Process

Major value addition is to pick up the nodes which can be considered as a trustworthy node in a social network in order to pass confidential data and provide the TrV to the underline MANET to process trustworthy routing. Prior to that, it is detected whether a social node is trustworthy or not, according to the introduced trust-based mechanism.

System implementation was done using an iterative procedure, where it manages to minimize the error of the system values. While considering the closeness of the predicted values for the trust which is estimated by means of the structure centered on the rankings of the trust assumed by the user (anonymized data set) in system evaluation, accuracy of the system has been obtained. It is decided that the developed Bayesian Belief Network has high accuracy if the predicted TrVs are much similar.

If not, doing furthermore modifications to the structure of the BBN is very necessary to acquire the accurate results. In similar manner, for minimizing the number of errors, the novel changes to the BBN have been done number of times and in various circumstances. Further, according to the maximum user satisfaction obtained from the anonymized data set, subsequent to playing out a number of iterations, the edge values of trust have been concluded.

To simulate the concept research will use the popular Facebook social network for this study. Preceding to the implementation, gathering the information was done through an unknown Facebook dataset, by utilizing the Netvizz Facebook Application. Altogether twenty-two (22) anonymized datasets were used on behalf of system implementation and evaluation purposes. Randomly selected two third of datasets have been used for implementation, and the other one third is used in the evaluation and testing. Corresponding detailed information is mentioned in subsequent sections in the thesis.

3.1.7 Parameter Extraction

Social Network Parameters

Using 'Netvizz tool' which abstracts facts from diverse segments of the 'Facebook' manifesto (personal profile, clusters or groups, Facebook pages) for exploration determinations, network data has been extracted. File outputs are being saved in Graph Description Format (GDF) file arrangement and can be effortlessly examined in typical software. Fifteen user (friendship) network datasets were being utilized, which have been downloaded via the Netvizz application,

which creates a network file for a friendship network, with all friendship connections in the particular network. Facebook social network data which gathered from Netvizz has been analyzed with the idea of filtering necessary parameters. The GDF files are imported into Gephi, which is an excellent tool for visualizing and analyzing networks.

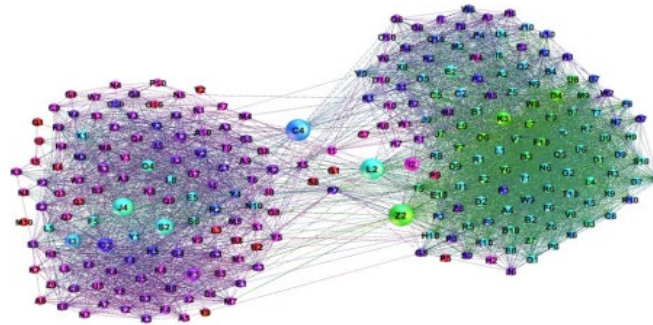


Figure 3.5: Analyzed social network using Gephi

Mohsen and Hassan explain the different aspects of social network analysis including graphical relations to represent the social relationships and other important metrics to represent the social relations as well as statistical models for social network analysis. Further, the related social network properties such as degree and centrality and other sub-structures are explained- by them [54]. Hence the following factors have been considered as parameters in the process of calculating the social network trust.

By considering the above literature-based knowledge, the filtered parameters with the use of Gephi tool are listed below.

Parameters Extracted using Gephi

- Strongly connected components
- Degree
 - In-degree
 - Out-degree
- Centrality
 - Betweenness centrality
 - Closeness centrality
 - Eigenvector centrality

- Eccentricity
- Locale
- Gender
- Authority
- Node

3.1.8 Literature Based Parameters Extracted

Common Friends

Mutual associates of 'A' and 'B' to be all the associates of the neighborhood that are associates of in cooperation 'A' and 'B.'

Shortest Path

It models the distance of the direct route in the middle of two nodes inside an association [55].

Accept unknown friend requests

If an unknown friend request is accepted, it is being granted that stranger access to a wealth of personal information. The account security is compromised if the unknown person is a hacker or a malicious computer program [56]. The percentage of accepting friend requests from unknown people is around 55 [55].

Trusted Contacts

A parameter introduced by Facebook to give its users more control over their account security. Trusted contacts are friends that can securely help the user get back into his account. The user can personally choose and manage his trusted contacts (i.e., close friends he really trust to help him and can be called for help to get back into his account) anytime through his or her Security Settings, instead of only when s/he is having trouble accessing to his or her own account. A user can select three to five friends to be his trusted contacts. To select good trusted contacts, on Facebook it says:

- You should select people who are trustworthy to you, such as one of your most trustworthy companion, where you can give the spare key to your house.

-Select people that you can reach preferably using a telephone extension or meet them in person instead of reach through the Facebook because you may get a need to connect with them when you are unable to log in.

-Selecting people should be done considering the fact that who can help you. If you choose more friends when you need help, there will be many people to help you. The more friends you choose, the more people who can help you when you need it.

- Secure Account
this is a feature introduced by Facebook. Using the particular wizard user can reset all his passwords and information, in case he suspects that his account might be compromised [57].
- Posted Photographs
- Private/ Public Profiles

Sites are esteemed private if the accompanying message shows up on the site of intrigue:

“___ just offers certain data with everybody. In the event that you know ___, include him/ her as a companion on Facebook” [58].

- Facebook Fake Pages
Fake pages are a kind of bogus or false pages. There are 8% ‘Facebook Fake Pages’ [59].
- Phishing
- Malware
- Hacked and Compromised Accounts

3.1.9 Implementation of the model

A Bayesian Belief Network is developed resulting in the process of extracting the related parameters. The degree of trust level on various nodes has been considered mainly in developing the trust-based system. Given a network \mathbf{N} , the goal is to declare a TrV for any given node \mathbf{u} within the network. Given any information having about a specific node, the BBN is enlightening what can be legitimately concluded for a certain problem, in view of the best mathematical and statistical knowledge to date, while updating the hypothesis variables. By

utilizing the Bayes' theorem within BBN, a TrV for every social network node is evaluated with regarding that mechanism.

Calculation of the TrV

Relationships between nodes in a Facebook friendship network is signified by means of the edges of the network diagram. To develop the trust network, those edges must be improved with values are revealing the trustworthy connections between individual nodes. This unit defines the practice of the addition of the TrVs within social network users.

Definition of Trust

T_N = TrV generated from neighbour s for node N . It is associated with a direct link with neighbours of node N_1 and Node N_2 . Trust can be described as,

$$T_N \in [0, 1]$$

In the above equation, 1 signifies fully trust and 0 signifies no trust. TrV should be a positive value in the range of 1 and 0. If the TrV is higher, it signifies a higher trust level. Hence,

- If TrV of node N is 1, that signifies that N 's all neighbours reveal 100% trust for N .
- If TrV of node N is 0, then all of his neighbours are totally mistrust N .

For a given network, TrV which generated will be specific for a given user or node. Therefore, TrV computed for an individual in one system is not the same as the TrV ascertained for a similar individual in another system. In the above-mentioned system, it is computed just the TrVs of first-arrange neighbours to be specific direct neighbours, using the BBN. That implies for a specific user, the related TrVs of his direct neighbour are computed by the BBN [14].

In Figure 3.6 it is depicted the distribution of trust relationships in an example network. First order neighbours (direct friends), second-order neighbours, and third-order neighbours of node A are separated by three corresponding circles. First-order neighbours are the immediate friends; second-order neighbours are two hops away; third-order neighbours are three hops away; and so on.

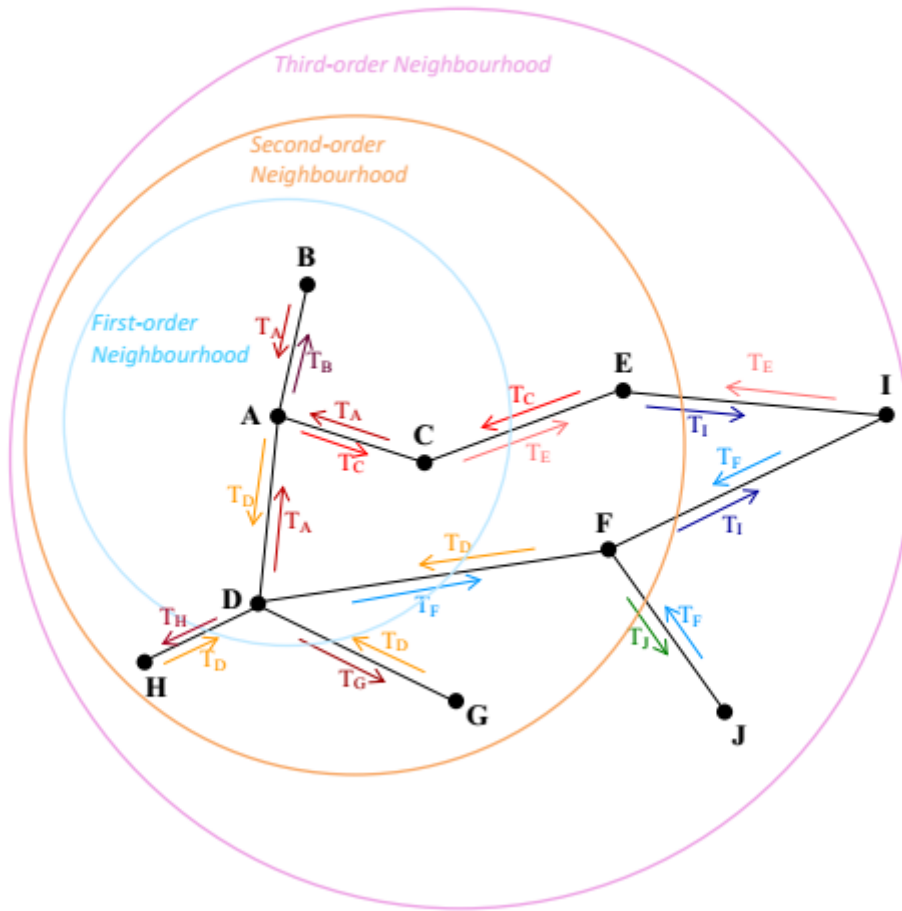


Figure 3.6: Trust relationship between neighbours

3.1.10 Network Structure

The network structure, with a corresponding hypothesis, evidence, and intermediate variables, has been designed towards the implementation of the Bayesian Belief Network. In order to correctly represent the probabilistic relationships among those variables, proper variables for each type have been selected with corresponding relationships.

Hypothesis Variable (Guess Variable)

Initially, the hypothesis variable has been observed. This is the variable for which the probability distribution is being determined while guessing the value of trust. TrV is the verified hypothesis variable, which is defined as having two possible states; Yes and No. This demonstrates the trustworthiness and preciseness of data which can be obtained from

the related person (node). The theoretical definition of the TrV is described in previous section.

Evidence Variables (Information Variables)

Subsequently, evidence variables or the information variables are added to the network structure. These are allowing the information to be entered into the network when events are observed. From the data visualizer, Gephi evidence variables such as Betweenness Centrality, Closeness centrality, Eigenvector Centrality and Degree are observed. Each evidence variable captured above consists of states such as Low, Medium, and High. By means of figure 3.7, selected evidence variables are discussed below.

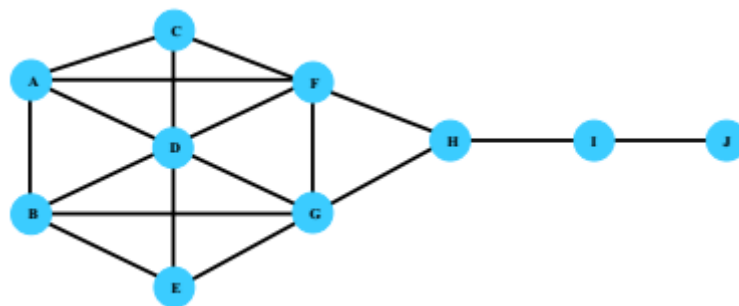


Figure 3.7: Hypothetical graph

Degree

The quantity of edges that are adjacent to the node declares the degree of a node. For a fixed system, it is illustrated as two discrete portions of degree centrality such as ‘in-degree’ and ‘out-degree’. The degree can be utilized to decide the network activity for a node by ascertaining its number of direct associations. The person with the most elevated degree is the most dynamic node inside the network since it turns into a connector or a center point in the specific system. For graph $G: = (V, E)$ with $|V|$ vertices and $|E|$ edges, the degree of vertex v is;

$$C_D(v) = \text{deg}(v) \quad (3.2)$$

A node with a higher degree involves many interactions. In the network represented in Figure 3.7, node D has most direct connections while obtaining the highest degree of value.

Eigenvector Centrality

This calculates the importance of node given, based on connections of a node. Eigenvector centrality can be estimated by evaluating how very much associated an individual is to the parts of the system with the best availability. A node which has higher eigenvector centrality is having an expansive number of associations, and its associations have numerous associations, and their associations have numerous associations until the finish of the system [60].

Closeness Centrality

Closeness centrality is being assessed on geodesic separation, particularly deciding the normal social separation from every node to each and every node in the system. An individual with the base normal most limited way from it to every other individual inside the system includes the best closeness centrality esteem. Consequently, it is having the best visibility on the network whereas it is placed in a better spot to monitor the network traffic flow [60]-[62]. A node with higher closeness centrality is normally lying close to, and may quickly communicate with other individuals in the network. The highest closeness centrality value is achieved by nodes F and G in figure 3.7 [14].

Betweenness Centrality

This is a measure of centrality of a node in the network which is developed by sociologist Linton Freeman. It quotes how frequently a node shows up on most limited ways between nodes in the system. Consequently, betweenness centrality is the quantity of briefest techniques from every single vertices in the graph to all the other vertices that connected with that precise node. This is not only an estimation of association, but then again an estimation of the load and significance of the node. A node with a higher betweenness centrality may control the outcomes in a network because it is located in one of the best spots in the network. Hence it can play an influential role in the specific system while having an excessive effect on what is flowing over there.

Below expression gives the betweenness centrality of node v ;

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \quad (3.3)$$

σ_{st} = total number of direct routes from node s to node t

$\sigma_{st}(v)$ = number of those routes that pass-through node v.

A node with higher betweenness centrality is lying on the huge direct routes count in the middle of other sets of nodes in the network. For instance, in figure 3.7, node H has the highest betweenness centrality as it links with two separate networks and it is between two important constituencies.

Intermediate Variables

In the wake of deciding the four proof factors, three intermediate variables have been presented as Node Popularity, Node Importance, and Influence on the network. Each intermediate variable is having three conceivable states; High, Medium, Low. In view of the fact that these variables are not providing any additional information, they may not be necessary. But because of the reasons that intermediate variables are assisting the network structure too correctly and meaningfully represent the modelled system and reducing the size of the conditional probability tables, those are utilized.

Node Popularity

When an individual is prominent in its system, it might be presumed that he/she is an all-around enjoyed, celebrated character among different users inside the system.

Node Importance

When an individual is critical, or it is a more significant individual for its group, he/she can assume a key part of the system.

Influence on Network

In the event that one can have an immense effect on its group, he/she has impressive power. This will raise his/her effect on the specific system. Connections between the proof and intermediate factors with the explanations behind those connections are examined beneath.

- The degree is a neighbourhood measure of a node. Ordinarily, it is informed that “the more associations, the better,” which is not generally valid. With a high degree, the node turns into a midpoint in organize. Thus it is a prominent node inside its own particular clique or in the entire system, which expands Node Popularity.
- Node significance is measured by the factor which is called as Eigenvector Centrality. This parameter gives the reply for the request, 'How well is this individual related with other all around related people?' A client with high eigenvector centralities are pioneers or the all-inclusive community figures of the framework while making them popular nodes. Hence, eigenvector centrality could cause Node Popularity and Node Importance [14].
- Nodes with better Closeness Centrality are fundamental influences inside their close by composing gathering. Those are not open figures to the entire system, but instead frequently regarded locally. Because of the reason that they are having the best deceivability and get to all the more quickly to any node in the system; it can be extended the Node Significance [14].
- Individuals who are going about as extensions between groups having high Betweenness Centrality esteem. They do not have the briefest normal way to others as closeness centrality. However, the greater part of the briefest way is experiencing them while making them intermediaries crosswise over essential groups since they are in very much situated best areas in the whole system. By methods for the capable part they play in the system, the Node Importance gets expanded. And furthermore, a node with better betweenness centrality has a high Influence in Network since it is a solitary purpose of disappointment [14].

Hence;

- Eigenvector Centrality and Degree is causing Node Popularity
- Eigenvector Centrality, Closeness Centrality, and Betweenness Centrality is causing Node Importance
- Influence on Network can be caused by Betweenness Centrality

Relationships among intermediate variables and the hypothesis variable are mentioned below. In a real-world scenario, if an individual is highly trusted by others in his community, then he becomes a more popular, more important person in his network. And also he can make a high influence on his network since others highly trust him. Conversely, if an individual is very popular, more important, and he can make a huge influence on his network, it can be guessed that he may be a trusted person inside his network. With the use of that technique, TrV has been estimated. Hence given the Node Popularity, Node Importance, and Influence on Network values of a particular node, its TrV may be measured by the constructed BBN [14].

Therefore:

- Node Popularity, Node Importance, and Influence on Network could cause Trust Value. By linking these three types of parameters with directed arrows, the network structure is formed while ensuring that the links are following the direction specified by causality. In BBN, the nodes symbolize hypothetical variables, and the arcs symbolize fundamental associations between the interconnected variables. Above mentioned evidence and intermediate variables have been included in the system with the intention of effectively predict the TrV of a precise node in the social network. By using the review of the literature and the performed social network analysis, the evidence can be gathered [14].

GeNIe 2.0 module which is established at the ‘Decision Systems Laboratory, University of Pittsburgh’ is being utilized to construct the Bayesian Belief Network. It is an adaptable and a comprehensible development setting for ‘graphical decision-theoretic’ prototypes [63]. The constructed network structure is illustrated below.

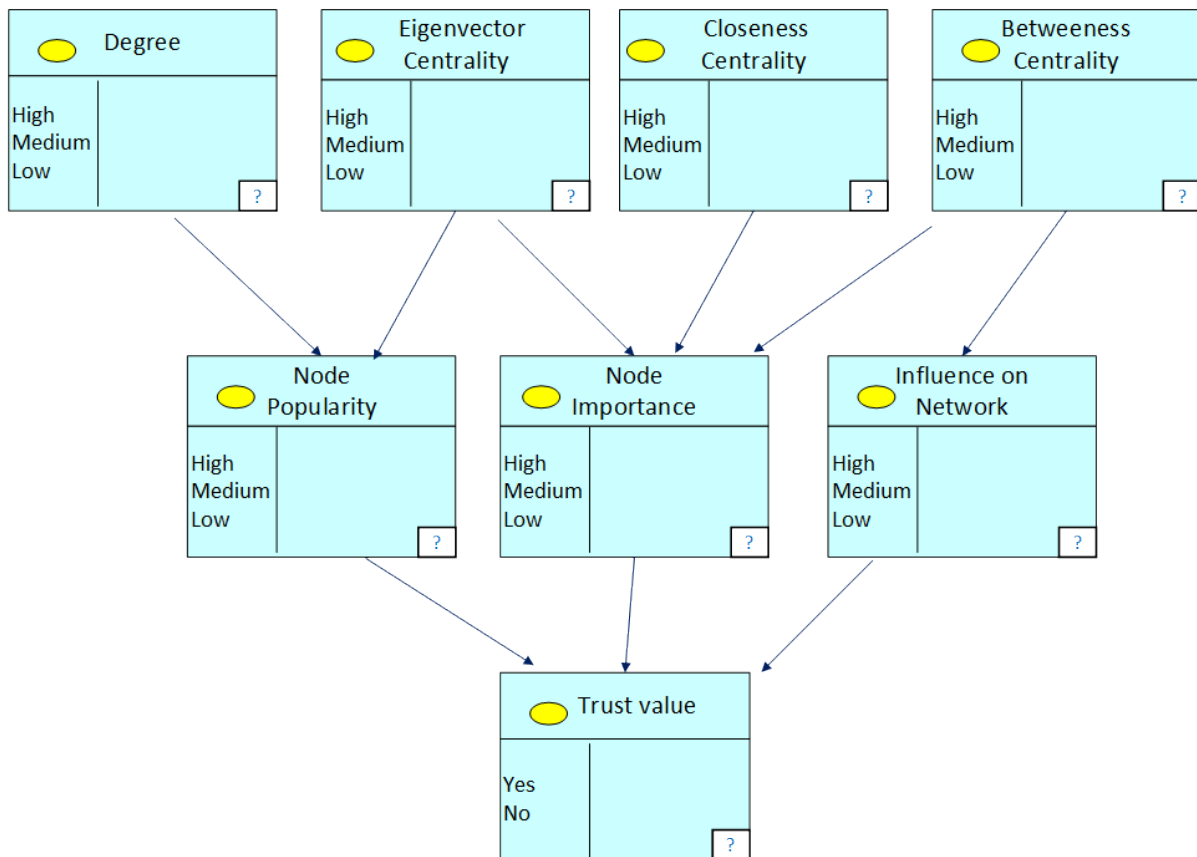


Figure 3.8: Bayesian Belief Network Structure

3.2 Entropy-based Spiral Trust AODV (ESTAODV)

3.2.1 MANET Trust

Generally, a MANET is consisting of many interconnected free and autonomous nodes, which are often composed of mobile devices. Since MANETs are mobile, the network topology is changing rapidly and unpredictably. Because of this nature of the mobility of the nodes in MANETs, the main problems occurred are unreliable communications and weak security where the data can be compromised or misused easily.

With the growth of the ubiquitous computing, the concept of the Internet of Things (IoT) security or Wireless Network security has become an interesting research area throughout the past two decades. With the improvement of wireless communication technologies MANETs setup at an important role, however there are some concerns attributable to the straight influence of mobility of the nodes to the system setting such as unreliability of communication

and weak physical protection. This may encounter an opportunity for an adversary to steal or misuse the data. It gives the idea of such that a reliable connectivity depends basically on the trust of each node. Because of these issues, many research works had been conducted in the recent past to enhance the trust within MANETs. Shalabh Jain describes the confidence in MANET basically as the loyalty of a node to a certain procedure where it can be an facilitator of communication and corporation [48]. When compared to the infrastructure based networks, since having a dynamic topology, error-prone communication media and energy constraining nodes, these mobile networks are more susceptible to malicious occurrences and arbitrary breakdowns [64].

In MANET, the equivalent functions for instance network management, packet forwarding and routing are carried by all available nodes without having a set of dedicated nodes for functioning. Hence one node can be captured by an adversary which may lead to a node misbehaviour or non-cooperated behaviour with all single nodes in the network and aims at damaging other nodes by becoming a malicious node. Therefore, a node in a MANET can be objected to various forms of outbreaks for instance ‘node capture’, ‘eavesdropping’, ‘worm-hole’, ‘Sybil attack’, ‘sink-hole’ and ‘denial of service’ etc. By reason of these occurrences, valuable confidential data can be compromised, misused or, even a total interruption of the system may have effect. Therefore the major challenge in MANETs is the need of trust since reliable connectivity is considered as the backbone of a network.

Therefore, considering the above-explained fact, the corresponding research work proposes a trust routing protocol named ESTAODV for enhancing security in MANET routing.

3.2.2. Proposed ESTAODV model

Entropy-based Spiral Trust AODV (ESTAODV) is the routing protocol, proposed in this research in order to achieve the ultimate goal of creating a trust framework for MANETs based on Reinforcement Learning. Reinforcement learning will be used on top of the ESTAODV model to predict TrVs for given nodes in a MANET. Hence more accurate security decision can be taken within the network. Further, it will remove the need for any specific technology to test authentication and authorization. Because with the trust protocol there will be zero need for authentication.

The entire Trust calculation process can be divided into 4 subparts in the level of implementation.

1. Direct Trust Calculation based on Control Packets and Data Packets
2. Indirect Trust Calculation based on 5 specific Axioms / Entropy Based Indirect Trust
3. Trust Level Identification process
 - Trustworthy Nodes
 - Partially Trustworthy Nodes
 - Selfish Nodes
 - Malicious Nodes
 - Collaborative Malicious Nodes
4. Spiral model for detecting collaborative malicious nodes

To calculate the trust and identify trust levels basically, there are three main processes other than direct trust calculation process. Three phases of this model are trust level identification phase, collaborative malicious nodes discovery process using “spiral model” (explained in 3.2.6) and after transmission phase where model take the actions of the misbehaving nodes.

Implementation of the model uses a cluster-based approach for setting up the trust calculation. According to the ESTAODV protocol, every node calculates trust for each neighbour node. Calculated TrV needed to be fed into the cluster nodes for feeding into the existing RL model (Deep Reinforcement model). Each TrV will be propagated to the central locations and the Q value will be propagated back to each individual node.

All the cluster nodes will be processing RL model and decided the predicted Q values and it will update all the regular nodes in the cluster. The cluster will be defined to reduce the overload generated from the additional control packets within the network.

3.2.3 Trust Model Classification

3.2.3.1 Basic classification

Basically, there are two diverse tactics to appraise trust:

1. Policy-based trust management

2. Reputation-based trust management

Policy-based trust management is positioned considering solid and target security plans, for example, intelligent views and unquestionable properties encoded in marked accreditations to get to control of information assets. Likewise, the model is a rule based on components having a very much characterized trust administration that has confirmation and verification. Such an arrangement based trust administration approach, for the most part, settles on a double choice as per which the requester is trusted or not, and in like manner, the entrance asking for is permitted or not. Because of the paired idea of confiding in assessment, approach based trust administration has less adaptability. Moreover, the accessibility of (or access to) trusted certificate authorities (CA) cannot usually be ensured, especially for conveyed frameworks, for example, MANETs.

Then again, reputation-based trust administration uses computational and numerical modules to assess trust. Frequently, trust is figured by gathering, conglomerating, and spreading reputation among the contributors in those kinds of frameworks. All in all, the denominate '*trust management*' is conversely utilized with the denominate '*reputation management*' [65]. Nonetheless, there is a minor dissimilarity amongst reputation and trust. As described in some approaches, faith is dynamic whereas reputation is non-resistant [66]. To be precise, trust is a node's confidence in the put stock in characteristics of a companion, in this manner being reached out from a precise node to its associate. 'Reputation' is the sensitivity that acquaintances edging about a member in the network. Likewise, the suggestion is every now and again utilized as an approach to gauge confidence or reputation. The reference is just an endeavour by the side of imparting a party's fame beginning by way of single cluster setting then onto the next [67], [68].

A reputation-based structure utilizes coordinate perceptions and second-hand data disseminated in the middle of nodes in a scheme to gauge a node. A confidence foundation structure assesses immediate nodes in light of straight or direct perceptions while trust connections among a pair of nodes instead of earlier direct connections are worked through a mix of feelings from intermediate nodes.

3.2.3.2 Structure-based classification

Trust calculations comprise three segments: 'experience', 'recommendation', and 'knowledge' [69].

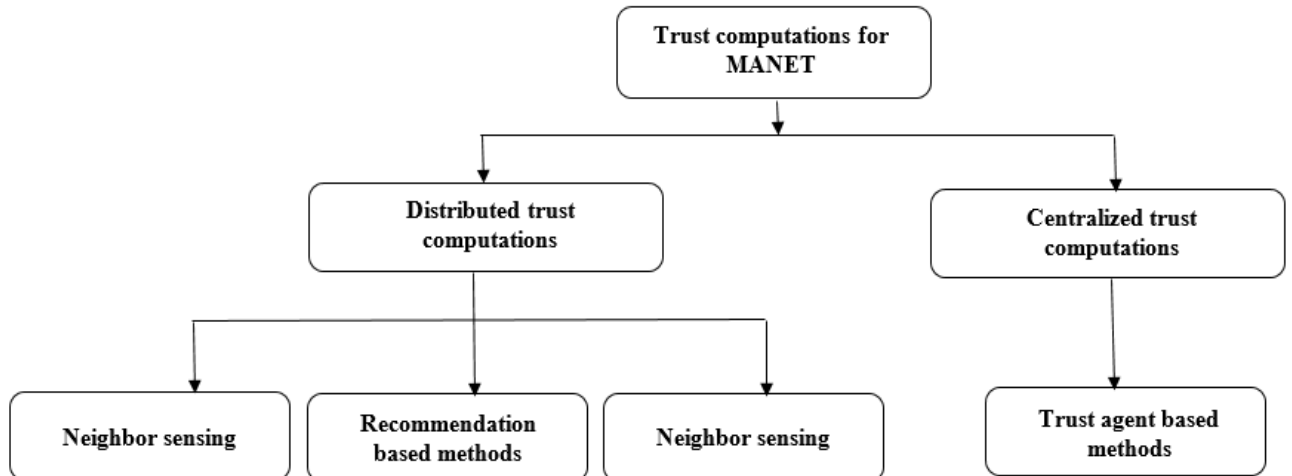


Figure 3.9: Trust Computing Methods Classifications

The work on trust calculations can be extensively grouped into the accompanying classes in light of the structure:

- Distributed trust calculations: Each and every node in the network estimates its own trust rate on its acquaintance/neighbouring nodes on the same network
- Centralized trust calculations: a central agent, which is selected from the network as the major performer, manages/ helps in trust calculation

Distributed trust calculations:

There are three major distributed trust computation procedures, namely Neighbour sensing (Direct trust), Recommendations based trust (Indirect trust), and Hybrid technique.

1. Neighbour sensing (Direct trust):

Distributed trust calculation in light of neighbour detecting is shown in Figure 3.10 (a), where each node watches its own neighbour's actions or the behaviours for their interactions and acquires the reports in 'information' reserve. A node, which is confided in estimating node

(trustor node) will investigate its own perception and be responsible for facts concerning the occurrence by way of the observation summary it regulated by the node which one trust should be estimated (trustee node) and moreover from one more relatively neighbours. Trust feature will be established in light of the assessment of alterations of the observation reports [43].

In this proposed model direct trust in between A_i and A_j is specified by:

$$T_{A_i A_j}(n) = \frac{[CP_{A_i A_j}(n) + DP_{A_i A_j}(n)]}{2} \begin{cases} i = 1 \\ n = 1, 2, 3.. \\ j = 1, 2, 3.. \end{cases} \quad (3.4)$$

Where A_i symbolizes the evaluating node and A_j symbolizes the evaluated node by A_i . CP signifies control packets (transferring or reacting proportion), and DP means data packets transferring proportion after some time with n number of collaborations with the one-hop neighbouring nodes.

2. Recommendation based trust (Indirect trust):

Distributed trust estimations in light of recommendation systems appear in Figure 3.10 (b). Here, trust associates among nodes in the network are enhanced in observation of commendations alone.

A trust inaugurate procedure constructed on local elective for ad hoc setups is conferred in [70]. A trust system illustration G is moulded wherever members in the system are associated if those neighbours are one hop away (immediate/ adjacent neighbours) by means of physical transportations. At this time, each and every node in the relevant context has trust rate/value either +1 or -1 (+1 for fully trustworthy node and -1 for fully un-trustworthy node) by means of the confidence of $c \in [+1, -1]$ on every single member in the system. In this elective strategy $c_{ij} = 1$ symbolizes wholly positive confidence i has on j , $c_{ij} = -1$ symbolizes entirely trust associates are asymmetric, i.e., $c_{ij} \neq c_{ji}$. In the voting regulation, assume node i is considered as the object for the confidence determination, each and every approximation cost on i from neighbouring node s will be accumulated to construct a trust worth. Ever since the node, which has the responsibility of giving the recommendation, himself may perhaps be a misbehaving node or having collaborative malicious behaviour with the precise node, rather

than just applying summary as accumulation this procedure is recommending an adequate elective strategy. The adequate confidence rate between i and j is represented by the negative confidence and $c_{ij} = 0$ implies exclusively ambiguous (uncertain), i.e., i and j have no associations.

$$\hat{c}_{ij} = \frac{c_{ij} + c_{ji}}{2} \quad (3.5)$$

Assuming $S_i(k)$ is the trust rate of i at the k^{th} occurrence and the trust rate at the $(k + 1)^{\text{th}}$ occurrence is denoted by:

$$S_i(k + 1) = \begin{cases} 1 & \text{if } m_i(k) > \eta \\ -1 & \text{if } m_i(k) < \eta \end{cases} \quad (3.6)$$

Where η is certain threshold and $m_i(k)$ is denoted by:

$$m_i(k) = \sum_{j \in N_i} \hat{c}_{ji} S_j(k) \quad (3.7)$$

Where N_i is the total nodes count in the limited network in which each and every member of the given network is associated. Furthermore, this correspondingly states a comprehensive polling specification whereas an alternative just N_i companions, the approximations from every single node in the system is deliberated in figuring the trust worth.

In our case, each one in the network has a TrV in between 0 and +1 (+1 for fully trustworthy node and -1 for fully un-trustworthy node) by means of the confidence of $c \in [0, +1,]$ on every single neighbour in the network. In this recommendation procedure, $c_{ij} = 1$ symbolizes wholly positive confidence i has on j , $c_{ij} = 0$ implies exclusively ambiguous (uncertain), i.e., i and j have no associates.

3. Hybrid method:

For instance, in Figure 3.10 (c), in the Hybrid method, trust for a precise node is estimated in the opinion of straight facts and moreover references receiving from the rest of the neighbour nodes in the system. A trust devising grounded on direct consolidation of self-estimated trust worth ($0 \leq T_s \leq 1$) and adjacent neighbour deliberated trust worth ($0 \leq T_o \leq 1$) for mobile networks is advised in [45]. The node x 's confidence on node y is indicated by:

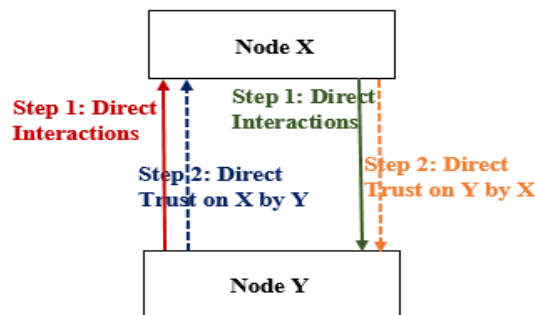
$$T_{x,y} = \alpha T_s + \beta T_o \quad (3.8)$$

Where the perpetual α and β are such that $\alpha + \beta = 1$. T_s is determined by direct perceiving y for whole packets lost by y , packet transmitting lag by y , wrongly transmitted packets by y and packets erroneously implanted by y . T_o is the comprehensive or the global trust evaluation through every single nodes in the network on y .

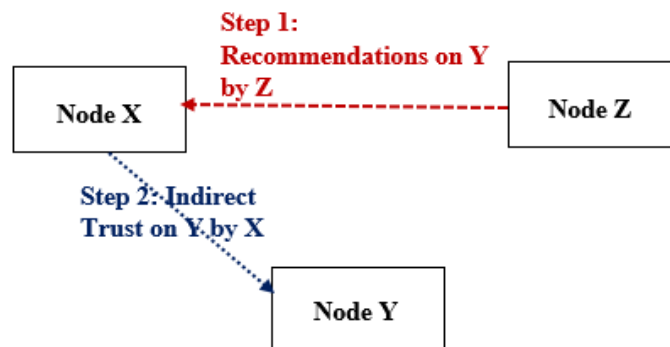
By using direct trust (DT) which is mentioned in, direct trust method and using indirect trust (IDT) method which is calculated based on recommendations of the one-hop neighbours, global trust (GT) will be calculated. Other than the direct trust and indirect trust it will consider relative layer trust (RLT) where we can get the TrV from the relative layers such as social network trust. This is optional to take the relative layer trust. Now, global trust is given by:

$$GT = DT + IDT + RLT \quad (3.9)$$

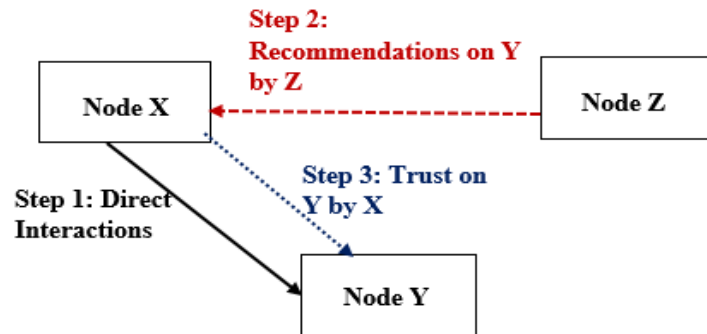
where GT is Global Trust, DT is Direct Trust, IDT is Indirect Trust, and RLT is Relative Layer Trust(optional).



(a) Direct trust formation based on neighbour sensing



(b) Indirect trust formation based on the recommendation



(c) Hybrid trust formation based on both direct interactions and recommendation

Figure 3.10: Pictorial demonstration of the discrete calculating strategy

Trust Strategies	Diverse Assaults							
	'DOS'	'BMA'	'OOA'	'CBA'	'SA'	'CA'	'CoA'	'NCA'
Distributed trust calculations								
Neighbour sensing	✓	✓	✓	✓	x	x	✓	x
Recommendation based methods	x	x	✓	✓	x	x	x	✓
Hybrid methods	x	✓	✓	✓	x	✓	✓	✓
Centralized trust calculations								
Trust agent-based method	x	✓	✓	✓	x	✓	x	✓

Table 3.1: Comparison of diverse trust calculating approaches with respect to numerous attack prototype

3.2.4 Trust MANET Security

Here, explore the formulation of trust in the system. The notion of trust can complement and often reduce the requirements from other cryptography-based security schemes. Trust can be seen as a gauge of a precise node's adherence to a given protocol. Thus, trust aptly captures the behavioural aspects of system elements. In fact, it may be argued that the notion of trust is critical in distributed systems and a fundamental requirement for collaboration between nodes.

For robust design, we utilize the parameters available at different layers. We view the network communication protocol as a composition of different components, rather than a single system

entity. One of the salient, yet critical advantages of the trust-based approach is the ability to quantify the influence of adversarial behaviour on individual components or layers of a system. This provides two significant advantages over the classical view.

Firstly, trust metrics developed for individual components may be utilized in multiple systems which share those components. Secondly, the component-based view also provides flexibility in mitigating adversarial behaviour. It may be easier to repair or replace malicious components, rather than the entire system.

3.2.5 Proposed Trust Model

Proposed Trust Model Traditionally routing protocols are designed to cope with routing operation, but in practice, they may be affected by misbehaving nodes so that they try to disturb the normal routing operations by launching different attacks with the intention to minimize or collapse the overall network performance. Therefore, detecting a trusted node means ensuring authentication and securing routing can be expected.

3.2.5.1 Trust Model for Direct Trust

In this model, research proposed a Trust and Q-learning based Security model to detect the misbehaving nodes over Ad Hoc On-Demand Distance-Vector (AODV) routing protocol. In the network each node plays a dual role such as an ordinary node; to perform network operations and router; to forward packets, hence there is no specialized router for forwarding the packet. Every single member of the particular network can 'join' and 'leave' at any time leading to dynamic topology. Such a special characteristic makes the network eligible various applications. At the same time providing security in such an environment is difficult due to distinct nature hence probability rate of failure is very high compared to a traditional network.

Typically trust can be evaluated based on direct and indirect trust, where the direct means, the information is gathered from one hop. But both trust information exhibits the historical interactions of nodes with respect to each other.

In this model, we make use of a Q-learning algorithm which is proposed by Watkins in the year 1989 in order to enrich the proposed model. The algorithm involves an agent, state s and a set of actions per state a . The state of the setting will alter once it acknowledged the action a . After

executing an action in a specific state, the agent gets a reward. The objective is to discover an ideal strategy that inspires the negotiator (agent) to acquire the aggregate reward during the whole operation and based on the total reward decision will be taken [25], [27]. The algorithm is defined as,

$$Q(s, a) = r(s, a) + \text{MAX}_{a'} \gamma(Q(s', a')) \quad (3.10)$$

Where $r(s, a)$ is an immediate reward, γ is a discount factor that governs the significance of forthcoming rewards. The cost of discount between 0 and 1 range, s' represents the new state after action a , a' represents the action in state s' and a and s represent the current state and action respectively.

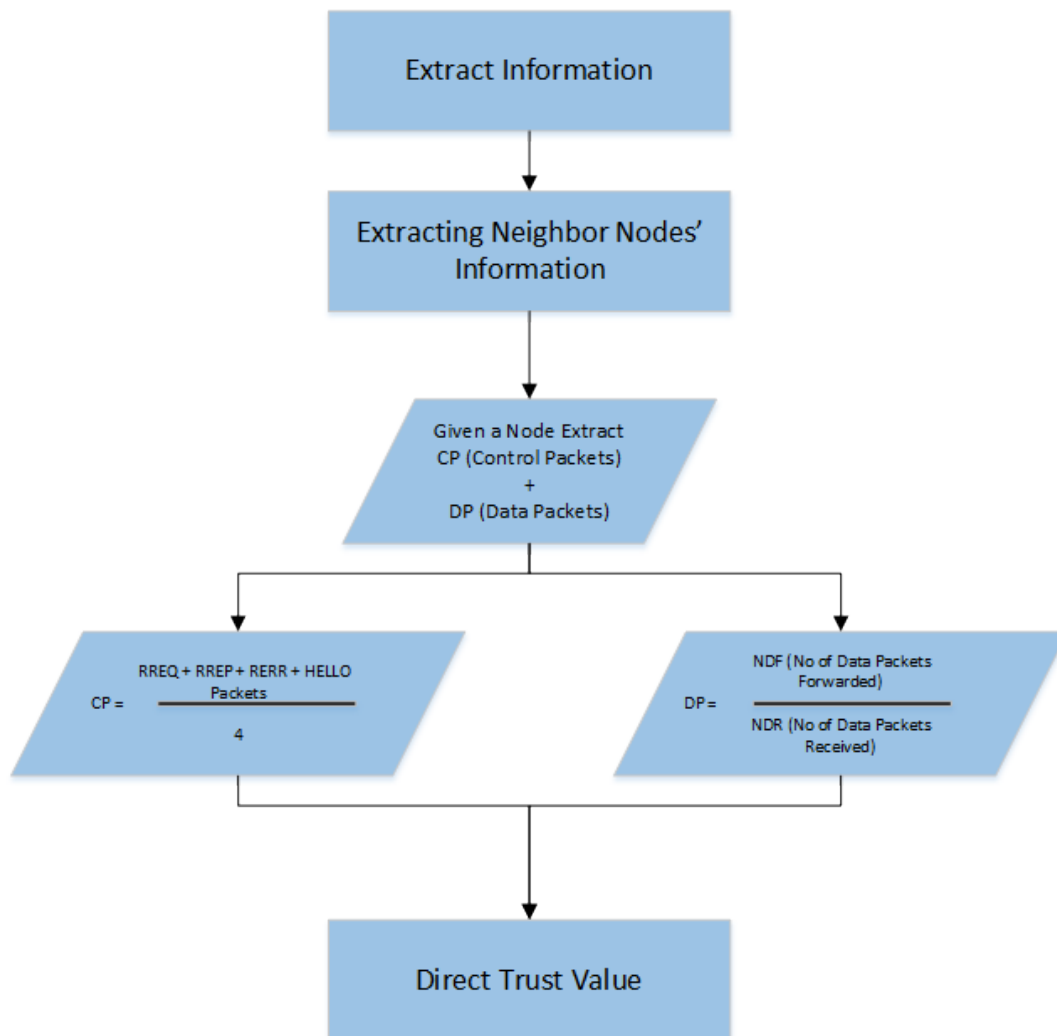


Figure 3.11: Direct Trust Calculation Process

To calculate the direct trust, we consider the Trust and Q-learning based Security (TQS) prototype. The key aspiration of this procedure is to guarantee authentication and detect misbehaving nodes by measuring the historical interactions of neighbouring nodes using direct trust evaluation mechanism. But here in this model, research is using this only to calculate the direct TrV, and in Fig 3.11 it shows the flow of calculating the direct trust of the neighbouring nodes of a given node in the same network in the given range.

Proposed prototype is constructed on the consecutive inferences. For simplicity, it is supposed that the system is small-scaled. Every single node in the network are behaving well at the time of preliminary network placement since all the nodes are authentic and all the nodes are having well-defined assets for instance battery power, bandwidth, and memory. Over the period, they may change their behaviour and become a misbehaving node.

Every single node in the system upholds three separate tables named ‘Trust table’, ‘Recommendation table’, and ‘Backup table’. In trust table direct trust, indirect trust, interactions, blacklist and global TrVs of their neighbour nodes can be stored. Every single trust entry/record on the Trust Table is allied through a time break (timeout). For that reason, on every occasion the node associated with that trust record is not valid or as soon as it expires it is going to remove the trust record or the entry from the particular table. The format of the Trust table is presented in Table 3.2.

Neighbour node	Direct Trust (DT)	Indirect Trust (IDT)	Global Trust (GT)	Interactions (In)	Blacklist (BL)

Table 3.2: Trust Table

Research also assumed that node’s TrV as continuous real numbers in the range 0 to 1 and based on the TrV we have categorized each node into five different categories or trust levels.

1. Trustworthy Nodes
2. Partially Trustworthy Nodes
3. Selfish Nodes
4. Pure Malicious Nodes
5. Collaborative Malicious Nodes

Secondly, in the recommendation table is recommending the nodes along with the recommendation value, reputation value or maturity level, and blacklist can be stored. Here the blacklist value is a Boolean value where we can check whether that the given node is a collaborative node or not. Structure of the recommendation table is shown in Table 3.3.

Neighbour node	Recommending nodes	Reputation Value/ Maturity level	Blacklist	Recommendation value

Table 3.3: Recommendation Table

Lastly, in the Backup table, the global TrV, backup time, and analyzed result value can be stored. Here we have the same values as a trust table global value in the trust table. Periodically it will update the backup table according to the recent values in the trust table, and after for a predefined time duration or timeout, it will delete the records which exceed the given time. For example, if we define the backup time duration as 2 hours, for each and every 2 hours it will check whether that backup time duration \Rightarrow 2 and if so it will delete the particular record from the backup table permanently. This will control the unnecessary memory wastage and expansion of the table. Structure of the Backup table is shown in Table 3.4.

Neighbour node	TrV	Time duration/ Backup time	Analyzed results

Table 3.4: Backup Table

As point out earlier, at the start, all the nodes are collaborating fine. Over the period of time, a node desires to transmit a packet to a precise endpoint. According to this model, primarily, all the nodes in the network flood the HELLO packets as an alternative for originating path discovery procedure or inspecting their individual routing table in order to choose the appropriate route. Hence that each node guarantees it's one hop adjacent nodes eventually only single hop acquaintances answer back to the HELLO packs since those nodes are at the equivalent range or the same network. From that, every single node can accomplish how many nodes are residing as single-hop acquaintances. In AODV the following four varieties of

control packs are used. In route discovery, as the control packets ‘route request’ (RREQ), ‘route reply’ (RPLY) packets are used. Now path maintenance procedure, as the control packets ‘route error’ (RERR) and HELLO packs are used. While appraising trust those four types of control packets are also contemplated, as they accommodate a momentous involvement towards the routing procedures. Though misbehaving nodes can also consume such packets but consume likelihood of such packets are comparatively less in number correlated with well-behaving nodes.

Where NDF denotes the quantity of data packets in reality forwarded, and NDR denotes the quantity of packets truly acknowledged against to the time with n quantity of communication links. Correspondingly, every single node could estimate the trust rate of all its single-hop acquaintances and renovate its Trust table according to the updated value. Every single node can observe its single hop neighbouring nodes’ transmitting manners by using inactive concession. After getting the CP and the DP, the algorithm can calculate the direct trust. Specially algorithm is not going to consider the Control packets and Data packets for only one transaction for calculating the direct trust, instead of that periodically within a given time period we have to get the summation of CP and DP and finally get the average TrV as the direct trust. It is given by,

$$DT = \frac{\sum_{i=1}^n (CP+DP)}{2n} \quad (3.11)$$

where DT is direct TrV within time duration T, CP is Control Packets for T time duration, DP is Data Packets for T time duration, and n is the number of interactions or transactions for T time duration.

Procedure 1: Calculate Direct Trust (DT) for time period T

- 1 Identify neighbour nodes
- 2 for every neighbour node do
- 3 get count of RREQ
- 4 get count of RPLY
- 5 get count of ERR
- 6 get count of HELLO

- 7 calculate CP for T time duration
- 8 get number of data packets sent within T
- 9 get number of data packets received within T
- 10 calculate DP for T time duration
- 11 calculate DT for T time duration
- 12 add DT to Trust table
- 13 end for

Algorithm 1: Calculate Direct Trust (DT) for time period T

```

1  BEGIN
2      FOR each neighbour node do
3          Get count of RREQ
4          Get count of RPLY
5          Get count of ERR
6          Get count of HELLO
7           $CP = (RREQ + RPLY + ERR + HELLO)/4$ 
8          Sent = number of Data Packets Sent
9          Received = number of Data Packets Received
10          $DP = \text{sent data packets} / \text{received data packets}$ 
11          $DT = \frac{\sum_{i=1}^n (CP+DP)}{2n}$ 
12         Save DT in trust table DT to Trust table
13     END FOR
14 END

```

3.2.5.2 Trust Model for Indirect Trust

Nodes in a self-coordinated dispersed system can confer ‘positive’ or ‘negative’ endorsements regarding other nodes in the network either a well-behaving node or misbehaving node with some self-intrigue. These standpoints are practically corresponding to circumstances in composite frameworks by means of game theoretical collaborations [71]. There can be non-collaborative where each and every member of the network collaborates the game

autonomously or collaboratively where a cluster of members compose subsections and participate to the game collectively compared to the rest of the members in the system [72].

Giving a confidence metric to every single node in the network is cooperative, at the time when nodes behave badly (misbehave) besides, at the time when nodes transmit data. Hence, it is imperative to just act together with honest neighbouring nodes, meanwhile interfacing with disobedient neighbouring nodes can trade-off the independence of MANETs.

The proposed approach does not necessitate the distribution of the trustworthy data within the entire network. Rather, keeping and transmitting trustworthy information of the nodes within the given range should be done. Without the necessity for a comprehensive trust awareness, this scheme escalates fine for massive schemes whereas still lessening the quantity of conveyed messages and consequently the energy consumption.

Projected prototype declares an adjustable trust prototype based on the theory of human belief and afterwards employ this prototype to ad hoc networks. This prototype constructs, for every single node, a trust association to all the other neighbours in the network. The trust is established on prior distinctive maturities of the node (direct trust) and on the endorsements (indirect trust) of its acquaintances. The endorsements/recommendations enhance the trust appraisal procedure for nodes that unsuccessful in perceiving their acquaintances attributable to capability limitations or association interruption. The capability to evaluate the TrV of its associated neighbouring nodes carries numerous benefits. First, a node has the ability to distinguish and isolate malevolent practices, escaping transmitting packets to misbehaving neighbours. Furthermore, collaboration is roused by choosing the acquaintances with greater trust ranks. Nodes in the network study built on the facts transferred with honest fellow nodes to construct an acquaintance plane [21], [23]. An additional effect is that endorsements/recommendations are simply swapped among fellow nodes, explicitly, 'recommendations are not forwarded'.

This tactic similarly decreases the possibility of fabricated endorsements as the amount of acknowledged recommendations is considerably minor, and there is no intermediary node to boost the uncertainty or the ambiguity of the data. Moreover, a particular node can continuously stabilize the acknowledged endorsements with its direct trust (own experiences) to estimate the TrV, for the reason that nodes do not compute the trust of nodes that are not fellow nodes.

In this effort, it describes trust as the expense that echoes the history of the actions that a node has regarding a precise fellow node. This fact is used as an anticipation of its fellow node's upcoming actions. Fellow nodes can additionally contribute their individual beliefs to enhance the trust appraisal. The communication of a belief regarding a precise node i is determine as an endorsement. Fellow nodes contemplate this recommendation whereas computing the trust for node i . The major objective of the endorsements is to reimburse for the lack of observing abilities due to resource restrictions. Generally, a precise node in the network is not capable to detect the whole actions of a certain acquaintance over time. Recommendations from alternative acquaintances in the network are beneficial in this situation for a precise trust allocation. Furthermore, the usage of recommendations can accelerate the consolidation of the trust assessing procedure.

The research considers the concept of maturity level same as the human-related maturity where it gives priority to node which is having long-term relationship or the interactions for a long time with the evaluating node and later on based on this maturity level of each and every node it can give a weight without getting the recommendation TrV as it is. This thought permits nodes to contribute additional concern to recommendations referred by long-standing acquaintances more willingly than short-range acquaintances. If it considers equation 3.12 Relationship maturity or the Maturity level of B is given by,

$$ML_B = \frac{IN_{AB}}{IN_{AB} + IN_{AC}} \quad (3.12)$$

where ML_B is the maturity level of B, IN_{AB} is the number of interactions between A and B, and IN_{AC} is the number of interactions between A and C within a given time period.

For illustration, if investigation contemplates the sample network in the Figure 3.12 namely A, B, C, D, E nodes linked via a dotted arrow are fellow nodes and the amount specify for the maturity level in between each and every node, to be precise, the association maturity criterion. A usual arrow denotes a recommendation, and the letter on top of the arrow specifies the objective fellow node. The principal object to acknowledge is that recommendations involve single frequent fellow node of diverse nodes. So therefore, node D is a frequent fellow node of node A, B, and C. Node B and C forward their recommendation regarding node D to node A. Node A will contemplate the recommendation or the reference from node B further significant

than the recommendation or the reference acknowledged from node C since node B has an extensive or the strong connection with node A (high maturity level). It is noteworthy that recommendations referred by node D regarding node E will be unnoticed by the other node in the network, since node E is not a directly connected neighbour of A.

In the suggested prototype, trust is a count of exposure with its rate indicated by entropy. Here it utilized four axioms in one of the existing work and one other new axiom that focus on the crucial understanding of trust and the principles for trust dissemination [73]. In view of these axioms, this model has a trust model: entropy-based model, which fulfils every axiom.

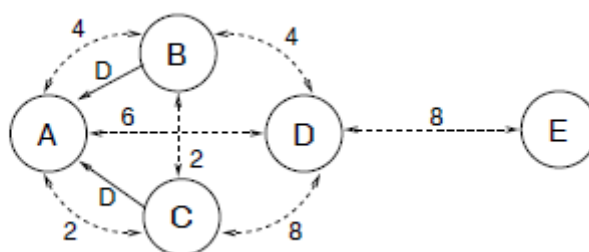


Figure 3.12: Example: node A obtains recommendations regarding node D

A dispersed scheme is intended to secure, sustain, and refresh trust entries associated with the nodes' packet dispatching manners and the practices of constructing references regarding fellow nodes in the same network. From this awareness of trust, it forms axioms that focus on the necessary philosophies to accumulate trust through an arbitrator (concatenation propagation) and through recommendations from numerous bases (multipath propagation). In light of these axioms, exploration forms procedures that compute trust rate from inspection and compose the prototype that discuss the integration and multiple path trust diffusion complication in ad hoc networks. Simulations are accomplished to judge the viability of the suggested prototype in ad hoc networks. Distinct clients acquire the trust rate by considering transmitting packets and composing recommendations or references in a decentralized manner. The malevolent fellow nodes in the network can be distinguished, in addition, their varieties can also be acknowledged. The suggested prototype can also acknowledge the dynamics of the structures obediently. Without trust estimation, the suggested prototype can pick the appropriate path with advanced arranged excellence to facilitate the degree of packet abandoning are enormously diminished.

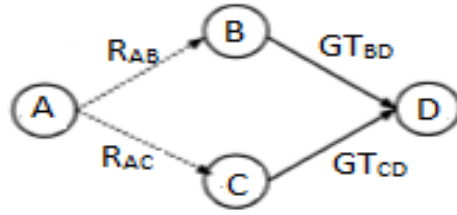


Figure 3.13: Node A requesting recommendations about D from B and C

Indirect trust model, each node maintains a trust table containing neighbour nodes' information. This is where the calculated 'Indirect Trust' value will be recorded. For the ease of demonstration, we will define 4 nodes as 'A', 'B', 'C' and 'D'.

R_{AB} - B recommendation about C for A.

GT_{BD} - Global trust between B and D.

In - Interactions between the two nodes.

W_B, W_C - Weight factors.

Indirect TrV which A records for D (IDT_{AD}) can be calculated as follows.

$$IDT_{AD} = W_B [R'_{AB} * GT_{BD}] + W_C [R'_{AC} * GT_{CD}] \quad (3.13)$$

Where,

$$W_B = R'_{AB} / (R'_{AB} + R'_{AC}) \quad (3.14)$$

$$W_C = R'_{AC} / (R'_{AB} + R'_{AC}) \quad (3.15)$$

$$R'_{AB} = ML_B * R_{AB} \quad (3.16)$$

$$R'_{AC} = ML_C * R_{AC} \quad (3.17)$$

$$ML_B = In_B / (In_B + In_C) \quad (3.18)$$

$$ML_C = In_C / (In_B + In_C) \quad (3.19)$$

We assume that R_{AB} is equivalent to DT_{BD} which is Direct Trust between Node B and D. A weight factor is defined in order to make the indirect TrV between 0 and 1.

Procedure 2: Indirect Trust

```
1  Get trust table entries as node_entry_list
2  for every node in node_entry_list do
3    calculate indirect TrV for the node
4    update the indirect TrV in the trust table
5    update Global Trust for the node
6  end for
7  Function calculateIndirectTrust
8  consider the node
9  get trust table entries for the given node
10 for each nei_node in node_entry_list do
11  calculate the weight for a given node
12  calculate final recommendation trust for a given node
13  get Direct Trust and Global Trust for a given source to the destination node
14  Calculate the weight term and get the summation of the weight
15 end for
16 Return the summation of the weight
17 .End function
18 Function calculating maturity level for a given node
19  get number of interactions for the given node
20  get trust table entries for the given node
21  for each nei_node in node_entry_list do
22  get no of interactions for each node
23  end for
24  calculate maturity level
25 End Function
26 Function calculating final recommendation trust for a given node
27  calculate maturity level for a given node
28  get Direct Trust and Global Trust for a given source to the destination node
29  final recommendation trust = maturity level * DT
```

```

30 Return final recommendation trust
31 Endfunction
32 Function calculate weight for a given node
33     calculate new recommendation trust for a given node
34     get trust table entries for the given node
35     for each node in node_entry_list do
36         Calculate new recommendation trust for a given node
37         Get the summation of new recommendation trust
38     end for
39 Return the ratio of new recommendation trust from the summation of new
    recommendation trust
40 End Function
41 Function get DT and GT for a given source to destination node
42     send TRR to the neighbour node and get DT and GT for the destination node
43 End function
44 get trust table entries
45 for each node in node_entry_list do
46 calculate indirect TrV for the node
47 update the indirect TrV in the trust table
48 Update Global Trust
49 end for
50 End

```

Algorithm 2: Indirect Trust

```

1  BEGIN
2  Function calculate indirect TrV for a given node
3  Pass In: in_node
4  node_entry_list = getTrustTableEntries()
5  w_sum = 0
6  FOR each nei_node in node_entry_list DO
7      W = calculate W for a given node

```

```

8         R'_nei_node = calculating R' for a given node
9         Rec[] = get DT and GT for a given source to destination node
10        cal_w_term = W * (R'_nei_node * rec[1])
11        w_sum = w_sum + cal_w_term
12    END FOR
13    Pass Out: w_sum
14    Endfunction
15    Function calculating maturity level for a given node
16    Pass In: p_node
17    Intr_p_node = p_node.getInteractions()
18    node_entry_list = getTrustTableEntries()
19    In_all = 0
20    FOR each entry in node_entry_list DO
21        In_all = In_all + entry.getInteractions()
22    END FOR
23    ML = Intr_p_node / In_all
24    Pass Out: ML
25    Endfunction
26    Function calculating R' for a given node
27    Pass In: p_node
28        ML = calculating maturity level for a given node
29        rec[] = get DT and GT for a given source to destination node
30        DT = rec[0]
31        R' = ML * DT
32    Pass Out: R'
33    Endfunction
34    Function calculating W for a given node
35    Pass In: p_node
36        R'_nei_node = calculating R' for a given node
37        R'_all = 0
38        node_entry_list = getTrustTableEntries()

```

```

39 FOR each node in node_entry_list DO
40     R'_node = calculating R' for a given node
41     R'_all = R'_all + R'_node
42 END FOR
43     W = R'_nei_node / R'_all
44 Pass Out: W
45 Endfunction
46 Function get DT and GT for a given source to destination node
47 Pass In: nei_node, in_node
48     rec[] = sendTRR(nei_node, in_node)    //both DT and GT stored
         inside rec
49 Pass Out: rec[]
50 Endfunction
51 node_entry_list = getTrustTableEntries()
52 FOR each node in node_entry_list DO
53     calculate indirect TrV for the node
54     update the indirect TrV in the trust table
55     updateGlobalTrust(node)
56 END FOR
57 END

```

In here, trust is defined by means of a degree of vulnerability in addition, the fundamental comprehension of trust is recapitulated as given below.

1) Trust can be defined as an interconnection set up among two entities for a particular activity. Explicitly, a single object confides the other object in order to accomplish a precise task. Furthermore, in this tactic, the principal object is known as the **requestor**; the subsequent object is acknowledged as the **neighbour**. Hereafter we discuss the notation {requestor: neighbour, action} to distinguish a trust association.

2) Trust is an element of vulnerability. Explicitly, if the requestor trusts that the neighbour will play out the given action or the task without a doubt or any failure, the requestor completely

“trusts” the neighbour to play out that activity in addition to that there is no ambiguity (uncertainty); if the requestor have faith in that the neighbour will not accomplish the specific action for certain, the requestor will not “trusts” the neighbour and not allow him to perform the precise action, besides there is no whichever vulnerability either. Then the requestor has the greatest significant uncertainty for this situation.

3) The rate of trust can be computed by a continuous real number, raised to as the TrV. Vulnerability should be represented by TrV.

4) The requestors may perhaps have diverse trust rates with the equivalent fellow node aimed at the equivalent act or the task. Trust is not certainly coherence. The point that A believes B do not certainly anticipate that B correspondingly beliefs A, where A and B are two objects in the same network.

Grounded on human knowledge of trust, in this model more established fundamental axioms for inaugurating trust association from end to end either straight communications or from end to end indirectly as recommendations in the middle of the neighbour and the requestor [73].

Axiom 1: Ambiguity or the Uncertainty is one of the Trust Measurement: The theory of trust explains the confidence or certainty of whether the neighbour will accomplish a specific action in the requestor’s viewpoint. Let T {requestor: neighbour, action} symbolize the trust rate of the association {requestor: neighbour, action}, and P {requestor: neighbour, action} symbolize the likelihood that the neighbour will accomplish the specific action in the requestor’s viewpoint. It is significant that this likelihood is not definite, nevertheless the belief of a precise requestor. Accordingly, diverse requestors can allocate diverse likelihood values for the identical neighbour as well as the identical action. Information theory conditions that entropy is an ordinary portion for ambiguity or the uncertainty [74]. Therefore, we explain the entropy-based trust as

$$T\{\text{requestor: neighbour, action}\} = \begin{cases} 1 - H(p), & \text{for } 0.5 \leq p \leq 1 \\ H(p) - 1, & \text{for } 0 \leq p \leq 0.5 \end{cases} \quad (3.20)$$

$$\text{Where } H(p) = -p \log_2(p) - (1 - p) \log_2(1 - p) \quad (3.21)$$

And $p = P \{ \text{requestor: neighbour, action} \}$. Here we consider, the TrV is a continuous real number in $[0, 1]$. Given notation contains below described properties. When $p = 1$, the requestor fully faiths the neighbour, ever since the highest TrV is 1. When $p = 0$, the requestor fully disbeliefs the neighbour, ever since the lowest TrV is 0. When $p = 0.5$ (Average trust level) the requestor either does not fully trusts the neighbour nor fully disbeliefs the neighbour beside the TrV is 0.5. Generally, TrV is negative for $0 \leq p < 0.5$ and positive for $0.5 < p \leq 1$. TrV is a cumulative function with p . It is distinguished that (3.20) is a one-to-one representing in the middle of $T \{ \text{requestor: neighbour, action} \}$ and $P \{ \text{requestor: neighbour, action} \}$.

Axiom 2: Integration Circulation of Trust Does Not Increase Trust: Once the requestor inaugurates a trust association with the neighbour via the reference from a fellow node, the trust cost in the middle of the requestor and the neighbour should not exceed the trust cost in the middle of the requestor and the recommender, in addition to the trust cost in the middle of the recommender and the neighbour. In this axiom, it explains that ambiguity boosts through distribution. A trust association can be symbolized using the diagram presented in Figure 3.14, where the trust worth is given by the heaviness of the edge in the diagram. Dashed lines signify giving recommendations, and solid lines signify the accomplishment of a specific action. Once association $\{A: B, \text{action}_r\}$ and $\{B: C, \text{action}\}$ are existing, trust association $\{A: C, \text{action}\}$ can be recognized if the given two circumstances are fulfilled.

1) The action r is to give a recommendation of different nodes in the network regarding accomplishment of a specific action.

2) The trust worth of $\{A: B, \text{action}_r\}$ is positive.

The first and major circumstance is compulsory for the reason that the objects that accomplish the specific action do not certainly send the truthful references or the recommendations [25]. Next circumstance ensures that the recommendations from objects with low degree trust worth would not be considered for the trust estimation process. Above described circumstance creates the circulation of trust among the neighbouring nodes in decentralized systems resilient to misbehaving individuals who can influence their endorsements (recommendations) with the intention of effect highest harm towards the network. Furthermore, the second circumstance is not crucial for various additional states where the misbehaving individuals' activities of

building endorsements (recommendations) are foreseeable. The mathematical demonstration of Axiom 2 is

$$T_{AC} \leq \min(R_{AB}, T_{BC}) \quad (3.22)$$

Where $T_{AC} = T \{A: C, \text{action}\}$, $R_{AB} = T \{A: B, \text{action}_r\}$, and $T_{BC} = T \{B: C, \text{action}\}$. This is identical to data transforming in information theory: the information or the facts cannot be amplified by way of circulation. In our demonstration, the trust constructed upon neighbouring nodes' recommendations is not over and above the recommenders' trust besides the faith in the recommenders.

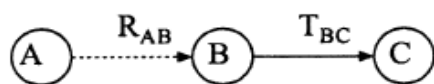


Figure 3.14: Concatenation trust propagation

Axiom 3: Multipath Circulation of Trust Does Not Lessen Trust: If the requestor obtains the equivalent endorsements (recommendations) for the neighbouring nodes from numerous bases, the trust worth should not be lower than that in the circumstance where the requestor obtains a lesser amount of endorsements. Furthermore, as demonstrated in Figure 3.15, A_1 and C_1 entities inaugurates trust from end to end by using single concatenation route, and A_2 , C_2 entities inaugurates trust from end to end by using two identical routes. Let $T_{A_1C_1} = T \{A_1: C_1, \text{action}\}$ and $T_{A_2C_2} = T \{A_2: C_2, \text{action}\}$. The mathematical demonstration of Axiom 3 is

$$\begin{aligned} T_{A_2C_2} &\geq T_{A_1C_1} \geq 0, \text{ for } R_1 > 0 \text{ and } T_2 \geq 0 \\ T_{A_2C_2} &\leq T_{A_1C_1} \leq 0, \text{ for } R_1 > 0 \text{ and } T_2 < 0 \end{aligned} \quad (3.23)$$

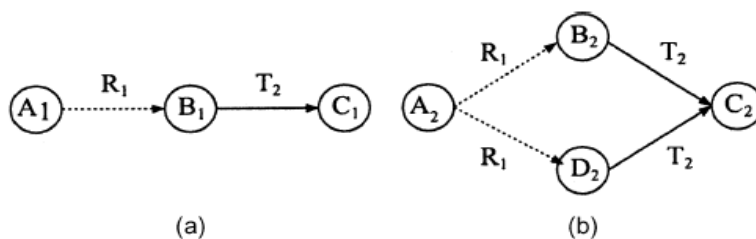


Figure 3.15: Combining trust recommendations

Where $R_1 = T \{A_1: B_1, \text{action}_r\} = T \{A_2: B_2, \text{action}_r\} = T \{A_2: D_2, \text{action}_r\}$ and $T_2 = T \{B_1: C_1,$

action} = T {B₂: C₂, action} = T {D₂: C₂, action}. Above described circumstance builds that multipath endorsements will not rise ambiguity or the uncertainty. Axiom 3 holds if and only if numerous bases produce the equivalent endorsements. This is for the reason that the cooperative grouping of diverse endorsements is a tricky purely that can produce diverse trust worth corresponding to diverse trust prototypes.

Axiom 4: Trust Established on Numerous Recommendations from a Solitary Basis Should Not Be Greater than That from Autonomous Bases: As soon as the trust association is recognized cooperatively via concatenation as well as multipath trust circulation, it is conceivable to have various endorsements on a solitary basis, as presented in Figure 3.16 (a). From the time when the endorsements from a solitary basis are extremely associated, the faith constructed on those associated endorsements should not be exceeding the faith constructed upon endorsements from autonomous bases. Specifically, let $T_{A_1C_1} = T \{A_1: C_1, \text{action}\}$ symbolize the trust worth acknowledged in Figure 3.16 (a), and $T_{A_2C_2} = T \{A_2: C_2, \text{action}\}$ symbolize the trust worth acknowledged in Figure 3.16 (b). Above axiom declares that

$$\begin{aligned} T_{A_2C_2} &\geq T_{A_1C_1} \geq 0, & \text{if } T_{A_1C_1} &\geq 0 \\ T_{A_2C_2} &\leq T_{A_1C_1} \leq 0, & \text{if } T_{A_1C_1} &< 0 \end{aligned} \quad (3.24)$$

Where $R_1, R_2,$ and R_3 are altogether positive values. The physical intention of the above axiom is that the endorsements from autonomous bases can lessen ambiguity or the uncertainty more efficiently than the endorsements from simultaneous bases.

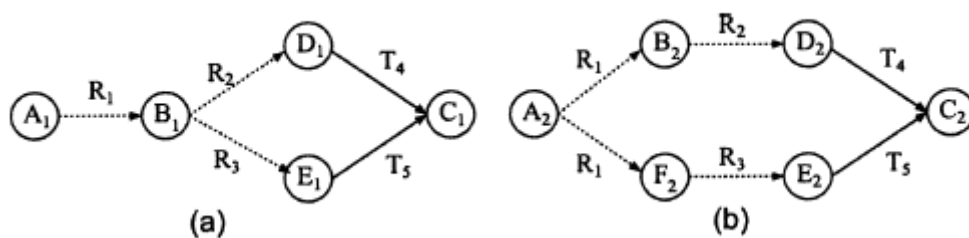


Figure 3.16: One entity provides multiple recommendations

Axiom 5: Recommended TrV Should Be Higher for High Interpreted Relationship Maturity Nodes: when the node establishes a trust relationship the neighbour node, it indirectly checks the maturity level or the relationship maturity of the recommending node. The maturity level

is the number of historical transactions completed by targeted nodes.

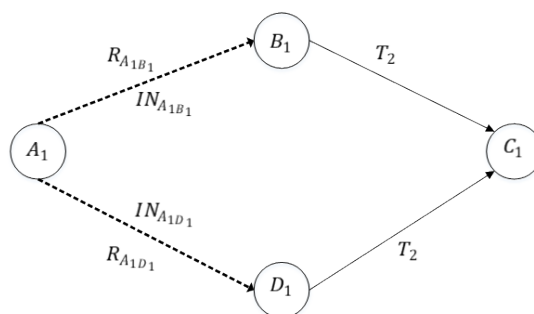


Figure 3.17: A1 requesting a recommendation about C1 from B1 and D1

Maturity level for B1 and D1 is given by,

$$ML_{B1} = \frac{IN_{A1B1}}{IN_{A1B1} + IN_{A1D1}} \quad ML_{D1} = \frac{IN_{A1D1}}{IN_{A1B1} + IN_{A1D1}}$$

Where IN_{A1B1} is the number of interactions between A1 and B1 and IN_{A1D1} is the number of interactions between A1 and D1.

Given example, A1 needs a recommendation about C1 from B1 and D1. Trust recommendation value would be higher once given the fact $ML_{B1} > ML_{D1}$. Here we assume that both recommending values are the same. So, the final recommendation totally depends on the maturity level and the resulted recommendations R_{A1B1}' and R_{A1D1}' are given by, $R_{A1B1}' = R_{A1B1} * ML_{B1}$ and $R_{A1D1}' = R_{A1D1} * ML_{D1}$,

$$R_{A1B1}' > R_{A1D1}' \geq 0 \quad \text{if } ML_{B1} > \quad (3.25)$$

$$R_{A1D1}' > R_{A1B1}' \geq 0 \quad \text{if } ML_{D1} > ML_{B1}$$

3.2.6 Trust Architecture for MANETS

In order to calculate the trust and identify trust levels basically, there are three main processes other than direct trust calculation process. Three phases of this model are trust level identification phase, collaborative malicious nodes discovery process using spiral model and after transmission phase where we take the actions of the misbehaving nodes.

1. Trust Level Identification phase

After calculating the direct trust and the indirect trust of the neighbours, we are getting the global trust and based on the global TrV we have defined five trust levels. Next challenge is to identify the trust level of the neighbours, and by considering the trust level, we have to get the routing decision. According to the TrV, five trust levels are listed down in below table. When we are dividing the trust level, we have to consider a threshold value for each and every level. In the AODV context we only calculating the global trust using direct trust and indirect trust model which is mentioned in the previous section under proposed trust model and as the input for the RL model we are going to give this TrV in order to get the reward or the Q value for the next process, where an evaluating node can take routing decision based on the reward of each of its neighbouring nodes. In order to communicate with these two modules, there is another module called flow monitor in-between the AODV module and the RL module.

Trust Level	Threshold	Meaning
1	$\geq TH1$	Trustworthy node
2	$< TH1$ and $\geq TH2$	Partially Trusted node
3	$< TH2$ and $\geq TH3$	Selfish node
4	$< TH3$ and $\geq TH4$	Pure Malicious node
5	$< TH4$	Collaborative Malicious node

Table 3.5: Threshold Table

After the reward computation phase, now an evaluating node can take a decision based on the reward of each of its neighbouring nodes.

This reward will be checked against the predetermined threshold value which is mentioned in Table 3.5. Here the threshold values can also be changed according to the user specification.

These values are classified into five categories, first one is trusted nodes; we can allow those nodes in normal routing operation and data processing is actual data forwarding and receiving, second is partially trusted nodes; we can allow those nodes in normal routing operation and data processing is actual data forwarding and receiving same as trusted nodes, third level is selfish nodes; we allow those nodes to take part in the normal routing operation but they will not be involved in actual data processing, and their TrV will be reduced by considering their maturity level or the reputation, next level is, pure malicious nodes; those nodes are isolated from the network and information about the misbehaving nodes can be broadcast by evaluating

nodes before actual route discovery commences. Therefore, those nodes are deleted from all the trust tables, recommendation tables, and backup tables. By the way, such nodes are isolated from the route discovery process, and therefore authentication is ensured by excluding those nodes.

Finally, we have the collaborative malicious nodes; same as the malicious nodes for these nodes also do the same process and take the same actions and apart from that special action will be taken for the collaborative malicious nodes such as other than broadcasting and isolation of the node particular node will be added to the blacklist and declare it as a blacklisted node. After blacklisting the node, it will broadcast and notify the neighbouring nodes in order to collaboratively convey the collaborative malicious behaviour of the particular node and take the decision about the node. When discovering the collaborative nodes, we have to go through the Spiral model which we discuss in the next section. Every node can execute the spiral model over the period of time or when needed. After getting the same blacklisted node second time from another neighbour evaluating node can confirm the collaborative node and it will be deleted from the tables.

As shown in the Figure 3.18 first we use the TH2 threshold value to separate the nodes to two different categories and using these two different node categories we can easily identify three basic trust levels; trustworthy nodes partially trusted nodes and finally selfish nodes. But the challenge is to differentiate Pure Malicious nodes and Collaborative Malicious nodes based on their dynamic behaviour within the network.

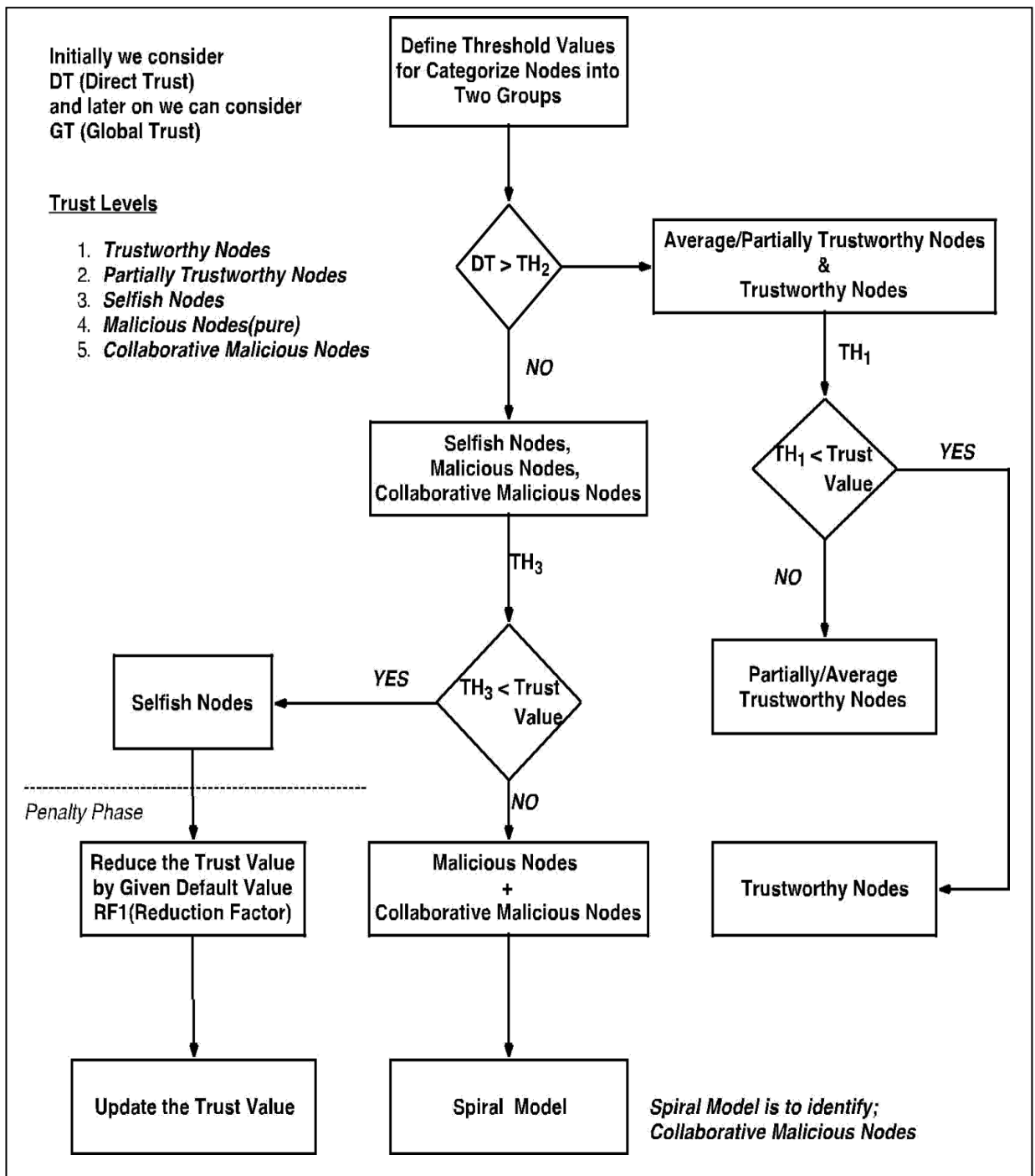


Figure 3.18: Basic Trust Level Identification Process

After categorizing these nodes into two different groups, first, we consider the nodes where the TrV of the node is greater than the given threshold value TH2. According to our table 3.5 values that can be either partially trusted node or a Trustworthy node. Secondly, we have re-categorized the group according to the TH1 value and if the TrV is greater than the given threshold value TH1 we identify it as a Trustworthy node and obviously other nodes will be partially trusted nodes according to our argument. Now, we have identified basic two trust levels, and as the next step, we are going to consider second basic category, which was the group where we have the nodes with fewer TrVs than the TH2 value. Here, we have three main trust levels; selfish nodes, pure malicious nodes and collaborative malicious nodes. Among those three we can easily identify selfish nodes directly using TH3 value. Then we identify the nodes where they have greater TrV than TH3, and obviously, those are in between TH2 and TH3 because our basic categorization was based on TH2. According to the trust level table, this category we can directly identify as selfish nodes and other nodes can be either pure malicious nodes or collaborative malicious nodes. In the first phase of this model, we can only identify the first three basic trust levels using TH1, TH2, and TH3. For the selfish nodes, we apply the penalty phase where we reduce the TrV of the selfish node by given value or a reduction factor. Reduction factor will be decided based on the maturity level or the reputation of the node. This process we called as penalty phase, and each and every time we found a selfish node it is a must to apply the penalty phase for the particular node. Immediate after the penalty, old TrV should update with the new value in the trust table.

A major challenge and the goal will be the collaborative malicious node identification from the malicious nodes cluster. Initially, we consider both pure malicious nodes and collaborative malicious nodes as one cluster called malicious nodes or misbehaving nodes for simplicity of the process. Eventually, we have four main trust levels after completing the first phase of the model. Another basic assumption of this phase is for an initial network where they have fewer communication links or interaction with each other it is not worthy for consider indirect trust or recommendation from the neighbours and the global trust for each node will be only the direct TrV. Initially, all the nodes will get a default value as the TrV, and as they behave in the network, the value can be changed with the time goes.

Procedure 3: Identify trust levels for each node

```
1  Considering a node, select the set of global TrVs (Z) for its neighbours from the trust
   table.
2  for every element  $B_i \in Z$  do
3   $B_i$  is checked with predefined threshold value ranges
4      if  $B_i > 0.6$ , then
5          if  $B_i > 0.8$ , then
6              set  $B_i$ 's trust level as 1 in the trust table.
7          else
8              set  $B_i$ 's trust level as 2 in the trust table.
9          end if
10     else
11         if  $B_i > 0.4$ , then
12             set  $B_i$ 's trust level as 3 in the trust table and reduce the indirect TrV
               from the trust table by the given reduction factor.
13         else
14             send to the spiral model to find out if the node is 'Pure malicious'
               or 'Collaborative malicious'.
15         end if
16     end if
17 end for
```

Algorithm 3: Identify trust levels for each node

```
1  BEGIN
2  Node A selects the set of global TrVs (Z) for its neighbours from the trust table.
3      FOR every element  $B_i \in Z$  DO
4          IF  $B_i > 0.6$ 
5              IF  $B_i > 0.8$ 
6                  set  $B_i$ 's trust level as 1 in the trust table.
7              ELSE IF
```

```

8           Set Bi's trust level as 2 in the trust table.
9           END IF
10          ELSE
11           IF Bi>0.4
12           Set Bi's trust level as 3 in the trust table.
13           Reduce the indirect TrV from the trust table by the given
reduction factor.
14          ELSE
15           Send to spiral model to find out if Bi is 'pure malicious'
or 'collaborative malicious'.
16          END IF
17          END IF
18          END FOR
19 END

```

2. Spiral Model for advanced categorization of malicious nodes

As the advanced categorizing of the malicious nodes, we have to go to the second phase, which is a spiral model where we have the collaborative malicious node discovery process. In the spiral model, mainly, there are three different phases.

Collaborative malicious node discovery process

This is the phase where we do the advanced categorization for the malicious nodes and identify the collaborative malicious nodes by analyzing the dynamic behaviour of the nodes. Only using one record we cannot predict a collaborative malicious behaviour, and we have to have more historical records or trust records. For this purpose, mainly we are maintaining a backup table where we store the recent records of the trust table and each entry on the backup table is associated with a timeout. Initially, we have predetermined range for the trust with high TrV (HT) and low TrV (LT) and using the backup table records and current trust record we can compare the values against the time. For a given time period we can analyze the TrVs, and after getting the analyzed report or plot, we can check for outliers within the given range HT –

LT. If it contains any outliers or there are any sudden dynamic changes of the TrVs, we can suspect it as a collaborative malicious node. Otherwise, it can be a pure malicious node without any dynamic changing behaviour. The range can be changed according to the user specification.

For the malicious nodes; those nodes are isolated from the network and information about the misbehaving nodes can be broadcast by evaluating nodes before actual route discovery commences. Therefore, those nodes are deleted from all the trust tables, recommendation tables, and backup tables. By the way, such nodes are isolated from the route discovery process, and therefore authentication is ensured by excluding those nodes. End of this phase we can identify the collaborative malicious nodes.

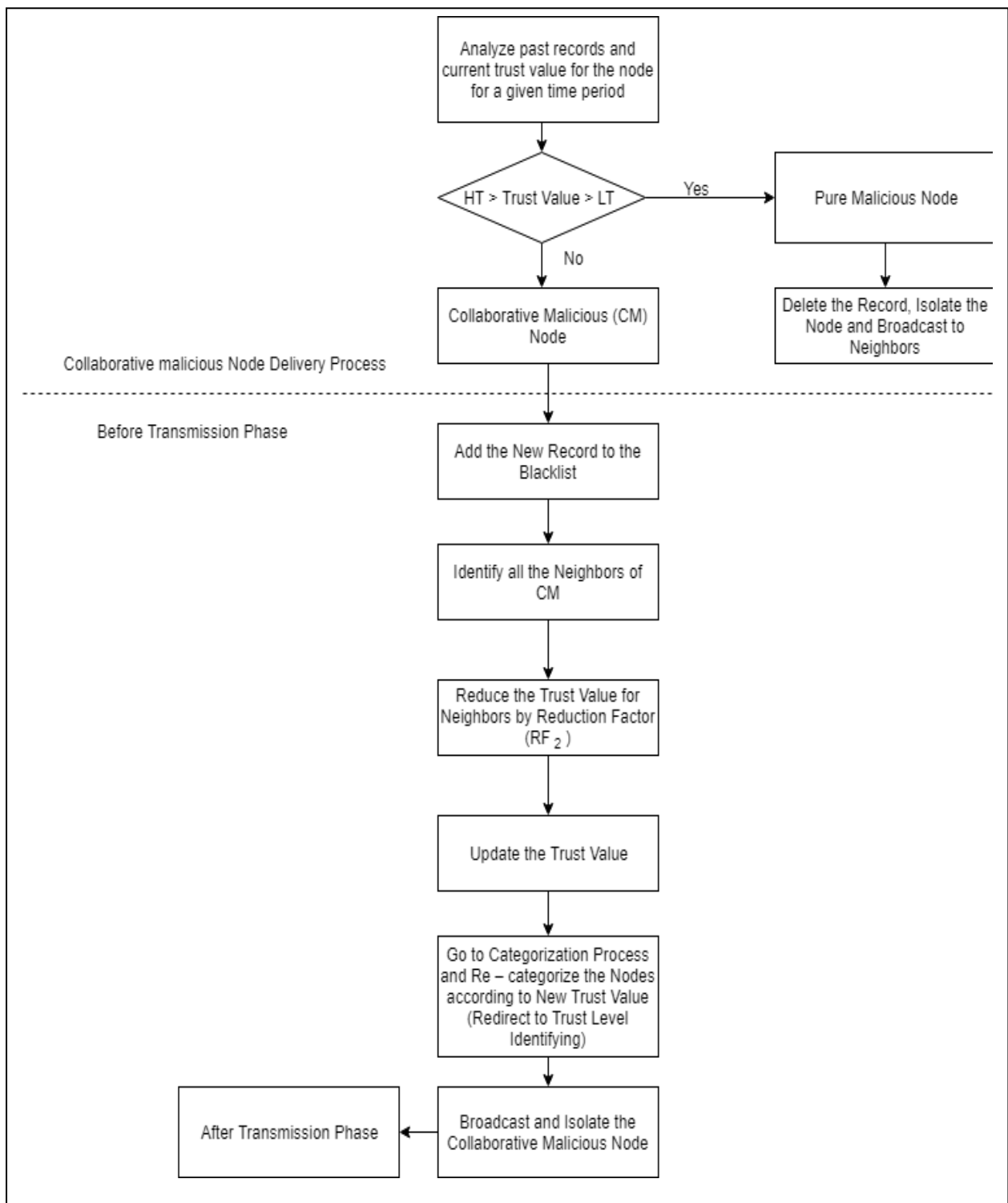


Figure 3.19: Spiral Model

Before Transmission Phase

Initially for every node of the network will consider as partially trusted nodes and when they do the communication according to the behaviour, it will update the TrV along with the trust level of the particular node. In this phase after identifying the collaborative malicious nodes that node should declare as a blacklisted node as shown in the Figure 3.19 by adding a new record to the blacklist which is available in the recommendation table temporarily. This blacklist field always contains a Boolean value and using that value it will declare as a blacklisted node. Before confirming the particular node as a collaborative malicious node evaluation node will not actively interact with the blacklisted node. After blacklisting the node, it should broadcast and notify the other neighbours about the blacklisted node.

When we are adding a new record to the recommendation table, we have to add the neighbour along with the recommending nodes which give the recommendation for the particular neighbour with their recommendation values. Here, for the blacklisted node we have to find all the recommending nodes who gave the higher recommendations. If the recommendation value is greater than the threshold value, those nodes will be forwarded to the penalty phase. This value can be changed according to the user specification same as the other threshold values.

Penalty Phase

Same as the trust level identification phase here also we reduce the TrV of the selecting recommending node by given value or a reduction factor. Reduction factor will be decided based on the maturity level or the reputation of the node. This process must apply to each and every selected recommending node and immediately after the penalty old TrV should update with the new value in the trust table. According to the updated TrV, those nodes should redirect to the trust level identification phase in order to re-categorize the nodes using the trust levels.

The node which is evaluating will wait to get another recommendation from one of its neighbour for the same blacklisted node as a collaborative malicious node to confirm it as a collaborative node. That process will discuss under the After-Transmission phase.

Procedure 4: collaborative malicious node discovery Algorithm (Spiral model)

- 1 Get highest TrV and lowest TrV for the node for a given time range and marked them as value boundary for outliers
- 2 Then compare current TrV is in between the range or not.
- 3 If the current TrV is in between the range, it's categorized as a pure malicious node.
- 4 Then it (the node who execute this) can delete that record from all of its tables and can broadcast message to aware others.
- 5 So that will terminate the pure malicious identified process.
- 6 If current TrV is not in between outliers it's categorized as collaborative malicious (CM) node.
- 7 Then it (the node who execute this) can edit its trust table blackList flag to true.
- 8 Identify all the neighbours of the identified CM node and reduce their TrV since they have given the incorrect recommendations.
- 9 Broadcast to the other nodes
- 10 Go to the Identifying_trust_levels algorithm again.

Algorithm 4: collaborative malicious node discovery Algorithm (Spiral model)

```
1  BEGIN
2      p_M =passed-in malicious node
3      IF TrV is not an outlier
4      THEN
5          Delete from trust table
6          Send Broadcast to delete node
7      ELSE
8          Mark the node as a blacklist in trust table
9  FOR each node which recommended p_M DO
10     Calculate reduce factor
11     Recalculate indirect trust
12     Update global trust
```

```

13 END FOR
14   Broadcast neighbours about p_M node
15   GOTO Identifying_trust_levels
16 END IF
17 END

```

After Transmission Phase

In this phase, each neighbouring node will receive the collaborative node detail through the broadcast, and it has to check its own neighbour set. For the simplicity of the process, we consider some notations to represent a different set of nodes as shown in the Figure 3.20.

Collaborative malicious node	CM1
Neighbour set	NE
Blacklisted node set	BL

Firstly, it will check whether the received collaborative malicious node CM1 is in its blacklisted node set BL and if $CM1 \in BL$ then it will delete the record from the blacklist and from every table. This is the place where the evaluating node confirms the collaborative node and take action for the discovered CM1. After deleting from the tables, CM1 will isolate from the network, and others also will notify through the broadcast.

Secondly, the node will look into the neighbouring node set, NE and if the CM1 is not in the BL but still it is in the neighbour set, $CM1 \in NE$ a new record will be added to the blacklist and declare CM1 as a blacklisted node. Finally, all neighbours of the CM1 will go through the penalty phase which we mentioned in the spiral model, and after all these actions it will broadcast the notification and isolate the CM1 from the network.

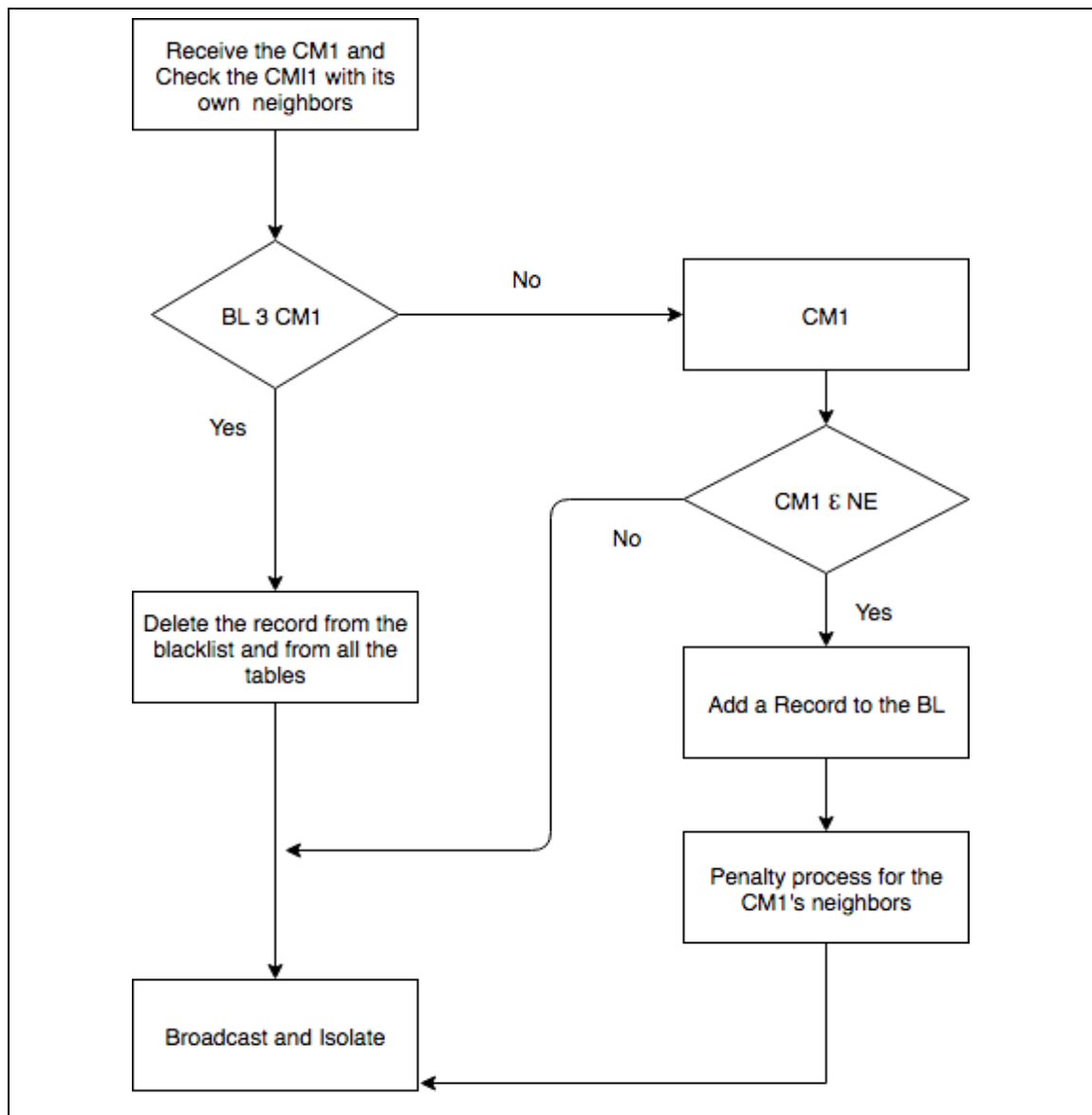


Figure 3.20: After Transmission Process

Procedure 5: After Transmission Process

- 1 read the received broadcast message and identify the p_CM
- 2 for every element $B_i \in \text{BlackListed nodes}$ do
- 3 if B_i equals p_CM then
- 4 delete p_CM in trust table
- 5 delete p_CM in recommendation table

```

6  break
7  end if
8  end for
9  for each neighbour node do
10 if p_CM found then
11 Mark as blacklist in trust table
12 Mark as blacklist in recommendation table
13 End if
14 End for
15 Send broadcast message about p_CM
16 For each node which recommended p_CM do
17 Calculate the reduction factor
18 Update the indirect trust in the trust table
19 End for
20 End

```

Algorithm 5: After Transmission Phase algorithm

```

1  BEGIN
2      FOR every element  $B_i \in \text{BlackListed nodes}$  do
3          IF p_CM equals  $B_i$  then
4              Delete p_CM in Trust Table
5              Delete p_CM in Recommendation Table
6              BREAK;
7          END IF
8      END FOR
9      FOR each neighbour node do
10         IF p_CM found then
11             Mark as blacklist in trust table
12             Mark as blacklist in recommendation table
13         END IF
14     END FOR

```

```

15 Send broadcast message about p_CM
16     FOR each node which recommended p_CM do
17         Calculate a reduction factor
18         Recalculate indirect trust in trust table
19     END FOR
20 END

```

3.2.7 Trust Calculation

Trust estimations in fixed networks are comparatively uncomplicated since the trust worth at this time resolves mostly caused by performance of the nodes in the network. Afterwards, by doing sufficient inspections, performances of the mentioned nodes can be predicted. Yet, in MANET trust estimations are requiring great effort.

There could be diverse forms of ability to move in MANETs for instance lesser ability to move (human walking with sensors) or excessive ability to move (mobility of sensors attached on a vehicle). The network configuration possibly will expressively modify with time in a random way as a result of this ability to move (mobility). Once the neighbouring node frequently changes, it turn out to be challenging to investigate and acquire sufficient occasions for collaborations to assess the trust. Data acknowledged from the MANET nodes are further admirable and truthful if they can be linked to where and when the appraisals initiated [28]. However, as soon as the position is continuously varying, it is tough to correlate the facts and node performance with situations.

In the lack of consolidated governor location, observing the actions of nodes in the network is challenging. The complication of trust calculations raises non-linearly lacking the consolidated knowledge center. The worst-case complication of attaining the trust worth on every single node through each additional nodes in the network of N associated neighbouring nodes is $O(N^2)$ [24].

3.2.7.1 Direct Trust Calculation

In place of point out before, at the start entirely the nodes are collaborating fine. When time passes, one of the nodes wishes to forward a packet to a specific endpoint. According to

suggested prototype, primarily, each and every node in the network flood the HELLO packets as an alternative of originating direction encounter procedure or inspecting their personal routing table designed for the preferred direction. As a result, that every single node guarantees it is single hop adjacent nodes eventually only single hop nodes answer back towards hello packets since those nodes are in equivalent radio range. After that, every single node can determine the number of nodes are continuing as single-hop neighbours. Once that, each node accomplishes the trust appraisal process on every single immediate node built on the given calculation 3.26.

$$T_{A_iA_j}(n) = \frac{[CP_{A_iA_j}(n) + DP_{A_iA_j}(n)]}{2} \begin{cases} i = 1 \\ n = 1,2,3.. \\ j = 1,2,3.. \end{cases} \quad (3.26)$$

Where A_i represents the appraising node and A_j represents appraised node by A_i . CP represents control packet (sending or replying proportion), and DP represents data packets sending proportion over time with n interactions count with the single-hop adjacent nodes.

In AODV the following control packets are used. In route discovery, route request (RREQ), route reply (RPLY) packets are used. Route error (RERR) and HELLO packets are used in route maintenance procedure. However, appraising trust, above-mentioned packets are correspondingly deliberate for the reason that they offer an essential involvement headed for the transmitting processes. However misbehaving nodes can also consume these types of packets, however, consume likelihood of these types of packets are comparatively low when comparing with well-behaving nodes. Hence the proportion of Control Packet transmitting (CP) or replying is estimated over the epoch of time-based on the calculation 3.26 with n interaction count with the single-hop immediate nodes.

$$CP_{A_iA_j}(n) = \frac{RREQ_{A_iA_j}(n) + RPLY_{A_iA_j}(n) + RERR_{A_iA_j}(n) + HELLO_{A_iA_j}(n)}{4} \begin{cases} i = 1 \\ n = 1,2,3.. \\ j = 1,2,3.. \end{cases} \quad (3.27)$$

The Data Packet (DP) transmitting a proportion of every single node is estimated based on the calculation 3.27.

$$DP_{A_i A_j}(n) = \frac{NDF_{A_i A_j}(n)}{NDR_{A_i A_j}(n)} \begin{cases} i = 1 \\ n = 1,2,3.. \\ j = 1,2,3.. \end{cases} \quad (3.28)$$

Where *NDF* denotes the number of data packets actually forwarded, and *NDR* denotes the number of packets truly acknowledged over time with *n* interaction count. After getting the CP and the DP, we can calculate the direct trust. Especially here we are not going to consider the Control packets and Data packets for only one transaction for calculating the direct trust, instead of that periodically within a given time period we have to get the summation of CP and DP and finally get the average TrV as the direct trust. It is given by,

$$DT = \frac{\sum_{i=1}^n (CP+DP)}{2n} \quad (3.29)$$

where DT is direct TrV within time duration T, CP is Control Packets for T time duration, DP is Data Packets for T time duration and *n* is number of interactions or transactions for T time duration. Likewise, every single node could determine the trust rate of all its single-hop neighbouring nodes and bring up to date its Trust table. Every single node can observe its adjacent nodes' communication manners through inactive acknowledgment indirectly.

3.2.7.2 Indirect Trust Calculation

Entropy-based indirect trust

In the suggested prototype, the trust circulations are designed straight from trust worth determined in (3.20). For concatenation trust circulation presented in Figure 3.14, node B detects the actions of C node then gives an endorsement to the node as $T_{BC} = T \{B: C, \text{action}\}$. In our context, this value is always will be the global TrV, GT_{BC} between B and C.

Node A faiths node B by $T \{A: B, \text{making recommendation}\} = R_{AB}$ and as the recommendation cost each time B sends the direct trust rate towards C, that is

$$R_{AB} = DT_{BC}$$

Since there can be more than one recommending nodes for the C, and according to the knowledge A gained throughout the time it will consider the relationship maturity value. Based

on Axiom 5 when it has a higher maturity value final recommendation value also get a higher value. Here it will mainly consider the interactions between the evaluating node A and the recommending node B, and for the B, the maturity level is given by,

$$ML_B = \frac{IN_{AB}}{IN_{AB} + IN_{AD}}$$

So, the final recommending value, R_{AB}' from the B is,

$$R_{AB}' = R_{AB} * ML_B$$

The problem is, a number of nodes or the neighbours count A should trust node C to accomplish the achievement. To justify second axiom, one approach to compute $T_{ABC} = T \{A: C, action\}$ is

$$T_{ABC} = R_{AB}' * GT_{BC} \quad (3.30)$$

Note that if node B does not have a clear idea regarding node C (i.e., $GT_{BC} = 0$) or if node A does not have a clear idea regarding node B (i.e., $R_{AB}' = 0$), the trust in the middle of, A and C is zero, i.e., $T_{ABC} = 0$.

Intended for multipath trust circulation, let $R_{AB} = T \{A: B, making\ recommendation\}$, $T_{BC} = T \{B: C, action\}$, $R_{AD} = T \{A: D, making\ recommendation\}$, $T_{DC} = T \{D: C, action\}$. Thus, A can inaugurate faith to C from end to end using two diverse routes A-B-C and A-D-C. In the direction of merging the trust inaugurated from end to end using diverse routes, here suggest practicing the highest proportion merging as

$$T\{A: C, action\} = w_1(R_{AB}T_{BC}) + w_2(R_{AD}T_{DC}) \quad (3.31)$$

According to our context, this should update and finally, the indirect trust between A and C, IDT_{AC} is represented as,

$$IDT_{AC} = w_1(R_{AB}' * GT_{BC}) + w_2(R_{AD}' * GT_{DC}) \quad (3.32)$$

Where

$$w_1 = \frac{R_{AB}'}{R_{AB}' + R_{AD}'} \quad (3.33)$$

$$\text{And, } w_2 = \frac{R_{AD}'}{R_{AB}' + R_{AD}'} \quad (3.34)$$

3.2.8 Mathematical Analysis of the Model

In order to assess our proposed model, ESTAODV, we make use of a simple network topology, which is displayed in the Figure.5.54, which involves 5 nodes. S and D indicate the source and destination nodes correspondingly and A, B and C are middle nodes. Over the period of time, according to our ESTAODV model source node S desires to communicate with the destination D.

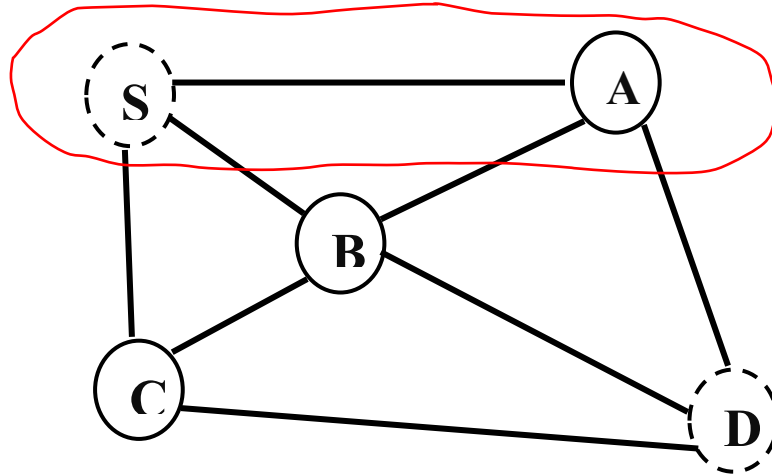


Figure 3.21: Example Network

At the start node S broadcast a *Hello* packet, so that node A, B and C can reply the *Hello* packet because they are one-hop neighbours. Then node S get to know that there are three one hop neighbours are staying in touch with. After discovering one-hop neighbours, it executes a trust calculation process based on 3.26.

$$T_{A_i A_j}(n_i) = \frac{[CP_{A_i A_j}(n_i) + DP_{A_i A_j}(n_i)]}{2}$$

Hence, we assume, the node S assesses a TrV of one of its immediate node A overtime with a number of collaborations say 3 so it is marked in the above figure. So the above calculation is

modified as,

$$T_{A_i A_j}(n_1) = \frac{[CP_{SA}(n_1) + DP_{SA}(n_1)]}{2} \text{ for } n = 1$$

$$T_{A_i A_j}(n_2) = \frac{[CP_{SA}(n_2) + DP_{SA}(n_2)]}{2} \text{ for } n = 2$$

$$T_{A_i A_j}(n_3) = \frac{[CP_{SA}(n_3) + DP_{SA}(n_3)]}{2} \text{ for } n = 3$$

from the above calculations, first, every node estimates the Control Packet (CP) forwarding and replying ratio for 3 collaborations based on the 3.27 so that,

$$CP_{SA}(n_1) = \frac{(RREQ_{SA}(n_1) + RPLY_{SA}(n_1) + RERR_{SA}(n_1) + HELLO_{SA}(n_1))}{4}$$

$$CP_{SA}(n_2) = \frac{(RREQ_{SA}(n_2) + RPLY_{SA}(n_2) + RERR_{SA}(n_2) + HELLO_{SA}(n_2))}{4}$$

$$CP_{SA}(n_3) = \frac{(RREQ_{SA}(n_3) + RPLY_{SA}(n_3) + RERR_{SA}(n_3) + HELLO_{SA}(n_3))}{4}$$

Assume $RREQ_{SA}(n_1) = 0.4$, $RPLY_{SA}(n_1) = 0.4$, $RERR_{SA}(n_1) = 0.0$ and $HELLO_{SA}(n_1) = 0.5$. Therefore,

$$CP_{SA}(n_1) = (0.4 + 0.4 + 0.0 + 0.5)/4 = 0.3$$

Likewise, assume $CP_{SA}(n_2) = 0.5$ and $CP_{SA}(n_3) = 0.8$

Then, Data Packet (DP) forwarding ratio based on the 3.28 with number of collaborations say 3. Therefore,

$$DP_{SA}(n_1) = \frac{NDF_{SA}(n_1)}{NDR_{SA}(n_1)}$$

$$DP_{SA}(n_2) = \frac{NDF_{SA}(n_2)}{NDR_{SA}(n_2)}$$

$$DP_{SA}(n_3) = \frac{NDF_{SA}(n_3)}{NDR_{SA}(n_3)}$$

Assume $NDF_{SA}(n_1) = 0.5$ and $NDR_{SA}(n_1) = 0.8$

Therefore,

$$DP_{SA}(n_1) = 0.5/0.8 = 0.6$$

Likewise, assume $DP_{SA}(n_2) = 0.5$ and $DP_{SA}(n_3) = 0.7$.

The Control Packet (CP) and Data Packet (DP) forwarding and responding ratio values are Substituted into 3.26. So that the solution will be,

$$\begin{aligned} T_{SA}(n_1) &= [CP_{SA}(n_1) + DP_{SA}(n_1)]/2 \\ &= (0.3+0.6)/2 \\ &= 0.5 \end{aligned}$$

$$\begin{aligned} T_{SA}(n_2) &= [CP_{SA}(n_2) + DP_{SA}(n_2)]/2 \\ &= (0.5+0.5)/2 \\ &= 0.5 \end{aligned}$$

$$\begin{aligned} T_{SA}(n_3) &= [CP_{SA}(n_3) + DP_{SA}(n_3)]/2 \\ &= (0.8+0.7)/2 \\ &= 0.8 \end{aligned}$$

After getting the TrVs for every transaction within a given time period T, it will take the average value as the direct TrV for the particular node by dividing the total TrV by the number of transactions N.

$$DT_{SA}(T) = \frac{(T_{SA}(n_1) + T_{SA}(n_2) + T_{SA}(n_3))}{N}$$

Therefore,

$$DT_{SA}(T) = \frac{0.5 + 0.5 + 0.8}{3} = 0.6$$

This is how we calculate direct trust for each and every interaction of one node A, and for other neighbours also we can calculate the direct trust same way and each time after calculating the TrV it should update with the previous value in every table.

Suppose the direct TrVs for A, B and C for the time T is as follows.

$$DT_{SA(T)} = 0.6$$

$$DT_{SB(T)} = 0.5$$

$$DT_{SC(T)} = 0.9$$

After calculating the direct trust for all the one hop neighbours, we have to get the recommendations from the neighbours of the one hop neighbours in order to calculate the indirect trust. When we calculate the indirect trust, neighbouring nodes will get recommendation request and all the neighbours should reply for the recommendation request and send the recommendation trust for the requested node S and for the simplicity of the model always this recommendation value will be the direct TrV between one hop neighbour and the recommending node. For an example if S need to get the recommendation about node A form the neighbours of the A node, all the neighbours of A should send the recommendation value to the S. According to our example topology neighbours of A is only B and as the recommendation trust it will send the direct trust between A and B. We can represent the relationship between these two values with the notation as follows,

$$R_{SB} = DT_{BA}$$

Where DT_{BA} is the direct trust that B is having about A and R_{SB} is recommendation value for A given by B. But S will not get the R_{SB} value as it is and this is the place where it considers the maturity level or the reputation value of the recommending node. Maturity level will be calculated based on the interactions that the particular node is having with the recommending node. The maturity level is given by,

$$ML_B = \frac{IN_{SB}}{IN_{SA} + IN_{SB} + IN_{SC}}$$

where ML_B is maturity level of B, IN_{SA} is the number of interactions between S and A, IN_{SB} is number of interactions between S and B and IN_{SC} is number of interactions between S and C within a given time period.

According to the maturity level of each and every node the recommendation value can be changed and after considering the maturity level and all the factors indirect TrV is given by,

$$IDT_{SA}(T) = w_l[(R_{SB} * ML_B) * GT_{BA}]$$

Where GT_{BA} is global trust, where B is having towards A node, and w_l is given by,

$$w_l = \frac{R_{SB}}{R_{SB}}$$

This w_l value here will always be 1, since the node B is the only recommending node that A is having for this interaction.

If we consider about node B path we have two recommending nodes A and C. Maturity levels of these two nodes are given as,

$$ML_A = \frac{IN_{SA}}{IN_{SA} + IN_{SB} + IN_{SC}}$$

$$ML_C = \frac{IN_{SC}}{IN_{SA} + IN_{SB} + IN_{SC}}$$

And recommendation values given by A and C nodes are R_{SA} and R_{SC} respectively. Weightage values of these two are given by,

$$w_1 = \frac{R_{SA}}{R_{SA} + R_{SC}}$$

$$w_2 = \frac{R_{SC}}{R_{SA} + R_{SC}}$$

Finally, the indirect trust of the node B can be given as follows.

$$IDT_{SB}(T) = w_1[(R_{SA} * ML_A) * GT_{AB}] + w_2[(R_{SC} * ML_C) * GT_{CB}]$$

Same way indirect trust of the node C is as follows.

$$IDT_{SC}(T) = w_1[(R_{SB} * ML_B) * GT_{BC}]$$

Then it will calculate the global TrV using direct and indirect TrVs calculated in the previous steps. According to our definition, always for the TrV, there should be a value in between 0 and 1. Normalization is the process we are going to use in order to get the final result for the global trust, which scales all numeric variables in the range [0, 1]. After taking the summation of the direct and indirect trust we are going to take the global trust using,

$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

where, X_{new} is the final global TrV for a particular neighbour node, X is the global TrV that we are getting using the below equations, where we both consider the direct trust and the indirect TrV, X_{min} is the minimal TrV that we can have for X and here that is 0 since we have to maintain the TrVs in-between 0 and 1 ($DT=0$ and $IDT=0$) and finally X_{max} is the maximum value that we can have for X and here that is 2 since the $DT=1$ and $IDT=1$.

$$GT_{SA}(T) = DT_{SA}(T) + IDT_{SA}(T)$$

$$GT_{SB}(T) = DT_{SB}(T) + IDT_{SB}(T)$$

$$GT_{SC}(T) = DT_{SC}(T) + IDT_{SC}(T)$$

For the purpose of demonstration, we will take the indirect TrVs as follows.

$$IDT_{SA}(T) = 0.7$$

$$IDT_{SB}(T) = 0.3$$

$$IDT_{SC}(T) = 0.8$$

Finally, as the global TrVs after normalization process,

$$GT_{SA}(T) = \frac{(0.6+0.7)-0}{2-0}$$

$$= 0.65$$

$$GT_{SB}(T) = \frac{(0.5+0.3)-0}{2-0}$$

$$= 0.4$$

$$GT_{SC}(T) = \frac{(0.9+0.8)-0}{2-0}$$

$$= 0.85$$

After that node S will look for each and every TrV of the neighbouring nodes and assign a trust level based on table 10. Here we assume the threshold values TH1=0.8, TH2=0.6 and TH3=0.4 TH4=0.2 so that, $GT_{SA}(T)$ = the trust level is partially trusted, $GT_{SB}(T)$ = the trust level is Selfish and $GT_{SC}(T)$ = the trust level is Trustworthy. According to the TrVs and the trust levels for the source S , in order to transmit the packet best path will be through the C since it has the highest TrV and it is the trust worthiest one-hop neighbour.

3.3 Deep Reinforcement Learning Approach

The utilization of reinforcement learning is set up a TrV using inbuilt protocol ESTAODV. The proposed reinforcement learning approach is consisting of major two tasks, which are, the MANET simulation for parameter extraction and the development of deep reinforcement learning agent in order to calculate the TrVs. Hence, the developed system RLTM (Reinforcement Learning Trust Manager) can be divided into two sets of algorithms which is Flow monitor for network and the trust prediction Reinforcement Learning (RL) model. The specified routing protocols are Ad-hoc On-demand Distance Vector (AODV) and mainly the above-stated Entropy-based Spiral Trust AODV (ESTAODV). Simply in the initial phase of the process, hyper-parameters extracted from the MANET is captured by Flow monitors and the information of each flow is passed to the RL Model as an array. Calculation of the Q-value is done by RL model, and the Flow monitor is informed by the evaluated Q-value in order to find the optimal route path by calculating the total discount reward and to identify whether a certain path is consisting of trustworthy nodes or reputed nodes or selfish nodes or malicious

nodes.

RL model mainly consists of a Recurrent Neural Network which has 3 LSTM (Long Short-Term Memory) layers and default Input Layer and the Dense Layer. To build the MANET environment for the parameter extraction, Flow monitor module is used which is an inbuilt set of algorithms within NS-3. The `wifi_flow_monitor.py` file is modified accordingly with the use of ESTAODV routing protocol and by creating the simulation network which has 49 nodes where it uses the random waypoint model as the mobility scenario. Using flow monitors of NS-3, a set of network parameters are extracted which regards to calculate a value for the trust in each route. The extracted hyper-parameters include the jitter sum, delay sum, transfer bytes, receive bytes, number of transfer packets, number of receive packets and the number of lost packets. With the use of evaluated values-sets of hyper-parameters, reward or the initial TrV is calculated using the reward function. The calculated reward is passed to the Recurrent Neural Network in the RL model, and with the use of Rectified Linear (RELU) and Linear activation function, it gives the Q-value per each flow and then to a given node. Back propagation is done in order to check whether the calculated results are accurate or whether it includes an error. Then based on the evaluated Q-values, total discount reward or the justified TrV is calculated with the aid of Bellman equation where it identifies the action as to be the next best hop or the optimal route path to be connected. Based on the given Q-values per each flow the model identifies the routes have maximum Q-value as the optimal route path which includes trustworthy nodes and reputed nodes and the routes having minimum Q-values as the untrustworthy routes which are consisting of malicious nodes. Based on the evaluated Q values ESTAODV routing protocol decide the Routing procedure where it acts as a Q-routing protocol.

3.3.1 Deep Reinforcement Learning Oriented Trust Protocol

When it comes to reinforcement learning (RL), there is no whichever exact response, nevertheless the RL agent has to choose in what way to perform in order to achieve some precise actions. In the lack of current training records, the RL agent acquires from knowledge. It collects training instances as a trial-and-error as it efforts to its duty, by means of the objective of exaggerating the long-term award. A learning agent has to predict the best route to reach the destination to maximize its utility from the rewards generated through each and every route in

the network. For example, if we need to find the shortest optimized path from source to destination, we can assign positive rewards and negative rewards for each action taken by the agent. Hence finally, according to the final reward gain from the routes, we can find the best route to transmit packets. Routing protocol of a network also much important when finding the optimized route in a network. In this chapter further discuss routing protocols, neural networks, how the reward function process and so more about the Q-learning routing protocols.

A distinctive component of using reinforcement learning is, it is important to learn from direct collaboration with the environment, instead of depending on supervision or finish models of the environment. The utilization of reinforcement learning in this implementation is to set up a TrV using inbuilt protocol ESTAODV. A learning agent needs to discover a Q value to select a path to amplify its utility (accepting each progression gives some reward along the way). With a specific end goal to obtain performance, the reward component should be painstakingly planned. For instance, the objective is to locate the secure path to the exit, at that point, it's easy to provide each progression negative reward while giving the exit vast positive reward. Then again, if the reward function is ineffectively composed to such an extent that the model consists of a series of positive rewards, at that point being caught permanently in the routing network will be very possible for the learning agent.

3.3.2 Architecture and Implementation

Implementation of the model uses a cluster-based approach for setting up the trust calculation. According to the ESTAODV protocol, every node calculates trust for each neighbour node as it was explained in chapter 3.2. Calculated TrV needed to be fed into the cluster nodes for feeding into the existing RL model (Deep Reinforcement model) which was explained in the previous section. Following figure 3.22 explains to the model how each TrV will be propagated to the central locations and how the Q value will be propagated back to each individual node.

All the cluster nodes will be processing RL model and decided the predicted Q values and it will update each and every regular node in the cluster. The cluster will be defined to reduce the overload generated from the additional control packets within the network.

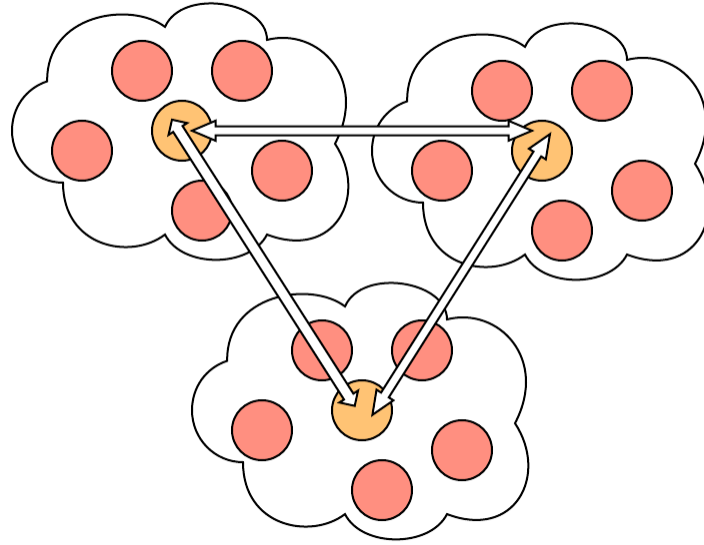


Figure 3.22: Cluster-based Deep Reinforcement Learning architecture

3.3.3 System Analysis

Deep Q-learning is normally explained as a model-free RL technique which defines a policy handling rewards at each state-action pair [75]. With regards to trustworthy routing, research should first characterize the all the actions and the states. In advance, forwarding packet can be effortlessly demonstrated as an MDP: every packet must travel through intermediate nodes to reach the goal for every flow, stochastic transmission brings about an unaltered result about the state. At this point, a state is simply the position of every single packet while the set of accomplishment is the conclusions of the immediate node for each node.

The learning agent is referred to as the decision maker for the next hop by considering the previous transmission results. If wireless nodes are examined as learning agents, at that point there is certainly not once more whichever state transformation requesting as soon as a packet is transmitted to the immediate node, it is transferred to additional learning agent. Toward relevantly apply Q-learning transmitting, every single node must be consolidated to frame different learning agents. Each learning agent is in charge of one precise objective, which carries roughly three-dimensional Q values:

$$Q: S \times A \times D \rightarrow R \quad (3.35)$$

Where D is the procedure of the endpoint. Along these lines, the relating Q estimation of node picks an activity a to accomplish precise task d can be specified as $Q_i(d, a)$. Meanwhile, in the transmitting concern, the action a and state s are linked by means of a certain node, for accessibility, here it specifies the immediate node of the particular node as a , as well as the current node as s . For instance, $Q_i(d, j)$ indicates the Q value for given hop j as the immediate node to accomplish the endpoint d .

Entirely the Q values are assigned for every single node as appeared in Figure. 3.23. In the meantime the state s is consistently the particular node, every single node just desires to keep the Q estimations of end point-action sets $D \times S$. Certainly, for the purpose of the proper space utilization, subsequently, the node can just accomplish its neighbouring nodes Adj_i , just $Q_i(d, a) \forall a \in Adj_i$.

$$Q'_i(d, a) = Q_i(d, a) + \alpha (R_a(i) + \gamma \max_j Q_a(d, j) - Q_i(d, a)) \quad (3.36)$$

Where $\max_j Q_a(d, j)$ is the best forthcoming worth if the immediate node a is nominated.

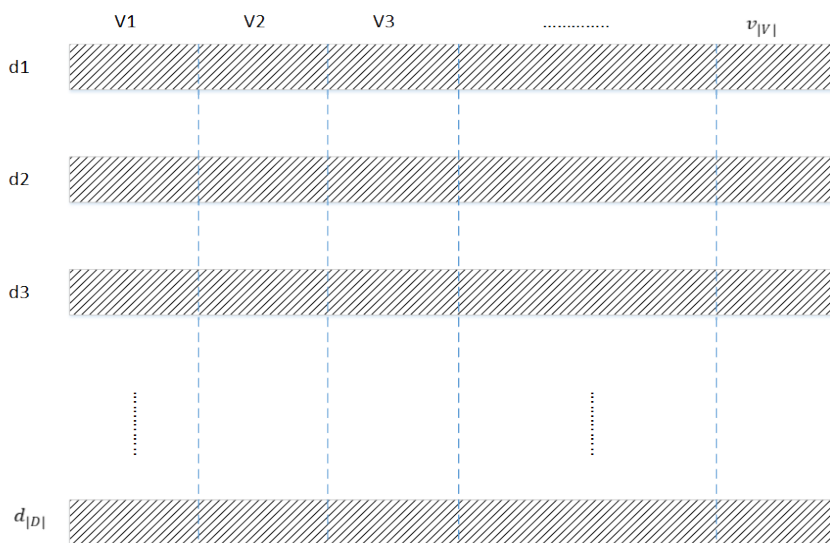


Figure 3.23: Relationship between learning agents and nodes

A learning agent for each destination is represented by every tuple. Entirely the learning agents are kept in every node in the network $\max_j Q_a(d, j)$ can only get back from node a .

One more imperative selection for the Q -learning tactic is the conclusion of the reward

operation $R_a(s)$. Diverse reward capacities demonstrate distinctive inclinations in the path determination process. It is hard to contend which compensate work is the best on the grounds that the last execution is generally in view of the application and the system characteristics. In any case, it is as yet conceivable to outline a reward function which is of more vigorous and adaptable relevance.

$R_a(s)$ value will be represented mainly through the generated TrV from ESTAODV protocol. Further, throughput and delay metrics will be added to the equations using different weights. Trust is an important indication of how good the link security will be: greater trust typically outcomes in more trustworthy and protected associations. Throughput is a significant QoS constraint demonstrating the proportion of effective information transfer, which is essential for real-time interactive program facilities. Interruption or the delay assesses the endwise latency of packet sending from the starting node to the endpoint. The extreme latency is probable to be affected by incompetent transmitting or uncompromising congestion. As a result, $R_a(s)$ is represented as:

$$R_a(s) = F(w_1.TRUST)F(w_2.Throughput).w_3.Delay \quad (3.37)$$

Where TRUST, Throughput, and Delay are the normalized feedback with their respective weights w_1 , w_2 and w_3 . F is a function used to convert values of Trust and Throughput to a variety of (0, 1]. There are certain fascinating facts regarding $R_a(s)$:

- Delay is looked upon in a different way since it is obsessive. For instance, by including the latency from node i to j then for the one from node j to k, we can realize the latency from i to k. As a result, those nodes perform as a discount factor to regularize latency.
- A usual preference of F is $F(x) = e^{-x}$ for the reason that the amount of an exponential function decays at a degree proportional to its present worth. Accordingly, a fast drop when Trust and Throughput are minor can be predictable due to the minus sign in front.
- Trust, Throughput, and Delay are completely transformed into [0, 1] by re-scaling:

The subsequent stage is to settle on the routing decision of the values of Q. That might give the

impression partway that the record of the value of Q with the greatest value has to be the value preferred as the following repetition. In any case, from the time once the value of Q must be restored if the action accomplished, a greedy approach preserves the investigation procedure from finding a superior path. A straight procedure is to regularize the value of Q through the ‘Boltzmann probability distribution’ to permit investigation [58]. The optimal tactic can be exemplified as

$$P(R_i(d) = a) = \frac{e^{\beta Q_i(s,a)}}{\sum_{j \in Adj_i} e^{\beta Q_i(s,j)}} \quad (3.38)$$

In the above equation, β is the controller of covetousness of selection procedure. When β goes to infinity, entry where it leads the highest Q value is selected as next hop. And also when β goes to zero, gets a random selection from the Q values.

3.3.4 System Architecture & the Process Summary

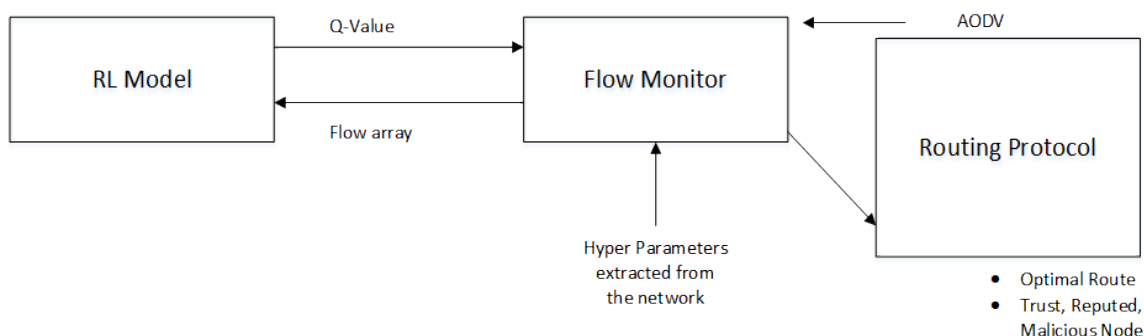


Figure 3.24: System Architecture

The proposed approach is consisting of major two tasks, which are, the MANET simulation for the purpose of parameter extraction and the development of deep reinforcement learning agent in order to calculate the predicted TrVs. Hence, the developed system RLTM (Reinforcement Learning Trust Manager) can be divided into two sets of algorithms which is Flow monitor for network simulation within NS-3 and the trust prediction Reinforcement Learning (RL) model. ESTAODV is used as the specific routing protocol. Represented in Figure 3.24, hyper-parameters including the TrV from ESTAODV extracted from the MANET is captured by the NS-3 Flow monitors and the information of each flow is passed to the RL Model as an array.

Calculation of the Q-value is done by RL model, and the Flow monitor is informed by the evaluated Q-value in order to find the optimal route by calculating the total discount reward and to identify whether a certain path is consisting of trustworthy nodes or reputed nodes or malicious nodes.

RL model mainly consists of a Recurrent Neural Network which has 3 LSTM (Long Short-Term Memory) layers and default Input Layer and the Dense Layer. To build the MANET environment for the parameter extraction, Flow monitor module is used which is an inbuilt set of algorithms within NS-3. The `wifi_flow_monitor.py` file is modified accordingly with the use of AODV routing protocol and by creating the simulation network which has up to 100 nodes where it uses the random waypoint model as the mobility scenario.

Figure 3.25 describes the step by step process of extracting the network parameters, evaluating the Q-value and the output predictions from Recurrent Neural Network. Using flow monitors of NS-3, a set of network parameters are extracted which regards to calculate a value for the trust in each route. The extracted hyper-parameters include the jitter sum, delay sum, transfer bytes, receive bytes, number of transfer packets, number of receive packets and number of lost packets (Delay and Throughput values) and TrV from ESTAODV protocol. With the use of evaluated values-sets of hyper-parameters, reward or the initial TrV is calculated using the reward function. The calculated reward is passed to the Recurrent Neural Network in the RL model, and with the use of Rectified Linear (RELU) and Linear activation function, it gives the Q-value per each flow and then to a given node. Back propagation is done in order to check whether the calculated results are accurate or whether it includes an error.

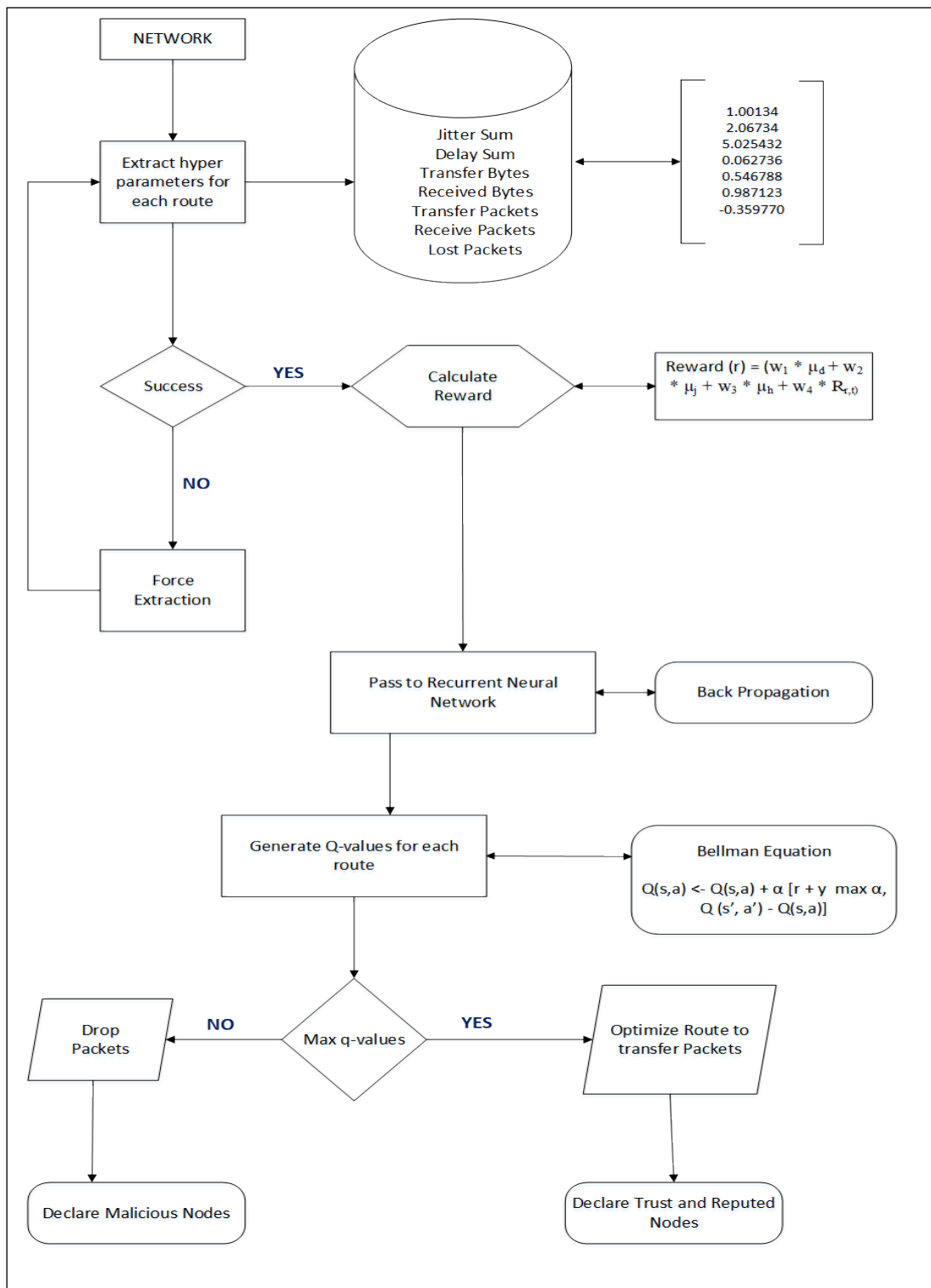


Figure 3.25: Flow of the process of generating output through RNN

Then based on the evaluated Q-values, total discount reward or the justified TrV is calculated with the aid of Bellman equation where it identifies the action as to be the next best hop or the optimal route path to be connected. Based on the given Q-values per each flow the model identifies the routes have maximum Q-value as the optimal route path, which includes trustworthy nodes and reputed nodes and the routes having minimum Q-values as the untrustworthy routes which are consisting of malicious nodes.

Following Algorithm 6 describes the process of predicting the action for the given set of normalized flow stats per each flow to the RL model. An epsilon value is set, and an epsilon-greedy method is used for exploration during training. What happens there is when an action is selected in the training process, it is chosen as either as the action with the highest Q-value or a random action. Deciding between these two is random, and it is based on the value of epsilon. The epsilon value is annealed during the training, such that lots of random actions are taken initially which is the exploration where it seeks the environment. But with the progress of the training, lots of actions are taken which are having max Q-values which is usually called as exploitation. During the testing also this epsilon-greedy method is used, but with a very low value of epsilon where it can have a strong bias towards exploitation over exploration which means it has a favour on choosing the action with the highest Q-value over a random action.

Hence, in RLTM when it comes to the action prediction until the epsilon time expire it is giving the random predictions based on the flow stats, and after the expiration of the epsilon time it passes the flow stats to the recurrent neural network and gives the real-time predictions by giving actions which are having highest Q-values or in our case the next best route path to be connected. Most of the time the epsilon procedure is adapted to minimize the possibility of over fitting during the evaluation process.

Algorithm 6: Predicting the action by RL model

- 1 **Function** (act)
- 2 **Initialize:**
- 3 $\epsilon = 0.01$
- 4 **Input :**

```

5      N : number of flows
6      flow id
7      Expireexpect
8  for n → 0 to N do
9      get flow stats by flow id
10     Txb : Transferred Bytes
11     Rxb : Received Bytes
12     Txp : Transferred Packets
13     Rxp : Received Packets
14     Lp : Lost Packets
15     μd : Mean delay
16     μj : Mean jitter
17  if (Expireexpect) < (epsilon) then
18      return random.act
19  end
20  if (Expireexpect) >= (epsilon) then
21      return actValues
22  end
23  Return optimal action

```

3.3.5 Reinforcement Learning and Q-Learning

As it was mentioned above, RLTM is a reinforcement learning based approach to enhance trust in wireless networks which is based on Q-learning and routing. In the routing context, initially, need to consider the states and action in the network. In our approach states are the hyper-parameters extracted for each route from the wireless network and action is the Q-value prediction for every route. In order to predict Q-values for each and every route research used Q-learning, and after finding the optimized route, to update the routing protocol ESTAODV protocol is chosen by us.

3.3.5.1 Machine Learning

Machine learning is a sort of artificial intelligence (AI) that enables programming applications

to end up noticeably more precise in foreseeing results without being expressly modified.

3.3.5.2 Reinforcement Learning

An objective of learning in view of association with the environment. Reinforcement Learning is said to be the expectation of artificial intelligence. Furthermore, it is properly said as much, the fact that the potential that Reinforcement Learning is immense. Reinforcement Learning will realize what to do and how to delineate to activities. The final product is to maximize the numerical reward value. The agent is not advised which move to make, yet rather should find which activity will yield the maximum reward. It learns through trial-and-error.

Reinforcement learning contains two parts which going to explain in this section. First part is about the environment that should be modified. The next part detailed about the agent that is in charge of learning. In the learning procedure, in the wake of accepting environment state (s_t), the learner will choose an action (a_t) for the given state and pass it to the environment. The present state (s_{t+1}) of environment will be modified by the action and raises the reward (r_{t+1}) for the impact. Next, the learner will update the action-value function for a new state (s_{t+1}) and (r_{t+1}), and pick an action for the new state as indicated by the action-value function. And afterwards, continue circling above advances. In the learning procedure, the impact of each action is the premise of conjoined choices. Through the learning encounters, the learner can choose the optimal action for various state, to reach the end goal.

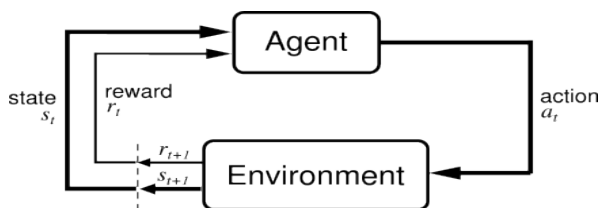


Figure 3.26: Reinforcement learning module

3.3.5.3 Deep Reinforcement Learning

DRL can be used in deep networks to represent policy, value function, model and to optimize these policies value function and model end-to-end using stochastic gradient descent. DRL also can be used in the field of developing games such as Atari, poker, Go, etc., to explore 3D worlds, to control physical systems like swim, walk, and also to interact with users.

3.3.5.4 MDP

Markov Decision Process (MDP), which is a practice that has quantified transition probabilities state wise. MDPs consist of:

1. A finite collection of states. These are the conceivable situations of the particular network.
2. An identified collection of activities or actions which are reachable for every single state. {left, right, up, down, movements}
3. Transitions between states. These can be a collection of likelihoods that association to over and above single conceivable state.
4. Rewards associated with each transition.
5. A discount factor γ in the middle of 0 and 1. This measures the dissimilarity of significance, in the middle of instant rewards and upcoming rewards.
6. Memorylessness. In other words, “the future is independent of the past given the present.”

Markov decision is one of the applications used in the reinforcement learning, and it shows a mathematical framework to take the decisions.

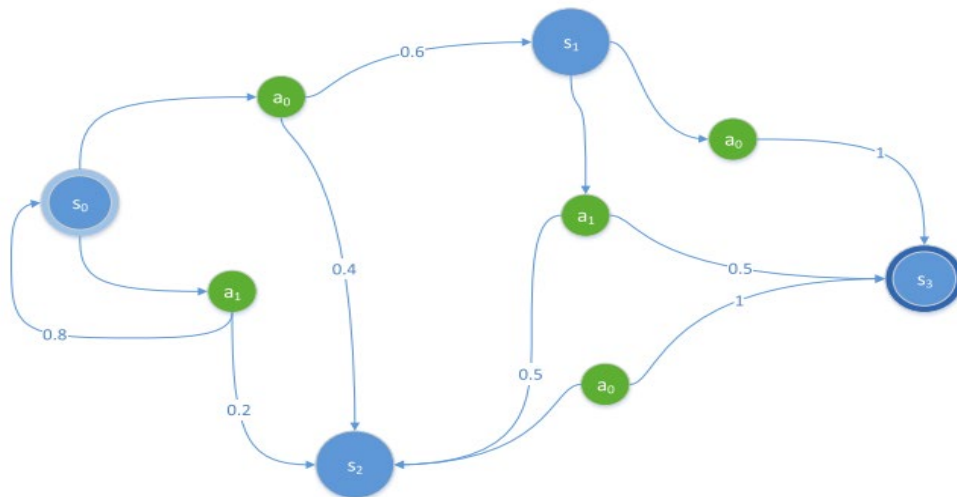


Figure 3.27: MDP State Model

There are 4 tuples in the Markov decision scenario as $(S, A, Pa(s, s'), Ra(s, s'))$. In here S and

A are State and Action and $P_a(s, s')$ which is the probability of the transmission of states change to the s' when confirmed action A and $R_a(s, s')$ is the reward that is given for that action 'a.'

In RL, the value of Q is generated to identify the excellence of a precise accomplishment with respect to the precise state. To update the Q value, the reward is needed.

What is basically happened in RL model in this research is first an agent is created, and it is the one who is doing all the action and replay functions. The neural network is created, and then as the state, all of the 3 hyper-parameters are taken from the flow monitor and by doing the action Q value is generated. To do that first of all need to explore the environment.

3.3.5.5 Exploitation and exploration

When focusing on the actions first of all need to explore the environment. There are two main phases of Reinforcement Learning which are in exploitation and exploration.

Why explore?

To learn how to deal with an agent to identify all the possible ways in the environment. In supervised learning, there is a supervisor who can do and manage all the actions. But in reinforcement learning environment problem can access the environment only through its own actions. To learn about good policy agent needs to access right experience. When selecting the action, there are so many approaches that can be taken in reinforcement learning.

3.3.5.6 Greedy Policy

In reinforcement learning algorithms, all the reward values are always tried to increase with the time. In this greedy policy, they always choose the highest value from the action set. As an example, for the exploitation, we can take giving a best Q value to the best action in the current moment. That means the agent is exploiting the current state to get the reward based on the action.

In below figure 3.29, each value is taken from the highest value of Q values.

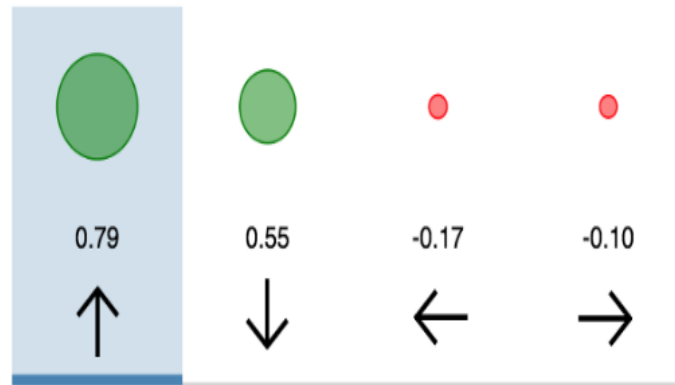


Figure 3.28: Highest Q Values

3.3.5.7 Reward Function

Considering the extracted hyper-parameters from the simulation of the network a reward function was developed with the aggregated weights.

$$R_a(s) = F(w_1.TRUST)F(w_2.Throughput).w_3.Delay \quad (3.39)$$

In the above reward equation (3.39), Delay is Mean delay which is mean of the summing up all end-to-end delays of all received packets of a given route path, and Mean jitter which is mean of the sum of all end-to-end delay jitter (delay variation) values for all received packets of the route path, Throughput is the $R_{r,t}$ is the Transfer ratio which is the Ratio between received packets and transfer packets and L_p is the number of lost packets. Weights are set where the summation of them to be equal to 1, where $w_1=0.5$, $w_2=0.3$, $w_3=0.2$. The weights have been set according to the priority for the network convergence. Following Algorithm 7 shows the reward calculation using these weights and parameters.

Algorithm 7: Algorithm for calculating the reward

- 1 **Function** (reward)
- 2 **Input :**
- 3 μ_d : Mean delay
- 4 μ_t : Mean Trust

- 5 μ_{Tr} : Mean Throughput
- 6 Reward = $F(w_1 * \mu_t) F(w_2 * \mu_{Tr}) (w_3 * \mu_d)$
- 7 **Return** maximum reward

This calculated reward value, action, state, next state has to be stored in the memory array to view in the learning phase to get the optimized results.

Algorithm 8: Storing the reward by index

- 1 **Function** (remember)
- 2 **Initialize:**
- 3 memory_array[]
- 4 **Input :**
- 5 s: state
- 6 a: action
- 7 r_m: maximum reward
- 8 s_n : next state
- 9 memory_array.append \rightarrow (s, a, r_m, s_n)
- 10 **return** memory_array

3.3.5.8 Recurrent Neural Networks

The central element of a Recurrent Neural Network (RNN) is that the system contains no less than one criticism association, so the enactments can flow round in a loop. That empowers the systems to do temporal classification and learn arrangements. As an example perform grouping acknowledgement/proliferation or fleeting affiliation/expectation. RNN structures can have a wide range of structures. One regular sort comprises of a standard Multi-Layer Perceptron (MLP) in addition to included loops. These can abuse the effective non-straight mapping capacities of the MLP, and furthermore, have some type of memory. Others have more uniform structures, conceivably with each neuron associated with all the others, and may likewise have stochastic actuation capacities.

RNNs are capable, in light of the fact that they join two properties. Circulated shrouded layer enables them to store a considerable measure of data about the past effectively, a Non-direct flow that permits to refresh their concealed state in muddled ways. Repetitive neural systems (RNNs) can specifically pass data crosswise over succession steps while handling consecutive information one component at any given moment. For our network implementation, we used RNN as it requires much less training data to reach the same level of performance as other models and also because of the long short-term memory (LSTM) layers it can remember wanted data, and it forgets unwanted data from memory, and it avoids the memory overflow. Hence, the memory and performance increases.

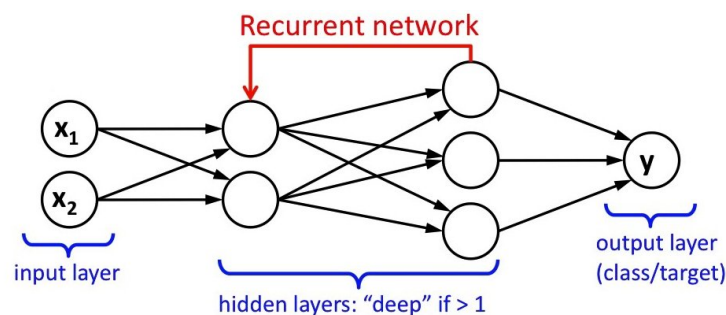


Figure 3.29: Recurrent Neural Network

3.3.5.9 Long Short-Term Memory (LSTM)

Basically, an RNN is a neural network which is having loops where those loops confess the neural network to access and use the information from previous layers, which is acting as the memory. Size of the memory in the network will be defined on several factors. If memory is degraded most of the information will be less usable. LSTM can be considered as a special classification of RNN. The distinction is that LSTM can effectively keep up self-associating loops without them corrupting and it is refined through a to some degree favour actuation, including an extra "memory" output for the self-looping association. Then the network is trained to select what data should be selected. Since the network is trained explicitly to select what to remember, there is no issue of old information getting destroyed. Further, the vanishing gradient does not affect the information that is decided to keep.

An LSTM block is consisting of four components which are;

- Input gate

- Output gate
- Cell/Remember gate
- Forget gate

LSTM itself is a recurrent neural network, because of having recurrent connections. Each of the three gates could be considered as a conventional artificial neuron as in a multilayer neural network, that it computes the weighted sum, q-values using an activation function. Following Figure 3.31 represents the basic formation of Long Short-Term Memory (LSTM) block.

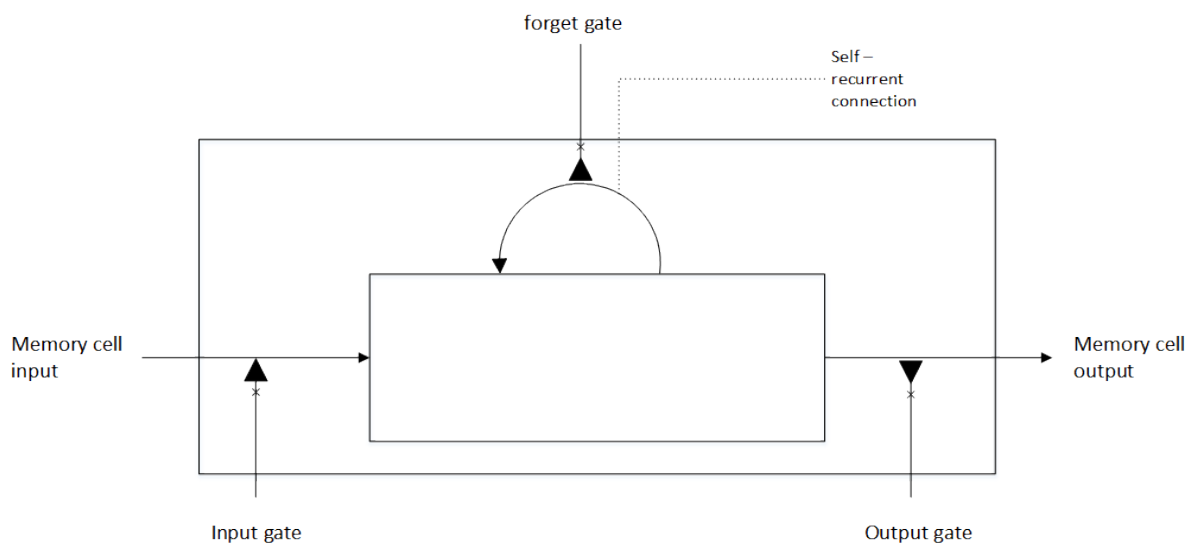


Figure 3.30: Basic formation of an LSTM block

Choosing the better structure of the RNN depends on various factors such as the number of nodes needed for the problem, the amount of feasible data, and how far back that we want our network's memory to reach. Since LSTM is the better choice with that given infinite data, computing speed, and has sequential processing with a given time series, LSTM layers are used within the proposed RLTM approach where it dealt with an infinite flow of a set of data with a time sequence and needed with the dynamic evaluation and updates.

3.3.5.10 Implementation of the Recurrent Neural Network

The RLTM consists of, 3 LSTM layers and default input layer or the embedded layer and dense layer in the implemented Recurrent Neural Network (RNN). Rectified linear (Relu) activation function is used as the activation function within the LSTM layers. Generally, in

deep learning networks rectified linear units (ReLUs) are used for the hidden layers. To output, the justified Q-value Linear Activation function is used within the dense layer. Adam function is used as the optimization function, and Dropout mechanism is used within the model in order to reduce the model over fitting. Stochastic Gradient Descent (SGD) is used to resolve the issues occurred due to batch optimization process where it helps to reduce the computational cost and improve the processing speed.

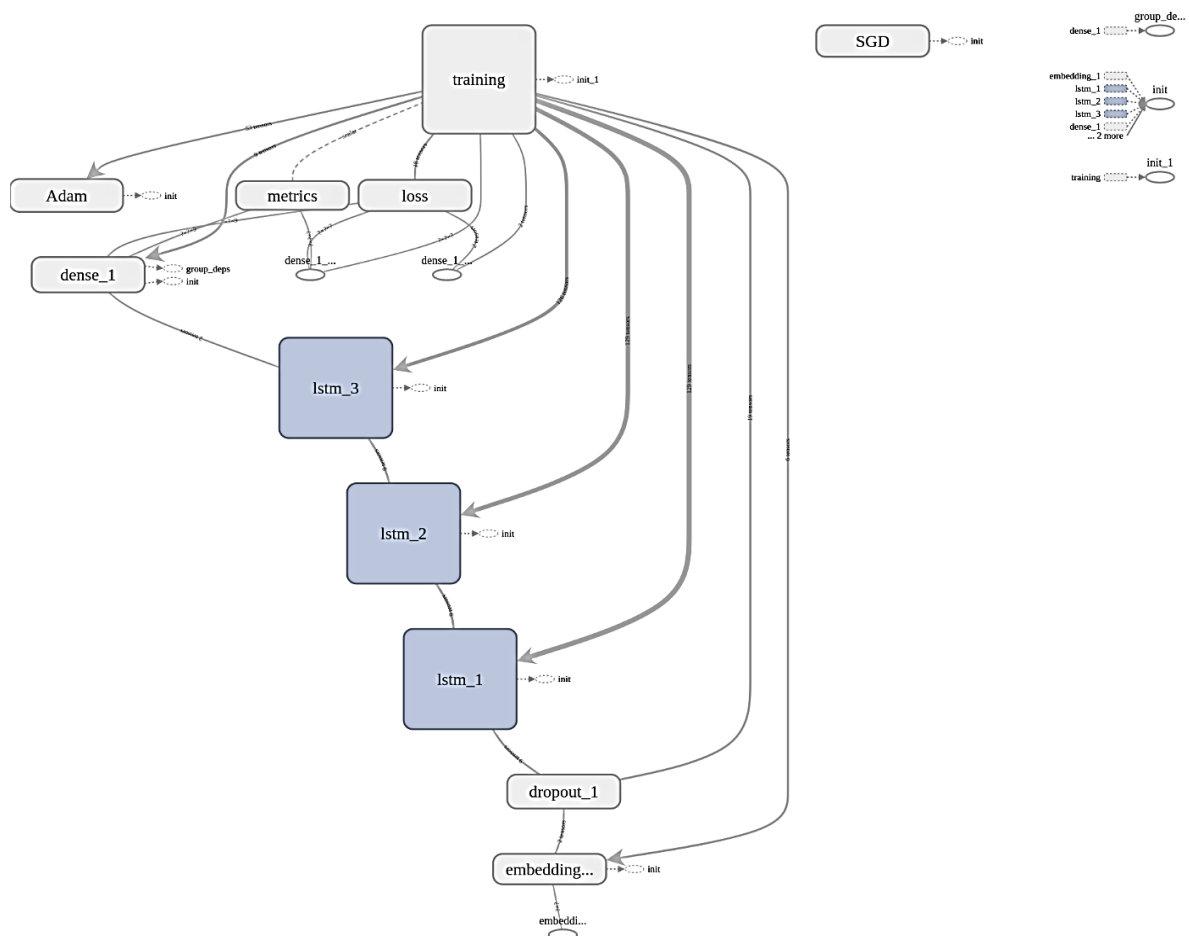


Figure 3.31: Structure of the developed Recurrent Neural Network

And there is a model called the build module in the RL model, it is not about reinforcement learning, directly about the implementation of the neural network. There are three hidden LSTM layers apart from the input layer and the output layer. There is no any limitation on the amount of the layers. But if it is less than 2 or 3 then it is good. That means when the number

of layers is increased it will affect the complexity of the network as well. LSTM layer can store previous states in memory cells. The speciality of the LSTM layer it has two gates called forget gate and remembers gate. Remember, the gate stores the previous states, and it decides what needs to be read into memory and what needs to be removed. Forget gate removes all the unnecessary states. Otherwise, all the states are stored and eventually memory can be full and overflow. That's why a recurrent network is used to this. Dense layer is the final output layer, and it predicts the Q value. The Linear activation function is used to give real value. Normally, Softmax is used as the output layer. Softmax defines prediction categories by applying probabilities to them. As an example, if two classes are given as zero and one, either zero or one is predicted based on probability. In Dense layer, linear activation is used to get the Q value. When indexes of IP addresses are input, indexes are learned one by one, and a Q value is given to indexes. That is the time when IP addresses input as an array. Then the Q value is given to each index in that array. From those Q values, the highest Q value is taken as the next point. For that, the greedy policy is used. Using the greedy policy, the highest value is taken.

3.3.5.11 Activation Function

What an activation function does: it simply put an artificial neuron to calculate a weighted sum of input, adds bias and then decide whether it should be fired or not. A rectified linear unit has output 0 if the input is less than 0, and raw output otherwise. That is, if the input is greater than 0, the output is equal to the input. And also, it makes efficient gradient propagation, no vanishing or exploding gradient problems.

Following Figure 3.33 shows the comparison of activation functions and Relu activation function makes the best performance.

There are two main benefits of using Relu:

ReLU is computationally much simpler. The forward and backward passes through a ReLU are both just a simple if statement. When comparing to the sigmoid activation, which requires computing an exponent. This advantage is huge when dealing with big networks with many neurons, and can significantly reduce both training and evaluation times.

Sigmoid activations are easier to saturate. ReLUs only saturate when the input is less than

0. And even this saturation can be eliminated by using leaky ReLUs. For very deep networks, saturation hampers learning, and hence ReLUs provide a nice workaround.

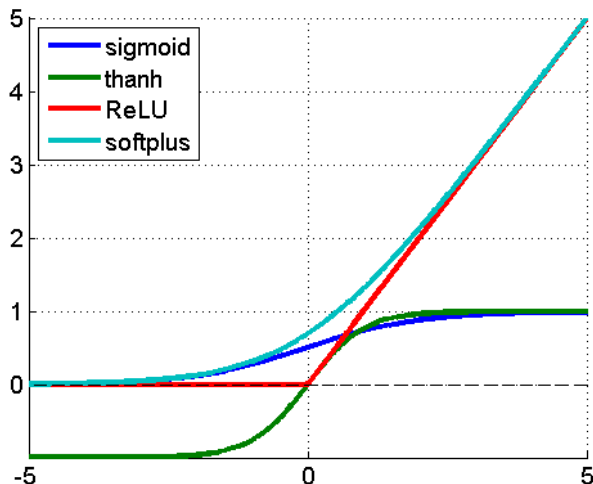


Figure 3.32: Comparison of Activation Functions

3.3.5.12 Q-Value Prediction

This prototype or the model has a function Q that receipts as an input individual state and individual action and rebounds the predictable reward of aforementioned action at that state. In advance prototype discover the surroundings, Q contributes the identical (arbitrary) static cost. However at that point, as prototype discover the surroundings further, Q contributes an enhanced estimate of the action value ‘ a ’ at a state s . Here, the prototype modernizes function Q with the time passes. In the below-explained calculation demonstrates the way of updating the Q value built on the reward from this context.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma * \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (3.40)$$

Q value is specified as the comprehensive summary of rewards once choosing action Q besides the ideal actions subsequently. When the time to take an action at state S_t , that should update the $Q(s_t, a_t)$ value through including a term. This encloses:

- Learning rate alpha: is how quickly a network abandons old beliefs for new ones. A value used by the *learning* algorithm to determine how quickly the weights are adjusted. When

alpha is near to 0, we are not modernizing very quickly. When alpha is near to 1, which should basically exchange the past value with the modernized value.

- The reward: is the reward we acquired through taking action at a_t state s_t . So the reward should be added to the old approximation.
- Assessed upcoming reward, which is the extreme attainable reward Q for entirely existing actions at x_{t+1} .

Lastly, it should deduct the past Q value to verify that we are only incrementing or decrementing by way of the modification in the approximation.

Algorithm 9: Predicting the action by RL model

```
1  Function (act)
2  Initialize:
3  epsilon = 0.01
4  Input:
5      N: number of flows
6      flow id
7      Expireexpect
8  for n → 0 to N do
9      get flow stats by flow id
10     Txb: Transferred Bytes
11     Rxb: Received Bytes
12     Txp: Transferred Packets
13     Rxp: Received Packets
14     Lp: Lost Packets
15     μd: Mean delay
16     μj: Mean jitter
17 if (Expireexpect) < (epsilon) then
18     return random.act
19 end
```

```

20 if (Expireexpect) >= (epsilon) then
21     return actValues
22 end
23 Return optimal action

```

Now that we have a value estimate for each state-action pair, we can select which action to take according to our action-selection strategy; we do not necessarily just choose the action that leads to the most expected reward every time.

Algorithm 10: Q-value calculation with experienced replay

```

1  Function (replay)
2  Initialize:
3      Replay memory → capacity N
4      Action-value Q function → random weights
5      Activation function → Rectified linear
6      M: number of episodes
7      epsilon: 0.01
8  for episode → 1 to M do
9  Initialize:
10     sequence  $s_1$ 
11     preprocessed sequence  $s_0$ 
12     t: time sequence
13 for t → 1 to T do
14     if (epsilon) < 0.01 then
15         return random_action( $a_t$ )
16     end
17     if (epsilon) >= 0.01 then
18         return maxQ_action( $a_{max}$ ), state( $s_t$ )
19     end
20     Execute random_action( $a_t$ ) observe reward( $r_t$ )
21     Set state( $s_{t+1}$ ) = ( $s_t$ ), ( $a_t$ ) and preprocess

```

```

22 Store transition in replay memory → Random sample transition minibatch
23 for state, action, reward, next_state, done in minibatch
24     target → reward
25     if not done
26         Target = reward + gamma * max(predict(next_state))
27         if (Target < 0.307 and Target > 0.303) then
28             Path consist of trustworthy nodes
29         end
30         if (Target < 0.302 and Target > 0.300) then
31             Path consist of reputed nodes
32         end
33         if (Target < 0.290 and Target > 0.270) then
34             Path consists of malicious nodes
35         end
36     end
37 Return optimal routes

```

The Q-value will be generated through the Rectified linear (RELU) function which is the activation function of the LSTM layers. No. of episodes will be run, and after the completion of training of the RL agent, the predictions will be given for the trained dataset. The calculated Q-value will be updated by sending through the Bellman equation in order to get the optimal Q-value by running several epochs. Finally, the decisions will be made according to the predicted values regarding a given state and an action where the RL agent can identify the optimal routing path which is consists of trustworthy nodes or reputed nodes or the malicious nodes.

3.3.6 Comparison with other RL approaches

Reinforcement learning can be identified as learning by trial-and-error by the agent to predict actions for the given states. There is various kind of reinforcement learning algorithms available in this field, but RLTM research is focused on the category of temporal difference (TD) learning as the environment is frequently changing. Other reinforcement learning

algorithms like Monte Carlo and dynamic programming (DP) are not suited for RLTM since they need a model of the environment, and they must wait until an episode or task to be completed. In RLTM, the agent will learn to work in environments like military, disaster environment and networks which are interconnected with nodes to enhance the trust in those networks in order to protect lives, properties, and resources in those environments.

3.3.6.1 Monte Carlo Approach

When the environment route replies with a negative reward for action by selecting a route with a malicious node to forward the packets to the given destination, then the agent will recognize the set of actions as a negative response. This process is named as bootstrapping. Hence, in such situation, non-bootstrapping methods used in Monte Carlo mechanism would slow down the learning process of the agent. So in RLTM when enhancing the trust in MANETs needed to be highly efficient, therefore the Monte Carlo approach based reinforcement learning algorithms are not suitable for our research work [34].

3.3.6.2 Temporal difference (TD) learning and SARSA learning

When performing reinforcement learning need to concern about policy (π) which is a function that takes the current environment state to return action.

$$\pi(s): S \rightarrow A \quad (3.41)$$

These consist of two different environments to be considered as the deterministic environment and stochastic environment, in the deterministic environment both state transition model and reward model are deterministic functions. If the agent in a given state repeats a given action, it will always go the same next state and receive the same reward value. In a stochastic environment, when the agent repeats doing the same action in a given state, the new state and received reward may not be the same each time. Therefore, in our research work, it leads to a stochastic environment. A policy in a stochastic environment is used to select an action to a given state. So the π^* which returns the optimal policy can be generated from the total discount reward when finding the path with maximum total reward from the initial state. The goal of the agent is to choose the best policy that will maximize the total reward from the environment.

Hence, to generate total reward let's assume that the environment is initially at state s_0 . Agent observes the environment state s_0 and chooses an action a_0 , then perform its action, environment state becomes s_1 , and the agent receives a reward r_1 .

$$\text{Total Reward} = r_1 + r_2 + r_3 + r_4 + \dots \quad (3.42)$$

However, it is common to use a discount factor to give higher weight to near rewards received near than rewards received further in the future.

$$\text{Total Discount Reward} = \sum_{i=1}^T \gamma^{i-1} r_i \quad (3.43)$$

Hence, after calculating the optimal policy, this can be applied to an on-policy method which is a temporal difference method which attempts to find the optimal policy and follow this policy while exploring the environment. One of the most popular example in temporal difference learning algorithm is SARSA which stands for State-Action-Reward-State-Action. SARSA learning also looks mostly same as the Q-learning, in Q-learning we update our Q-function by assuming taking action a that maximizes post-state Q-function $Q(s_{t+1}, a)$. But in SARSA, we use the same policy, epsilon-greedy that generated the previous action a_t to generate the next action a_{t+1} which we run through our Q-function for updates, $Q(s_{t+1}, a_{t+1})$ [34].

3.3.6.3 SARSA learning Vs. Q-learning

The major change within Q-learning and SARSA is that Q-learning is off-policy, and SARSA is on-policy. The equations below show the updated equation for Q-learning and SARSA:

$$\text{Q-learning: } Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma * \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (3.44)$$

$$\text{SARSA: } Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (3.45)$$

They look mostly the same except that in Q-learning, we update our Q-function by assuming we are taking action a that maximizes our post-state Q Function $Q(s_{t+1}, a)$.

In SARSA, we use the same policy (epsilon-greedy) that generated the previous action a_t to generate the next action, a_{t+1} which we run through our Q-function for updates, $Q(s_{t+1}, a_{t+1})$. Intuitively, SARSA is on-policy because we use the same policy to generate the current

action a_t and the next action a_{t+1} . We then evaluate our policy's action selection and improve upon it by improving the Q-function estimates. For Q-learning, we have no constraint on how the next action is selected, only that we have this “optimistic” view that all hence-forth action selections from every state should be optimal.

Algorithm 11: comparison procedure of q learning and Sarsa algorithm

- 1 **Initialize** $Q(s, a)$ arbitrarily
- 2 **Repeat** (for each episode):
- 3 Initialize s
- 4 Choose a from s using policy derived from Q
- 5 (e.g., ϵ - greedy)
- 6 Repeat (for each step of episode):
- 7 Take action a , observe r, s'
- 8 Choose a' from s' using policy derived from Q
- 9 (e.g., ϵ - greedy)
- 10 $Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma Q(s', a') - Q(s, a)]$
- 11 $S \leftarrow s'; a \leftarrow a'$;
- 12 **until** s is terminal

To identify next state and action, there are two action selection steps. The parameters α and γ have the same meaning as in q learning.

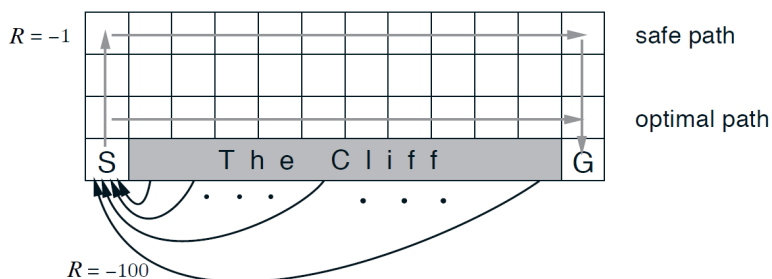


Figure 3.33: Cliff world example

Cliff world is used to describe the difference between Sarsa and q learning.

There is a small grid in the world and the goal is G on the lower right-hand corner. S is the start which is in the left-hand corner. When moving out from the cliff, it will give negative 100 and when in top row it will give negative 1. Along with edges q learning learn the best path and due to greedy action, falls off.

The best path is learned by Sarsa with the top row using action selection method. Hence it receives the highest reward than q learning. Below graph shows the rewarded episode for both Sarsa and Q learning.

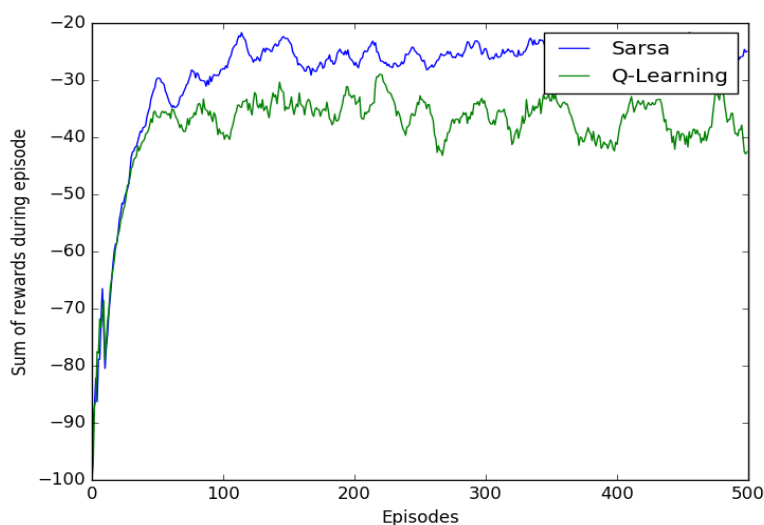


Figure 3.34: Reward episode

3.3.7 Reinforcement Learning Framework

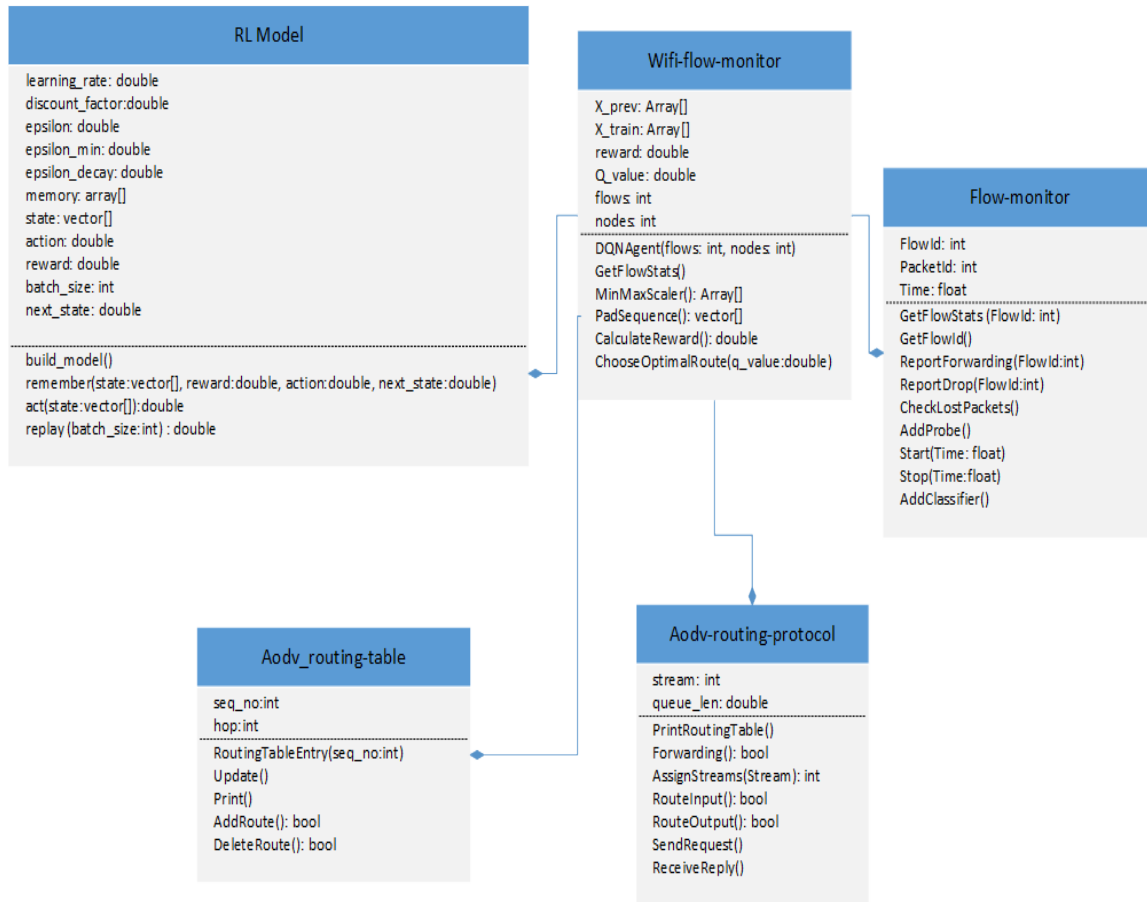


Figure 3.35: RLTM Framework class diagram

The agent module was developed with the content of Reinforcement Learning when implementing RLTM, and class diagram in Figure 3.36 represents the classes used when implementing the RLTM.

In the RL model class, it performs the building of recurrent neural network model and using the hidden layers in the network model, generates Q_values for the given states for each and every route as described in above sections. In the flow-monitor class, it generates transfer bytes, received bytes, transferred packets, received packets, jitter sum, delay sum and lost packets for every route and those parameters extracted are passed to the Wi-Fi-flow-monitor class to generate reward value using reward function implemented.

Then after the q_value predictions, the aodv-routing-table class is updated with the q_value which assign to each route and finally using aodv-routing-protocol class send and receive the

data packets through trusted routes in the network.

3.3.8 ACT (Actor-Critic Trust) Approach

For the creation of the algorithm, research used the Actor-Critic Trust (ACT) model and the Q learning concepts. Following are some of the parameters we identified that we would be using for the purpose of developing the algorithm.

- No of connections
- No of transactions
- No of packets forwarded
- Packet delay
- No of successful transactions and acknowledgements
- Availability
- Reputation
- Recommendations from other nodes

As an example, following Figure 3.37 will illustrate a basic idea of an evaluation method for whether a route path is trustworthy or not by using ACT model.

- Values for the direct trust and indirect trust is evaluated separately in order to calculate the total or the overall TrV.
- Direct trust calculation is done based on the factors including trust metrics and other node properties except using the interaction based properties only.
- Indirect trust calculation is done considering the recommendations by neighbouring nodes.
- According to the principles of reinforcement learning a reward, the function is generated corresponding to a node in order to learn the credibility of neighbouring or witness node.
- Determine a value function for learning the weights for direct and indirect trust.
- Update witnesses' credibility and weights for trust.
- Aggregate testimonies from neighbour nodes or witnesses with the highest witnesses' credibility value.

- Find total TrV by combining the direct and indirect TrVs.
- Selecting the route path having the highest total TrV as a trustee.
- Receive the interaction outcome again regarding the selected trust path, and continue with the process iteratively.

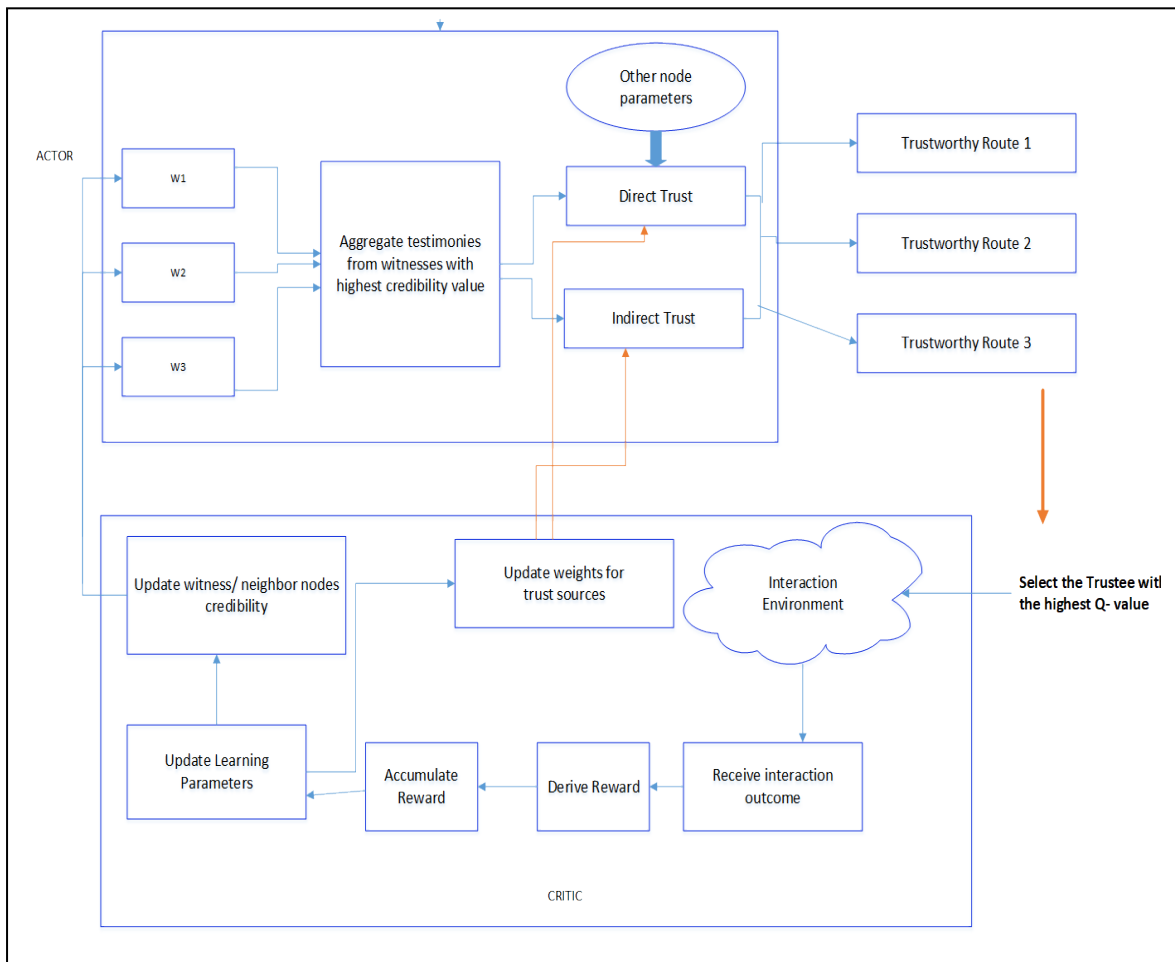


Figure 3.36: ACT model for identifying Trustworthy Paths

Chapter 4 - Simulation results and Discussions

This chapter details the Trust implementation framework using social network analysis and second major contribution of this research, namely, the Entropy-based spiral trust development framework. This model is implemented using reactive AODV MANET protocol. Finally discusses 3rd and final major contribution of using Deep Reinforcement learning as an intelligent trust prediction model using the above mentioned two techniques. The first section explains the monitoring mechanism of how this was used to identify malicious nodes.

4.1 Experimental Results and evaluations in Social Network Trust

4.1.1 Constructing the Probability Tables

After the Bayesian Belief Network (BBN) has been organized, the primary assignment is to fill the probability table for every factor. Subsequently, the Bayesian math is being utilized here to decide the probability values.

Firstly, the initial probabilities are added to the states of parentless parameters. All the parameters with no parents are assigned equal probability to each state initially, i.e., all states are equally likely

Evidence Variable	High	Medium	Low
Degree	0.33333333	0.33333333	0.33333333
Eigenvector Centrality	0.33333333	0.33333333	0.33333333
Closeness Centrality	0.33333333	0.33333333	0.33333333
Betweenness Centrality	0.33333333	0.33333333	0.33333333

Table 4.1: Initial Probabilities of Evidence Variable

Subsequently, the probability tables of parameters with parents have been filled. Those are much complicated and filled using probabilities based on gathered data and statistical methods. In this BBN, all the tables are small enough to be populated by hand; all the conditional probabilities are calculated manually.

Node Popularity							
Degree	High			Medium			
Eigenvector Centrality	High	Medium	Low	High	Medium	Low	High
High	1	0.5	0	0.5	0	0	
Medium	0	0.5	1	0.5	1	0.5	
Low	0	0	0	0	0	0.5	
Node Importance							
Eigenvector Centrality	High						
Betweenness Centrality	High			Medium			
Closeness Centrality	High	Medium	Low	High	Medium	Low	High
High	1	0.7	0.3	0.7	0.3	0	0
Medium	0	0.3	0.7	0.3	0.7	1	0
Low	0	0	0	0	0	0	0
Influence on Network							
Betweenness Centrality	High	Medium	Low				
High	1	0	0				
Medium	0	1	0				
Low	0	0	1				
TrV							
Node Popularity	High						
Node Importance	High			Medium			
Influence on Network	High	Medium	Low	High	Medium	Low	High
Yes	1	0.83	0.67	0.83	0.67	0.5	0.6
No	0	0.17	0.33	0.17	0.33	0.5	0.33

Table 4.2: Probability Tables of Intermediate and Hypothesis Variables

4.1.2 Compile the Bayesian Belief Network

After assigning the initial probabilities for all the parameters, the constructed Bayesian Belief Network may be applied to all individuals (nodes) in a social network and observe the TrV for each individual (i.e., the individual is how much trusted by his neighbours).

Ahead of that, the possible variations of evidence have been entered into the system. The evidence is entered into the BBN by selecting one of the states of an information variable. For all the possible combinations of evidence, it has been taken the value of the trust (Trust Value) as the output of the constructed BBN.

Below figures (Figure 4.1 and Figure 4.2) illustrate the BBN while sample evidence is entered and the probabilities are updated.

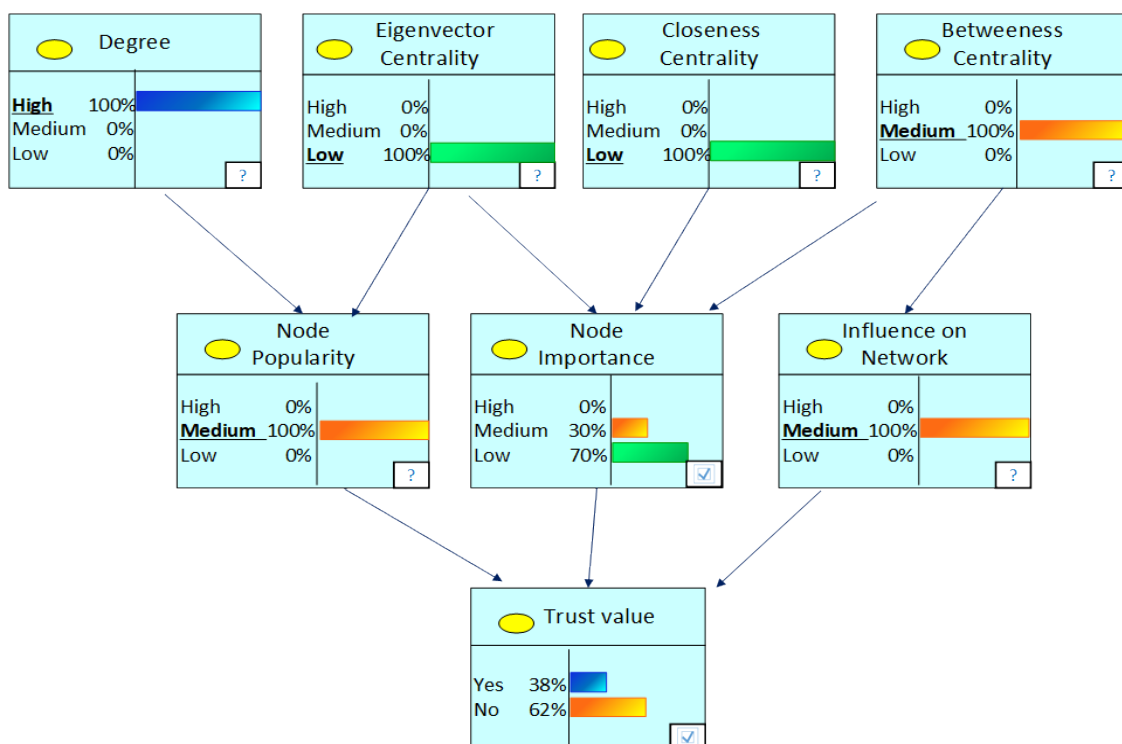


Figure 4.1: Sample Evidence I

Sample evidence (I):
 Degree = High
 Eigenvector Centrality = Low
 Closeness Centrality = Low
 Betweenness Centrality = Medium

Output: TrV
 Yes = 0.38
 No = 0.62

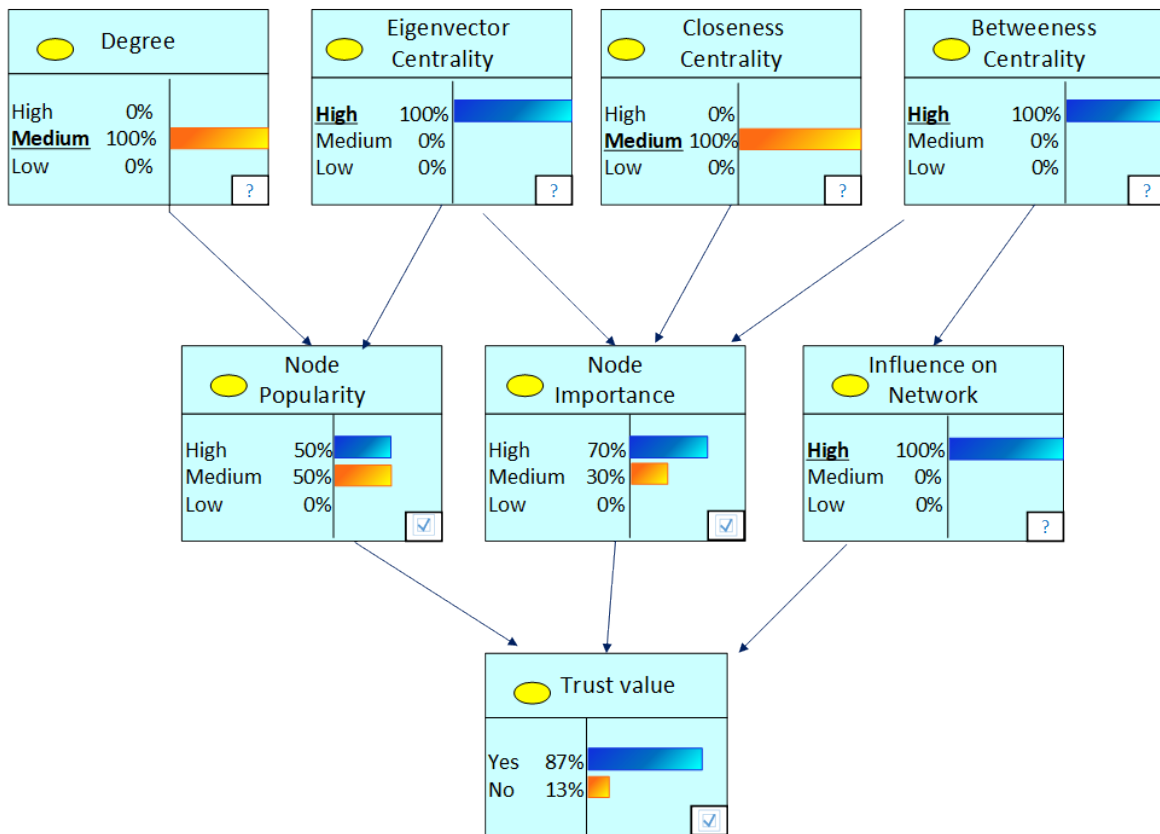
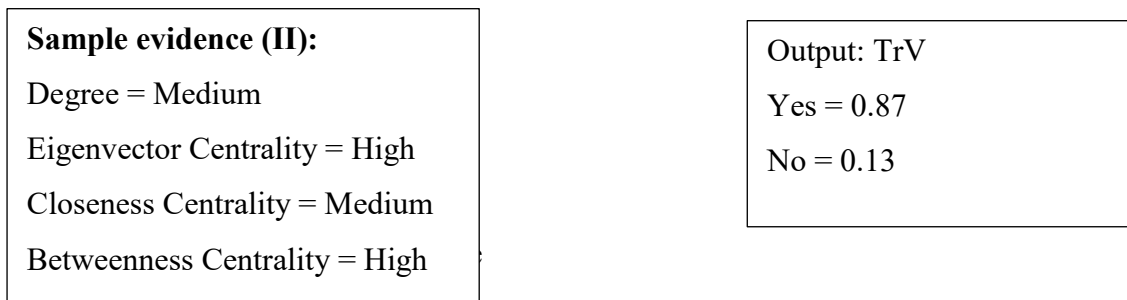


Figure 4.2: Sample Evidence II



Degree, Eigenvector Centrality, Closeness Centrality, and Betweenness Centrality parameters of various nodes in a network, the TrV state are changing according to the BBN as illustrated below.

Degree	Eigenvector Centrality	Closeness Centrality	Betweenness Centrality	TrV	
				Yes	No
H	H	H	H	100	0
H	H	H	M	78	22
H	H	H	L	55	45
H	H	M	H	95	5
H	H	M	M	72	28
H	H	M	L	50	50
H	H	L	H	88	12
H	H	L	M	67	33
H	H	L	L	45	55
H	M	H	H	87	13
H	M	H	M	63	37
H	M	H	L	42	59
H	M	M	H	80	20
H	M	M	M	59	42
H	M	M	L	37	63
H	M	L	H	75	25
H	M	L	M	53	47
H	M	L	L	30	70
H	L	H	H	72	28
H	L	H	M	50	50
H	L	H	L	28	72
H	L	M	H	67	33
H	L	M	M	45	55
H	L	M	L	22	78
H	L	L	H	62	38
H	L	L	M	38	62
H	L	L	L	17	83
M	H	H	H	92	9
M	H	H	M	70	30
M	H	H	L	47	53
M	H	M	H	87	13
M	H	M	M	63	37
M	H	M	L	42	59
M	H	L	H	80	20
M	H	L	M	59	42
M	H	L	L	37	63
M	M	H	H	78	22
M	M	H	M	55	45
M	M	H	L	33	67
M	M	M	H	72	28
M	M	M	M	50	50

M	M	M	L	28	72
M	M	L	H	67	33
M	M	L	M	45	55
M	M	L	L	22	78
M	L	H	H	63	37
M	L	H	M	42	59
M	L	H	L	20	80
M	L	M	H	59	42
M	L	M	M	37	63
M	L	M	L	13	87
M	L	L	H	53	47
M	L	L	M	30	70
M	L	L	L	9	92
L	H	H	H	83	17
L	H	H	M	62	38
L	H	H	L	38	62
L	H	M	H	78	22
L	H	M	M	55	45
L	H	M	L	33	67
L	H	L	H	72	28
L	H	L	M	50	50
L	H	L	L	28	72
L	M	H	H	70	30
L	M	H	M	47	53
L	M	H	L	25	75
L	M	M	H	63	37
L	M	M	M	42	59
L	M	M	L	20	80
L	M	L	H	59	42
L	M	L	M	37	63
L	M	L	L	13	87
L	L	H	H	55	45
L	L	H	M	33	67
L	L	H	L	12	88
L	L	M	H	50	50
L	L	M	M	28	72
L	L	M	L	5	95
L	L	L	H	45	55
L	L	L	M	22	78
L	L	L	L	0	100

Table 4.3: TrV Changes

Trusted Network

Ahead of development of the filter plugin, two essential conditions were fulfilled.

- Assigning ranges of numerical values to High, Medium, and Low states of Degree, Eigenvector Centrality, Closeness Centrality, and Betweenness Centrality parameters
- Defining trust threshold values, fifteen real datasets collected from Facebook account holders were exploited to train the constructed model.

Above mentioned two tasks had been conjointly completed to come up with better values in order to minimize errors. While comparing with the personal recommendations, those tasks were done repeatedly. Parameter ranges and some other properties of the gathered fifteen Facebook personal friend networks are represented below.

Network #	Number of Nodes	Degree (Value Range)	Eigenvector Centrality (Value Range)	Closeness Centrality (Value Range)	Betweenness Centrality (Value Range)
1	18	1 – 17	0 - 1	0 – 1.769	0 – 18.05
2	159	1 – 113	0 - 1	0 – 3.089	0 - 1522.392
3	311	0 – 163	0 - 1	0 – 4.371	0 – 6061.985
4	412	0 – 145	0 - 1	0 – 5.102	0 – 4126.27
5	453	0 – 138	0 - 1	0 – 5	0 – 5652.892
6	570	0 – 210	0 - 1	0 – 4.707	0 – 6526.109
7	579	0 – 450	0 - 1	0 – 4.189	0 – 10695.725
8	676	0 – 344	0 - 1	0 – 5.038	0 – 17093.799
9	701	0 – 287	0 - 1	0 – 4.866	0 – 19180.882
10	715	0 – 422	0 - 1	0 – 4.594	0 – 15079.666
11	778	0 – 318	0 - 1	0 – 4.598	0 – 16778.634
12	802	0 – 438	0 - 1	0 – 4.397	0 – 15605.855
13	889	0 – 492	0 - 1	0 – 4.357	0 - 20523.219
14	947	0 – 349	0 - 1	0 – 5.281	0 – 17254.002
15	1253	0 - 638	0 - 1	0 – 6.17	0 – 36472.899

Table 4.4: Properties of trained datasets

These datasets are having a variety of characteristics with the smallest network consists of 18 nodes and the largest is having 1253 nodes. And also there is a considerable amount of increase in the maximum values of Degree, Closeness Centrality, and Betweenness Centrality of larger networks and hence the values are not spread gradually. Eigenvector Centrality value is always ranging from 0 to 1 for any network.

Because of those characteristics, it is not favourable to define similar lower and higher margin values for states of the same variable for different networks. A much favourable the solution for any size of networks has been defined.

For Degree, Closeness Centrality, and Betweenness Centrality parameters;

- High state: Values greater than or equal to one-third of the maximum value of that variable.
- Medium state: Values greater than or equal to one-sixth of the maximum value of that variable.
- Low state: Values less than the Medium state

For Eigenvector Centrality parameter;

- High state: Values greater than or equals 0.2.
- Medium state: Values greater than or equals 0.1.
- Low state: Values less than 0.1.

4.1.3 Threshold Values of Trust

In favour of passing the confidential data, two edge values (threshold) for trust is proposed. As indicated by the privacy level of such data, the trust edges are presented. If the TrV for node N is TN;

Two edge values (threshold) for trust are characterized as shown in the below (Table 4.5). To pass very important data, yes condition of the TrV probability must be more prominent than or equivalent to 0.8.

Type of Info	Confidentiality Level	TrV Threshold
<ul style="list-style-type: none"> • Personal identified data/info <ul style="list-style-type: none"> ○ Psychotherapy Notes ○ Information about a Mental illness or Developmental Disability etc. 	Highly confidential (Highly Trust)	TN(Yes) ≥ 0.8
Aggregated data Non-personal data	Confidential (Medially Trust)	TN(Yes) ≥ 0.65

Table 4.5: Threshold Values for Trust

Thus according to the Table 4.3, to pass highly confidential data, there must be a TrV with Yes state more prominent than or equivalent to 0.8. Nine possible combinations have been selected. (Highlighted in green)

	Degree (Value Range)	Eigenvector Centrality (Value Range)	Closeness Centrality (Value Range)	Betweenness Centrality (Value Range)
i	High	High	High	High
ii	High	High	Medium	High
iii	High	High	Low	High
iv	High	Medium	High	High
v	High	Medium	Medium	High
vi	Medium	High	High	High
vii	Medium	High	Medium	High
viii	Medium	High	Low	High
ix	Low	High	High	High

Table 4.6: Highly Trust / confidential combinations

Next, to pass private (partially trust) data, there must be a TrV with Yes state more prominent than or equivalent to 0.65. The conceivable mixes are (featured in blue, in Table 4.3);

	Degree (Value Range)	Eigenvector Centrality (Value Range)	Closeness Centrality (Value Range)	Betweenness Centrality (Value Range)
i	High	High	High	Medium
ii	High	High	Medium	Medium
iii	High	High	Low	Medium
iv	High	Medium	Low	High
v	High	Low	High	High
vi	High	Low	Medium	High
vii	Medium	High	High	Medium
viii	Medium	Medium	High	High
ix	Medium	Medium	Medium	High
x	Medium	Medium	Low	High
xi	Low	High	Medium	High
xii	Low	High	Low	High
xiii	Low	Medium	High	High

Table 4.7: Medially Trust / confidential combinations

4.1.4 Trusted Network Filter Plugin

Once the trusted network has been implemented in a given social network, it must only contain trusted nodes. On behalf of that, the constructed Bayesian Belief Network has been applied to the social network. The nodes in the BBN represent concepts which are being mapped to attributes of nodes or links in the social network.

A new Gephi plugin named “Trusted Network” has been built to fulfil the task, which filters the highly trusted nodes and the partially trusted nodes according to the trust threshold values. By means of this node filtering mechanism, users can easily determine the individual nodes which may be highly or medially trusted.

In addition to the Trusted Network Filter Plugin, another Gephi Filter Plugin was generated for the evaluation and verification purposes of the system accuracy. The second Filter Plugin named “Most Untrusted Nodes” is filtering the mostly untrusted nodes in the network; nodes with $TrV = 0$.

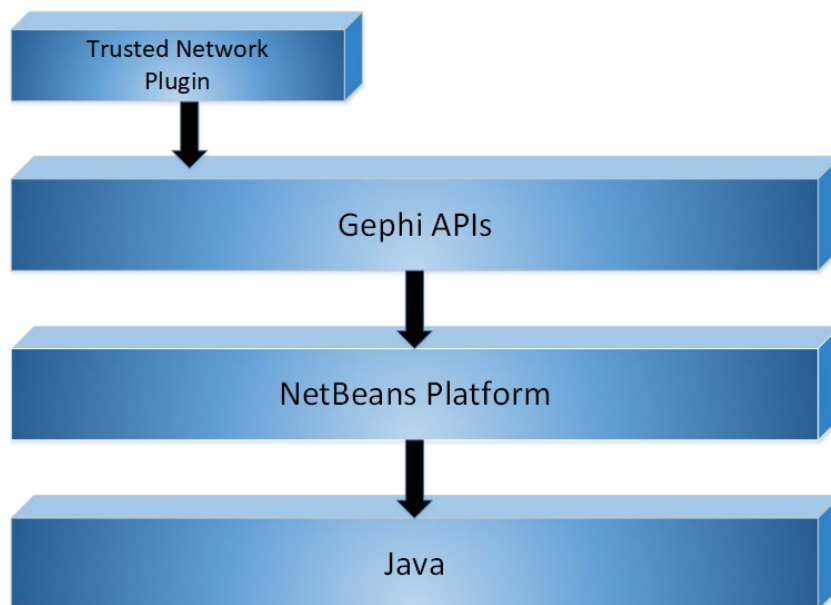


Figure 4.3: Gephi Architecture with new Trusted Network Plugin

Gephi is coded in Java Net Beans. In Gephi software, ahead of the primitive functionalities provided in its core, it consists of a flexible plugin architecture. By advancing that, developers are able to add their own layouts, network metrics, graph generators, filters, and tools hence it is free and open-source [76]. In order to implement the new Gephi plugin, the repository `gephi-plugins` which is an out of the box development environment for Gephi plugins has been downloaded. A new Net Beans module has been created inside that project, and the new filter has been coded in that module. Developed source codes are being appended to the Appendix section.

Gephi Filters

While satisfying filter conditions, filters are used to reduce the graph. These are predicates or functions which remove nodes and edges from the graph that unsatisfied particular filter conditions. Predicates return true or false, whereas functions input a graph and output a graph.

Gephi has a filter pipeline which is working with graph copies and executed on a separate thread and does not block the rest of the application. When filtering is being utilized, a copy of the complete graph is structure identified by an ID and where filters can work on it without disturbing other views. Gephi view concept in Graph API enhances this functionality. According to that method, a filter removes nodes on a copy of the graph structure while the main view (the complete graph) remains the same. When a filter is enabled, the copy is set as the visible view, which the graph visualized in the graph window. The main view is set as the visible view once the filtering is disabled. Filters may be bound into queries with the use of nested query concept as well [77].

Once the user has installed the Trusted Network filter plugin in Gephi software, it is being displayed under the Filters Tab. Ahead of filter the Trusted Network, three statistics under the Statistics tab must be run. Those are; Average Degree which calculates the degree values, Eigenvector Centrality which measures the eigenvector centrality values, and finally the Avg. Path Length which computes betweenness centralities and closeness centralities. Subsequently, the Trusted Network filter has to be dragged into the Queries area. By pressing the Filter button, the user may come up with the trusted network.

With the expectation of assessing the executed framework, seven Facebook close companion arrange datasets have been used, which were separated by means of Netvizz. The execution estimation of the presented trust-based system is proficient in the utilization of customized suggestions from the data set.

Personalized recommendations are a key feature in many online systems, which are generated from known opinions and trust relationships. User preferences, general acceptance of items, and influence from social friends are few types of personal recommendations[78], [79]. Thirty-three percentage of datasets (seven datasets) have been used in the assessment process, from the whole datasets assembled. Each and every network is analyzed using Gephi and filtered by the newly implemented filter plugins; Trusted Network Filter Plugin and Most Untrusted Nodes Filter Plugin. The two result networks from each personal network given in Table 4.8 and Table 4.9 were given to its particular owner (user). Hence the user may give his/her recommendations on the nodes of both networks separately.

The clients are solicited to rate the nodes from his/her own Trusted Network and the Most

Untrusted Nodes as Trust or Distrust, as per any bits of proof for choosing about the dependability of a related node. A short time later, it is analyzed the closeness of the trust esteems which were anticipated by the framework with the trust appraisals given by the user. Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) values have been estimated to get the precision of the framework. For the reason that both the MAE and RMSE are contrarily arranged scores, datasets scoring with low MAE, RMSE esteems are giving a superior reaction.

First, MSE and RMSE are basically the same thing as far as where to stop, one is just the square root of the other, and as far as comparisons go, the square root is exactly the same as the original. It just "compresses" the answers to be closer together; and they require different thresholds, but if X and Y are two positive real numbers, then if $X < Y$, then $\sqrt{X} < \sqrt{Y}$. Comparison is preserved.

With MSE (or RMSE) we are measuring the magnitude of the errors, and trying to minimize that average. Consider if candidate X has an absolute error of 4; and candidate Y has an absolute error of 5. If you use MSE, these are 16 and 25, respectively: You are giving 9 extra "points" worth of error candidate Y, for one extra unit of error; which means you are emphasizing (giving higher weight) to the error on outliers (items with larger error). MSE, or RMSE, is only useful if we are particularly worried about the large errors, and would rather your model have larger errors on the rest of the sample.

Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |f_i - y_i| = \frac{1}{n} \sum_{i=1}^n |e_i| \quad (4.1)$$

MAE is a quantity used to measure the system accuracy by determining how much predictions are close to the real values. The average absolute deviation between the predicted value \hat{f}_i of the system and the user rating y_i is calculated by MAE [75].

Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(f_i - y_i)^2}{n}} \quad (4.2)$$

The contrast between values anticipated by a model and the actually observed values can be measured by the consistently used measure RMSE. The sample standard deviation of the contrast between the predicted values and observed values is represented by this RSME. Hence, the errors are squared before averaged, this gives a relatively high weight to large errors. RMSE is most helpful when vast mistakes are particularly undesirable and it is a decent measure of precision [80], [81].

All the resulted networks and other details related to below-given Table 4.8 and Table 4.9 can find in appendix 2.

According to below table when untrustworthy nodes count became 0 always both MAE and RMSE values become 0.

	Total nodes count	Nodes count in the Trustworthy Network	Recommendation from Data Set		MAE	RMSE
			Trustworthy nodes count	Untrustworthy nodes count		
Test Dataset I	581	19	19	0	0	0
Test Dataset II	1064	4	4	0	0	0
Test Dataset III	378	13	9	4	0.307692	0.5547
Test Dataset IV	93	8	8	0	0	0
Test Dataset V	691	16	12	4	0.25	0.5
Test Dataset VI	10	3	3	0	0	0
Test Dataset VII	394	12	10	2	0.166667	0.408248

Table 4.8: Trusted Network Filter Plugin results

	Total nodes count	Most Untrustworthy Nodes count	Recommendation from Data Set		MAE	RMSE
			Untrustworthy nodes count	Trustworthy nodes count		
Test Dataset I	581	35	32	3	0.085714	0.29277
Test Dataset II	1064	83	79	4	0.048193	0.219529
Test Dataset III	378	28	18	10	0.357143	0.597614
Test Dataset IV	93	12	10	2	0.166667	0.408248
Test Dataset V	691	45	37	8	0.177778	0.421637
Test Dataset VI	10	0	0	0	0	0
Test Dataset VII	394	21	6	15	0.714286	0.845154

Table 4.9: Most Untrusted Nodes Filter Plugin results

As shown in the Table 4.8 and Table 4.9 always there can be correlation between trustworthy and untrustworthy nodes in a network that is,

$$\text{Untrustworthy nodes count} = \text{Total nodes count} - \text{Trustworthy nodes count}$$

Figure 4.4 and Figure 4.5 demonstrate the plotted diagram of MAE and RMSE of Trustworthy Networks against the number of nodes in the social network individually.

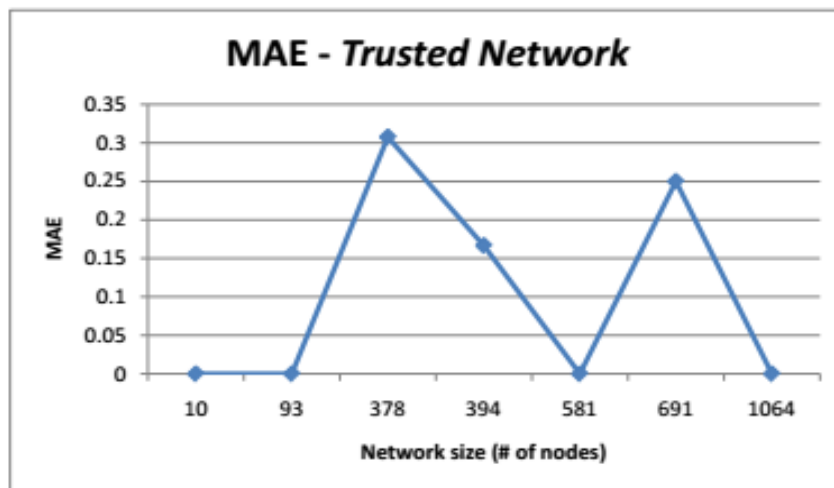


Figure 4.4: Test Results – MAE – Trusted Network

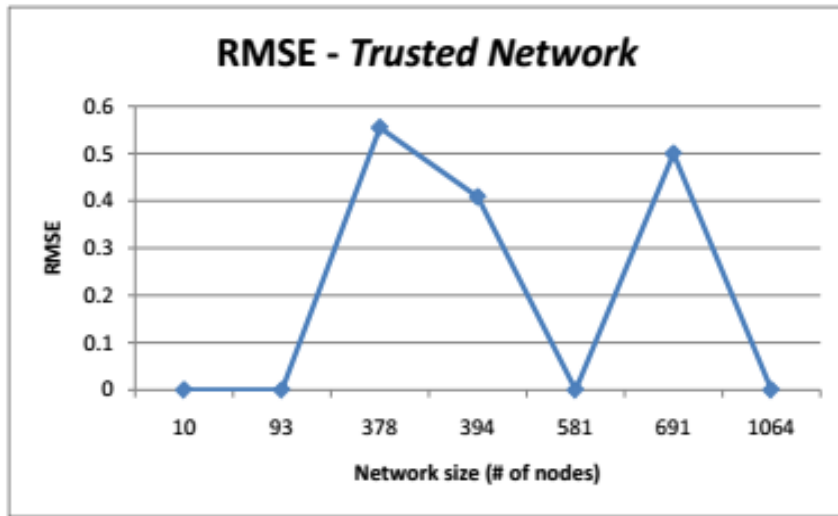


Figure 4.5: Test Results – RMSE – Trust Network

Figure 4.6 and Figure 4.7 show the plotted diagram of MAE and RMSE of Most Untrustworthy Nodes against the number of nodes in the informal organization separately. Since both the MAE and RMSE blunder esteems for Trustworthy Network and Most Untrustworthy Nodes are having low sums for all the test datasets, it could be resolved that the framework means an abnormal state of precision paying little heed to the system measure.

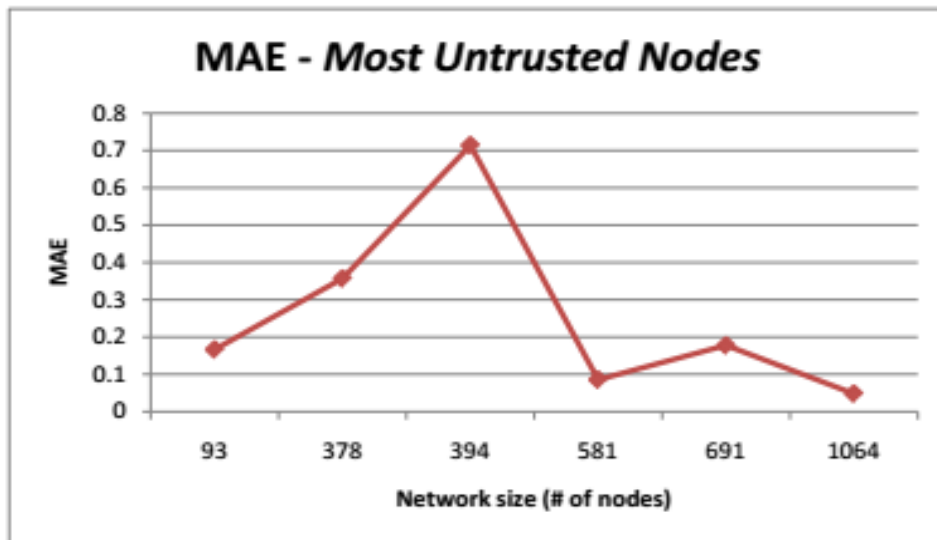


Figure 4.6: Test Results – MAE – Most Untrustworthy Nodes

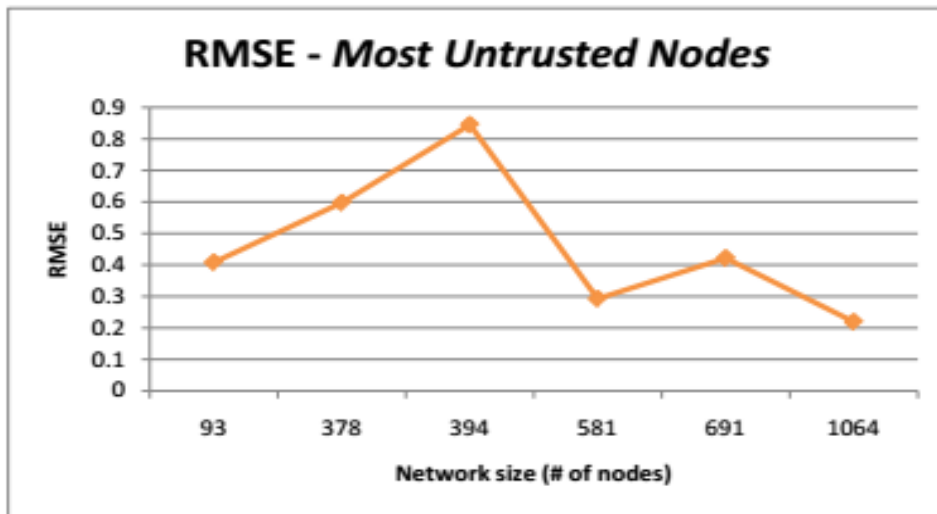


Figure 4.7: Test Results – RMSE – Most Untrustworthy Nodes

In Figure 4.8, it is delineated the framework exactness rates against each test dataset. This was figured by separating 'number of precise nodes in Trustworthy Network and Most Untrustworthy Nodes' from 'add up to the number of nodes in Trustworthy Network and Most Untrustworthy Nodes' in each dataset. Exactness rates are over 80% for five datasets. From that, one dataset has 100% exactness while another three datasets are acquiring over 90% of precision.

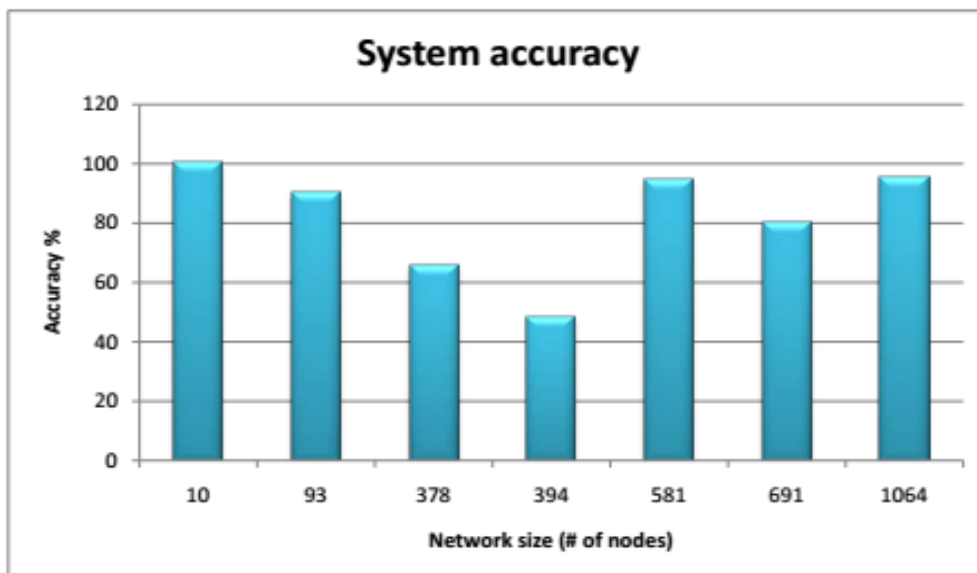


Figure 4.8: System accuracy

Accuracy mean value and Error mean value of the network calculated by utilizing all the seven test datasets which used in the assessment.

Mean estimation of the Error = 0.173913

Mean Estimation of the Error

$$= \frac{\text{Total number of incorrect nodes in Trustworthy Networks and Most Untrustworthy Nodes}}{\text{Total number of nodes in Trustworthy Networks and Most Untrustworthy Nodes}} \quad (4.3)$$

Mean value of the Accuracy = 0.826087

Mean Value of the Accuracy

$$= \frac{\text{Total number of correct nodes in Trustworthy Networks and Most Untrustworthy Nodes}}{\text{Total number of nodes in Trustworthy Networks and Most Untrustworthy Nodes}} \quad (4.4)$$

As delineated in Figure 4.8, it can be resolved that the framework has a high precision for any size of the system. And furthermore, the mean estimation of the framework precision is 0.826087; which shows to a greatly improved exactness, while the mistake implies is 0.173913; consequently, the framework error is low.

4.2 Simulation Results of ESTAODV and DRL

4.2.1 Simulation Setup

Simulations are done by NS3 (Network Simulator 3) version 3.2.0 (ns-3.2.0) in Ubuntu 16.04. The created network is consisting of multiple mobile nodes. ESTAODV and proposed Q learning based mechanism Reinforcement Learning Trust Manager (RLTM), Trust model ESTAODV is used to evaluate the results and the performance of the network.

The simulation network involves numerous Wi-Fi enabled mobile nodes. Here in this exploration, by assuming that the IEEE 802.11b DCF manner is used with RTS/CTS supported. AODV is picked as the benchmark to assess the exhibitions [82]. This was a simple choice because of the broadened convention was created utilizing AODV.

Exploration simulates a dense MANET involving of up to 100 Wi-Fi enabled mobile nodes equivalently scattered in a 300 m × 1500 m setting. Flows are produced by means of arbitrary

starting-end node combinations. An individual node can be in cooperation starting node and endpoint node.

For the period of the simulation, exploration accepts that altogether nodes are not collaborative: there can be malicious and collaborative malicious nodes. Each and every time there is an arriving packet, a receiving node attempts to transmit it to the endpoint regardless of its present circumstances. But malicious and other collaborative nodes will drop packets intentionally or unintentionally. The summary of simulation parameters is shown in Table 4.10.

MAC Layer	IEEE 802.11 DCF with RTS/CTS
Simulation area	1500 m × 300 m
Simulation time	400 seconds or longer
Maximum speed	2-10 m/s
Packet size	512 bytes
Mobility pattern	Random way-point model
Flow rate	10 packets/sec
Flow number	5-40
Traffic flow	Constant Bit Rate (CBR)
Raw stream data rate	2 Mbps
Large-scale propagation model	Lon-distance path loss model, $\eta_1 = 3.0$
Fast fading model	Ricean fading, $K = 13\text{dB}$

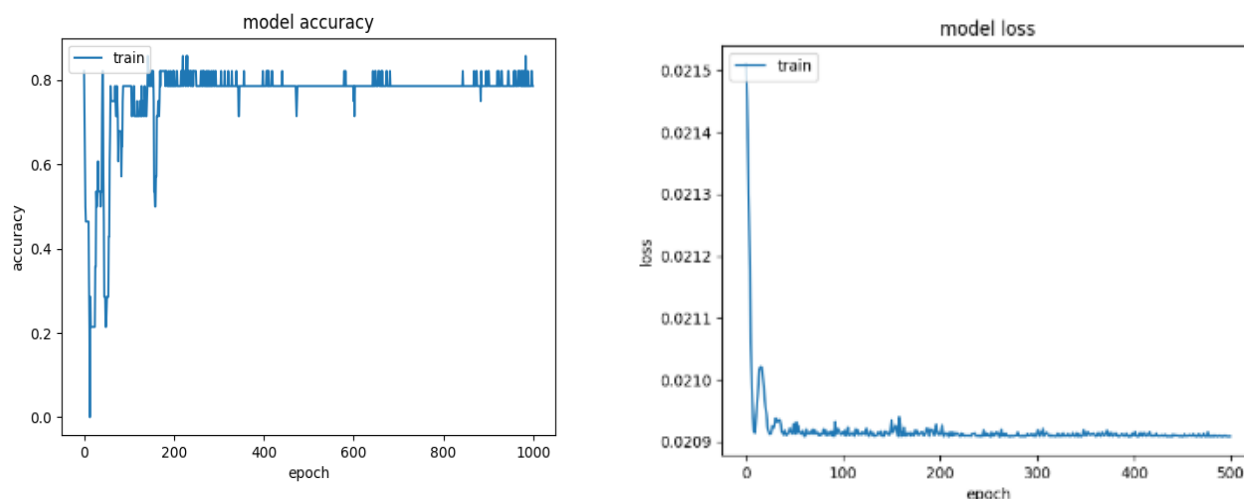
Table 4.10: Simulation Parameters

Node and flow accomplishments are assessed under diverse node densities, traffic loads, and mobility and link qualities. Every single dimension is an average of 60 iterations with diverse arbitrary amount seeds. Using continuously perceiving the effects, every single simulation

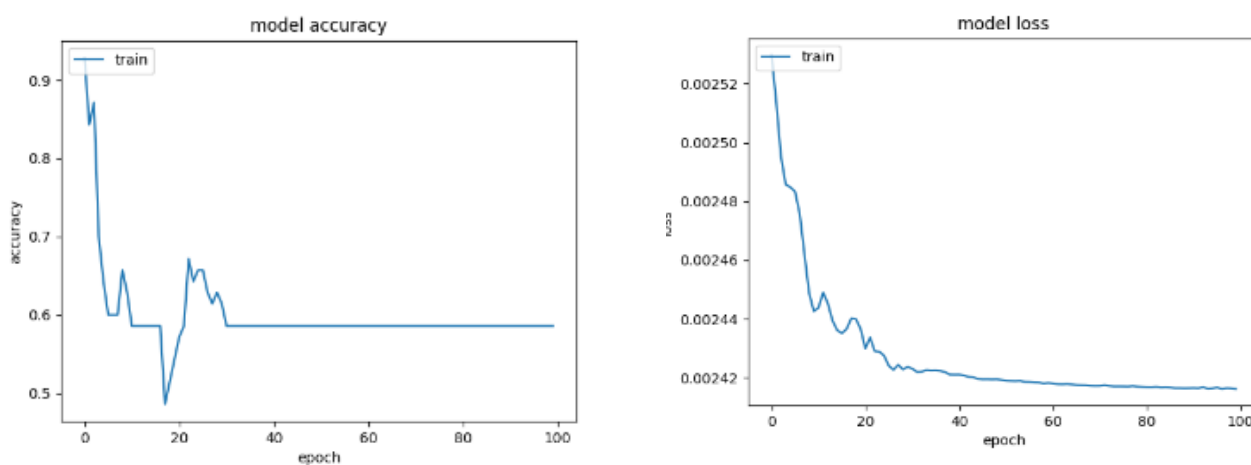
dismisses at 400 seconds or the period as soon as two contiguous inspections are within 5% variances, whichever is lengthier, to guarantee convergence. The outcomes are presented with 95% confident interval.

4.2.2 Simulation Results of Reinforcement Learning Algorithm

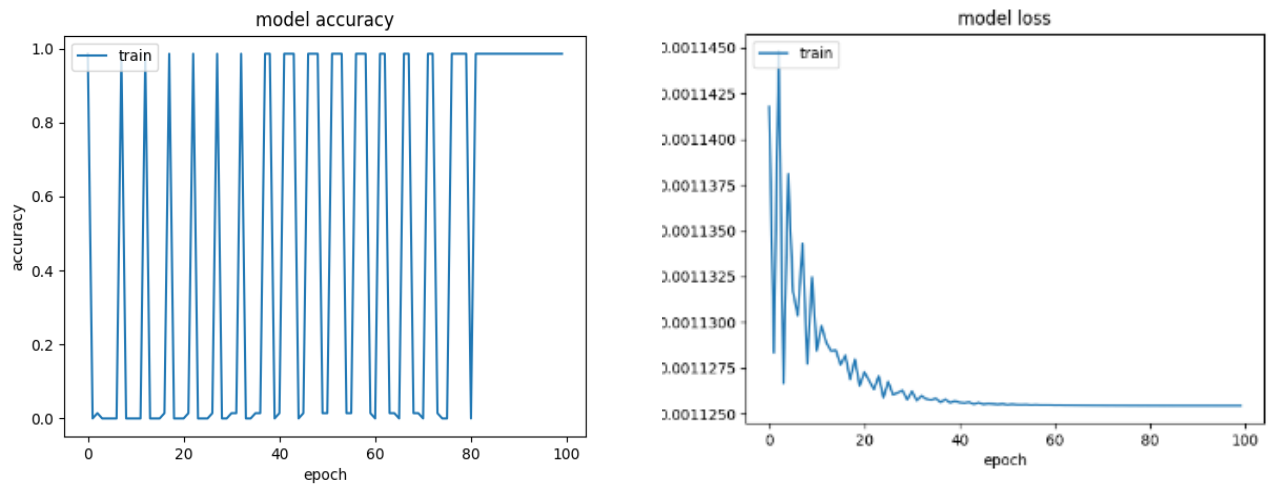
In Reinforcement Learning Trust Manager (RLTM), there are some parameters namely learning rate (α), discount factor (γ) and weights in the reward function. The following Figure shows the comparison of different learning rates and discount factors to adjust the accurate parameter value.



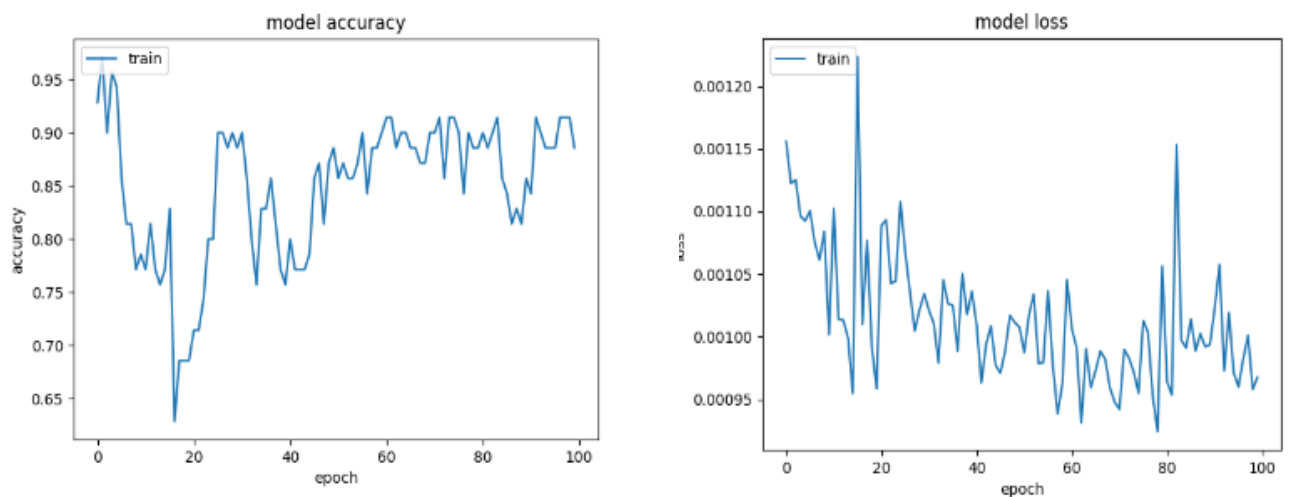
(a) accuracy and loss evaluation of model when $\alpha = 0.95$ and $\gamma = 0.001$



(b) Accuracy and loss evaluation of model when $\alpha = 1.45$ and $\gamma = 0.005$



(c) Accuracy and loss evaluation of model when $\alpha = 0.55$ and $\gamma = 0.020$



(d) Accuracy and loss evaluation of model when $\alpha = 0.01$ and $\gamma = 0.001$

Figure 4.9: Comparison of accuracy and loss of the model

The RLTM model is basically a general Q-learning model which involves with a specific recurrent neural network that generates q-values for different scenarios. Similar to the general reinforcement learning models this also has the two common phases of exploitation and exploration, where exploitation is about simply making the best decisions for the given current information and exploration is about gathering more information. In the Q-learning context,

when agent begins to learn we want it to make some random actions for the purpose of exploring more paths. Once the agent become trained better, we need model to exploit paths which are having highest q values. This is the scenario that Epsilon greedy policy works on, which helps to select random actions from a set of available actions. Hence, maintaining the optimal epsilon-probability (ϵ) to take the reward should be considered. Further, to differentiate between the optimality of the actions taken in the exploration and exploitation process, the other two training parameters which are learning-rate, alpha (α) and the discount-factor, gamma (γ) have to be considered.

The learning rate (α) can be defined as the step-size which determines the ability of overriding the old data of the network with the new information. Since a recurrent neural network has been used here, in the context of a neural network, this is one of the hyper-parameters which determines how fast it is going to be adjust the weights. Setting the learning rate to '0' means that q-values are never get updated and the agent is not going to be learn anything. In most of the learning models learning rate is set between '0' and '1' where setting a high value for the learning rate makes the learning of the agent more quick.

Gamma (γ) is the discount factor which measures the value of importance we give for the future rewards. Gamma is also set between '0' and '1' and if gamma is more closed to '0', the q learning agent is more tend to consider immediate rewards, and either if it is more close to '1' , it will be making more consideration about the future rewards with great weights. Further in order to converge the prediction process, the discount factor needed to be set for value which is less than zero.

Since all these things are happening in the aforementioned two phases which are exploration and exploitation, and also the convergence taking place in this situation, maintaining the epsilon (ϵ) value should be done. Making the predictions and getting the rewards can be done considering the epsilon-probability and the fact to be considered is in order to gain a maximum reward value epsilon value have to be set into the '(1- ϵ)'.

Considering those variations in the above mentioned hyper-parameters which are learning rate (α), discount-factor (γ) and the value of epsilon-probability (ϵ) following parameter tuning has been took place and according to that the relevant results have been filtered.

In order to tune parameters in the model on behalf of learning rate and discount factor (a) represents the model accuracy and loss when the values are set to 0.95 and 0.001 respectively, in (b) it shows the graphical evaluation according to 1.45 (α) and 0.005 (γ), (c) evaluations respect to the 0.55 (α) and 0.020 (γ) and finally, evaluation using 0.01(α) and 0.001 (γ). According to the above graphical evaluation of learning rates and discount factors in the model according to the accuracy and loss, research can conclude that the optimized values for learning rate as 0.95 and discount factor as 0.001.

W_1	W_2	W_3	W_4	W_5	Reward
0.3	0.3	0.1	0.2	0.1	0.302
0.2	0.3	0.2	0.2	0.1	0.401
0.1	0.2	0.2	0.4	0.1	0.601
0.1	0.1	0.2	0.5	0.1	0.721

Table 4.11: Weight Tuning of Reward Function

According to the table evaluation, weights w_1, w_2, w_3, w_4, w_5 adjusted for the mean delay, mean jitter, mean hop count, transmission ratio and lost packets of each and every route in the model respectively. Hence research can conclude from above comparison of weights in the reward function, weights having $w_1 \rightarrow 0.1, w_2 \rightarrow 0.1, w_3 \rightarrow 0.2, w_4 \rightarrow 0.5$ and $w_5 \rightarrow 0.1$ generate the maximum reward of 0.721. Hence, those weights are the best weights tuned for the reward function.

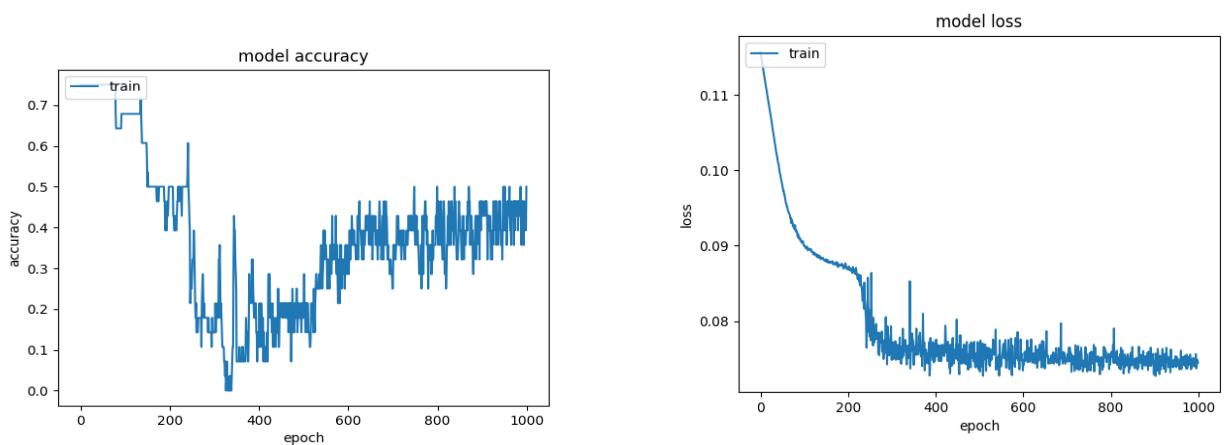


Figure 4.10: Loss and accuracy of the model when the number of nodes = 16

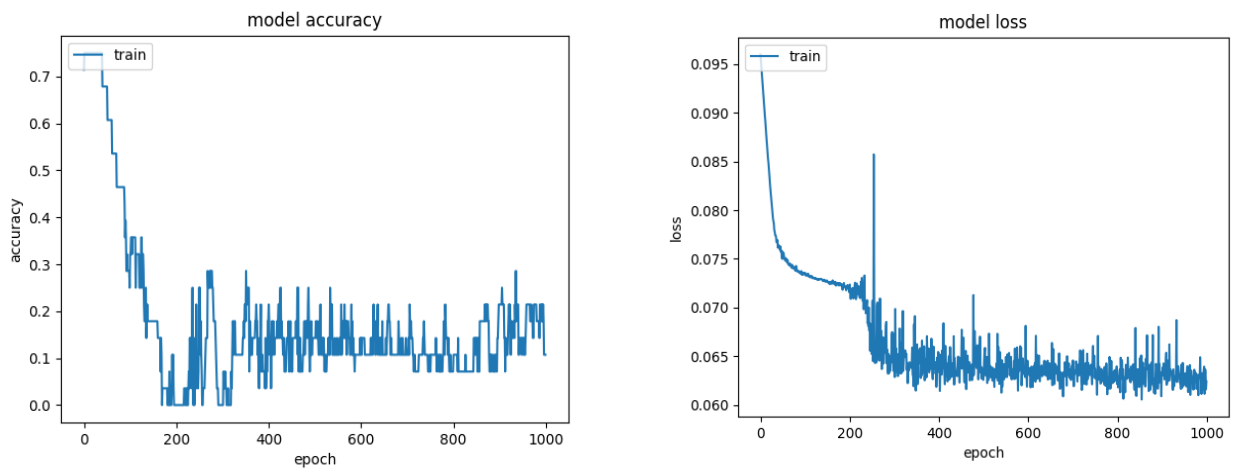


Figure 4.11: Loss and accuracy of the model when the number of nodes = 25

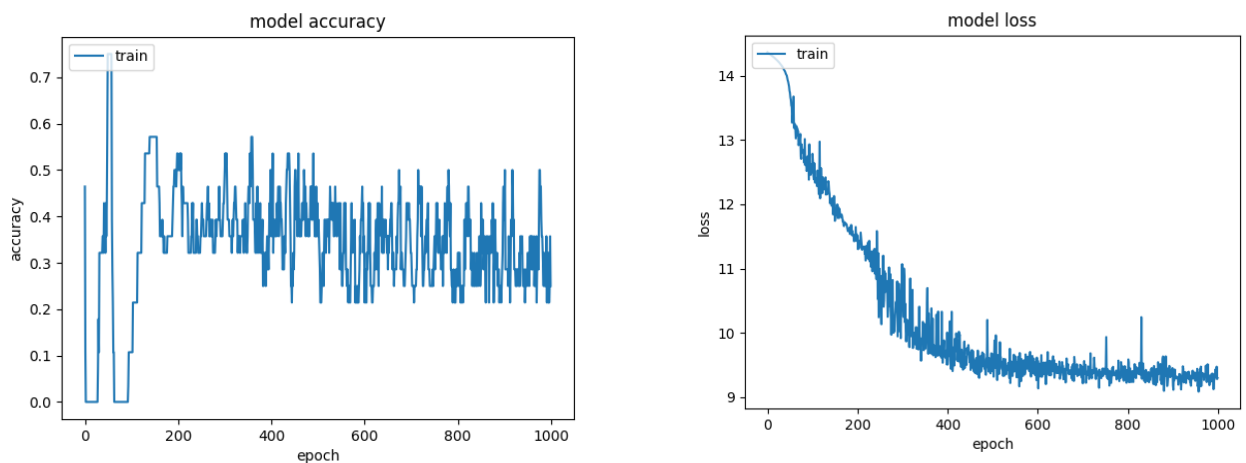


Figure 4.12: Loss and accuracy of the model when the number of nodes = 50

From figure 4.12 its evident that there is no clear evidence of model accuracy changes in different node configurations. Comparatively in the case of 50 nodes, which is the largest number accuracy remains the strongest. This indicates that the model gets better when more nodes are present. Looking at the model loss, the experiment clearly shows that when the training gets deeper, the loss becomes less and consistent, which is a positive aspect.

RLTM (Reinforcement Learning Trust Manager) is a routing algorithm in view of Q learning which is a set of reinforcement algorithms that creates a trustworthy network. It uses the discount factor to differentiate the optimal paths and the trustworthy behaviour of neighbouring

nodes where the discount factor can be expressed as,

$$\gamma = \text{mobility factor} * \text{bandwidth factor} * \text{energy factor} \quad (4.5)$$

Performance of the network after applying RLTM is evaluated for dissimilar node densities, traffic loads, and mobility and link qualities. By always concluding the outcomes, each simulation abolishes at 300 seconds to ensure convergence.

Following Figure 4.13 shows an example scenario, in which the process of transmitting and receiving of the packets between the 36 nodes of the developed network.

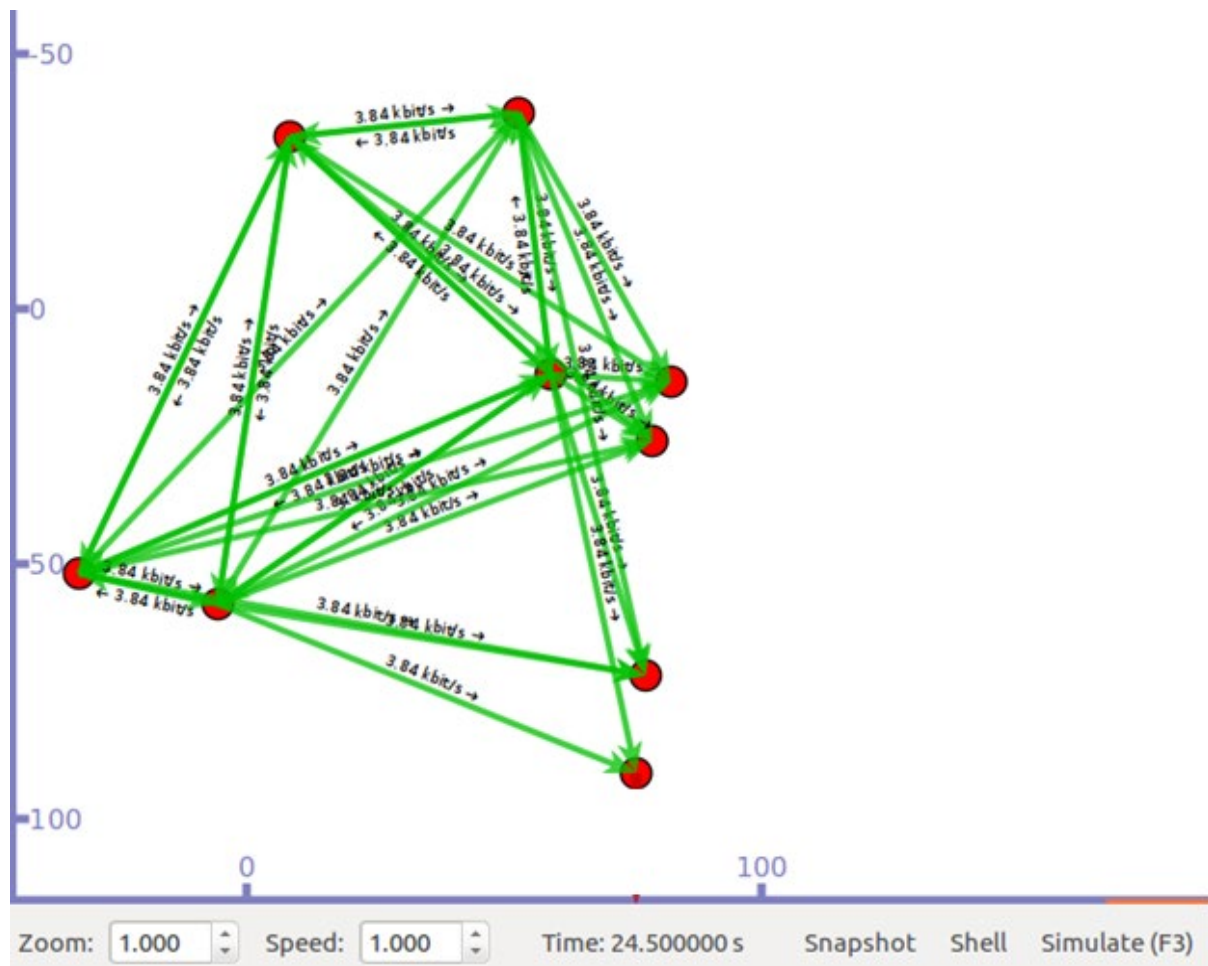


Figure 4.13: Visualization of the data transactions of the developed network

When residual energy levels are low, it is called a threshold value of the selfish node. In here threshold value is set to 20%.

Flow Id	Source Address	Destination Address	Q value	Reward
1	10.0.0.1	10.0.0.2	0.0138	1
2	10.0.0.1	10.0.0.3	0.0271	-1
3	10.0.0.1	10.0.0.4	0.0125	1
6	10.0.0.1	10.0.0.5	0.1216	1
4	10.0.0.1	10.0.0.6	0.8672	1
5	10.0.0.1	10.0.0.7	0.3218	1

Table 4.12: Q Table

Summarizing the results, it can be assumed that the path which has selfish nodes are analyzed by the TrV and from that can find out what is the best path that can be used to forward the packets.

After adding hyper-parameters to the recurrent neural network, it will generate the Q value using activation function as rectified linear. The reason behind the usage of the rectified linear function is it will generate the real value as the output. Other activation functions such as softmax will create kind of a probability value. And using the output layer as the dense layer it will give the Q value as the output. To identify best Q value here used the greedy policy and by using that get the maximum Q value from the array. And then based on the Q value reward values are generated. Finally using all the previous details again train the trained epochs and identify the loss and accuracy of the test and train epochs. And route information details are taken to the table and using that can identify what is the next hop and based on the Q value can identify what is the optimal path that nodes can go through. And also using below simulation research come up with the idea that if Q value is less than 0.168, then that path may consist some selfish nodes. Because paths which have less Q value than 0.168 has gone through selfish nodes which we implemented earlier.

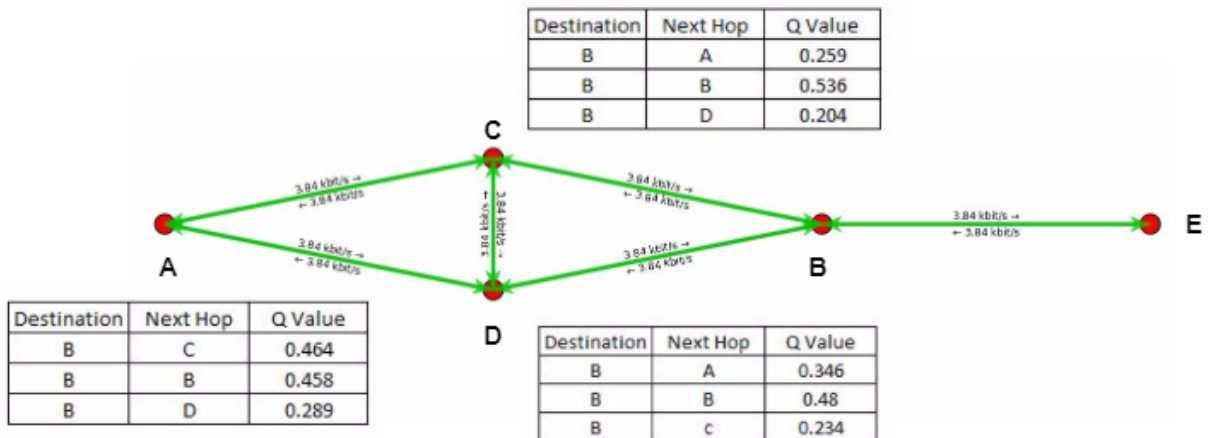


Figure 4.14: Node A, C, and D trade facts in between and modernize Q value

As shown in the above (figure 4.14) final result will display the destination with their next hop and the related Q values. From that can identify the best path which is suitable to do the data transmission.

```

deeplearning@deep-learning-virtual-machine: ~/Desktop/ns-allinone-3.20/ns-3.20
reward is 20
episode: 52/100, score: 20
Tx Bytes: 48
Rx Bytes: 48
Tx Packets: 1
Rx Packets: 1
Lost Packets: 0
Mean(Delay): 1.000578267
Mean(Jitter): 0.0
Mean(Hop Count): 1.0
Throughput: 0.000365999292187
FlowID: 32 (UDP 10.0.0.1/654 -> 10.0.0.5/654)
60
reward is 20
episode: 52/100, score: 20
Tx Bytes: 60
Rx Bytes: 60
Tx Packets: 2
Rx Packets: 2
Lost Packets: 0
Mean(Delay): 0.002494877
Mean(Jitter): 0.002332596
Mean(Hop Count): 1.0
Throughput: 4.5509372487e-05
FlowID: 33 (UDP 10.0.0.1/654 -> 10.0.0.9/654)
90
reward is 20
episode: 52/100, score: 20
Tx Bytes: 90
Rx Bytes: 90
Tx Packets: 3
Rx Packets: 3
Lost Packets: 0
Mean(Delay): 0.002049055
Mean(Jitter): 0.001916432
Mean(Hop Count): 1.0
Throughput: 6.2190347629e-05

```

Figure 4.15: Results of trained 100 epochs

```

+-----+-----+-----+-----+-----+-----+
| Flow id | Source | Destination | Q-value | reward | hopcount |
+-----+-----+-----+-----+-----+-----+
| 192 | 10.0.0.3 | 10.0.0.2 | 0.293 | 1.316 | 1 |
+-----+-----+-----+-----+-----+-----+
| Flow id | Source | Destination | Q-value | reward | hopcount |
+-----+-----+-----+-----+-----+-----+
| 208 | 10.0.0.3 | 10.0.0.6 | 0.295 | 1.303 | 1 |
+-----+-----+-----+-----+-----+-----+
| Flow id | Source | Destination | Q-value | reward | hopcount |
+-----+-----+-----+-----+-----+-----+
| 209 | 10.0.0.3 | 10.0.0.4 | 0.294 | 1.526 | 1 |
+-----+-----+-----+-----+-----+-----+
Selfish Nodes Found
+-----+-----+-----+-----+-----+-----+
| Flow id | Source | Destination | Q-value | reward | hopcount |
+-----+-----+-----+-----+-----+-----+
| 246 | 10.0.0.3 | 10.0.0.14 | 0.16 | -0.769 | 2 |
+-----+-----+-----+-----+-----+-----+
Parameters extracted for node: 4
+-----+-----+-----+-----+-----+-----+
| Flow id | Source | Destination | Q-value | reward | hopcount |
+-----+-----+-----+-----+-----+-----+
| 7 | 10.0.0.4 | 10.0.0.13 | 0.286 | 1.308 | 1 |
+-----+-----+-----+-----+-----+-----+
| Flow id | Source | Destination | Q-value | reward | hopcount |
+-----+-----+-----+-----+-----+-----+
| 33 | 10.0.0.4 | 10.0.0.8 | 0.292 | 1.371 | 1 |
+-----+-----+-----+-----+-----+-----+
| Flow id | Source | Destination | Q-value | reward | hopcount |
+-----+-----+-----+-----+-----+-----+
| 34 | 10.0.0.4 | 10.0.0.9 | 0.292 | 1.306 | 1 |
+-----+-----+-----+-----+-----+-----+

```

Figure 4.16: final simulation output

When the Q value is less than 0.169 that path is considered as a selfish path which has selfish nodes. Hence reward also should be decreased.

```

Node: 1 Time: 8.00s Ipv4ListRouting table
Priority: 100 Protocol: ns3::aodv::RoutingProtocol
Node: 1 Time: 8.00s
AODV Routing table
Destination      Gateway          Interface      Flag    Expire    Hops
10.0.0.1         10.0.0.1        10.0.0.2      UP      1.10     1
10.0.0.3         10.0.0.3        10.0.0.2      UP      1.07     1
10.0.0.4         10.0.0.4        10.0.0.2      UP      1.01     1
10.0.0.6         10.0.0.6        10.0.0.2      UP      1.05     1
10.0.0.7         10.0.0.7        10.0.0.2      UP      12.05    1
10.0.0.8         10.0.0.8        10.0.0.2      UP      12.10    1
10.0.0.10        10.0.0.10       10.0.0.2      UP      12.10    1
10.0.0.11        10.0.0.11       10.0.0.2      UP      1.05     1
10.0.0.12        10.0.0.12       10.0.0.2      UP      1.05     1
10.0.0.14        10.0.0.14       10.0.0.2      UP      1.11     1
10.0.0.15        10.0.0.15       10.0.0.2      UP      0.01     1
10.0.0.16        10.0.0.16       10.0.0.2      UP      1.05     1

```

Figure 4.17: Routing table

Using the above table, all the routing information can be identified, and the next hop also can be identified using gateway address.

Following 4.18 graph display the plot of accuracy on the training and validation datasets over training epochs. Figure 4.19 shows the plot of loss on the training and validation datasets over training epochs.

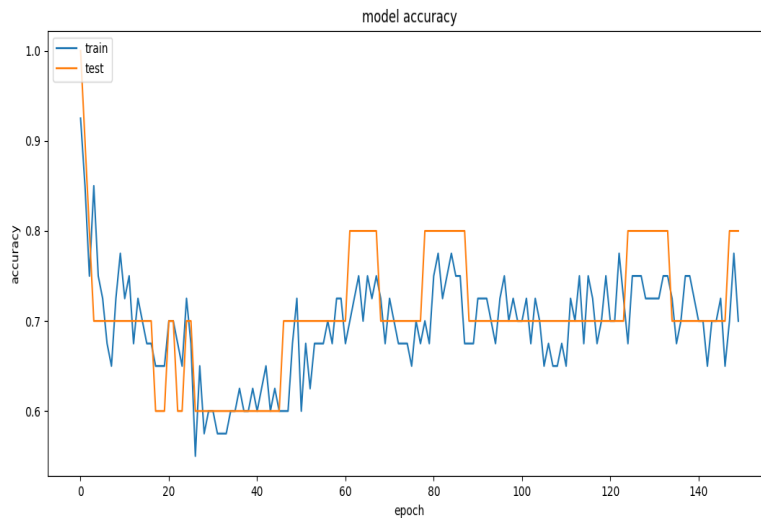


Figure 4.18: Data accuracy with epochs

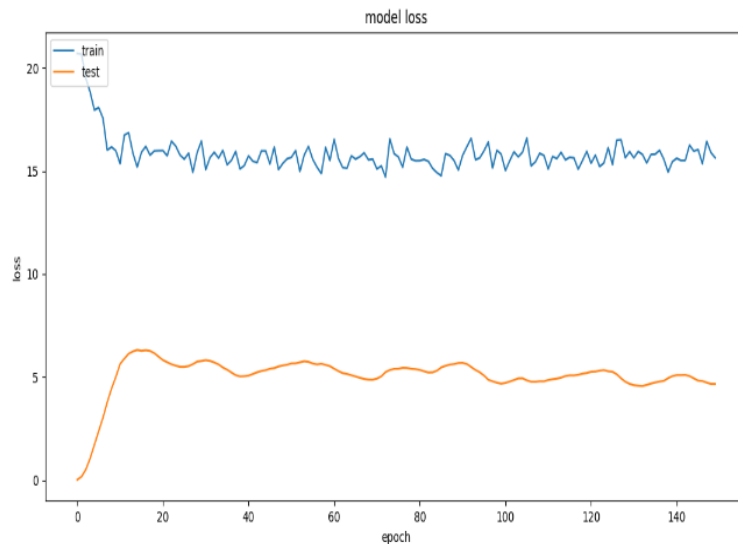


Figure 4.19: Model loss with epochs

The accuracy of a model is usually determined after the model parameters are learned and fixed and no learning is taking place. Then the test samples are fed to the model and the number of mistakes (zero-one loss) the model makes is recorded, after comparison to the true targets. Then the percentage of misclassification is calculated.

It is considered that the lower the **loss**, the better a model (unless the model has over-fitted to the training data). The loss is calculated for **training** and **validation** and its interpretation is how well the model is doing for these two sets. Unlike the accuracy, the loss is not a percentage. It is a summation of the errors made for each example in training or validation sets.

In the case of neural networks, the loss is usually negative log-likelihood and residual sum of squares for classification and regression respectively. Then naturally, the main objective in a learning model is to reduce (minimize) the loss function's value with respect to the model's parameters by changing the weight vector values through different optimization methods, such as back propagation in neural networks.

Loss value implies how well or poorly a certain model behaves after each iteration of optimization. Ideally, one would expect the reduction of loss after each, or several, iterations.

The output of the RNN will be generated through the Rectified linear (RELU) function which is the activation function of the LSTM layers. Since RELU is non-linear back propagating the errors is easy and able to activate the neurons in multiple layers. No of episodes will be run, and after the completion of training of the RL agent, the predictions will be given for the trained dataset. Action will be predicted by the bellman equation according to the generated Q-Value and the initially calculated Q-value will be updated by sending it continuously through the Bellman equation in order to get the optimal Q-value by running several epochs. Finally, the decisions will be made according to the predicted values regarding a given state and an action where the RL agent can identify the optimal routing path which is consists of trustworthy nodes or reputed nodes or the malicious nodes. The routing table of the ESTAODV protocol will be updated dynamically with the Q-values for each flow id, and the ESTAODV routing protocol will decide the best path for route according to the Q-values. Figure 4.17 the routing table updates per each node regarding each flow id with the Q-values.

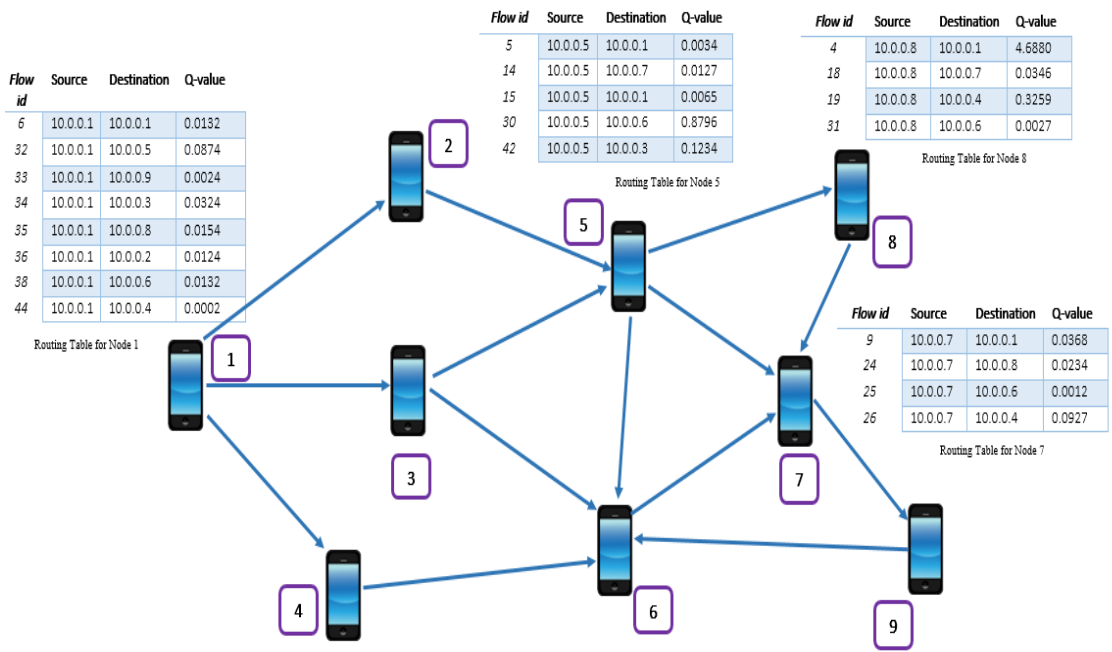


Figure 4.20: Routing table visualization with updated Q-values

The generated Q-Learning algorithm is applied to the Network implemented within the NS-3 and results was evaluated relevant to the 4 major areas, which classifies the nodes in the network whether they are trustworthy or not, reputed, whether a particular node has a malicious behaviour and the classification of the node that identifies belonging group and let AODV protocol to identify the best route path according to the evaluated results. Figure 4.20 represents the Q-values per each node in the network where the nodes are having max Q-values are considered as trustworthy and reputed nodes while the nodes having min values are identified as malicious ones.

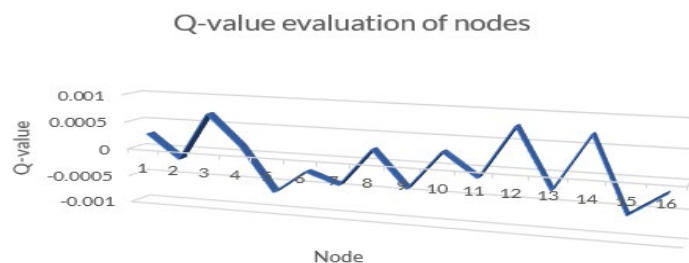


Figure 4.21: Q-value evaluation of the nodes

The performance of the network is evaluated in different perspectives including accuracy of the model, transmission rate and loss in the model.

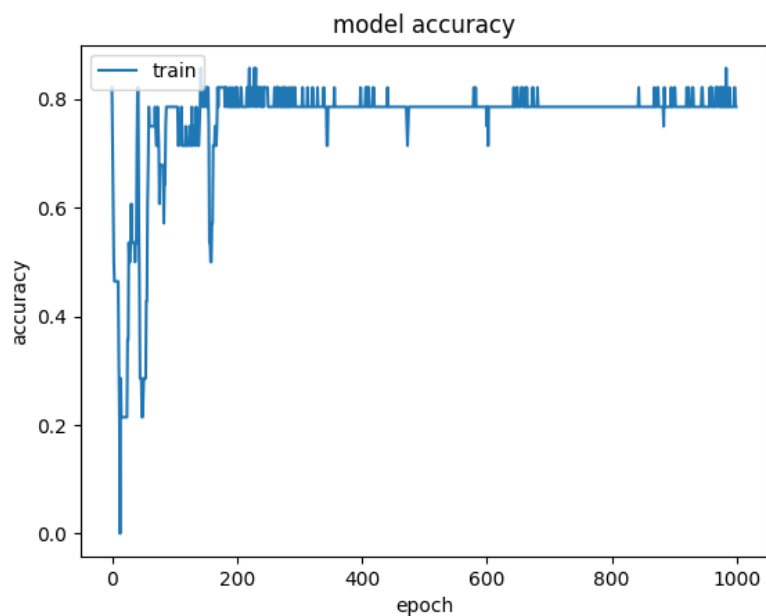


Figure 4.22: Accuracy of the model

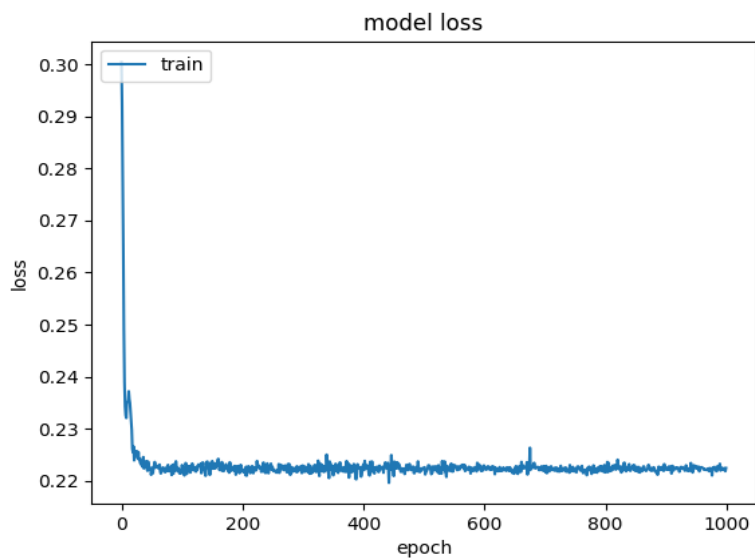


Figure 4.23: Loss of the model

Epochs Comparison

When the number of epochs is increased, then the variation of the loss and accuracy also increased and as shown in below figures. When the epochs amount is 1000 at that time, the clear accuracy and less loss can be identified. Hence it is good to have an average amount of epochs in the network environment.

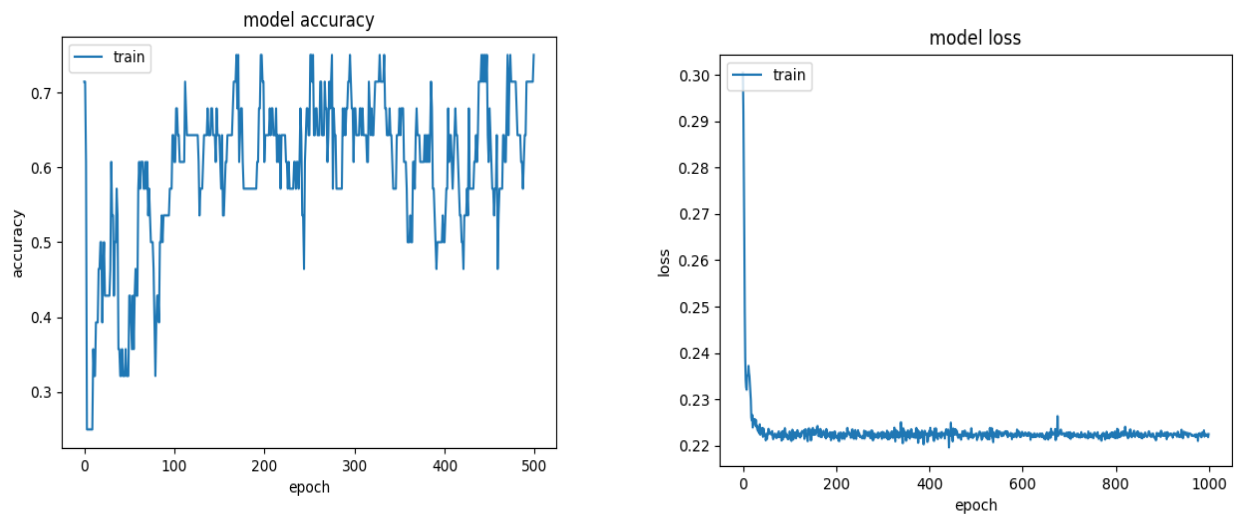


Figure 4.24: Model accuracy and loss for 500 epochs

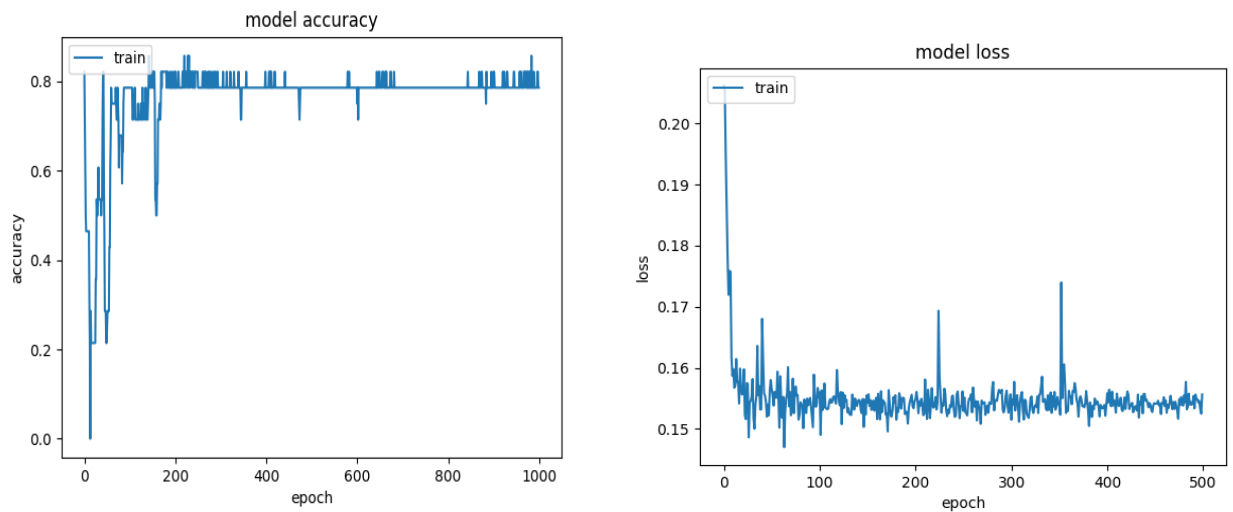


Figure 4.25: Model accuracy and loss for 1000 epoch

4.2.3 Simulation Matrices

As given below, endwise QoS metrics are selected to assess the performances:

- Packet Delivery Ratio (PDR): This is the proportion of the calculated data packets acknowledged by the endpoint node to the calculated packets transferred by the starting end. Most of the transmission PDR is applied as a constant rate. Hence packet delivery ratio is regarded to be an important measurement of effective throughput about the network communication.
- Delay: It is the duration taken in between data transfer from the sender to the receiver. Ideally, the retransmitted packet delay is not taken, and the first attempt is recorded. The simulation performance facts are composed by the NS3 flow monitors.
- Malicious nodes: Detecting malicious nodes was the major intention of the research. Hence this parameter would be used to analyze the efficiency of identifying malicious nodes using the implemented algorithm.

4.2.4 Traffic Load

The amount of mobile nodes is limit to 50, though the measure of flows varies from 5 to 40. Investigation discontinues the estimation at 40 streams. Even though the number of flows can be increased, there is a negative effect for Packet Delivery Ratio. Hence simulation results were limited to these numbers. The activity of each stream is produced at 10 data packets for every second. The delay variation and PDR can be seen in Fig. 4.26 and Fig. 4.27 respectively.

AODV is a protocol, which can be affected when heavy traffic is involved. It can be seen in Fig. 4.26. As it is shown in the figure, the variation of delay is heavy once the flows count rises in the variety of 5 to 10. This form of outcome is noticeably attributable to the nature of shortest direction routing used in AODV. Further shortest path algorithm is affected in the dense areas of the routing network. Both AODV and ESTAODV tend to select the existing paths to send data to reduce the overloaded communication channels. Due to this reason, more flows can add more delay to the network. Research results indicate that no major deviation was introduced with the additional burden of control packets to the network. Application of ESTAODV and RLTM had not given any major delay concern to the network.

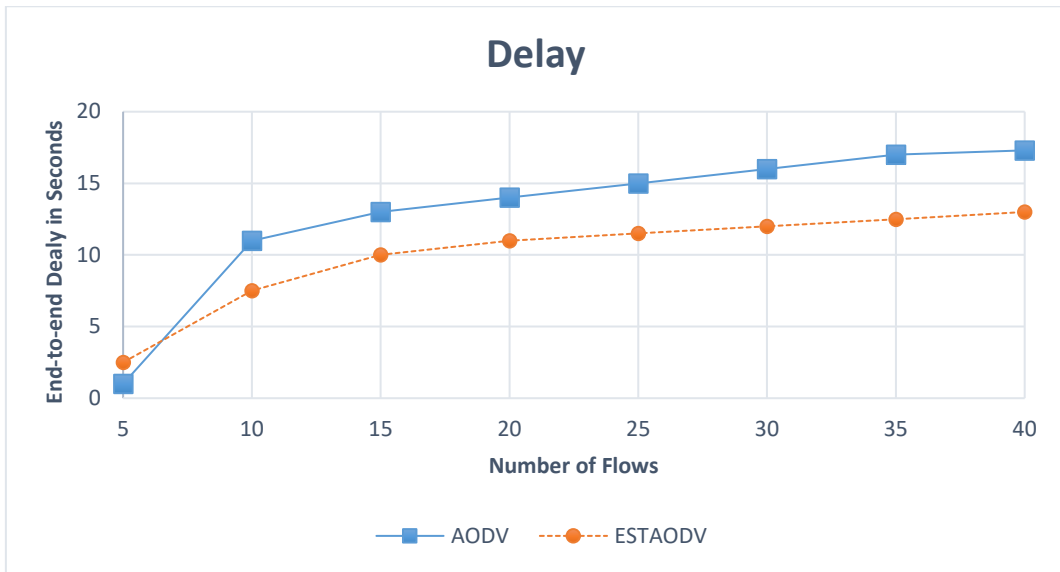


Figure 4.26: Delay

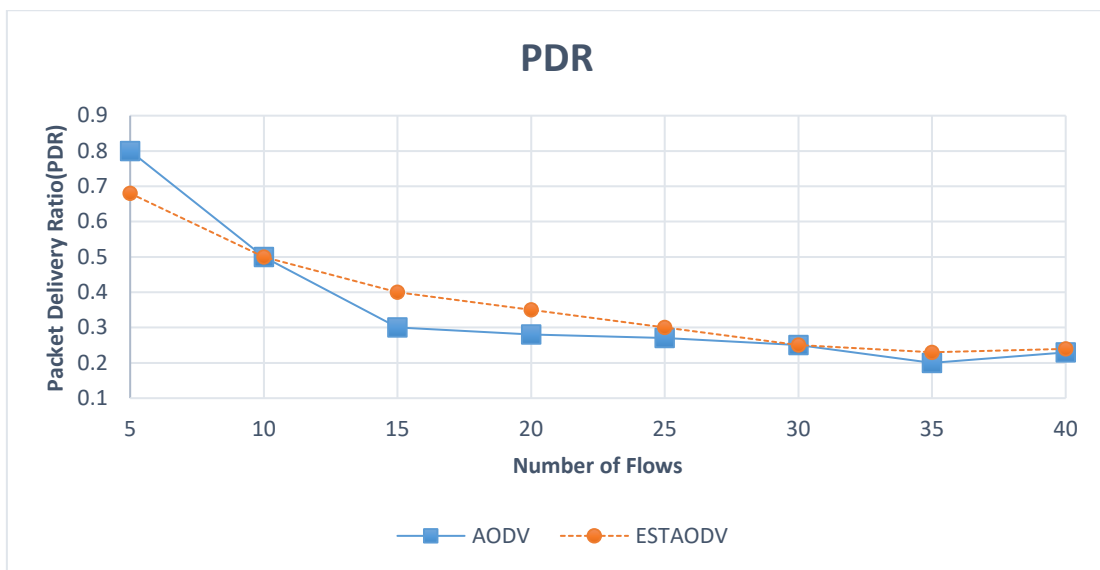


Figure 4.27: Packet Delivery Ratio

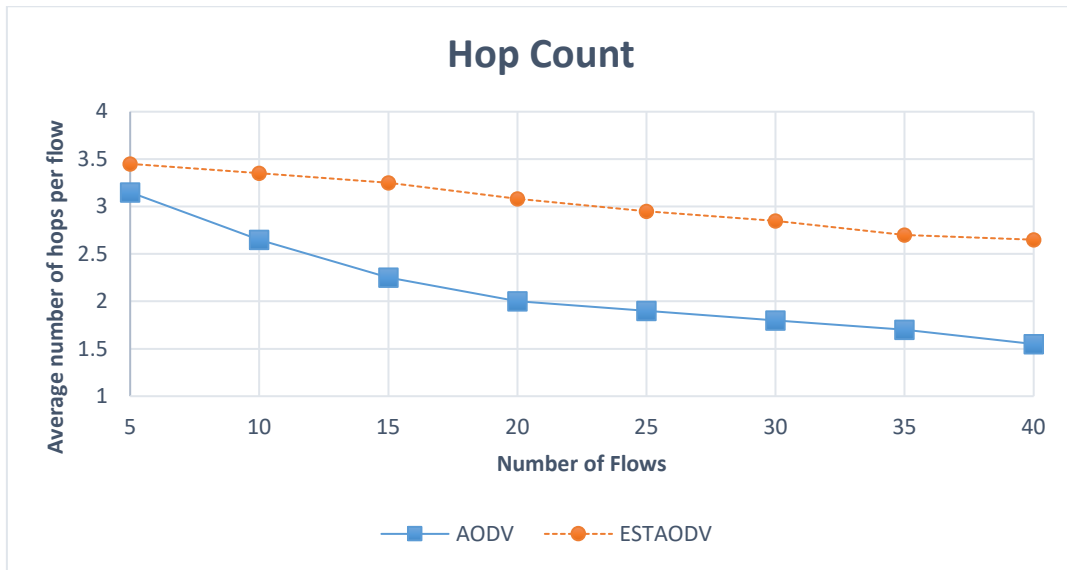


Figure 4.28: Average Number of Hops per flow

Given kind of situation ESTADOV also will be using more direct paths which will add more a negative congestion level.

Next at figure 4.27 shows the effect of more flows on Packet Delivery Ratio. The graph shows both protocols go through a loss in PDR due to the increasing number of flows. ESTADOV will perform negatively in terms of PDR not only for the effect on bandwidth but other excessive contentions. But researcher is confident that the addition of several control layers in ESTADOV and RLTM have not majorly affected the control conditions like PDR and Delay. Moreover, the addition of several control packets and cycles makes it difficult to manage the bandwidth. For example, ESTADOV adds several control behaviours, which add few control broadcasts to the environment. AODV has a good performance when the number of flows is 5 because short paths usually perform well when congestion is not formed. ESTADOV does not have a great performance issue in PDR.

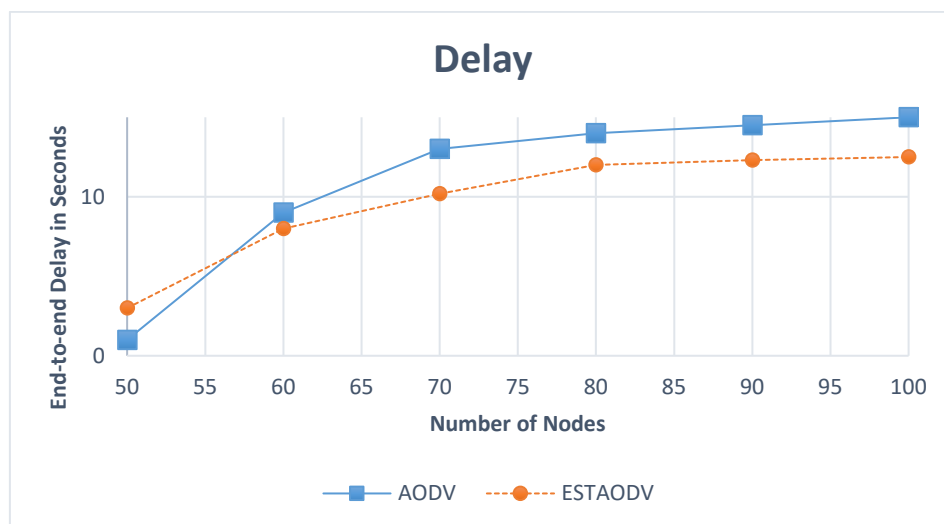
The average number of hops is a good measure to understand the nature of operation under heavy traffic. In figure number 4.28 shows both AODV and ESTADOV use a lesser number of hops when more flows added to the network. This scenario only works with successfully received packets within the network.

4.2.5 Node Density

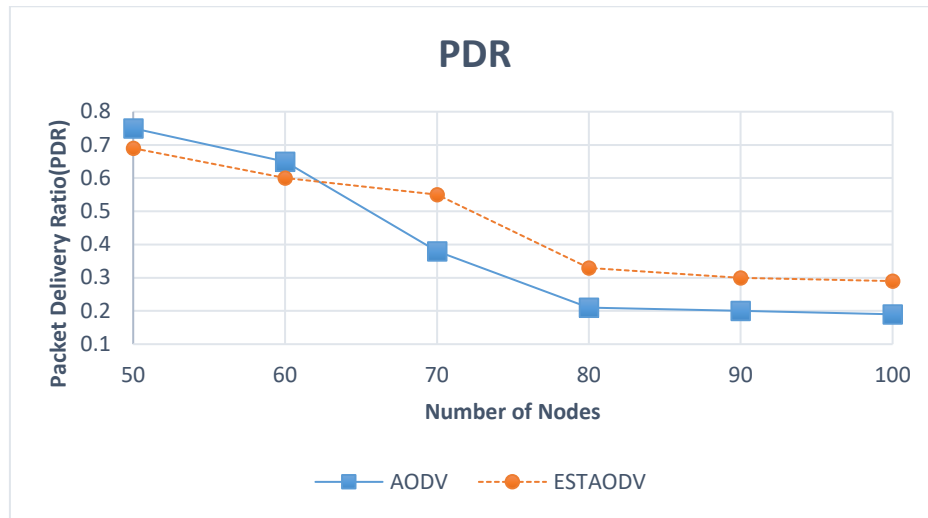
Research testing was done with increasing node numbers from 50 up until 100. The number of flows were kept fixed when increasing the node numbers. After the testing, the results are depicted as in the figure numbered 4.29.

When applied with higher node density there are more nodes around each other. This may result in more communication blockages. At the same time having more nodes will allow in more forwarding paths which provides a better connective environment within the network. But the performance of the network nodes (delay and packet delivery ratio) mainly depends on information sharing mechanism and additional routing options provided. These routing options and sharing mechanisms should provide facilities to minimize the routing overheads.

Due to the nature of AODV adding more nodes will not have much of a greater improvement in the performance aspect. One reason is it uses AODV is a reactive protocol in nature, which will look for new paths if there is an issue for the current path. The same applies to ESTAODV. Even though the security will be increased in terms of authentication, availability and etc. there will be no big value addition in terms of adding more nodes. As the next step, the ESTAODV also looks for nodes at the edge of transmission range with poor link quality. This also provides no help to improve the delay or packet delivery ratio.



(a) Delay Under Different Node Densities



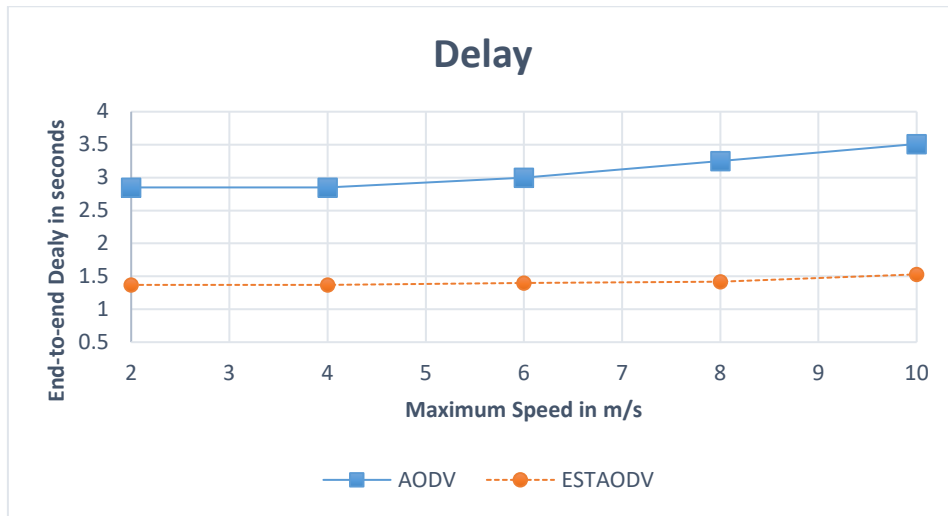
(b) PDR under different node densities

Figure 4.29: QoS Performance under Different Node Densities

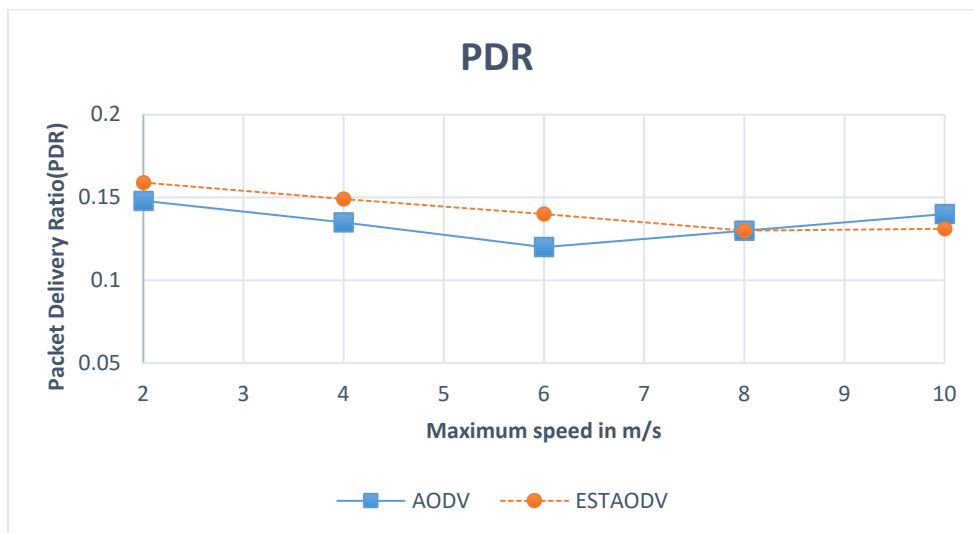
4.2.6 Mobility

Next consideration would be how the frequent change in the network will handle the QoS parameters. The test model was developed restricting to 40 network flows. Further, mobility model was designed with the random way-point model. Model parameters were changed from 2 m/s to 10 m/s.

Both protocols showed a high performance with node mobility. This was clearly can be seen in the figure listed following. Topology change of the network was managed by the routing mechanisms both in AODV and ESTAODV. However, the learning process of ESTAODV using RLTM relies on how the neighbours are structured within the routing network topology. ESTAODV will always change the learning process due to the constant value changes within the global trust and next it will shift to the RLTM to generate an accurate value. ESTAODV had some minor performance degradation issues due to the architectural complexity within the RLTM module. ESTAODV trust mechanism was not affected in this situation for any negative performance.



(a) Delay Under Different Motilities



(b) Packet Delivery ratio under Different Motilities

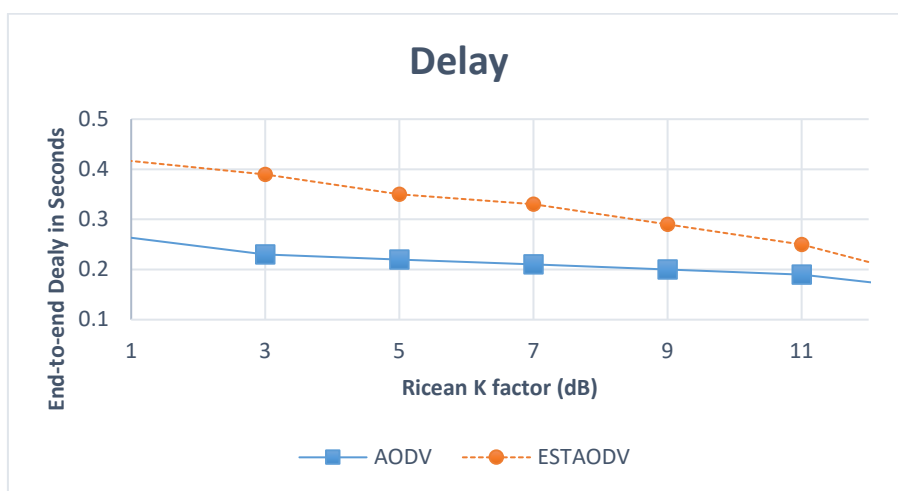
Figure 4.30: QoS Performance under Different Motilities

4.2.7 Link Quality

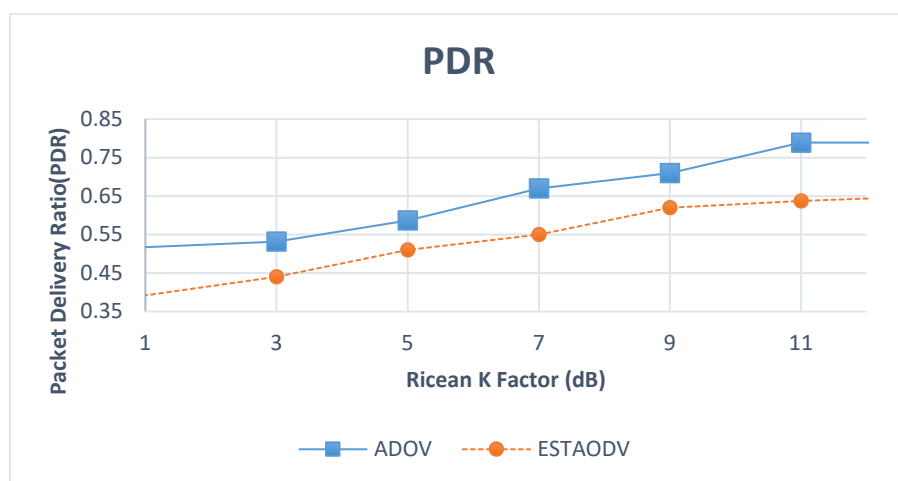
Quality of links straightforwardly decides the steadiness of a transmission. Better Quality in links can guarantee better Packet Delivery ratio. This particular stage calculates the impact of the quality of links by using factor K of the Ricean engendering model. For the testing number of nodes were restricted to 50 whereas the number of flows was restricted to 20. K is number,

which is a ration between the power in direct and indirect paths. If the K value is small, it indicates a lesser quality transmission link. In figure 4.31 shows the result of link quality changes affecting the performance of the two protocols.

As represented in Fig. 4.31a, ESTADOV is lesser vulnerable than AODV because it uses SINR information within the reward function of reinforcement learning implementation. Hence link quality is considered as part of the trust creation process. That really added a value in getting lower delay than in the case of AODV.



(a) Delay Under Different Ricean K Factors



(b) PDR Under Different Ricean K Factors

Figure 4.31: QoS performance Under Different Ricean K Factors

When needed to find a route AODV and ESTAODV both, sends a Route Request (RREQ) data packet which floods the network until the needed destination is found. Then a Route Reply (RREP) packet is sent to clarify the destination path. Hence, the path reliability is tested already. That concludes that PDR will not be affected much within the link quality changes. This is shown in figure 4.31b.

According to the results it is clearly indicated that ESTAODV performance matrices are better than benchmarked AODV matrices. This was due to following practical reasons,

- Even though, ESTAODV is more computationally complex than AODV it has fewer computational steps in routing decisions than AODV.
- Since we classify the nodes into trust levels, when it comes to routing selecting the optimal and secure path is done by means of these trust levels of the neighbours other than TrV. That reduces the time taken for the routing decision.
- Every time we calculate the trust it will automatically update the TrV along with the trust level and according to proposed model, we isolate and remove the malicious nodes from the network itself. This will reduce the possible routing paths and increase the performance.
- When we identified the nodes or the neighbours with collaborative malicious behaviour, along with the collaborative malicious node all the other collaborative neighbours will be blacklisted and isolated. It also increases the performance.
- The key idea of ESTAODV algorithm is to construct the new trust-entropy and select the secure and stable path with the help of entropy to reduce the number of route reconstruction so as to provide QoS guarantee in the network.

4.2.8 Varying number of malicious nodes

Next section compares how AODV and ESTAODV act when malicious nodes are present within the network. Data traffic for the network set as it defines section 4.2.1. Further, Random way point model is being used with nodes speed of 0.5 m/s were maintained.

Fig. 4.32 shows that ESTAODV performs better than AODV in terms of packet delivery ratio. The packet delivery ratio of ESTAODV is not 100%, which can be explained by the fact that

malicious nodes act as forwarders until trustworthy routes are established.

Fig. 4.33 illustrates that the average latency of ESTAODV and AODV increases sharply with the number of malicious nodes increasing. In order to avoid malicious nodes, ESTAODV may establish longer paths, which leads to a slight increase of its average latency.

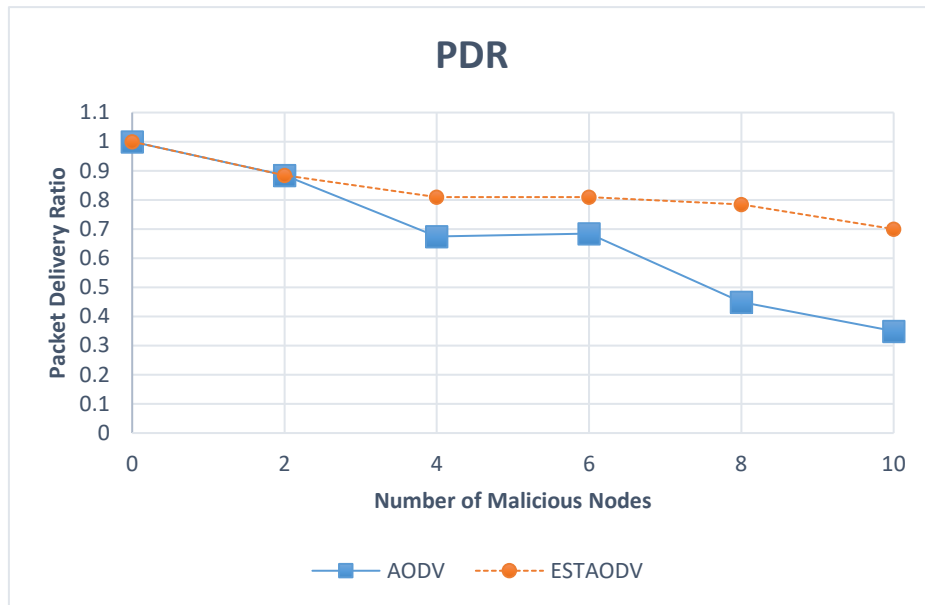


Figure 4.32: Packet Delivery Ratio

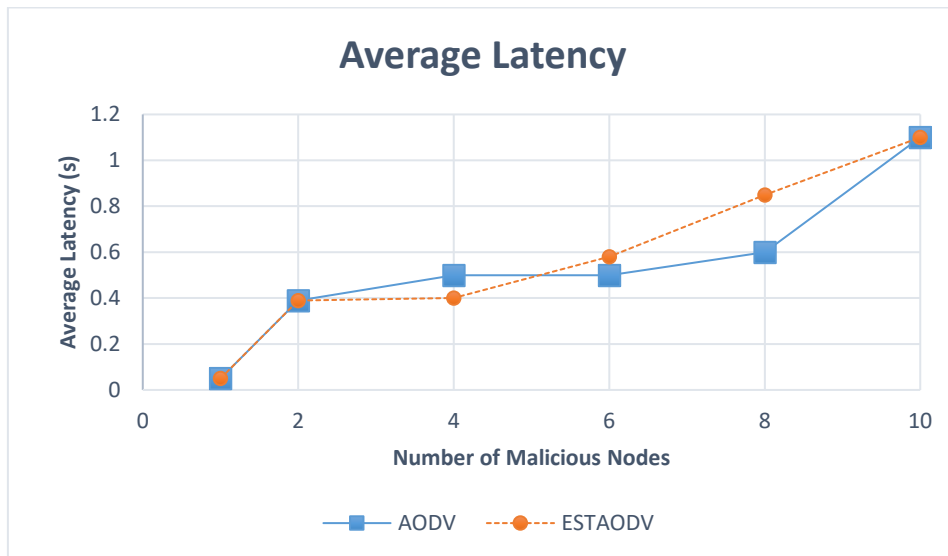


Figure 4.33: Average Latency

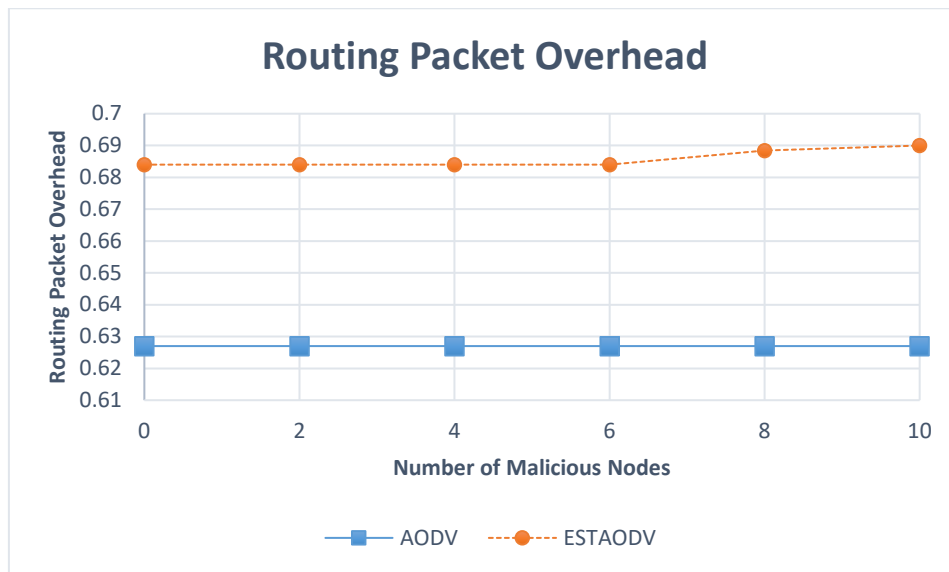


Figure 4.34: Routing Packet Overhead

Fig. 4.34 clearly shows the routing packet overhead is more in ESTAODV, which is obvious with the number of packets which been used for trust creation and reinforcement learning. With more malicious nodes more operational action will be taken for better accurate results.

4.2.9 Impact of the framework on the detection of malicious nodes

Experimental results were recorded on analyzing the performance of detection capability of malicious nodes based on the protocol implementation. As shown in the Figure 4.35, malicious node detection rate is increasing when the number of test nodes increase. In this phase, both pure malicious and the collaborative malicious nodes will take as one category ‘Malicious’.

The results show that the performance of the framework is satisfactory in identifying malicious nodes with high true positives and low false positives. This can be seen clearly in figure numbers 4.36 and 4.37. Results indicate that even at higher malicious nodes, the true positive rate remains at very high levels where classic AODV fails to identify. Further, false positive rate remains below 3%, which is a good indication of the detection results.

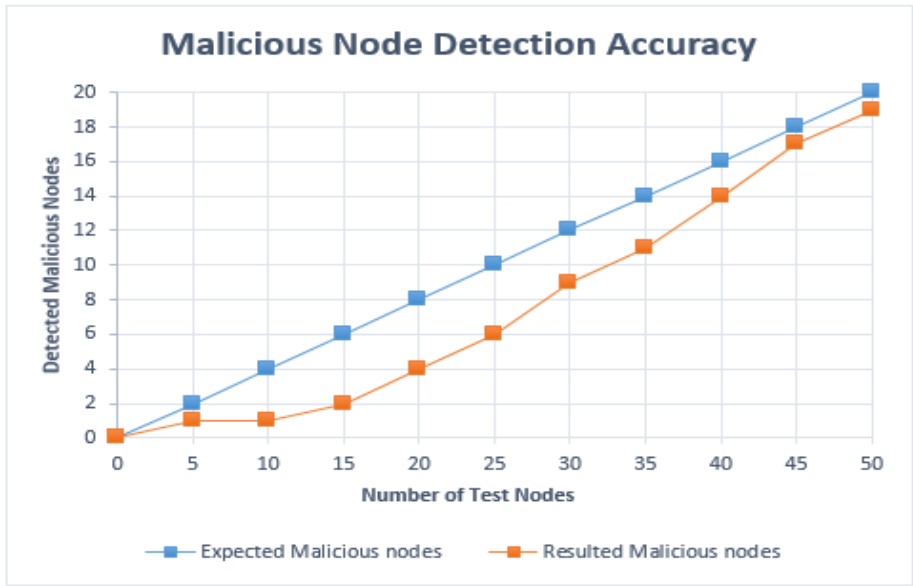


Figure 4.35: Malicious Node Detection for different Test nodes

From the above figures, it is represented that the framework efficiently detects malicious nodes in the MANET with an overall accuracy of up to 93%. Further, the system only gets very low false positive rates, which are under 4% for tested cases.

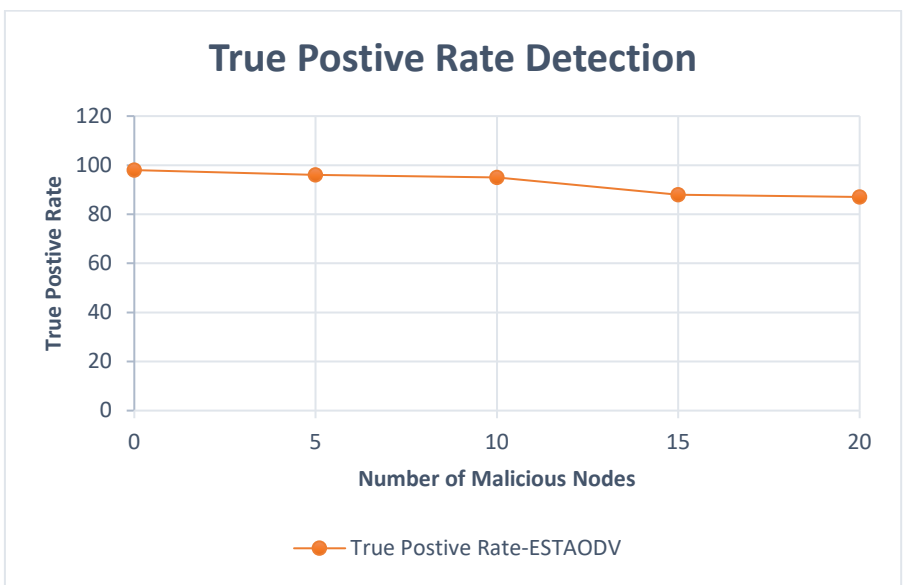


Figure 4.36: True Positive Rate for Different Malicious Nodes

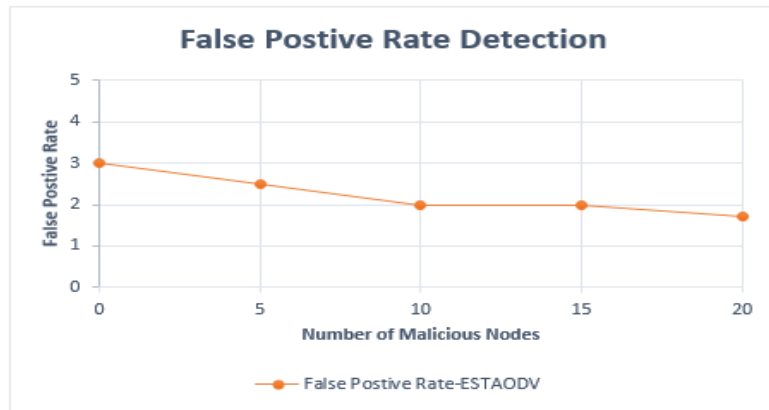


Figure 4.37: False Positive Rate for Different Malicious Nodes

4.2.10 Impact of the framework on detection of Collaborative malicious nodes

Experimental results were recorded on analyzing the performance of detection capability of collaborative malicious nodes based on the protocol implementation. Same data set for the Malicious Node Detection for different Test nodes in Figure 4.35 is used for the malicious node classification in this stage. As shown in the Figure 4.38 when the number of test nodes increasing collaborative malicious node detection rate is also high. There are no detected collaborative nodes in the resulted output up to 10 number of test nodes due to a small number of nodes in the network and low communication links among the nodes. For the second test case where it has 5 test nodes, after considering there are only two malicious nodes either both nodes can be malicious or collaborative malicious nodes since there should be at least two collaborative nodes in a cluster.

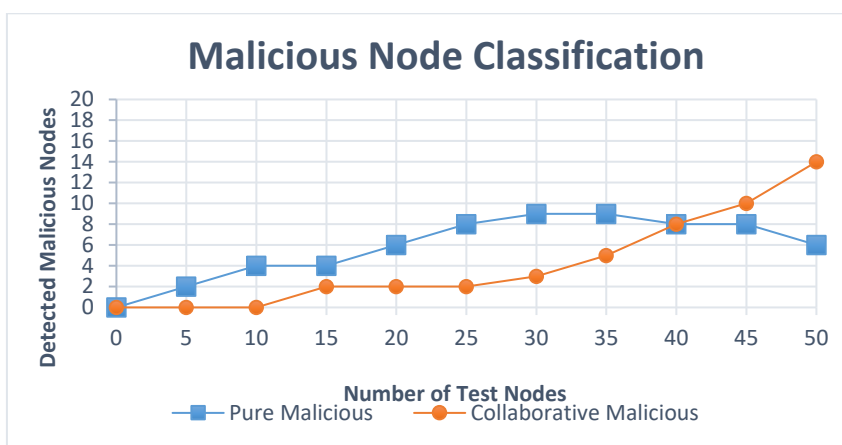


Figure 4.38: Malicious node classification for different test nodes

After some time, nodes in the network are going to build more collaborations and form clusters based on their interactions within the network. This collaboration can be either a good collaboration or the malicious behaviour in the cluster. When it has malicious collaborative behaviour nodes within the cluster is going to act as trustworthy nodes within the network but originally they have malicious behaviour within the cluster. So, they can act differently for the cluster and for the network. At the same time, there can be multiple malicious collaborative clusters as well. That is why in the Figure 4.38 after 30-35 test nodes pure malicious nodes count is low and collaborative malicious node rate is increasing.

The results show that the performance of the framework is satisfactory in identifying collaborative malicious nodes with high true positives and low false positives. This can be seen clearly in figure numbers 4.39 and 4.40. Results indicate that even at a higher number of collaborative malicious nodes, the true positive rate remains at very high levels where classic AODV fails to identify. Further, false positive rate remains below 4%, which is a good indication of the detection results.

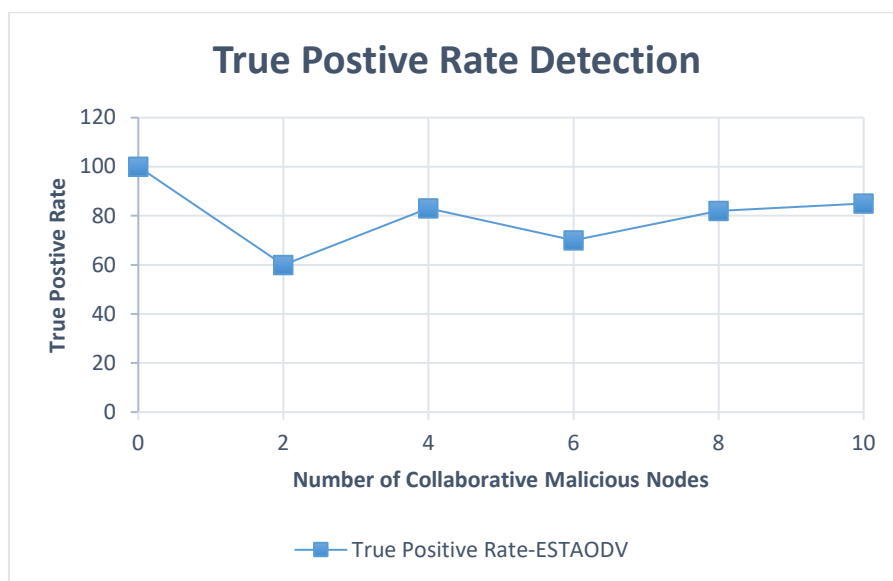


Figure 4.39: True Positive Rate for different Collaborative Malicious nodes

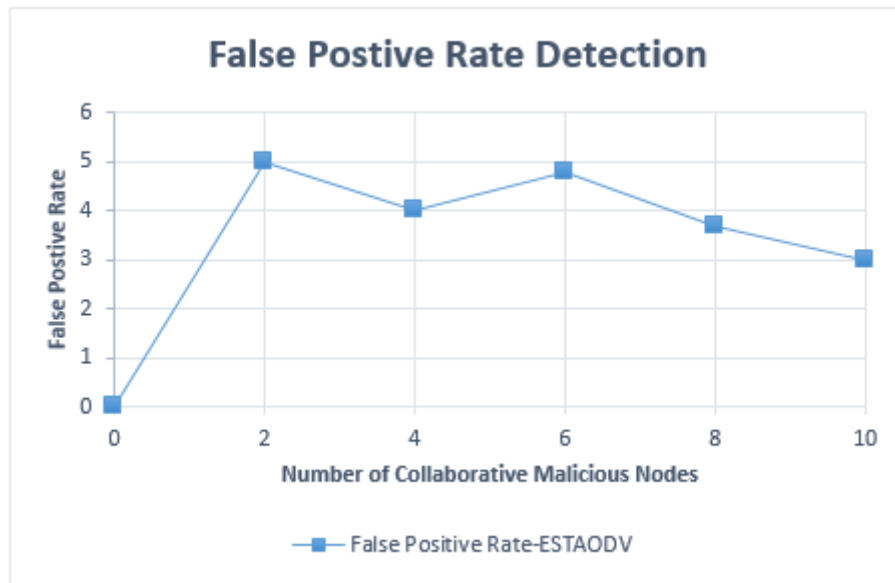


Figure 4.40: False Positive Rate for different Collaborative Malicious nodes

From the above figures, it is represented that the framework is having an identifiable difference in detecting collaborative malicious nodes. This can be seen in figure number 4.39, when collaborative node size is two there is a drop in the True Positive Rate. This may be due to the dynamic change of the TrVs or the dynamic behaviour of the collaborative malicious nodes. Time to time the behaviour of the collaborative nodes can be changed. For some of the nodes it is going to act as trustworthy nodes and for some of the nodes, it reacts as a malicious node. In the spiral model there is a way to track their dynamic behaviour and through that, it can suspect it is a collaborative node. According to the mechanism in the spiral model if one node going to evaluate another node it is going to consider direct trust and the recommendations getting from the neighbours. In that case, if those two values are more different from each other we can suspect some neighbours giving fake recommendations because of the malicious collaboration among those nodes with that particular node. Same time this can be due to their interactions count or the knowledge with the node which is going to evaluate. By considering all these factors collaborative malicious node detection is challengeable, and it can be changed time to time as we can see it in figure numbers 4.39 and 4.40.

Chapter 5 - Conclusion and Future Work

This chapter explains the conclusion of the research work and show some future directions for the research.

5.1 Conclusions

The first part of this research project is about a trust-based mechanism is introduced to identify trustworthy nodes in a social network. And here it is giving the opportunity to the nodes to introduce their own trust levels and apply those to this platform. Hence, it is easy for corresponding parties to identify the highly and medially trusted paths through particular nodes and links.

Twenty-two different anonymized Facebook datasets have been used in developing this mechanism. From that, randomly selected two-thirds of the datasets (15 datasets) were used to train, and the other one third (7 datasets) has been utilized for testing. As mentioned in chapter 5, the results were obtained, and system evaluations were performed. With the observed results, Mean Absolute Error, Root Mean Square Error, system accuracy percentage, mean value of the error, and the mean value of the accuracy were measured. Because of the reasons that the system has been tested with anonymized social network data and the system accuracy is evaluated by comparing its results with recommendations, it can be concluded that the system has a higher level of accuracy. And also, it is important to note that the system accuracy is relatively high for any size of the dataset.

Regardless of the size of the social network, the system is giving fairly low values for MAE and RMSE. To ensure the system accuracy, the second filter named Most Untrusted Nodes is also utilized. The purpose of this filter is to filter out the nodes which are considered as 'untrusted' inside the network. Most of the unknown nodes, as well as unclosed nodes, have been filtered out while giving the assurance of the accuracy of this implemented system. Despite the above-mentioned efforts, this mechanism may still face many challenging problems. When considering about trust in virtual worlds such as social networks, it is varying from that in the real world because of the limited personal knowledge in cyberspace. Not only that but with the inadequate personal relationships built on virtual interactions also, trust is hard to measure. Hence trust evaluation in online social networks is hard and careful analysis is

necessary for this task. In the Bayesian Belief Network constructed, three main factors have been considered which are affecting the TrV. Those are;

- Popularity of an individual within his network
- Importance of an individual to his network
- The influence he can do for his network

But there is a main factor to be considered while evaluating such a system. That is Human Factors, which cannot be measured or estimated through a computerized system. Here the system is filtering nodes relative to a particular network. Main factors considered to be a trusted node inside the network are mentioned above. All those factors are with respect to the whole network. Hence the TrV of an individual is also relative to the entire network. Because of that, the best way of system evaluation is to evaluate all users in the network separately and get the average value from that. But since it is very hard to perform such a process for user recommendations, here only the owner of the particular network has been joined to evaluate his/her own Trusted Network as the representative of the entire network. With that limitation of the evaluation process, some confusion has occurred.

The second framework and protocol (Entropy-based Spiral Trust Model) is a novel algorithm which provided a conclusive calculation of TrVs which is present in a MANET. Further, the protocol follows a direct and indirect trust calculation methodology, which is regarded to be more accurate. Next, the spiral recommendation model is a novel algorithm, which penalizes the collaborative malicious nodes which can be hard to detect and remove from a network of this nature. Results of this algorithm were forwarded to the DRL algorithm for predicting secure nodes.

Finally, an intelligent approach is taken to calculate the predicted TrVs which are important for detecting trustworthy nodes. Hence, research uses a novel deep reinforcement learning algorithm implementation. Even though there are similar reinforcement learning based algorithms being used in routing performance, QoS very little effort was taken to develop such for security. Hence the importance of using Deep reinforcement learning algorithm for this purpose is more relevant. Once the results were analyzed it gave a positive indication of the

framework. True positive rate and false positive rate were analyzed for identification of secure and malicious nodes which was successful. Further, introduction of these algorithms gave a minimal burden to the routing network in terms of delay, throughput and packet delivery ratio. Even though there were no big improvements in these parameters, the research managed to achieve the primary objective, which was to do secure routing.

The overall performance of RLTM/ESTAODV in terms of network load is better than that of AODV. When the network size is increased, it affects the performance of the novel algorithm due to the overhead it adds to the network. The end-to-end delay of the algorithm remains better in mobility and network size due to neglecting malicious nodes. In the case of Link quality, this algorithm is having less performance compared to AODV. This is one of the weaknesses of the current protocol.

5.2 Anticipated Benefits and Contribution to the Body of Knowledge

Mobile Ad-hoc Networks (MANETs) are fast spreading technology across the globe due to its attractive advantages. The main difference between a MANET and a conventional network is that a MANET does not have a predefined infrastructure. Due to MANET's dynamic topology and infrastructure less environment, building up routing protocols to security mechanisms have become an immense challenge.

The main advantage of a MANET is the main reason for the complexity faced when inventing routing protocols and security mechanisms which is not having a predefined infrastructure. It has cause to use different routing protocols and security mechanisms depending on the scenario where MANET is used according to the requirements of the network. In a rescue operation, most essential needs would be availability and reliability where security is not an immense concern. In contrast in a battlefield, security is a vital issue. This research addresses these issues and finds a solution which can be applied to a wide range of applications. Hence, the benefit of this research work is universal in creating secure MANETs.

Though there have been many researches on SNA, since Social Network security is in a primary stage, this will be an added value to the field since this is focusing on 4 major factors which impact on trust level and so as to the security aspect. Improvements of the trustworthiness evaluation, reputation evaluation, user classification and spammer detection will be added

benefits to the research area. Creating a holistic mechanism for which can be used within common social networks will be a significant contribution to the existing body of knowledge of research area.

Next, a novel trust calculation methodology with a MANET routing protocol was designed. It was designed using the information entropy methodology which helped to accurately retrieve and store trust-based information from the mobile nodes. Further, implemented a “spiral” model helped to identify collaborative malicious nodes which is difficult to identify. Both this method collaboratively helped to identify trustworthy nodes from untrustworthy once. But the major issue remained in the research domain about the mobility and changes. Mobility is a major issue because it can affect in changing the structure, topology and information within the network each second. Hence the Deep Reinforcement learning prediction algorithm will manage those changes and will predict the TrVs. This approach provides a novel principle of how a MANET can be configured to a secure environment without any or minimum need of cryptographic controls.

5.3 Research Constraints

- Extracting data is having limitations- provides access only to public data, no access to personal data, in terms of social network data.
- Solution to be implemented is expected to be tested in the real-time environment as well. Since performing in a real-time venture is difficult, have to make the simulation using NS -3.
- When extracting data from flow monitor, there is a limitation of the hyper-parameters, which can be used to do the prediction.
- To perform research output in real environment mobile devices do not have enough resources that can be accessed using reinforcement learning.
- Since we used a virtual machine as the operating system it may perform less speed and get more time to deliver the output results. Hence difficult to predict more amount of nodes.

5.4 Future Directions

The algorithms and frameworks used in the thesis can be modified to work with more intelligence. As an example, all the algorithms can be interpreted as multi-agent environments using game theory coupled up with reinforcement learning. This will provide cooperative game theory solutions, which provide an overall improvement in detecting malicious behaviours within the network. Further improvement can be made in overhead created by additional packets generated within the network by using the above-mentioned methodology. Next implementations can be more tightly integrated when there are more social friendly API's available to lower layer protocols. As an example, the social trust development can be further modified and get better accurate results with modern day social network API's. But currently, there are a lot of limitations available to use such API's freely.

Appendices

Appendix 1: Gephi Filter Plugin Code for “Trusted Network”

```
public boolean evaluate(Graph graph, Node node)
{
    int D;
    float EC;
    float BC;
    float CC;

    D=Integer.parseInt(node.getAttributes().getValue("Degree").toString());

    EC=Float.parseFloat(node.getAttributes().getValue("Eigenvector
Centrality").toString());

    BC=Float.parseFloat(node.getAttributes().getValue("Betweenness
Centrality").toString());

    CC=Float.parseFloat(node.getAttributes().getValue("Closeness
Centrality").toString());

    String statusD="";
    String statusEC="";
    String statusBC="";
    String statusCC="";
    String HighTrust="no";

    float Dmax=0.0f;
    float Dmax_Temp=0.0f;
    float BCmax=0.0f;
    float BCmax_Temp=0.0f;
    float CCmax=0.0f;
    float CCmax_Temp=0.0f;

    for(int i=0;i<graph.getNodeCount();i++)
    {
        try
        {
            Dmax_Temp=Float.parseFloat(graph.getNode(i).getAttributes().getValu
e("Degree").toString());
            if (Dmax_Temp>=Dmax)
                Dmax=Dmax_Temp;
        }catch(Exception ex)
        {
        }
    }

    for(int i=0;i<graph.getNodeCount();i++)
```

```

    {
        try
        {
            BCmax_Temp=Float.parseFloat(graph.getNode(i).getAttributes().getVal
ue("Betweenness Centrality").toString());
            if (BCmax_Temp>=BCmax)
                BCmax=BCmax_Temp;
        }catch(Exception ex)
        {
        }
    }

    for(int i=0;i<graph.getNodeCount();i++)
    {
        try
        {
            CCmax_Temp=Float.parseFloat(graph.getNode(i).getAttributes().getVal
ue("Closeness Centrality").toString());
            if (CCmax_Temp>=CCmax)
                CCmax=CCmax_Temp;
        }catch(Exception ex)
        {
        }
    }

    /////// Degree
    if(D >= (Dmax/3))
        statusD = "H";
    else if(D >= (Dmax/6))
        statusD = "M";
    else if(D < (Dmax/6))
        statusD = "L";

    /////// Betweenness Centrality
    if(BC >= (BCmax/3))
        statusBC = "H";
    else if(BC >= (BCmax/6))
        statusBC = "M";
    else if(BC < (BCmax/6))
        statusBC = "L";

    ///// Eigenvector Centrality
    if (EC >= 0.2)
        statusEC = "H";
    else if(EC < 0.2 && EC >= 0.1)
        statusEC = "M";
    else if(EC < 0.1)
        statusEC = "L";

    /////// Closeness Centrality
    if(CC >= (CCmax/3))
        statusCC = "H";
    else if(CC >= (CCmax/6))

```

```

        statusCC = "M";
    else if(CC < (CCmax/6))
        statusCC = "L";

    ////    Highly Trust Nodes (GREEN)    T>=.8

    if(("H".equals(statusD) &&"H".equals(statusEC) &&"H".equals(statusBC) &&"H".equals(statusCC)) ||
        ("H".equals(statusD) &&"H".equals(statusEC) &&"H".equals(statusBC) &&"M".equals(statusCC)) ||
        ("H".equals(statusD) &&"H".equals(statusEC) &&"H".equals(statusBC) &&"L".equals(statusCC)) ||
        ("H".equals(statusD) &&"M".equals(statusEC) &&"H".equals(statusBC) &&"H".equals(statusCC)) ||
        ("H".equals(statusD) &&"M".equals(statusEC) &&"H".equals(statusBC) &&"M".equals(statusCC)) ||
        ("M".equals(statusD) &&"H".equals(statusEC) &&"H".equals(statusBC) &&"H".equals(statusCC)) ||
        ("M".equals(statusD) &&"H".equals(statusEC) &&"H".equals(statusBC) &&"M".equals(statusCC)) ||
        ("M".equals(statusD) &&"H".equals(statusEC) &&"H".equals(statusBC) &&"L".equals(statusCC)) ||
        ("L".equals(statusD) &&"H".equals(statusEC) &&"H".equals(statusBC) &&"H".equals(statusCC)))
    {
        node.getNodeData().setColor(0,255,0);
        graph.getNode(HighTrust="yes");
    }

    ////    Medially Trust Nodes (BLUE)    T>=.65

    if(("H".equals(statusD) &&"H".equals(statusEC) &&"M".equals(statusBC) &&"H".equals(statusCC)) ||
        ("H".equals(statusD) &&"H".equals(statusEC) &&"M".equals(statusBC) &&"M".equals(statusCC)) ||
        ("H".equals(statusD) &&"M".equals(statusEC) &&"H".equals(statusBC) &&"L".equals(statusCC)) ||
        ("H".equals(statusD) &&"L".equals(statusEC) &&"H".equals(statusBC) &&"H".equals(statusCC)) ||
        ("M".equals(statusD) &&"H".equals(statusEC) &&"M".equals(statusBC) &&"H".equals(statusCC)) ||
        ("M".equals(statusD) &&"M".equals(statusEC) &&"H".equals(statusBC) &&"H".equals(statusCC)) ||
        ("M".equals(statusD) &&"M".equals(statusEC) &&"H".equals(statusBC) &&"M".equals(statusCC)) ||
        ("L".equals(statusD) &&"H".equals(statusEC) &&"H".equals(statusBC) &&"M".equals(statusCC)) ||
        ("L".equals(statusD) &&"H".equals(statusEC) &&"H".equals(statusBC) &&"L".equals(statusCC)) ||
        ("L".equals(statusD) &&"M".equals(statusEC) &&"H".equals(statusBC) &&"H".equals(statusCC)) ||
        ("H".equals(statusD) &&"H".equals(statusEC) &&"M".equals(statusBC) &&"L".equals(statusCC)))

```

```
quals(statusCC) ||
("H".equals(statusD) &&"L".equals(statusEC) &&"H".equals(statusBC) &&"M".e
quals(statusCC) ||
("M".equals(statusD) &&"M".equals(statusEC) &&"H".equals(statusBC) &&"L".e
quals(statusCC))
{
    node.getNodeData().setColor(0,0,255);
    graph.getNode(HighTrust="yes");
}

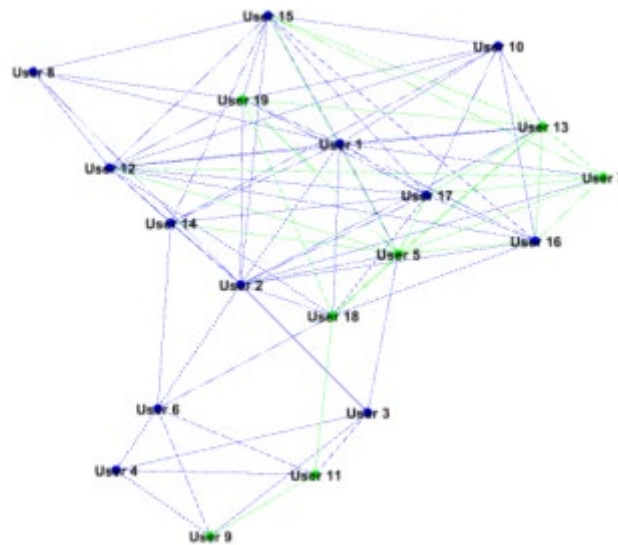
return "yes".equals(HighTrust);
}
```

Appendix 2: Test datasets used to evaluate ‘Gephi Filter Plugin’ and resulted networks

Test Dataset I: Trusted Network

Total nodes count = 581

Nodes count in the Trustworthy Network = 19



Trustworthy Network- Test Dataset I

Recommendation from Data Set:

Trustworthy nodes count = 19

Untrustworthy nodes count = 0

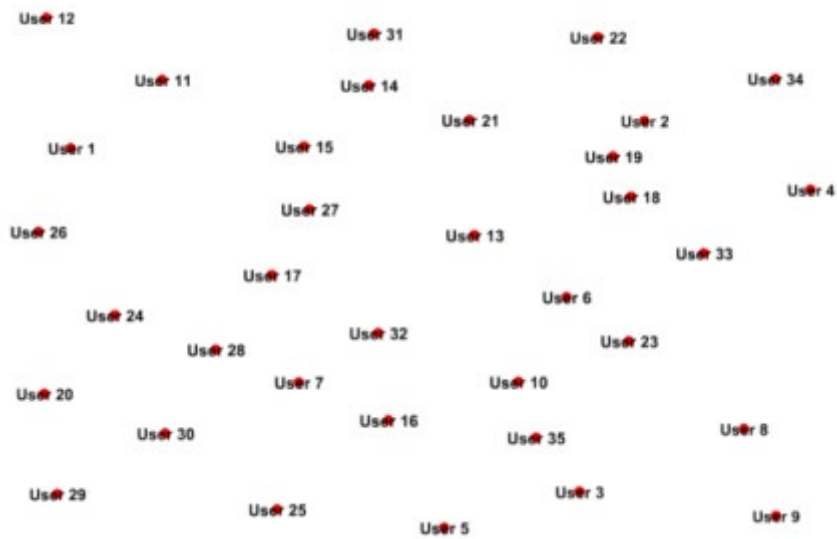
MAE = 0

RMSE = 0

Test Dataset I: Most Untrusted Nodes

Total nodes count = 581

Most Untrustworthy Nodes count = 35



Most Untrustworthy Nodes – Test Dataset I

Recommendation from Data Set:

Untrustworthy nodes count = 32

Trustworthy nodes count = 3

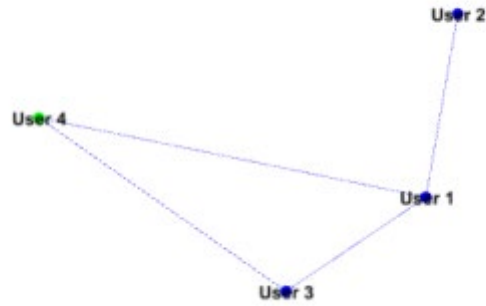
MAE = 0.085714

RMSE = 0.29277

Test Dataset II: Trustworthy Network

Total nodes count = 1064

Nodes count in the Trustworthy Network = 4



Trustworthy Network – Test Data II

Recommendation from Data Set:

Trustworthy nodes count = 4

Untrustworthy nodes count = 0

MAE = 0

RMSE = 0

Test Dataset II: Most Untrustworthy Nodes

Total nodes count = 1064

Most Untrustworthy Nodes count = 83



Most Untrustworthy Nodes – Test Dataset II

Recommendation from Data Set:

Untrustworthy nodes count = 79

Trustworthy nodes count = 4

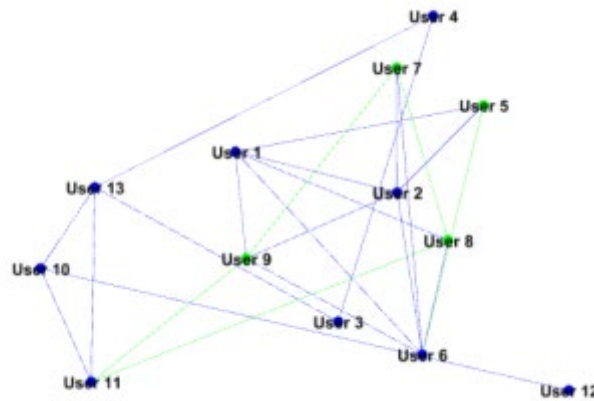
MAE= 0.048193

RMSE = 0.219529

Test Dataset III: Trustworthy Network

Total nodes count = 378

Nodes count in the Trustworthy Network = 13



Trustworthy Network – Test Dataset III

Recommendation from Data Set:

Trustworthy nodes count = 9

Untrustworthy nodes count = 4

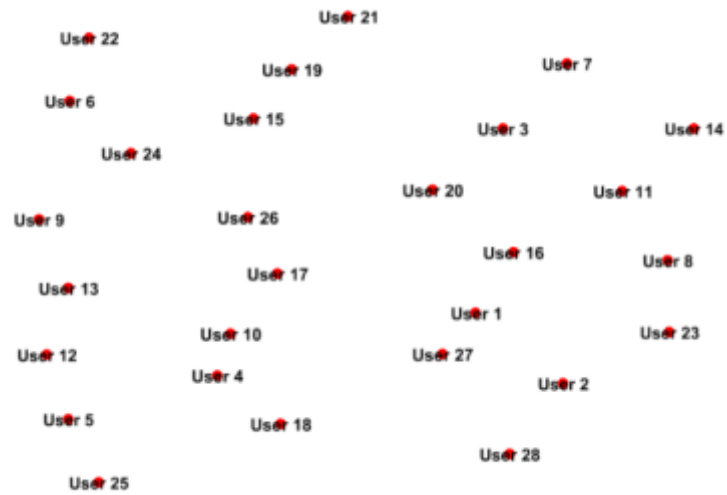
MAE = 0.307692

RMSE = 0.5547

Test Dataset III: Most Untrustworthy Nodes

Total nodes count = 378

Most Untrustworthy Nodes count = 28



Most Untrustworthy Nodes – Test Dataset III

Recommendation from Data Set:

Untrustworthy nodes count = 18

Trustworthy nodes count = 10

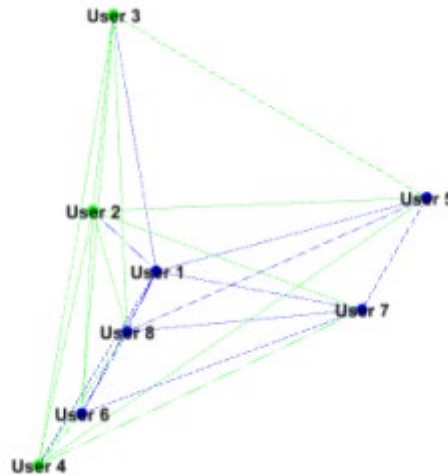
MAE = 0.357143

RMSE = 0.597614

Test Dataset IV: Trustworthy Network

Total nodes count = 93

Nodes count in the Trustworthy Network = 8



Trustworthy Network – Test Dataset IV

Recommendation from Data Set:

Trustworthy nodes count = 8

Untrustworthy nodes count = 0

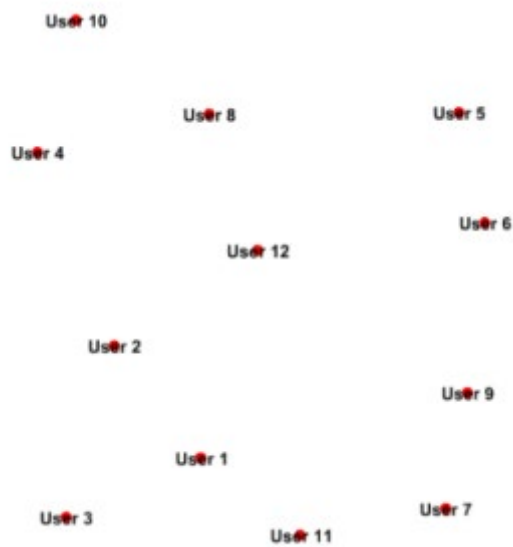
MAE = 0

RMSE = 0

Test Dataset IV: Most Untrustworthy Nodes

Total nodes count = 93

Most Untrustworthy Nodes count = 12



Most Untrustworthy Nodes – Test Dataset IV

Recommendation from Data Set:

Untrustworthy nodes count = 10

Trustworthy nodes count = 2

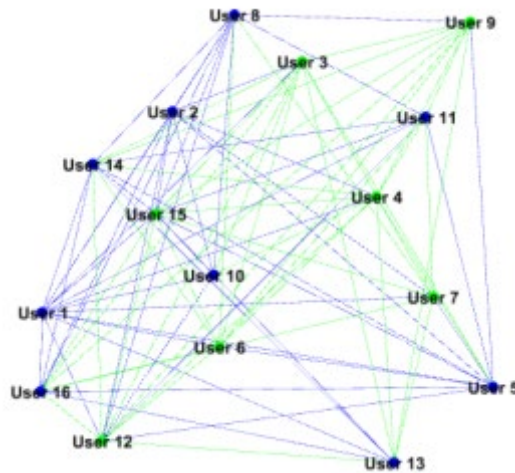
MAE = 0.166667

RMSE = 0.408248

Test Dataset V: Trustworthy Network

Total nodes count = 691

Nodes count in the Trustworthy Network = 16



Trustworthy Network – Test Dataset V

Recommendation from Data Set:

Trustworthy nodes count = 12

Untrustworthy nodes count = 4

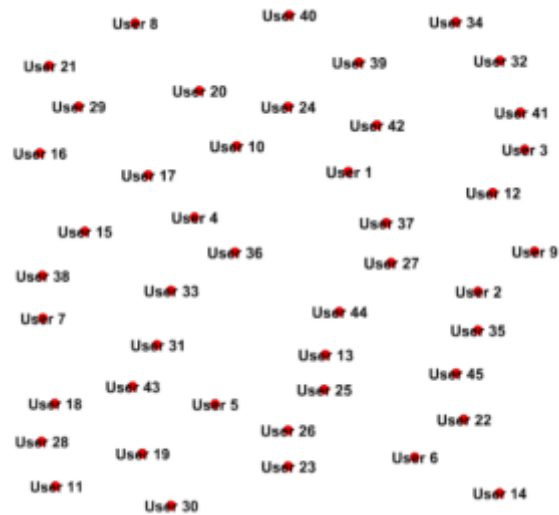
MAE = 0.25

RMSE = 0.5

Test Dataset V: Most Untrustworthy Nodes

Total nodes count = 691

Most Untrustworthy Nodes count = 45



Most Untrustworthy Nodes – Test Dataset V

Recommendation from Data Set:

Untrustworthy nodes count = 37

Trustworthy nodes count = 8

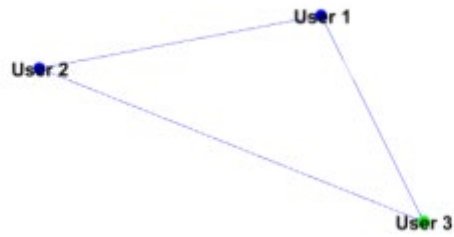
MAE = 0.177778

RMSE = 0.421637

Test Dataset VI: Trustworthy Network

Total nodes count = 10

Nodes count in the Trustworthy Network = 3



Trustworthy Network – Test Dataset VI

Recommendation from Data Set:

Trustworthy nodes count = 3

Untrustworthy nodes count = 0

MAE = 0

RMSE = 0

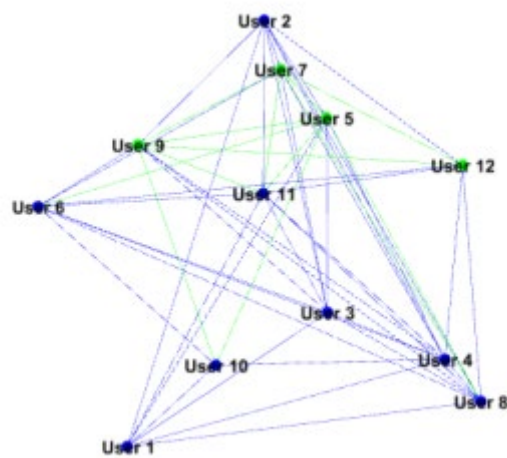
Test Dataset VI: Most Untrustworthy Nodes

No nodes have been filtered as Most Untrustworthy Nodes.

Test Dataset VII: Trustworthy Network

Total nodes count = 394

Nodes count in the Trustworthy Network = 12



Trustworthy Network – Test Dataset VII

Recommendation from Data Set:

Trustworthy nodes count = 10

Untrustworthy nodes count = 2

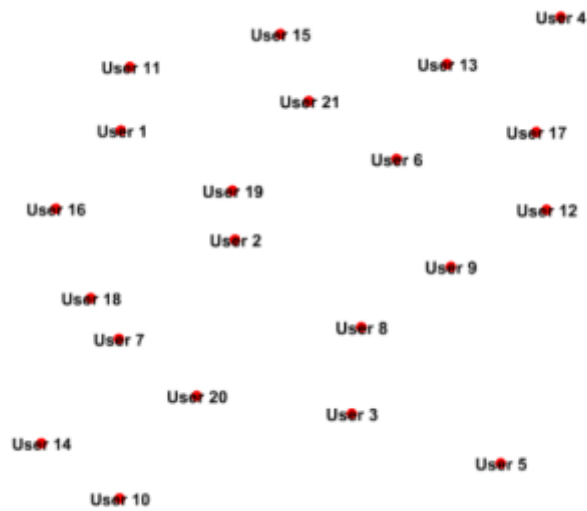
MAE = 0.166667

RMSE = 0.408248

Test Dataset VII: Most Untrustworthy Nodes

Total nodes count = 394

Most Untrustworthy Nodes count = 21



Most Untrustworthy Nodes – Test Dataset VII

Recommendation from Data Set:

Untrustworthy nodes count = 6

Trustworthy nodes count = 15

MAE = 0.714286

RMSE = 0.845154

Appendix 3: Source Code for RL Model

rlmodel.py

```
import random
import numpy as np
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM, Embedding, Dropout
from keras.optimizers import SGD, Adam
from beautifultable import BeautifulTable
import matplotlib.pyplot as plt
import pylab

class DQNAgent:
    def __init__(self, state_size, actions):
        self.state_size = state_size
        self.action_size = actions
        self.gamma = 0.95
        self.epsilon = 0.01
        self.epsilon_min = 0.01
        self.epsilon_decay = 0.995
        self.learning_rate = 0.001
        self.memory = []
        self.model = self.build_model()

    def build_model(self):
        self.model = Sequential()
        self.model.add(Embedding(self.state_size, 10))
        self.model.add(Dropout(0.2))
        self.model.add(LSTM(32, activation='relu', return_sequences=True))
        self.model.add(LSTM(16, activation='relu', return_sequences=True))
        self.model.add(LSTM(8, activation='relu', return_sequences=True))
        self.model.add(Dense(self.action_size, activation='linear'))
        sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
        self.model.compile(loss='mse', optimizer=Adam(lr=self.learning_rate),
                           metrics=['accuracy'])

        return self.model

    def remember(self, state, action, reward, next_state, done):
        self.memory.append((state, action, reward, next_state, done))
```

```

def act(self, state):
    if np.random.rand() <= self.epsilon:
        return np.random.randint(1, self.action_size)
    act_values = self.model.predict(state)
    return np.amax(act_values)

def q_values(self, state):
    if np.random.rand() <= self.epsilon:
        return np.random.randint(1, self.action_size)
    act_values = self.model.predict(state)
    return np.amax(act_values[0])

def replay(self, batch_size):
    minibatch = random.sample(self.memory, batch_size)
    for state, action, reward, next_state, done in minibatch:
        target = reward
        if not done:
            target = reward + self.gamma * \
                np.amax(self.model.predict(next_state)[0])
        target_f = self.model.predict(state)
        target_f[0][action] = target
        tensorboard = TensorBoard(log_dir='logs/{}'.format(time()),
                                   batch_size=batch_size, write_images=True,
                                   embeddings_freq=10)
        history = self.model.fit(state, target_f, epochs=1000, verbose=0,
                                  callbacks=[tensorboard])

    if self.epsilon > self.epsilon_min:
        self.epsilon *= self.epsilon_decay

```

Wifi_adv_flowmon.py

```
import sys
import matplotlib
import os
os.environ['TF_CPP_MIN_LOG_LEVEL']='2'
import tensorflow as tf

import pandas as pd
from keras.preprocessing import sequence
from sklearn import preprocessing

matplotlib.use('TkAgg')
import numpy as np
import scipy
import ns.applications
import ns.core
import ns.flow_monitor
import ns.internet
import ns.mobility
import ns.network
import ns.aadv
import ns.csma
import ns.wifi
from rlmodel import DQNAgent
from rlmodel import *
from beautifultable import BeautifulTable

try:
    import ns.visualizer
except ImportError:
    pass

DISTANCE = 100 # (m)
NUM_NODES_SIDE = 7
actor = []
action = 0
q_value = 0.0
X_train = []
discount_factor = 0.75

        actor.remember(X_prev, action, reward, X_train, True)
        actor.replay(32)

else:
    print monitor.SerializeToXmlFile(cmd.Results, True, True)
```

```

if cmd.Plot is not None:
    import pylab
    delays = []
    for flow_id, flow_stats in monitor.GetFlowStats():
        tupl = classifier.FindFlow(flow_id)
        if tupl.protocol == 17 and tupl.sourcePort == 698:
            continue
        try:
            delays.append(flow_stats.delaySum.GetSeconds() / flow_stats.rxPackets)
        except ZeroDivisionError:
            flow_stats.delaySum.GetSeconds() == 0

    pylab.hist(delays, 20)
    pylab.xlabel("Delay (s)")
    pylab.ylabel("Number of Flows")
    pylab.show()

if cmd.Plot is not None:
    import pylab
    jitters = []
    for flow_id, flow_stats in monitor.GetFlowStats():
        tupl = classifier.FindFlow(flow_id)
        if tupl.protocol == 17 and tupl.sourcePort == 698:
            continue
        try:
            jitters.append(flow_stats.jitterSum.GetSeconds() / flow_stats.rxPackets)
        except ZeroDivisionError:
            flow_stats.jitterSum.GetSeconds() == 0

    pylab.hist(jitters, 20)
    pylab.xlabel("Jitter (s)")
    pylab.ylabel("Number of Flows")
    pylab.show()

```


Appendix 4 : Statement of Contribution by Others

To Whom It May Concern I, Prabath Lakmal Rupasinghe, contributed the conceptual design, experimental design and the results analysis to the paper/publication entitled Evaluation of Trustworthiness for Online Social Networks Using Advanced Machine Learning, Hansi Mayadunne, **Rupasinghe P.L**, conference on International Conference on Advanced in Computing Technology, 2018, Sri Lanka



(Signature of Candidate)

I, as a Co-Author, endorse that this level of contribution by the candidate indicated above is appropriate.

Hansi Mayadunne

: 

To Whom It May Concern I, Prabath Lakmal Rupasinghe, contributed the conceptual design, experimental design and the results analysis to the paper/publication entitled A reinforcement learning approach to enhance the trust level of MANETs, Gihani Jinarajadasa, Wayomi Jayantha, Rupasinghe P.L, Iain Murray, conference on Smart Computing and Systems Engineering, 2018, Sri Lanka

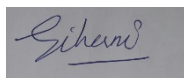


(Signature of Candidate)

I, as a Co-Author, endorse that this level of contribution by the candidate indicated above is appropriate.

Gihani Jinarajadasa

:



Iain Murray

:



To Whom It May Concern I, Prabath Lakmal Rupasinghe, contributed the conceptual design, experimental design and the results analysis to the paper/publication entitled Improving Trusted Routing by Identifying Malicious Nodes in a MANET Using Reinforcement Learning, Shanen Leen De Silva, Hansi Mayadunna, Iesha Wedage, Sasanka Pabasara, **Rupasinghe P.L**, Chethana Liyanapathirana, Krishnadeva Kesavan, Chamira Nawarathna, Kalpa Kalhara Sampath, conference on International Conference on Advances in ICT for Emerging Regions, 2017, Sri Lanka



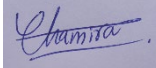
(Signature of Candidate)

I, as a Co-Author, endorse that this level of contribution by the candidate indicated above is appropriate.

Hansi Mayadunne

: 

Chamira Nawarathna

: 

Kalpa Kalhara Sampath

: 

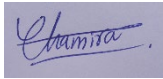
To Whom It May Concern I, Prabath Lakmal Rupasinghe, contributed the conceptual design, experimental design and the results analysis to the paper/publication entitled Enhancing the Security of OLSR Protocol Using Reinforcement Learning, **Rupasinghe P.L**, Chamira Nawarathna, Kalpa Kalhara Sampath, conference on National Information Technology Conference, 2017, Sri Lanka



(Signature of Candidate)

I, as a Co-Author, endorse that this level of contribution by the candidate indicated above is appropriate.

Chamira Nawarathna

: 

Kalpa Kalhara Sampath

: 

To Whom It May Concern I, Prabath Lakmal Rupasinghe, contributed the conceptual design, experimental design and the results analysis to the paper/publication entitled A Model to Represent Recommendation Based Trust for MANET Using Reinforcement Learning, **Rupasinghe P.L**, Chamira Nawarathna, Kalpa Kalhara Sampath, conference on National Information Technology Conference, 2017, Sri Lanka.

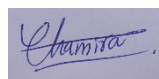


(Signature of Candidate)

I, as a Co-Author, endorse that this level of contribution by the candidate indicated above is appropriate.

Chamira Nawarathna

:



Kalpa Kalhara Sampath

:



To Whom It May Concern I, Prabath Lakmal Rupasinghe, contributed the conceptual design, experimental design and the results analysis to the paper/publication entitled Predictive Analytics with online data for WSO2 Machine Learner with the support of Ensemble method, **Rupasinghe P.L**, Heshani Herath, Ishani Pathinayake, Ashani Diaz, Indujayani Karthigesu, Krishnadeva Kesavan , Chethana Liyanapathirana, Sripa Vimuthi, IET Conference, 2016, Sri Lanka



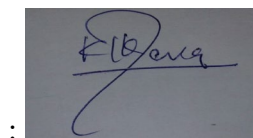
(Signature of Candidate)

I, as a Co-Author, endorse that this level of contribution by the candidate indicated above is appropriate.

Chethana Liyanapathirana



Krishnadeva Kesavan



To Whom It May Concern I, Prabath Lakmal Rupasinghe, contributed the conceptual design, experimental design and the results analysis to the paper/publication entitled SDN Based Security Solution for Legislative Email Communications, **Rupasinghe P.L**, Murray I, ICCCA, 2016, India



(Signature of Candidate)

I, as a Co-Author, endorse that this level of contribution by the candidate indicated above is appropriate.

Iain Murray

:



To Whom It May Concern I, Prabath Lakmal Rupasinghe, contributed the conceptual design, experimental design and the results analysis to the paper/publication entitled Trustworthy Provenance Framework for Document Workflow Provenance, **Rupasinghe P.L**, Murray I. ICCTICT 2016, India



(Signature of Candidate)

I, as a Co-Author, endorse that this level of contribution by the candidate indicated above is appropriate.

Iain Murray

:



To Whom It May Concern I, Prabath Lakmal Rupasinghe, contributed the conceptual design, experimental design and the results analysis to the paper/publication entitled Trust-Based Framework for Handling Communication Using Social Networks as Applied to Mobile Sensor Based Indoor Navigation System, **Rupasinghe P.L**, Murray I, IPIN, 2014, Korea



(Signature of Candidate)

I, as a Co-Author, endorse that this level of contribution by the candidate indicated above is appropriate.

Iain Murray

:



To Whom It May Concern I, Prabath Lakmal Rupasinghe, contributed the conceptual design, experimental design and the results analysis to the paper/publication entitled Authentication Algorithm to MANETs through Challenge-Response Based architecture, **Rupasinghe P.L**, Tennekoon R, Anushka B, Visagan S, Hettiarachchi B, Basnayake P, National Conference on Technology and Management, 2013, Sri Lanka

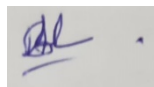


(Signature of Candidate)

I, as a Co-Author, endorse that this level of contribution by the candidate indicated above is appropriate.

Anushka B

:



To Whom It May Concern I, Prabath Lakmal Rupasinghe, contributed the conceptual design, experimental design and the results analysis to the paper/publication entitled Efficient, Authentication and Access control Implementation in Mobile Ad hoc Networks (MANET) as applied to Indoor Navigation Guidance System for Vision Impaired People, **Rupasinghe P.L**, Murray I, published under Peer-Reviewed Section, IPIN, 2012, Australia



(Signature of Candidate)

I, as a Co-Author, endorse that this level of contribution by the candidate indicated above is appropriate.

Iain Murray

:



REFERENCES

- [1] V. Gligor, “Security of emergent properties in ad-hoc networks (transcript of discussion),” in *International Workshop on Security Protocols*, 2004, pp. 256–266.
- [2] L. Zhou and Z. J. Haas, “Securing ad hoc networks,” *IEEE Netw.*, vol. 13, no. 6, pp. 24–30, 1999.
- [3] H. Wang, Y. Wang, and J. Han, “A security architecture for tactical mobile ad hoc networks,” in *Knowledge Discovery and Data Mining, 2009. WKDD 2009. Second International Workshop on*, 2009, pp. 312–315.
- [4] Abhayasinghe, K. N. (2016). "Human Gait Modelling with Step Estimation and Phase Classification Utilising a Single Thigh Mounted IMU for Vision Impaired Indoor Navigation" (Doctoral dissertation). Retrieved from https://espace.curtin.edu.au/bitstream/handle/20.500.11937/280/242452_Abhayasinghe%20K%202016.pdf?sequence=2&isAllowed=y.
- [5] D. Mane and D. Gothwal, “Improved Security for Attacks in MANET using AODV.”
- [6] R. K. Bar, J. K. Mandal, and M. M. Singh, “QoS of MANet Through Trust based AODV Routing Protocol by Exclusion of Black Hole Attack,” *Procedia Technol.*, vol. 10, pp. 530–537, 2013.
- [7] S. Choudhury, S. D. Roy, and S. A. Singh, “Trust management in ad hoc network for secure DSR routing,” *Nov. algorithms Tech. Telecommun. Autom. Ind. Electron.*, pp. 495–500, 2008.
- [8] A. K. Abdelaziz, M. Nafaa, and G. Salim, “Survey of routing attacks and countermeasures in mobile ad hoc networks,” in *Computer Modelling and Simulation (UKSim), 2013 UKSim 15th International Conference on*, 2013, pp. 693–698.
- [9] P. Goyal, V. Parmar, and R. Rishi, “Manet: vulnerabilities, challenges, attacks, application,” *IJCEM Int. J. Comput. Eng. Manag.*, vol. 11, no. 2011, pp. 32–37, 2011.
- [10] S. Kukliński and G. Wolny, “CARAVAN: a context-aware architecture for VANET,” in *Mobile Ad-Hoc Networks: Applications*, InTech, 2011.

- [11] K. Sampigethaya, M. Li, L. Huang, and R. Poovendran, “AMOEBa: Robust location privacy scheme for VANET,” *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, 2007.
- [12] S. Subramaniyan, W. Johnson, and K. Subramaniyan, “A distributed framework for detecting selfish nodes in MANET using Record-and Trust-Based Detection (RTBD) technique,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2014, no. 1, p. 205, 2014.
- [13] S. Marti, T. J. Giuli, K. Lai, and M. Baker, “Mitigating routing misbehavior in mobile ad hoc networks,” in *Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000, pp. 255–265.
- [14] L. Rupasinghe and I. Murray, “Trust framework for handling communication using social networks as applied to mobile sensor based indoor navigation system,” in *Indoor Positioning and Indoor Navigation (IPIN), 2014 International Conference on*, 2014, pp. 314–321.
- [15] X. Wu, X. Wang, and R. Liu, “Solving minimum power broadcast problem in wireless ad-hoc networks using genetic algorithm,” in *Communication Networks and Services Research Conference, 2008. CNSR 2008. 6th Annual*, 2008, pp. 203–207.
- [16] A. Kamerman and L. Monteban, “WaveLAN®-II: a high-performance wireless LAN for the unlicensed band,” *Bell Labs Tech. J.*, vol. 2, no. 3, pp. 118–133, 1997.
- [17] S. Chettibi and S. Chikhi, “Routing in Mobile Ad-Hoc Networks as a Reinforcement Learning Task,” pp. 128–135, 2011.
- [18] A. Biradar, R. C. Thool, R. Velur, and T. S. Indumathi, “Dual channel based multi-objectives genetic routing protocol for ad-hoc networks and optical networks using power aware clustered topology,” in *Optical Engineering (ICOE), 2012 International Conference on*, 2012, pp. 1–6.
- [19] T. C. Fischer, “Communications in Computer and Information Science,” 2017.
- [20] S. Misra, I. Zhang, and S. C. Misra, *Guide to wireless Ad Hoc networks*. Springer Science & Business Media, 2009.
- [21] M. Guizani, “Security and trust in mobile ad hoc networks,” in *Communication Networks and Services Research Conference, 2006. CNSR 2006. Proceedings of the*

- 4th Annual, 2006, p. 2--pp.
- [22] Y. Xiao, H. Chen, S. Yang, Y.-B. Lin, and D.-Z. Du, "Wireless network security." Springer, 2009.
- [23] P. N. Reddy, C. Vishnuvardhan, and V. Ramesh, "An Overview on Reactive Protocols for Mobile Ad-Hoc Networks," *Int. J. Comput. Sci. Mob. Comput.*, vol. 2, pp. 368–375, 2013.
- [24] D. Grigoras, D. C. Doolan, and S. Tabirca, "Scalability of Mobile Ad Hoc Networks," in *Handbook of Research on Scalable Computing Technologies*, IGI Global, 2010, pp. 705–717.
- [25] G. Santhi, A. Nachiappan, M. Z. Ibrahime, R. Raghunadhane, and M. K. Favas, "Q-learning based adaptive QoS routing protocol for MANETs," in *Recent Trends in Information Technology (ICRTIT), 2011 International Conference on*, 2011, pp. 1233–1238.
- [26] M. K. Nazir, R. U. Rehman, and A. Nazir, "A novel review on security and routing protocols in MANET," *Commun. Netw.*, vol. 8, no. 04, p. 205, 2016.
- [27] A. Alharbi, A. Al-Dhalaan, and M. Al-Rodhaan, "A Mobile Ad hoc Network Q-Routing Algorithm: Self-Aware Approach," *Int. J. Comput. Appl.*, vol. 127, no. 7, pp. 1–6, 2015.
- [28] S. Jain and J. S. Baras, "Distributed trust based routing in mobile ad-hoc networks," in *Military Communications Conference, MILCOM 2013-2013 IEEE*, 2013, pp. 1801–1807.
- [29] D. Djenouri, L. Khelladi, and N. Badache, "A survey of security issues in mobile ad hoc networks," *IEEE Commun. Surv.*, vol. 7, no. 4, pp. 2–28, 2005.
- [30] M. M. Ghonge, P. M. Jawandhiya, and M. S. Ali, "Countermeasures of network layer attacks in manets," *IJCA Spec. Issue Network Secur. Cryptogr. NSC*, 2011.
- [31] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad hoc networks*, vol. 1, no. 2, pp. 293–315, 2003.
- [32] A. K. Rai, R. R. Tewari, and S. K. Upadhyay, "Different types of attacks on integrated

- MANET-Internet communication,” *Int. J. Comput. Sci. Secur.*, vol. 4, no. 3, pp. 265–274, 2010.
- [33] R. I. M. Dunbar, V. Arnaboldi, M. Conti, and A. Passarella, “The structure of online social networks mirrors those in the offline world,” *Soc. Networks*, vol. 43, pp. 39–47, 2015.
- [34] W. J. Sause and M. Adviser-Laszlo, “A coordinated reinforcement learning framework for multi-agent virtual environments,” 2013.
- [35] Y. Li, “Deep reinforcement learning: An overview,” *arXiv Prepr. arXiv1701.07274*, 2017.
- [36] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.
- [37] R. S. Sutton, “Learning to predict by the methods of temporal differences,” *Mach. Learn.*, vol. 3, no. 1, pp. 9–44, 1988.
- [38] J.-H. Cho, A. Swami, and R. Chen, “A survey on trust management for mobile ad hoc networks,” *IEEE Commun. Surv. Tutorials*, vol. 13, no. 4, pp. 562–583, 2011.
- [39] T. Navaneethan and M. Lalli, “Security Attacks in Mobile Ad-hoc Networks--A Literature Survey,” *T. Navaneethan al, Int. J. Comput. Sci. Mob. Appl.*, vol. 2, no. 4, pp. 1–7, 2014.
- [40] D. Spiewak and T. Engel, “An overview of models applying trust as a component of security services in manets,” in *Proceedings of International Workshop on Research Challenges in Security and Privacy in Mobile and Wireless Networks (WSPWN 06)*, 2006.
- [41] J. Hu, *Trust management in mobile wireless networks: security and survivability*. The Florida State University, 2007.
- [42] C. R. S. Banerji, S. Severini, and A. E. Teschendorff, “Network transfer entropy and metric space for causality inference,” *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 87, no. 5, pp. 1–14, 2013.
- [43] T. Schreiber, “Measuring information transfer,” *Phys. Rev. Lett.*, vol. 85, no. 2, p. 461,

- 2000.
- [44] C. E. Shannon and W. Weaver, “The mathematical theory of communications--Univ,” *Illinois Press. Urbana*, 1949.
 - [45] B. Ishibashi and R. Boutaba, “Topology and mobility considerations in mobile ad hoc networks,” *Ad hoc networks*, vol. 3, no. 6, pp. 762–776, 2005.
 - [46] J. O. Kephart and D. M. Chess, “The vision of autonomic computing,” *Computer (Long. Beach. Calif.)*, vol. 36, no. 1, pp. 41–50, 2003.
 - [47] J. A. Barnes, “Graph theory and social networks: A technical comment on connectedness and connectivity,” *Sociology*, vol. 3, no. 2, pp. 215–232, 1969.
 - [48] S. Jain, “Security and trust in mobile ad-hoc networks,” University of Maryland, College Park, 2015.
 - [49] B. Rieder, “Studying Facebook via data extraction: the Netvizz application,” in *Proceedings of the 5th annual ACM web science conference*, 2013, pp. 346–355.
 - [50] M. Bastian, S. Heymann, M. Jacomy, and others, “Gephi: an open source software for exploring and manipulating networks.,” *Icwsn*, vol. 8, pp. 361–362, 2009.
 - [51] W. contributors, “Bayes’ theorem --- Wikipedia{,} The Free Encyclopedia.” 2018.
 - [52] W. contributors, “Bayesian network --- Wikipedia{,} The Free Encyclopedia.” 2018.
 - [53] D. Koelle, J. Pfautz, M. Farry, Z. Cox, G. Catto, and J. Campolongo, “Applications of Bayesian belief networks in social network analysis,” in *Proceedings of the 4th Bayesian modeling applications workshop during the 22nd annual conference on uncertainty in artificial intelligence*, 2006.
 - [54] M. Jamali and H. Abolhassani, “Different aspects of social network analysis,” in *Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on*, 2006, pp. 66–72.
 - [55] M. Fire, G. Katz, L. Rokach, and Y. Elovici, “Links reconstruction attack using link prediction algorithms to compromise social networks privacy,” 2012.
 - [56] F. Nagle and L. Singh, “Can friends be trusted? Exploring privacy in online social

- networks,” in *Social Network Analysis and Mining, 2009. ASONAM'09. International Conference on Advances in*, 2009, pp. 312–315.
- [57] E. Protalinski., “Facebook: 8.7 percent are fake users,” 2012. [Online]. Available: <https://www.cnet.com/news/facebook-8-7-percent-are-fake-users/>. [Accessed: 07-Dec-2017].
- [58] L. A. Thompson, E. Black, W. P. Duff, N. P. Black, H. Saliba, and K. Dawson, “Protected health information on social networking sites: ethical and legal considerations,” *J. Med. Internet Res.*, vol. 13, no. 1, 2011.
- [59] Orgnet.com, “Social Network Analysis: An Introduction by Orgnet,LLC,” 2017. [Online]. Available: <http://www.orgnet.com/sna.html>. [Accessed: 07-Dec-2017].
- [60] L. Matthews and P. Richard, “Who Is Central to a Social Network? It Depends on Your Centrality Measure,” *Act. Networks*, 2012.
- [61] Sites.google.com, “An introduction to Centrality measures - An Introductory Course on Network Analysis,” 2017. [Online]. Available: <https://sites.google.com/site/networkanalysiscourse/schedule/an-introduction-to-centrality-measures>. [Accessed: 07-Dec-2017].
- [62] Orgnet.com, “Social Network Analysis: An Introduction by Orgnet,LLC,” 2017. .
- [63] “BAYESFUSION,LLC,” 2017. [Online]. Available: <https://www.bayesfusion.com/tm>. [Accessed: 07-Dec-2017].
- [64] W. Li, A. Joshi, and T. Finin, “Sat: an svm-based automated trust management system for mobile ad-hoc networks,” in *MILITARY COMMUNICATIONS CONFERENCE, 2011-MILCOM 2011*, 2011, pp. 1102–1107.
- [65] J.-H. Cho and A. Swami, “Towards trust-based cognitive networks: A survey of trust management for mobile ad hoc networks,” 2009.
- [66] Z. Liu, A. W. Joy, and R. A. Thompson, “A dynamic trust model for mobile ad hoc networks,” in *Distributed Computing Systems, 2004. FTDCS 2004. Proceedings. 10th IEEE International Workshop on Future Trends of*, 2004, pp. 80–85.
- [67] S. Ruohomaa and L. Kutvonen, “Trust management survey,” in *ITrust*, 2005, vol.

- 3477, pp. 77–92.
- [68] E. Aivaloglou, S. Gritzalis, and C. Skianis, “Trust Establishment in Ad Hoc and Sensor Networks,” in *Critical Information Infrastructures Security: First International Workshop, CRITIS 2006, Samos, Greece, August 31 - September 1, 2006. Revised Papers*, J. Lopez, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 179–194.
- [69] J.-W. Ho, “Zone-based trust management in sensor networks,” vol. in IEEE In, 2009.
- [70] T. Jiang and J. S. Baras, “Trust Evaluation in Anarchy: A Case Study on Autonomous Networks.,” in *INFOCOM*, 2006.
- [71] J. S. Baras and T. Jiang, “Cooperative games, phase transitions on graphs and distributed trust in MANET,” in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, 2004, vol. 1, pp. 93–98.
- [72] J. S. Baras and T. Jiang, “Cooperation, trust and games in wireless networks,” *Adv. Control. Commun. Networks, Transp. Syst.*, pp. 183–202, 2005.
- [73] Y. L. Sun, W. Yu, S. Member, Z. Han, and K. J. R. Liu, “Information Theoretic Framework of Trust Modeling and Evaluation for Ad Hoc Networks,” vol. 24, no. 2, pp. 305–317, 2006.
- [74] J. A. Boyan and M. L. Littman, “Packet routing in dynamically changing networks: A reinforcement learning approach,” in *Advances in neural information processing systems*, 1994, pp. 671–678.
- [75] Eumetrain.org, “Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE),” 2017. .
- [76] GitHub, “gephi/gephi,” 2017. [Online]. Available: <https://github.com/gephi/gephi/wiki>. [Accessed: 07-Dec-2017].
- [77] GitHub, “gephi/gephi Filter,” 2017. [Online]. Available: <https://github.com/gephi/gephi/wiki/Filter>. [Accessed: 07-Dec-2017].
- [78] J. He and W. W. Chu, “A social network-based recommender system (SNRS),” in *Data Mining for Social Network Data*, Springer, 2010, pp. 47–74.

- [79] R. Andersen *et al.*, “Trust-based recommendation systems: an axiomatic approach,” in *Proceedings of the 17th international conference on World Wide Web*, 2008, pp. 199–208.
- [80] W. contributors, “Root-mean-square deviation --- Wikipedia{,} The Free Encyclopedia.” 2018.
- [81] Eumetrain.org, “Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE),” 2017. [Online]. Available: <https://www.bayesfusion.com/tm>. [Accessed: 07-Dec-2017].
- [82] S. Subramanian and B. Ramachandran, “Trusted AODV for trustworthy routing in MANET,” *Adv. Intell. Soft Comput.*, vol. 167 AISC, no. VOL. 2, pp. 37–45, 2012.

"Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged."