

**AN EFFICIENT TRUSTED FRAMEWORK FOR
CONTEXT AWARE SENSOR DRIVEN PERVASIVE
APPLICATIONS AND THEIR INTEGRATION USING
ONTOLOGIES**

Thesis

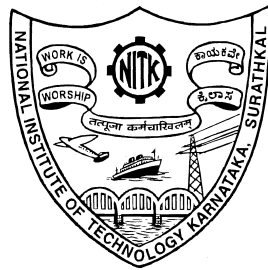
Submitted in partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

Mr. KARTHIK N

(145073IT14F01)

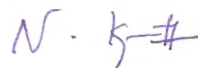


**DEPARTMENT OF INFORMATION TECHNOLOGY
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA
SURATHKAL, MANGALORE - 575025**

MAY 2020

Declaration

I hereby *declare* that the Research Thesis entitled “An Efficient Trusted Framework for Context Aware Sensor driven Pervasive Applications and their Integration using Ontologies” which is being submitted to the National Institute of Technology Karnataka, Surathkal in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy in Information Technology is a *bonafide report of the research work carried out by me*. The material contained in this thesis has not been submitted to any University or Institution for the award of any degree.



Mr. Karthik N
Register No.: 145073IT14F01
Department of Information Technology

Place: NITK, Surathkal

Date:

Certificate

This is to *certify* that the Research Thesis entitled “An Efficient Trusted Framework for Context Aware Sensor driven Pervasive Applications and their Integration using Ontologies” submitted by Mr. Karthik N (Register Number: 145073IT14F01) as the record of the research work carried out by him, is *accepted as the Research Thesis submission* in partial fulfilment of the requirements for the award of degree of Doctor of Philosophy.

Dr. Ananthanarayana V S
Research Guide
Professor
Department of Information Technology
NITK Surathkal - 575025

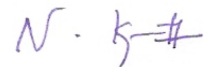
Chairman - DRPC
(Signature with Date and Seal)

Acknowledgements

Foremost, I express my sincere and deepest gratitude to my Research Guide, Head of the Department, Department of Information Technology, and Research Progress Assessment Committee members for their continuous support and encouragement.

I thank all my fellow doctoral students, teaching and non-teaching staffs of Department of Information Technology for their cooperation.

Last, but not least, I would like to thank my parents, wife, brothers, sisters, and friends for their emotional support.



(Mr. Karthik N)

Place: NITK, Surathkal

Date:

Abstract

Pervasive computing application consists of various types of sensors, actuators, set of protocols and services for monitoring physical, environmental circumstances and happenings by collecting data and act autonomously to serve the user. The pervasive computing is established on recent advancements of mobile computing, distributed computing, wireless communications, embedded systems and context-aware computing that makes computing devices smaller and earns more ability for perception, communication and computation operations. Sensor nodes play an important role in a pervasive computing environment. These sensor nodes are expected to be installed in various pervasive applications for detecting real-world events and respond consequently. Tiny sensor nodes are embedded in everyday objects invisibly that provides ubiquitous access to information services. Due to recent advancements of sensors and wireless technologies, pervasive computing is bringing heterogeneous sensors into our everyday life for providing better services. Massive amount of data is generated from sensor nodes of a pervasive environment, which is forwarded to the sink node through the gateway for data analysis and event detection. The sensed data from pervasive computing application suffers from data fault, missing data, due to the unfriendly, harsh environment and resource restriction.

In most of the cases, the generated data can be shared among different applications in the pervasive environment for increasing the user comfortableness, reliability of the application and achieving the full potential of the application. The shared data plays a vital role in critical decision making. The generated data from various sensors depict conflict in types, formats, and representations which arises problem for nodes to process and infer. Various types of sensor nodes and other devices would lead to the generation of heterogeneous data which constrains pervasive application to understand data and use efficaciously. Data interoperability problem occurs when different pervasive applications interact with each other. Furthermore, with the rise of several sensor node manufacturers, pervasive computing faces the problem in the data integration process. Because of data heterogeneity, the data cannot be shared with other application which leads to interoperability problem in the pervasive environment. The objective of the thesis is to share the trustworthy data and offer interoperability across different trusted context-aware pervasive applications. To deal with data faults, data loss and event detection, Trust Management Schemes (TMS) are proposed. To solve interoperability problem, hybrid ontology matching technique is proposed. Sensor data modeling is the basis for all TMS in sensor networks. An energy efficient hybrid sensor data modeling

for data fault detection, data reconstruction and event detection is proposed and analysis of energy consumption of data fault detection in various environment is also given.

This thesis introduces the Trust-based Data Gathering (TDG) in sensor networks, which focuses on trust-based data collection, trust-based data aggregation, and trust-based data reconstruction to show that the absence of trust in a sensor-driven harsh pervasive environment consumes more energy and delay for handling untrustworthy data, untrustworthy node and affects the normal functionality of the application.

This thesis presents the Hybrid Trust Management Scheme (HTMS) for sensor networks, which assign the trust score to node and data based on interdependency property. The correlation metric and provenance data are used to score the sensed data. The data trust score is utilized for making a decision. The communication trust and provenance data are used to evaluate the trust score of intermediate nodes and the source node.

The Context-Aware Trust Management Scheme (CATMS) is introduced in pervasive healthcare systems for data fault detection, data reconstruction and medical event detection. It employs heuristic functions, data correlation, and contextual information based algorithms to identify data faults and events. It also reconstructs the data faults and data loss for detecting events reliably. This work aims to alert the caregiver and raise the alarm only when the patient enters into a medical emergency.

Finally, this thesis investigates the hybrid ontology matching using upper ontology for solving semantic heterogeneity and interoperability problems. It combines direct and indirect matching techniques with upper ontology to share and integrate data semantically and establishes a semantic correspondence among various entities of pervasive application ontologies.

To find the efficiency of the proposed framework, we carried out experiments with INTEL Berkeley lab dataset, sensorscope dataset and data samples collected by medical sensor network prototype of pervasive healthcare application. The experimental results show that the proposed framework shares trustworthy data and offers interoperability across different trusted context-aware pervasive applications.

Keywords: Context Awareness; Data Fault Detection; Data Gathering; Data Reconstruction; Event Detection; Ontology Matching; Pervasive Environments; Sensor Data Modeling; Semantic Framework; Trust Management Scheme; Upper Ontology; Wireless Sensor Networks.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 2 |
| 1.1 | Trust Management Scheme | 3 |
| 1.2 | Sensor Data Modeling | 5 |
| 1.3 | Data Gathering | 6 |
| 1.4 | Semantic Web Technologies | 8 |
| 1.5 | Pervasive Healthcare System | 10 |
| 1.6 | Thesis Contributions | 11 |
| 1.7 | Datasets used in the thesis | 11 |
| 1.8 | Thesis Organization | 12 |
| 1.9 | Summary | 12 |
| 2 | Literature Survey | 14 |
| 2.1 | Trust Management Schemes in Wireless Sensor Network | 14 |
| 2.1.1 | Data fault detection in sensor networks | 19 |
| 2.1.2 | Data reconstruction schemes in sensor networks | 21 |
| 2.1.3 | Event detection in sensor networks | 21 |
| 2.2 | Trust Management Schemes in Medical Sensor Networks | 22 |
| 2.3 | Trust-based Data Collection and Trust-based Data Aggregation | 25 |
| 2.4 | Ontology Matching | 27 |
| 2.5 | Upper Ontology | 28 |
| 2.6 | Outcome of the Literature Survey | 28 |
| 2.6.1 | Research Motivation | 29 |
| 2.6.2 | Motivating Examples | 30 |
| 2.6.3 | Problem Definition | 30 |
| 2.6.4 | Research Objectives | 30 |
| 2.7 | General Methodology | 31 |
| 2.7.1 | Data Gathering | 31 |
| 2.7.2 | Trust Management Scheme | 32 |
| 2.7.3 | Upper ontology and ontology matching | 32 |

| | | |
|----------|--|-----------|
| 2.8 | Summary | 32 |
| 3 | Sensor Data Modeling for Data Trustworthiness | 34 |
| 3.1 | Preamble | 34 |
| 3.2 | Sensor Data Features | 34 |
| 3.2.1 | Data correlations and Data Provenance | 35 |
| 3.2.2 | Motivation | 35 |
| 3.2.3 | Assumptions | 35 |
| 3.2.4 | Data loss in WSN | 36 |
| 3.3 | Proposed Sensor Model | 36 |
| 3.3.1 | Localized data trustiness detection | 36 |
| 3.3.2 | Peer node data trustiness detection | 37 |
| 3.3.3 | Global data trustiness detection | 37 |
| 3.3.4 | Node trustiness and data provenance | 37 |
| 3.3.5 | Data reconstruction | 38 |
| 3.3.6 | Environmental disruption & environmental model | 39 |
| 3.3.7 | Event detection | 39 |
| 3.3.8 | Hybrid Sensor Data Modeling Algorithm | 39 |
| 3.4 | Results and Discussions | 41 |
| 3.4.1 | Detection Accuracy | 42 |
| 3.4.2 | Energy Consumption Analysis for Data Trustiness Detection | 43 |
| 3.4.3 | Analysis of Data Reconstruction | 46 |
| 3.4.4 | False Positive Rate for Event Detection | 48 |
| 3.5 | Summary | 49 |
| 4 | Trust-based Data Gathering in Wireless Sensor Network | 50 |
| 4.1 | Preamble | 50 |
| 4.2 | Trust-based Data Gathering | 50 |
| 4.2.1 | Proposed Trust model for Data Gathering | 51 |
| 4.2.2 | Trust-based data collection and data aggregation | 53 |

| | | |
|----------|--|-----------|
| 4.2.3 | Algorithm 4.1: Trust-based Data Collection | 55 |
| 4.2.4 | Algorithm 4.2: Trust-based Data Aggregation | 55 |
| 4.2.5 | Algorithm 4.3: Trust-based Data Reconstruction | 56 |
| 4.3 | Experimental setup and Simulation Environment | 56 |
| 4.4 | Results and Discussions | 57 |
| 4.4.1 | Energy consumption analysis | 58 |
| 4.4.2 | Network delay analysis | 61 |
| 4.4.3 | Data reconstruction analysis | 64 |
| 4.4.4 | Detection of data faults | 66 |
| 4.4.5 | Detection Accuracy of malicious nodes | 67 |
| 4.5 | Summary | 67 |
| 5 | Hybrid Trust Management Scheme for Wireless Sensor Networks | 68 |
| 5.1 | Preamble | 68 |
| 5.2 | Network Assumptions, Attack Model and Various Types of Trust . . | 69 |
| 5.2.1 | Network Assumptions | 69 |
| 5.2.2 | Attack model | 69 |
| 5.2.3 | Various Types of Trust | 71 |
| 5.3 | Proposed Trust Management Scheme | 72 |
| 5.3.1 | Structure of HTMS | 73 |
| 5.3.2 | Data Trust Evaluation | 74 |
| 5.3.3 | Node Trust Calculation | 78 |
| 5.3.4 | Provenance based node trust value evaluation | 81 |
| 5.3.5 | Algorithms for trust evaluation | 83 |
| 5.3.6 | Steps involved in trust evaluation of HTMS | 86 |
| 5.3.7 | Attack Resistant Direct Trust Evaluation | 86 |
| 5.3.8 | Attack Resistant Indirect Trust Evaluation | 87 |
| 5.4 | Results and Discussions | 88 |
| 5.4.1 | Trust evaluation comparison | 88 |
| 5.4.2 | Detection of untrustworthy data item | 93 |

| | | |
|----------|---|------------|
| 5.4.3 | Detection of malicious node | 95 |
| 5.4.4 | Detection of selfish nodes | 97 |
| 5.4.5 | Memory requirement analysis of HTMS | 97 |
| 5.4.6 | Detection rate comparison | 98 |
| 5.4.7 | Energy consumption comparison | 99 |
| 5.5 | Summary | 99 |
| 6 | Context-Aware Trust Management Scheme for Pervasive Healthcare | 100 |
| 6.1 | Preamble | 100 |
| 6.2 | Taxonomy of Trustiness in Pervasive Healthcare | 101 |
| 6.2.1 | Taxonomy of Trustiness in Pervasive Healthcare | 101 |
| 6.2.2 | Data Faults and Data Loss | 104 |
| 6.3 | Proposed Trust Management Scheme for Pervasive Healthcare | 106 |
| 6.3.1 | Structure of CATMS | 107 |
| 6.3.2 | Algorithms for Trust Evaluation | 115 |
| 6.3.3 | Steps required for evaluation of trust in CATMS | 122 |
| 6.4 | Results and Discussions | 123 |
| 6.4.1 | Data fault detection analysis | 123 |
| 6.4.2 | Data reconstruction analysis | 127 |
| 6.4.3 | Event detection analysis | 130 |
| 6.5 | Summary | 131 |
| 7 | Upper Ontology and Hybrid Ontology Matching for Pervasive Applications | 132 |
| 7.1 | Preamble | 132 |
| 7.2 | Upper Ontology for Pervasive Environments | 133 |
| 7.3 | Direct and Indirect Ontology Matching | 134 |
| 7.4 | Hybrid Ontology Matching | 135 |
| 7.5 | Results and Discussions | 136 |
| 7.6 | Summary | 138 |

| | |
|--------------------------------------|------------|
| 8 Conclusions and Future Work | 140 |
| References | 144 |

List of Tables

| | | |
|------|--|-----|
| 2.1 | Comparison of Trust Management Schemes for WSN | 18 |
| 2.2 | Comparison of TMS for MSN | 24 |
| 2.3 | Recent works of trust-based data aggregation and trust-based data collection | 26 |
| 2.4 | Comparison of instance-based ontology matching techniques | 28 |
| 2.5 | Comparison of Upper Ontologies | 28 |
| 4.1 | Amount of energy spent by MICAz node | 58 |
| 5.1 | Value and Provenance similarity | 77 |
| 5.2 | Case 1: Untrustworthy node C with four trustworthy neighbor | 91 |
| 5.3 | Case 2: Trustworthy node F with one untrustworthy node | 92 |
| 5.4 | Case 3: Trustworthy node B with one untrustworthy neighbor and three trustworthy neighbors | 92 |
| 5.5 | Case 4: Trustworthy node A with more number of trusted neighbors (<3) | 92 |
| 5.6 | Case 5: Trustworthy node H with only one trusted neighbor | 93 |
| 5.7 | Case 6: Trustworthy node G with only one trusted neighbor | 93 |
| 5.8 | Case 7: Trustworthy node E with one untrustworthy neighbor and two trusted neighbors | 93 |
| 5.9 | Case 8: Trustworthy node D with one trustworthy neighbor and one untrustworthy neighbor | 93 |
| 5.10 | Memory requirement for direct trust evaluation at sensor node | 97 |
| 5.11 | Memory requirement for indirect trust evaluation at sensor node | 98 |
| 7.1 | List of Ontologies used for experiments | 136 |
| 7.2 | List of applications and reference alignments | 137 |

List of Figures

| | | |
|------|---|----|
| 1.1 | Components of TMS | 3 |
| 2.1 | Taxonomy of trust management in WSN | 14 |
| 2.2 | Trusted semantic framework for context-aware pervasive applications | 31 |
| 2.3 | Organization of Thesis with respect to chapters and objectives | 33 |
| 3.1 | Proposed sensor data model | 38 |
| 3.2 | INTEL lab sensor deployment | 42 |
| 3.3 | Detection Accuracy | 43 |
| 3.4 | Energy consumption for case 1 | 44 |
| 3.5 | Energy consumption for case 2 | 45 |
| 3.6 | Energy consumption for case 3 | 45 |
| 3.7 | Element Random data loss | 46 |
| 3.8 | Block Random data loss | 47 |
| 3.9 | Element frequent data loss | 47 |
| 3.10 | Successive element data loss | 48 |
| 3.11 | False Positive Rate for Event detection | 49 |
| 4.1 | Proposed Trust Model | 51 |
| 4.2 | Data Collection with Trust mechanism | 54 |
| 4.3 | Data Aggregation with Trust mechanism | 54 |
| 4.4 | Sensor deployment in INTEL Lab Berkeley | 57 |
| 4.5 | Energy consumption analysis of DC and DCT for faulty data | 59 |
| 4.6 | Energy consumption analysis of DA and DAT for faulty data | 59 |
| 4.7 | Energy consumption analysis of DC and DCT for malicious nodes | 60 |
| 4.8 | Energy consumption analysis of DA and DAT for malicious nodes | 61 |
| 4.9 | Network delay analysis of DC and DCT for faulty data | 62 |
| 4.10 | Network delay analysis of DA and DAT for faulty data | 62 |
| 4.11 | Network delay analysis of DC and DCT for malicious nodes | 63 |
| 4.12 | Network delay analysis of DA and DAT for malicious nodes | 63 |

| | | |
|------|--|-----|
| 4.13 | Data reconstruction analysis with data density and neighbor node correlation based methods | 65 |
| 4.14 | Comparison of various data reconstruction schemes | 65 |
| 4.15 | Comparison of data fault detection schemes | 66 |
| 4.16 | Comparison of detection accuracy of the malicious nodes | 67 |
| | | |
| 5.1 | A simple WSN scenario | 73 |
| 5.2 | Structure of HTMS | 73 |
| 5.3 | Sliding window | 79 |
| 5.4 | Direct trust evaluation | 80 |
| 5.5 | Indirect trust evaluation | 81 |
| 5.6 | Interdependency property of data item and nodes | 82 |
| 5.7 | Filtering of recommendations | 88 |
| 5.8 | Sample scenario for trust evaluation | 89 |
| 5.9 | Data and node trust score using DBTA | 89 |
| 5.10 | Data and node trust score using proposed method | 90 |
| 5.11 | Node trust score evaluation by DBTA | 90 |
| 5.12 | Node trust score evaluation by HTMS | 91 |
| 5.13 | Detection accuracy of untrustworthy data items | 94 |
| 5.14 | Detection accuracy of untrustworthy data items | 95 |
| 5.15 | Detection accuracy of malicious nodes | 98 |
| 5.16 | Comparison of the energy consumption | 99 |
| | | |
| 6.1 | Taxonomy of Trustworthiness in Pervasive Healthcare | 102 |
| 6.2 | Data loss patterns | 106 |
| 6.3 | Pervasive healthcare system | 106 |
| 6.4 | Medical sensor network prototype | 107 |
| 6.5 | Structure of CATMS | 107 |
| 6.6 | Performance of Algorithm 6.1 | 125 |
| 6.7 | Performance of Algorithm 6.2 | 125 |
| 6.8 | Performance of Algorithm 6.3 | 125 |

| | | |
|------|--|-----|
| 6.9 | Performance of Algorithm 6.4 | 125 |
| 6.10 | Performance of Algorithm 6.5 | 126 |
| 6.11 | Comparison of TMS | 126 |
| 6.12 | Data reconstruction of HR | 127 |
| 6.13 | Data reconstruction of PR | 127 |
| 6.14 | Data reconstruction of BT | 128 |
| 6.15 | Data reconstruction of BP | 128 |
| 6.16 | Error rate of ERL | 129 |
| 6.17 | Error rate of EFL in row | 129 |
| 6.18 | Error rate of BRL | 130 |
| 6.19 | Error rate of SEL | 130 |
| 6.20 | Event Detection accuracy | 130 |
| 6.21 | FPR – Event detection | 130 |
| 7.1 | Upper Ontology for Pervasive Environments | 133 |
| 7.2 | Hybrid Ontology Matching | 135 |
| 7.3 | (a) Direct Ontology Matching (b) Indirect Ontology Matching (c) Hybrid Ontology Matching without Trust (d) Hybrid Ontology Matching with Trust | 137 |

List of Abbreviations

| Abbreviation | Meaning |
|---------------------|--|
| ADCT | Adaptive and dual Data-Communication Trust |
| BAN | Body Area Network |
| BSN | Body Sensor Network |
| BFO | Basic Formal Ontology |
| BN | Bayesian Network |
| BAN | Body Area Network |
| BRL | Block Random Loss |
| BT | Body Temperature |
| BP | Blood Pressure |
| BW | BandWidth |
| CATMS | Context Aware Trust Management Scheme |
| CT | Communication Trust |
| DA | Data Aggregation |
| DAT | Data Aggregation with Trust mechanism |
| DBTA | Distance Based Trust Assessment |
| DC | Data Collection |
| DCT | Data Collection with Trust mechanism |
| DoD | Depth of Discharge |
| DOLCE | Descriptive Ontology for Linguistic and Cognitive Engineering. |
| DT | Data Trust |
| DTM | Data Trust Model |
| DTMS | Data Trust Management Scheme |
| DoS | Denial of Service |
| ED | Event Detection |
| EDTM | Efficient Distributed Trust Model |
| EFL | Element Frequent Loss |
| ERL | Element Random Loss |
| FPR | False positive rate |
| FR | Fowarding Ratio |
| FTM | Fuzzy Trust Model |
| GTMS | Group based Trust Management Scheme |
| GPS | Global Positioning System |
| HR | Heart Rate |
| HT | Heuristic based Trust |

| | |
|-----------------|---|
| HTMS | Hybrid Trust Management Scheme |
| HTRM | Hybrid Trust and Reputation Management |
| IT | Initial Trust |
| IDT | InDirect Trust |
| KNN | K-Nearest Neighbor |
| LDTS | Lightweight Dependable trust System |
| LWTM | Light Weight Trust Model |
| MA | Medical Attention |
| MCC | Multi Correlation Coefficient |
| MSN | Medical Sensor Node |
| ML TRUST | Multi Level Trust |
| MDETM | Multi-Dimensional Evidence-based Trust Management |
| NT | Node Trust |
| OWL | Web Ontology Language |
| PBTA | Provenance Based Trust Assessment |
| PHS | Perasive Healthcare System |
| PR | Pulse Rate |
| RC | Rate of Change |
| RDF | Resource Description Framework |
| RDFS | Resource Description Framework Schema |
| RLT | Resource Level Trust |
| RT | Resource Trust |
| RMSE | Root Mean Square Error |
| SRM | Security Routing Model |
| SCT | Spatial Correlation based Trust |
| SEL | Successive Element Loss |
| SoC | State of Charge |
| ST | Spatio- Temporal |
| STA | Spatio- Temporal multi Attribute |
| SWT | Semantic Web Technologies |
| TA | Temporal Attribute |
| TAHM | Trust Aware Health Monitoring |
| TCT | Temporal Correlation based Trust |
| TDG | Trust based Data Gathering |
| TPR | True Positive Rate |

| | |
|-------------|---------------------------|
| TMS | Trust Management Scheme |
| TMM | Trust Management Model |
| TSA | Time Series Analysis |
| TSTM | Time Series Trust Model |
| TT | Total Trust |
| TWA | Trust Worthy Architecture |
| WQ | Waiting Queue |
| WSN | Wireless Sensor Network |
| WWW | World Wide Web |

Chapter 1

Introduction

The most fundamental technologies are those that vanish. They interweave themselves into the material of daily life until they are identified from it. The distinctiveness of the pervasive environment is its power to manage with any device and work autonomously for providing customized services to the user. Pervasive computing considers an environment with full of smart, intelligent devices with perception, computation and communication capabilities. The sensing, computation, and communication will take place anywhere, any time and at any device. Pervasive computing application consists of various types of sensors, actuators and smart devices for collecting data and act autonomously to serve user. Due to recent advancements of sensors and wireless technologies, pervasive computing is bringing heterogeneous sensors into our everyday life for providing better services. The Wireless Sensor Network (WSN) is an important component of the pervasive computing application, which is normally deployed in terrain for monitoring physical and environmental conditions. Sensor-driven pervasive applications rely on sensors for their main operations to achieve their goals.

Context refers to information about the environment, such as location, time and identities of nearby people and other computing objects. Context-aware pervasive computing system is a type of pervasive computing system that is aware of context and can automatically adapt and react to such context. The data generated by sensors should be shared among various applications for enabling them to reach their full potential. With the assistance of different modalities, a sensor-driven pervasive system can glean information about person's context without any denotative control from the person and it provides customized services. The concept behind this paradigm is being sensitive, context-aware and adaptive to the environment to learn about their occupants, connect to the work they are doing and their goals.

There are two problems confronted in the current scenario of the pervasive environment. They are untrustworthy data generation and interoperability. A significant amount of data from the real-time sensor-driven pervasive application suffers from data faults, data loss, and malicious attacks due to resource constraints, harsh and unfriendly environment. Data faults and data related malicious attacks lead to untrustworthy data. Data generated from heterogeneous sensors and an increasing number of node fabricators lead to data heterogeneity. The sensor generated data cannot be shared among various entities of pervasive application which leads to an interoperability problem. To deal

with untrustworthy data generation and interoperability, trust management schemes and ontology matching techniques are used respectively and explained in next subsections.

1.1 Trust Management Scheme

The conception of trust primitively comes from social sciences and is defined as the degree of confidence or belief about behaviour of a node or an entity (Cho *et al.* (2010)). Trust Management Scheme (TMS) in pervasive computing is required when communicating nodes without any previous interactions, desire to establish a link or to share the data with a satisfactory level of trust relationship among themselves. In addition, TMS has various applicability in many decision making including data fault detection, data reconstruction, event detection, data gathering, intrusion detection, isolating misbehaving nodes and other functions (Govindan and Mohapatra (2011)). The components of TMS is shown in Figure 1.1. TMS is the process of establishing trust among nodes, updating the trust and revocating it. Trust establishment is the process to deal with the collection of trust evidences, trust evaluation, generation, representation and distribution. Trust update is the process of updating the trust score periodically. Regular update of trust is required to reflect the present state of entities in the application. Trust revocation is the process of annulling or initialising the trust score of an entity.

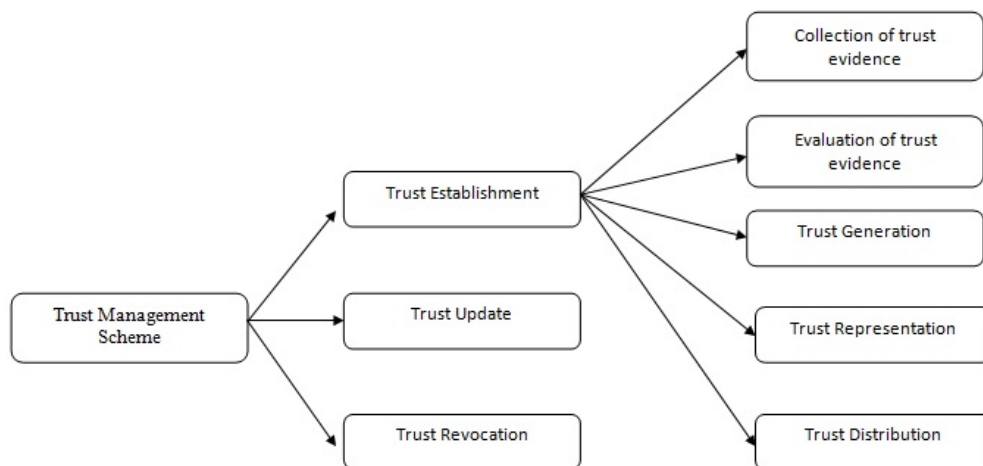


Figure 1.1: Components of TMS

Sensor nodes are randomly arranged in the terrain to identify the happenings by observing the conditions of physical phenomena (Akyildiz *et al.* (2002)). With the in-

crease of pervasive applications using WSN, security is a major concern in the environment. Due to the resource-constrained tiny nodes in the network, the implementation of the traditional cryptographic technique is not possible, since the attacker can capture the node and inject the invalid data into WSN (Zheng and Jamalipour (2009)). TMS is developed as an important supplementary mechanism for cryptographic techniques. A common problem associated with WSN is data fault due to the security threats and harsh environment. The malicious and faulty nodes are responsible for producing the erroneous data in the network. Untrustworthy data may lead to wrong decision making in the pervasive environment. Providing trustworthy data is a key issue to increase the reliability of the applications.

The TMS can make a pervasive environment tolerant to data faults, data losses, and malicious attacks thereby aiding the decision-making process. Faulty data results in incorrect estimation of the environment and causes unwanted utilization of network resources (Yu *et al.* (2015)). Data fault refers to the condition where the node in the sensor network behaves properly but produces erroneous data (Ni *et al.* (2009)). The faulty data can mislead the decision-making process which leads to incorrect action. Therefore it is mandatory to find out the faulty data in the network so that the data accuracy can be assured for decision making. To increase the reliability of the application, the trustworthy assessment of data item plays an important role. TMS solves this problem by assessing the trustworthiness of data item and scoring the data item according to its quality.

Data trust management (DTM) needs a real-time valuation process because the original data item might be disappeared when the DTM realizes that there are some untrustworthy data items in the observed data set (Bertino (2014)) and it should provide the facility to quickly achieve data recovery and correction actions for faulty data and missing values. However, the scoring only the data items which are generated by the sensor nodes is not enough in WSN, since there is an interdependency property between the data item and sensor node (Lim *et al.* (2010)). By considering only one trust element (either data trust or communication trust) (Momani *et al.* (2010)) for scoring the data item and nodes and to find out the total trust score of WSN is not enough. Untrustworthy nodes in terms of data trust might be asserted as trustworthy nodes based on communication potentials. If total trust score is established on one trust component, then the network might be misguided (Jiang *et al.* (2015)).

Nodes in the WSN are accountable for quite a few tasks like communication, generating data, computation, etc. Therefore the TMS should consider these tasks for de-

signing the different trust assessment technique for each task (Chen (2009)). TMS is developed to improve the security of WSN. However, when TMS protect against malicious attacks, they may also be violated. So, to increase the toughness of trust models, the associated attacks should be taken into account (Han *et al.* (2014)). TMS can be used as separate light weight component of security services in the network. It uses a trust score to aid an automated decision-making process. In recent times, it is introduced as an efficient security enhancement method for unprotected environments such as the internet, WSN, etc. Trust as subjective acts as an indicator for upcoming activities, and it has to adapt dynamicity due to the changes and communication between various parties. Past experiences may extremely affect the capacity of trust. Sensor data modeling is the basis for all TMS in WSN. It is explained in the next subsection.

1.2 Sensor Data Modeling

Sensor data is not random in nature; it is correlated with time and location. Data from the same sensor will not change much over the range; regularly the current data of a sensor will be close to its previous data. Two different sensors of the same phenomenon from a location give almost the same data. We can deploy the sensor to directly monitor the interesting phenomenon. Sometimes the interesting phenomenon cannot be observed directly, but it can be predictable from other sensor data. In the absence of ground truth, the sensor data model is used to find the data trustiness for decision making. The sensor data model comprises of the mathematical relationship between the variables. There are three main variables in the sensor data model (Hunkeler (2013)). The input variable of the data model represents the sensor data. The prediction of the future sensor data is represented in the output variable. The user threshold and some design parameters of the data model are represented as configuration variables. The sensor data model can represent the data of a single sensor over a period. Sometimes the sensor data model can be used to represent the mixture of different sensor data over some time.

In periodic monitoring application, sensor data are endlessly streaming. A large amount of sensor data streams are generated and forwarded to the sink node through the gateway for data analysis and further processing. Even though the gateway node and sink node has resources to process such amount of data, but the problem is data processing eats up time and network resources for data cleaning and data analysis. Forwarding of all data streams without pre-processing data lead to network traffic problem and response time issues in real time system. The sensor data streams of real-time mon-

itoring applications were affected by various data faults due to resource restriction and deployment of sensors in the harsh environment (Sharma *et al.* (2010)). To ease the load of transmitting all raw sensor data to the gateway and to reduce the processing cost and resource consumption, in-network data fault detection is employed at sensor node itself in the form of the sensor data model. The sensor data model not only detects data fault by considering the reliability of sensor node and interdependency property between node and data but also reconstructs data on gateway node to ensure the minimum amount of data availability for data analysis and decision making.

Modeling the sensor data can be done in different places. It can be modeled at the sensor node itself or in some cases; it can be modeled at a centralized server (Won and Bertino (2015)). The process of sensor data modeling in WSN is the basis for all DTM process (Reddy *et al.* (2017)). In large scale applications, the centralized data model causes big overheads and consumes more network resources. From the recent literature, we found in-network and local processing of sensor data reduces the energy consumption of communication (Fang and Dobson (2013)). However, the local processing of sensor data may affect the amount of information available at the sink node or base station for taking the critical decision from the application view point. The sensor data model should be designed in such a way that, the amount of available information at the sink node or base station is sufficient to take a critical decision. The hybrid sensor data model utilizes the temporal, spatial, attribute data features and data provenance for data trustiness detection, data reconstruction and event detection. The reason for applying hybrid approach is, there is a requirement for sensor node to process the data and take local decision to reduce the resource consumption since the communication of data consumes 94% of energy in WSN (Guestrin *et al.* (2004)). And there is a requirement for centralized sink node or base station to take decisiveness from the network viewpoint and to reconstruct the untrustworthy data and missing data. After sensor data modeling for trustiness, data gathering is an important process in WSN for data analysis and event detection, which is explained in the next section.

1.3 Data Gathering

In all pervasive applications, data gathering is the main procedure taking place in a sensor network, where the sink node gathers all sensor data for data analysis and decision making (Ji *et al.* (2014)). Data gathering is primarily employed for gathering interesting sensor data from environments, finding the size of the network, deciding mean system load and so on. Data gathering involves data collection without aggregation and data

collection with aggregation known as data collection and data aggregation respectively. Data aggregation normally calls for a coalition of data from many nodes at the intermediate node and forwards the aggregated data to the sink node in an energy efficient and delay aware manner. To carry out, data aggregation process, the aggregation schedule should be free from interference. The data aggregation should be designed in such a way that, it reduces network delay for data aggregation, increases application throughput, increases application lifetime and minimizes energy consumption.

In some of the pervasive applications, the sink node requires to gather all sensor data from nodes without aggregation. This process is called data collection. At particular time instant t , the union of all sensor data from the node is called snapshot. For continuous monitoring applications, a gathering of the uninterrupted snapshot is called continuous data collection. The performance metric called capacity of data collection is used to find how quick has sensor data items been gathered to the sink node. Since data collection transfers all sensor-generated data to sink node, it introduces more traffic, and it suffers from interference. It consumes more energy for data transmission and decreases network lifetime. Therefore it is important to design effective data collection technique which reduces energy consumption, increases the reliability of the application and network lifetime. The generation of faulty data, missing values, misbehavior of sensor nodes, incorrect data sampling and attacks on data and node are common in sensor driven pervasive applications due to resource constraints, harsh and unfriendly environments. Therefore it is necessary to design an effective data collection and data aggregation process to address faulty data, missing values, sensor node misbehavior, and unwanted resource consumption.

There are three main sources of untrustworthy data in wireless sensor networks (WSN): Errors, malicious attacks, and events. The detection methods of untrustworthy data are fault detection, intrusion detection and event detection. Related work in untrustworthy data detection has been found in trust management domain of WSN. Intensive examinations of outlier detection methods are presented in (Zhang *et al.* (2010b)) where most of them are related to trust management of WSN (Gwadera *et al.* (2014)). Trust-based data fault detection is found in (Karthik and Ananthanarayana (2016), Karthik and Ananthanarayana (2017a)). An extensive review of faulty behavior and malicious behavior detection in WSN is presented in (Han *et al.* (2014)), where all methods are devoted to trust management in WSN. The untrustworthy data and data losses should be reconstructed as trustworthy data to identify the events. Trust-based data reconstruction methods are found in (Gilbert *et al.* (2018)). Trust-based event detection methods are

found in (Chen *et al.* (2008), Illiano and Lupu (2015a)). Untrustworthy and data loss can be identified with the help of trust-based fault detection process, reconstruct the untrustworthy data and data losses with the help of trust-based data reconstruction process and identify the events with the help of trust-based event detection process. After data gathering process, the sensor data should be shared among various pervasive applications to achieve their full potential, and to increase the reliability of the applications. For data sharing, a common representation of sensor data among pervasive environment is required. Semantic Web Technologies (SWT) provides a common framework for representing the sensor data in machine understandable and processable format. SWT and ontology matching techniques play an important role in data sharing and establishing semantic correspondence in pervasive applications, which are explained in the next subsection.

1.4 Semantic Web Technologies

Sensor-driven pervasive computing has various potential applications in environmental monitoring, transport systems, smart objects, and smart spaces. Usually, pervasive applications are compiled of various types of sensors, actuators and other devices for monitoring the happenings in the environment and for data collection. Various types of sensor nodes and other devices would lead to the generation of heterogeneous data which constrains pervasive application to understand data and use efficaciously. Data interoperability problem occurs when different pervasive application interact with each other. Furthermore, with the rise of several sensor node manufacturers, pervasive computing faces the problem in the data integration process (Cao *et al.* (2016)). According to (Berners-Lee *et al.* (2001)), the semantic web is an extended version of the current web that gives a common framework to represent the data across heterogeneous applications. SWT are used to convert the unstructured and semi-structured data into structured data where the semantics of the data are explained in a machine-understandable format. SWT is used for data integration, data sharing and inferring new knowledge on the World Wide Web (WWW) (Shadbolt *et al.* (2006)). Lately, SWT is used for pervasive computing environment to achieve interoperability and to attain the full potential of the application. SWT like Resource Description Framework (RDF), RDF Schema (RDFS) and Web Ontology Language (OWL) are used to annotate data, to construct ontology and describe the relationship among various concepts of domains.

Sensor raw data are simple. It does not provide any knowledge about the environment. Metadata is required for raw sensor data to realize the situation and happenings

in the environment. Temporal information like time of data generation, spatial information like the location of the sensor node and description of data are required as metadata to enhance sensor data to understand the situation of the pervasive environment. The process of annotating sensor data with metadata is called semantic annotation. With semantic annotation, various types of sensor raw data can be represented in a single RDF data format. RDF data format representation of sensor data allows pervasive applications to interact with each other by sharing, reusing of sensor data and to achieve their full potential of the application. The advantages of semantic data modeling are the usage of the same sensor data for various pervasive applications and allow the user to interpret the event detected in the pervasive environment. It also deduces implicit knowledge from sensor data through the reasoning process and avoids the ambiguity problem. RDF data model composed of three elements: Resources; Properties and Statements. A Resource in the RDF model can be an object or real-world entities like sensor node or sensor data or Document in WWW or an element of the document. Property is used to describe the characteristics of resource or relation for defining a resource. A Statement is consists of subject, predicate, and object. The subject of the RDF statement must be a resource. The object of an RDF statement might be a resource or literal value. The predicate of the RDF statement can be a relation between subject and object.

Pervasive environments are established on acquiring real-time data from various sources and shared ad hoc by different applications to reach their full potential. For example, based on user location, a smart home application interacts with the traffic system to predict user arrival for controlling thermostat. When entities of different application try to share and exchange data, a semantic heterogeneity problem occurs. Ontologies are used to overcome the heterogeneous data integration problem ([Shvaiko and Euzenat \(2013\)](#)). Ontology supplies a set of vocabulary that describes the domain of interest and explicitly specifies the meaning of terms used in it ([Gruber \(2009\)](#)). It is also used to represent knowledge of concepts in a certain relationship with classes, properties, and rules. Ontology is used as a technique for sharing common knowledge and integration of pervasive applications. The ontology is used to have a common understanding of concepts and serves as a tool for interaction among heterogeneous application.

Ontology matching could supply a semantic connection between several ontologies for accessing and exchanging data semantically. There are three types of ontology matching techniques. They are direct, indirect and hybrid matching techniques. Direct matching uses multiple ontology architecture to find the set of correspondence among

concepts. Indirect matching uses global shared vocabulary as a background knowledge for finding semantic correspondence among various concepts. The combination of direct and indirect matching is called hybrid ontology matching. Most of the pervasive applications are developed and maintained by different developers who have diverse knowledge background, and different terms are used to describe the same concepts in the pervasive domain. Different developers contributed different ontologies for same domain concepts. This leads to semantic heterogeneity problem and limits interaction among entities. SWT and Ontology matching are introduced to overcome heterogeneity problem and interoperability problems among various pervasive applications. Pervasive Healthcare System (PHS) is an example for the research scenario used for experiments in this thesis, where heterogeneous sensors are deployed in the terrain for monitoring multiple events, which is explained in next subsection.

1.5 Pervasive Healthcare System

PHS consists of vital sign sensing, computation, and communication operations to monitor the day-to-day activities of patients and elders for providing context-aware services and medical diagnosis in time (He *et al.* (2012)). Medical Sensor Networks (MSN) consists of physiological sensors deployed on the human body for monitoring vital signs (heart rate, pulse rate, blood pressure, body temperature, oxygen saturation ratio, accelerometer) and periodically transmits such data to sink node. Sink node processes these data to perform several data related tasks on which efficacious activities can be taken (Yu *et al.* (2010)). The presence of abnormal data, data loss in pervasive healthcare system leads to incorrect identification of health status, activity recognition and wrong medical diagnosis.

The PHS uses the wireless medium for communication which is open in nature, intruders may modify, inject some wrong information and replays old messages. The traditional cryptographic techniques cannot handle the internal attacks, faulty data generation, and data loss and do not meet the requirements of pervasive healthcare applications (Boukerche and Ren (2009)). TMS is widely used in PHS to handle the data related attacks in sensor networks, faulty data generation and data loss. The MSN is data-centric in nature and important decisions are taken out from observed sensor data, the trust model for medical sensor data is important for the reliable pervasive healthcare systems. The process of sensor data validation Dondio *et al.* (2007) is extended for the trust process, in which the assisting tools and methods are used for checking the quality and validity of data. The data validation process cannot go beyond a compar-

ison of data items or analysis using simple statistics methods. It detects only the data anomalies and cannot grant trust. The trust process precisely starts after data validation ends (Dondio *et al.* (2007)). TMS involves the following steps: (1) Checking the quality and validity of data; (2) Checking the trustiness of data source which is responsible for producing data; (3) Checking the context in which data were generated; (4) Domain-specific analysis and malicious attacks analysis; (5) Checking the trustiness of data from better predictions with context, history of data source, intermediate nodes and their present evidence.

1.6 Thesis Contributions

The salient contributions of this thesis are listed as follows:

1. An energy efficient hybrid sensor data model is proposed for data fault detection, data reconstruction, and event detection. It also analyses the energy consumption of sensor data model in centralized, distributed and hybrid environment.
2. Trust-based Data Gathering is proposed in sensor networks, and it analyses the performance of trust-based data collection and trust-based data aggregation by varying the number of data faults and malicious nodes.
3. Hybrid TMS in WSN is proposed to detect data faults, malicious nodes and selfish nodes in real time by considering data provenance, interdependency property, and communication capabilities of a node.
4. Context-Aware TMS for pervasive healthcare is proposed to detect data faults, medical emergencies and data reconstruction by considering heuristic rules, data correlation and contextual information based algorithms.
5. An upper ontology for the pervasive environment with trust mechanism and hybrid ontology matching techniques are proposed to deal with faulty, missing data and ontology alignments among concepts of various ontologies.

1.7 Datasets used in the thesis

This thesis uses Intel Berkeley lab dataset (Madden *et al.* (2004)) and Sensorscope project dataset (Husein *et al.* (2016)) to validate the proposed approaches. Mica2dot nodes were deployed in Intel Berkeley lab to monitor the temperature, humidity and light parameters. The sampling time is 31 seconds. We used Intel Berkeley Lab dataset in chapter 3 and 4. Intel Berkeley lab dataset is a benchmarked dataset. It is observed that Intel Berkeley lab dataset is used by more than 300 published papers. Intel Berkeley

lab sensor deployment is an example for indoor sensor deployment and monitoring. In Sensorscope project, sensors were deployed between Switzerland and Italy in 2007 to monitor the temperature, humidity, soil moisture, rain and wind speed. The sampling time is 2 minutes. We used sensorscope project dataset in chapter 5, which is also a benchmarked dataset. It is observed that sensorscope dataset is used by more than 350 published papers. It is an example for outdoor sensor deployment and monitoring.

1.8 Thesis Organization

Chapter 2 talks about the literature review and its outcome, followed by defining the problem statements and research objectives. Chapter 3 briefly describes the sensor data modeling for data trustiness. TDG is introduced in chapter 4. The HTMS is elucidated in chapter 5. Chapter 6 introduces CATMS for pervasive healthcare. The upper ontology and hybrid ontology matching for the pervasive environment are explained in chapter 7 and followed by concluding remarks and future works in chapter 8.

1.9 Summary

This chapter introduced the basic concepts of pervasive computing, trust management scheme, data gathering, semantic web technologies, and ontology matching. The various steps of sensor data modeling, trust-based mechanisms, semantic data annotation, ontology matching, and healthcare system were briefly explained in this chapter. The challenges in pervasive environments, research motivations for the trust management schemes and ontology matching were discussed. The salient contributions of this thesis were listed out in this chapter. This chapter also gives the detailed organization of the thesis.

Chapter 2

Literature Survey

A TMS has been proposed in pervasive applications to overcome the internal attacks, data faults, data loss and identifying the events. The full potential of context-aware pervasive applications can be achieved by implementing the semantic interoperability. To share and integrate data semantically, ontology matching technique establishes a semantic correspondence among various entities of pervasive application ontologies. An extensive literature review has been carried out in TMS and ontology matching techniques for solving untrustworthy generation and interoperability problem. This chapter presents the review of existing TMS in WSN and MSN, Trust-based data aggregation, Trust-based data collection, and ontology matching techniques. Further, this chapter gives the research motivation, motivating examples, problem statement, and objectives of our research work.

2.1 Trust Management Schemes in Wireless Sensor Network

Trust models can be assorted into two main classes in WSN (Han *et al.* (2014)) as shown in Figure 2.1. They are node trust models and data trust models. Further the node trust models and data trust models can be grouped into three classes: centralized, hybrid and distributed. The centralized TMS use centralized trusted authority or base station to evaluate the trust score for node and data item. Due to the excessive resource consumption for transmitting, evaluating and exchanging the trust score, the centralized TMS are not desirable for large sensor networks. In distributed TMS, every node in the network evaluates the trust score themselves for all other nodes in the network. As the node needs to maintain the updated trust score for all other nodes, this distributed TMS are not suitable for large sensor networks (Han *et al.* (2014)). It is more reliable

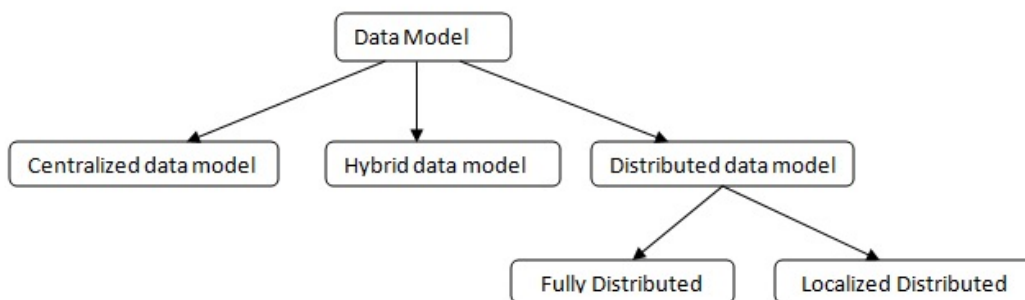


Figure 2.1: Taxonomy of trust management in WSN

than centralized TMS since there is no single point of failure. But it consumes more resources when maintaining the trust scores of all other nodes in WSN. This scheme is also referred as fully distributed scheme. To reduce resource consumption, localized distributed TMS are introduced, where the nodes evaluate and maintain only their neighbor's trust scores (Han *et al.* (2014)). The only disadvantage with localized distributed TMS is a delay in evaluating the distant node trust score. The hybrid trust management makes use of both centralized and distributed TMS advantages to reduce the resource utilization linked with trust evaluation in a distributed approach (Khalid *et al.* (2013)). The hybrid approach is reliable than centralized TMS and less reliable than distributed TMS. The hybrid scheme is used with a centralized trusted server, and the node evaluates and maintains only one neighbor's trust score which gives less memory consumption than the fully distributed schemes. Hence fully distributed and fully centralized TMS are not suitable for WSN.

In centralized data trust model, the trustworthiness of sensed data items are evaluated at a centralized server like a base station or sink node from the system perspective and application point of view. Each sensor node has its own trust model in distributed data trust model to compute the trustiness of sensed data item (Zahariadis *et al.* (2010)). In fully distributed data trust model, each sensor node has to evaluate the trustworthiness of all other nodes sensed data item in the network, whereas, in localized distributed data trust model, the sensor node has to evaluate only its neighbors' sensed data item (Hosseini *et al.* (2015)). The hybrid data trust model uses the advantages of both centralized and distributed data trust models. It is memory and energy efficient data trust model which evaluates only neighbors' node sensed data item. The difference between localized distributed trust model and hybrid trust model is the use of a centralized server in a hybrid model to take decisions from an application viewpoint.

The frequent incorrect sample spotlights the importance of automatized, online detection of data fault. To increase the reliability of the application, the TMS should filter out the faulty data in an online fashion. The online and in-network finding of data fault has benefit over centralized detection. Because the online and in-network detection provides real-time alerts when something goes wrong and initiate the corresponding corrective measures like replacing the faulty, malicious, selfish node and prediction of data, to avoid misleading of application. The centralized detections are not suitable and scalable (Fang and Dobson (2013)). In some cases, the data fault from several sensor nodes were found to be correlated since the battery of nodes exhausted at the same time (Sharma *et al.* (2010)). In event-based reporting application, the in-network error fil-

tering is the only option to filtrate the defective data, in which nodes do not report all data items back to the sink node. For large scale applications, the server side detection causes big overheads. The faulty data filtering is not possible with incomplete dataset at centralized sink node (Fang and Dobson (2013)). Only 49% of data samples were received at the sink node in (Tolle *et al.* (2005)). With this incomplete dataset at centralized sink node, the data filtering, data reconstruction is not possible. Communication is widely considered as the dominating energy cost in many sensor network applications. Around 94% of energy in motes is spent on some aspect of communication (Guestrin *et al.* (2004)). The power cost of sending 1 kb over of 100 m distance is more or less equate to executing 3 million instructions (Guestrin *et al.* (2004)). The local data processing is essential in reducing energy usage in a multi-hop sensor network (Akyildiz *et al.* (2002)). A fully distributed fuzzy based trust management system is implemented in (Hossein *et al.* (2015)) to increase the security and to determine the abnormal doings of nodes in the network. The advantage of this system is the fuzzy nature of trust evaluation, and it increases cooperation among the nodes. It observes the behavior of neighbors and evaluates the trust score based on observations. It combines direct and indirect trust to get robust trust value, and it considers multiple attributes like energy, bandwidth, buffer, accuracy and the reliability of data for trust evaluation.

An effective distributed trust model for WSN is introduced in (Jiang *et al.* (2015)). It considers communication trust, energy trust, and data level trust for direct trust evaluation. Trust familiarity and reliability are introduced to increase the efficiency of indirect trust evaluation. They have used probability distribution for data trust evaluation. The ray projection method is used to evaluate the energy trust. The communication trust is calculated from the successful and unsuccessful transmission of packets. The proposed model works for single-hop and multi-hop networks.

An algorithm is proposed in (Momani *et al.* (2010)) based on Bayesian fusion to infer the trust in WSN by combining communication and data trust. This paper also discusses the fact that only one trust element is not adequate to determine the trustiness of node in WSN. The proposed algorithm is simple so that trust elements can be summated or moved out from trust models.

A distance-based trust score assessment is introduced in (Won and Bertino (2015)) which uses a centralized server to evaluate the absolute trust score for sensor node and data item. The correlation between the data items and the physical distance between the nodes are taken into account for trust evaluation. The trustiness of data item is evaluated by considering the consistency of data items among its neighbors. The trust score of

sensor node is evaluated by considering its trust score of data item and its previous trust score of the sensor node.

A systematic, centralized model is introduced in (Lim *et al.* (2010)) for assessing the data trustworthiness and node trustworthiness. The proposed model uses data provenance and sensed data item to compute the trust score. A cyclical model which uses the inter-dependency attribute is considered for the evaluation of the trust score of the sensor node. The similarity of value and provenance are considered for the evaluation of trust score of data items. The result depicts that the suggested approach is worthy for finding the trustworthiness in sensor networks. They have modeled the data generation from the event as a normal distribution.

A novel centralized trust evaluation model for WSN is presented in (Hur *et al.* (2005)). It can handle and remove the inconsistent data and efficiently detect the faulty and malicious nodes. To find the location coordinates of neighbors, the proposed method uses ECHO protocol. Then it asserts the neighbor's data item with source data item to calculate the trust worthiness of data item. The consistency of sensed value, communication parameters, and energy parameter are considered for total trust evaluation. The proposed model is light weight, and it does not employ any certificate and cryptographic mechanisms.

To determine the faulty readings in WSN, a scheme is introduced in (Xiao *et al.* (2007)). The correlation network is built by exploring the correlation between the data items.

A hybrid trust computation is proposed in (Shaikh *et al.* (2009)) in which the whole cluster will be assigned a single trust score based on direct and indirect trust value. The direct trust value is calculated by time based past interaction. The recommendation trust is employed to compute the indirect trust score. The proposed trust evaluation is flexible and simple, and it does not require many resources to store and compute the trust at sensor nodes (Han *et al.* (2014)).

Table 2.1 compares the existing trust management schemes dedicated to hybrid, centralized and distributed WSN. We can say that the existing trust management schemes focus on either communication trust or data trust. No scheme considers communication and data trust with interdependency property to score the node and data item. Most of the trust models are vulnerable to malicious attacks and suffers from memory and computational overheads (Dhulipala and Karthik (2017)).

Table 2.1: Comparison of Trust Management Schemes for WSN

| Schemes | Architecture | Methodology | Communication Trust | Data Trust | Interdependency Property | Disadvantages |
|---|--------------|--|---------------------|------------|--------------------------|---|
| ACDT (Talbi <i>et al.</i> (2017)) | Hybrid | Adaptive trust function and past interaction based trust | Yes | Yes | No | Use of spatial correlation alone for data trust |
| DTMS (Hossein <i>et al.</i> (2015)) | Distributed | Available resources and data similarity based trust | Yes | Yes | No | Weakness to sybil and spoofing attacks |
| MULTIPRO (Dogan and Avincan (2017)) | Distributed | Kalman filtering and provenance-based trust | No | Yes | No | Do not consider attacks against trust model |
| EDTM (Jiang <i>et al.</i> (2015)) | Distributed | Communication, data, energy trust | Yes | Yes | No | Defining threshold, weightage |
| DBTA (Won and Bertino (2015)) | Centralized | Distance-based and correlation based trust | No | Yes | Yes | Communication trust is not considered |
| ML-TRUST (Zhang <i>et al.</i> (2014)) | Distributed | Multi-level trust, subjective objective trust | Yes | No | No | No trust sharing and renewal |
| LDTS (Li <i>et al.</i> (2013)) | Hybrid | cluster trust computations | Yes | No | No | Vulnerable to attacks |
| TBFTDA (Sun <i>et al.</i> (2012)) | Centralized | Data aggregation | Yes | Yes | No | Do not consider attacks against trust model |
| MDETM (Wang and Wu (2011)) | Hybrid | Multi dimensional trust | Yes | No | No | Choosing multi dimensional evidences |
| HTRM (Aivaloglou and Gritzalis (2010)) | Hybrid | Certificate and behavior based trust | No | No | No | Dependent on third party |
| TMAH (Zhang <i>et al.</i> (2010a)) | Hybrid | Past interaction based trust computation | Yes | No | No | Memory and computational overhead |
| PBTA (Lim <i>et al.</i> (2010)) | Centralized | Provenance based trust | No | Yes | Yes | Use of normal distribution |
| GTMS (Shaikh <i>et al.</i> (2009)) | Hybrid | Past interaction based Trust | Yes | No | No | Do not consider attacks against trust models |

From the recent literature of TMS, we found the following: 1. Most of the existing TMS are entity-centric (Han *et al.* (2014)), whereas the WSNs are data-centric. 2. Considering only data trust for scoring the data items and sensor nodes (Momani *et al.* (2010)) without communication trust is not sufficient and efficient in WSN (Jiang *et al.* (2015)). 3. We require an in-network real-time data trustworthiness assessment to find out the faulty data (Fang and Dobson (2013)). 4. TMS should be simple without any constraints on node resources and attack resistant.

Only a few existing mechanisms like (Karthik and Ananthanarayana (2016)), (Xiao *et al.* (2007)), (Gao *et al.* (2018)) use correlation among data items to compute the trust score. None of the existing approaches consider node trust, data security, interdependency property, data provenance together for trust evaluation. Only one trust component is not enough to decide the trustworthiness of node and data item in WSN (Momani *et al.* (2010)). And existing approaches suffer from computational, communicational overhead (Jiang *et al.* (2015)) and several attacks. TMS for data primarily focus on data fault detection which we elaborate in the next subsection with recent works on sensor networks.

2.1.1 Data fault detection in sensor networks

Data fault detection has earned much care, especially in WSN for providing better service. Data fault detection can be classified into four methods: Heuristic methods, learning methods, correlation methods and time series analysis.

2.2.1.1 Heuristic methods

It relies on the development of heuristic rules based on expert knowledge of WSN to identify the data fault. One of the recent heuristic methods in data fault detection (Karthik and Ananthanarayana (2017b)), (Karthik and Ananthanarayana (2016)) is defining the range of values for valid sensor readings and residual battery level of the node. Most of the heuristic methods for data fault detection work with periodic and real-numbered readings. However, heuristic methods demand expert knowledge on defining the range of values for MSN (Ye *et al.* (2016)).

2.2.1.2 Learning methods

According to (Ye *et al.* (2016)), learning methods integrate correlation and prediction methods. It requires a substantial amount of training data to train the data model for the classification of normal and faulty data. Neural networks (Paschalidis and Chen

(2010)), hidden Markov models (Paschalidis and Chen (2010)) and Bayesian models (Dereszynski and Dietterich (2011)) are commonly used for data fault detection. Even though these models give more accuracy in data fault detection, it requires quality training data for the modeling.

2.2.1.3 Correlation methods

There is a statistical correlation of sensor readings among heart rate, pulse rate, blood pressure and body temperature (Osman *et al.* (2013)), (Salem *et al.* (2013)). For example, the sensor values of heart rate and pulse rate show almost the same values of phenomena even though they are deployed at different places in the human body. Most of the existing techniques (Salem *et al.* (2013)), (Salem *et al.* (2014)) uses temporal and spatial correlations to identify and isolate faulty data.

2.2.1.4 Time series analysis

In time series analysis, the actual sensor measurement is compared with predicted value using previous sensor measurements. If the difference between the actual and predicted values goes over the threshold, then the sensor measurement is faulty. Autoregressive model, moving average model and autoregressive integrated moving average are commonly used in time aeries analysis for data fault detection (Ye *et al.* (2016)).

From existing works of data fault detection in sensor networks, we made the following observations:

1. For large scale applications, centralized detection of data fault is not suitable, and it suffers from larger overheads.
2. Distributed data fault detection has advantages over centralized detection; however, it does not guarantee the sufficient amount of data at the sink node for accurate event detection.
3. There is a need for distributed online data fault detection at sensor node to avoid the communication of faulty data from source to sink node for reducing resource consumption and we require a centralized approach at sink node to identify the faulty data and medical event from an application point of view.
4. The learning methods of data fault detection are useful when there is no spatial and temporal correlation among sensor readings.

After detecting data faults in MSN, the data reconstruction process is used to reconstruct the data faults and data losses which we list out with recent works on sensor networks in next subsection.

2.1.2 Data reconstruction schemes in sensor networks

The data reconstruction for data faults and data loss is very important in pervasive application to identify the environment condition and decision-making process. The resource restriction of sensor nodes, open wireless link, malicious attacks, and frequent mobility may cause data loss, data fault, and poor data quality. Those unqualified data for medical diagnosis process must be reconstructed to detect the real condition of the observed patient. K-nearest neighbor (KNN) (Cover *et al.* (1967)) is a traditional data reconstruction method which uses neighbor values to predict the data fault, data loss and poor quality data. Clustering based methods (Rajasegarar *et al.* (2006)) gather identical data into clusters to predict the data fault and data loss. Spatio-temporal correlated data is used in (Kong *et al.* (2013)) along with compressive sensing method to improve the data reconstruction process. Recently Bayesian network based data reconstruction (Zhang *et al.* (2016a)) uses the conditional probability of sensor data for data recovery. From related works of data reconstruction in sensor networks, we found that, State-of-the-art techniques use redundant information of a sensor node (by deploying many sensor nodes for single phenomena) for reconstructing the data for data loss. This type of redundant information is not possible in a medical sensor network.

There are two types of data outliers in sensor networks: data faults and event. Data faults should be detected, isolated and reconstructed for improving data quality which ensures the reliability of the application. In the same way, events also should be detected for medical diagnosis and treatment, which we explain in the next subsection and list out recent works in sensor network.

2.1.3 Event detection in sensor networks

Event detection is an important issue in WSN. The WSN collects data from environment and transmit to the sink node for detecting events. Detecting the medical emergency at the right time, requires a high true positive rate and less false positive rate event detection process (Nasridinov *et al.* (2014)). There are three categories of event detection in WSN. They are statistical, probabilistic and machine learning methods.

According to (Wittenburg *et al.* (2012)), there are four approaches in sensor networks to transmit data and detect events. The simple and basic approach in event detection is local detection, where local sensor node processes its data and decides about the event. The second approach is a decentralized approach, where the cluster head is responsible for event detection. The similar kind of nodes are grouped together to form clusters. The observed data from sensor nodes are transmitted to the cluster head. The third type

of event detection approach is a centralized approach, where all raw data from sensor nodes are transmitted to the sink node for processing and detection of events. The fourth and final type of event detection approach is distributed approach, in which node processes its data and communicate with other nodes and detect the event independently without the help of sink node.

There are three algorithmic approaches for data processing and event detection ([Wittenburg et al. \(2012\)](#)). The first approach in event detection approach is threshold based, where sensor values are compared against the threshold values defined by experts. The second approach is pattern recognition, in which substantial resources are consumed for data processing and event detection. We can detect a variety of complex events like vehicle classification ([Duarte and Hu \(2004\)](#)), fence surveillance and motion of humans ([Ghasemzadeh et al. \(2010\)](#)). The third and final algorithmic approach is anomaly detection where it detects the unusual events by learning the behavior of the application over time. Light tracking is an example for anomaly detection ([Wälchli et al. \(2007\)](#)). From recent works of event detection in the sensor network, we observed that statistical category is suitable for event detection in MSN. In data transmitting, processing and detecting events, the centralized approach is appropriate for MSN. The anomaly detection ([Wittenburg et al. \(2012\)](#)) is suited in algorithmic approaches of data processing and event detection in pervasive healthcare. Existing event detection methods of MSN ([Osman et al. \(2013\)](#)), ([Salem et al. \(2013\)](#)), ([Salem et al. \(2014\)](#)) detect the medical event when two or more sensor data has abnormal value. The patient activity is also a reason for abnormal values of two or more sensors. So event detection methods should consider the patient activity information with vital sign information for identifying the real medical event. The review of existing TMS in MSN are given in the next subsection.

2.2 Trust Management Schemes in Medical Sensor Networks

A Trust Management Scheme for MSN is an important topic in pervasive healthcare to overcome the internal attacks, data faults, data loss and identifying the selfish behavior of nodes. Research in TMS of pervasive healthcare is still in infancy state. In this section, we discuss the recent works of TMS for MSN, data fault detection, data reconstruction and event detection in sensor networks. A TMS for Body Area Networks (BAN) is proposed in ([Li and Zhu \(2014\)](#)). It uses collaborative filtering for finding the trust value of sensor nodes in BAN. Each node in BAN maintains the recommendation trust ratings for other devices in the network. Cosine similarity is used to measure the trust rating between nodes. To handle the problems like wrong data, eavesdropping, a

trust management model is proposed in (Bui (2011)). The proposed trust management model checks the trustiness of components and their usage of data.

Trust based secure routing protocol for the medical sensor is introduced in (Boukerche and Ren (2009)). The trust score of a node is identified with a voltage of battery terminal, the strength of the received signal and mobility model of the node in fault aware trust determination algorithm (Chitra (2018))for BAN. The proposed algorithm is used to classify the packet and transmit through a trusted path to sink. A trustworthy architecture for wireless body sensor network is proposed in (Kanaga (2018)) for having trustworthy communication among medical sensor nodes. An uncertainty based trust model is introduced in (Yu *et al.* (2010)) to mitigate the effects of malicious attacks on location tracking. To increase the robustness, to identify malicious attacks and to avoid the link contains untrustworthy nodes, a dynamic trust model is introduced in (Gao and Liu (2014)) which is based on Bayesian inference and Tsallis entropy.

A trust model based on fuzzy set is proposed in (Wu *et al.* (2014a)) for key distribution. The proposed trust model calculates the trust value of node based on four metrics: number of successful communication, packet drop ratio, and forwarding ratio and battery level of the node. To find the trustiness of sensor readings, a trust evaluation framework is introduced in (Bui *et al.* (2013)) for Body Sensor Networks (BSN). Based on the quality of sensor reading and opinions from others, the trustiness is evaluated. An attack resistant, lightweight trust model is proposed for two-tier architecture in (He *et al.* (2012))for BSN. It identifies malicious attacks, ignores faulty nodes and significantly increases the network performance. To monitor and evaluate the trustiness at component level, system level and application level, a trust management model is proposed for BSN (Bui *et al.* (2011)).

Table 2.2 compares the recent TMS of MSN in five aspects: Purpose, Node Trust, Data Trust, Advantages and Disadvantages.

Table 2.2: Comparison of TMS for MSN

| TMS | Purpose | Node Trust | Data Trust | Advantages | Disadvantages |
|---|---|------------|------------|--|---|
| BAN-TRUST (Li and Zhu (2014)) | Measuring trustiness of BAN | Yes | No | Recommendation based trust is calculated for interaction | Data trust is not considered |
| TMS-BSN (Bui (2011)) | To identify the trustiness of components | Yes | No | Enhances the system dependability and security | Data trust is not considered |
| SRM-TRUST (Gao and Liu (2015)) | To detect the trustworthy path from source to destination | Yes | No | Use of Tsallis entropy for the detection of trustworthy path | Interdependency property & data provenance are ignored. |
| FAT-WBSN (Chitra (2018)) | To evaluate the trust value of MSN | Yes | No | Enhances the lifetime of network | Data trust is not considered |
| TWA-WBSN (Kanaga (2018)) | For trusted and confidential communication | Yes | No | Use of finite state machine and Markov model | Data trust is ignored. |
| TAHM (Yu <i>et al.</i> (2010)) | To mitigate the malicious attacks against BAN | Yes | No | Trust-aware location tracking to identify malicious activities | Data related attacks are not considered |
| Be-TRUST (Gao and Liu (2014)) | Construction of trust model for node and path | Yes | No | Identification of dynamic node behavior, improving robustness and adaptability | Interdependency property & data provenance are ignored. |
| FTM-TRUST (Wu <i>et al.</i> (2014b)) | Fuzzy trust model to choose the trustworthy neighbors | Yes | No | Use of multiple metrics for trust evaluation | Data trust is ignored. |
| TEF-BASN (Bui <i>et al.</i> (2013)) | Evaluation of sensor reading | No | Yes | Use of subjective logic and opinion generations | Data loss and malicious attacks are not considered. |
| ReTRUST (He <i>et al.</i> (2012)) | To evaluate the trustiness of MSN | Yes | No | Detecting malicious behaviors, increases network performance | Data trust and data related attacks are not considered |
| TMM-BSP (Bui <i>et al.</i> (2011)) | Managing the trustiness of components in BSP | Yes | No | Improves system security, trust prediction | Data trust is ignored. |

From Table 2.2 and recent works of TMS in MSN, we made the following observations:

1. Most of the TMS focus on trust evaluation of sensor node (He *et al.* (2012)), (Yu *et al.* (2010)), (Boukerche and Ren (2009)), (Li and Zhu (2014)), (Kanaga (2018)), (Chitra (2018)), (Bui *et al.* (2011)) to find total trustiness of pervasive health.
2. Sensor network used in pervasive healthcare is data-centric in nature. The effective decision and necessary actions are taken from medical sensor data.
3. Considering only one element for trust evaluation is not sufficient, and it cannot represent the total trustiness of the pervasive healthcare system.

2.3 Trust-based Data Collection and Trust-based Data Aggregation

In this section, we look at recent research on trust-based data collection and data aggregation process. In all pervasive applications, data gathering is the main procedure taking place in a sensor network, where the sink node gathers all sensor data for data analysis and decision making. Data gathering involves data collection without aggregation and data collection with aggregation is known as data collection and data aggregation respectively. Data aggregation normally calls for a coalition of data from many nodes at the intermediate node and forwards the aggregated data to the sink node in an energy efficient and delay aware manner. To carry out, data aggregation process, the aggregation schedule should be free from interference. In some of the pervasive applications, the sink node requires to gather all sensor data from nodes without aggregation. This process is called data collection. In (Fasolo *et al.* (2007)), research on recent data collection, data aggregation techniques and their effects on resource consumptions are given. Furthermore, several surveys have been carried out in (Guo *et al.* (2011)), (Jesus *et al.* (2015)), (Sang *et al.* (2006)) on data collection, data aggregation process. We sum up the recent advancements of trust based data collection, data aggregation process from five aspects: Energy Efficiency; Delay Aware; Node Trust; Data Trust and Data Reconstruction which is given in the following Table 2.3.

Table 2.3: Recent works of trust-based data aggregation and trust-based data collection

| References | Approach | Energy Efficiency | Delay Aware | Node Trust | Data Trust | Data Reconstruction |
|--------------------------------------|-------------|-------------------|-------------|------------|------------|---------------------|
| (Taghikhaki <i>et al.</i> (2011)) | Aggregation | Yes | Yes | No | No | No |
| (Vijayalakshmi <i>et al.</i> (2013)) | Aggregation | Yes | Yes | No | No | No |
| (Liu <i>et al.</i> (2013a)) | Aggregation | Yes | No | No | No | No |
| (Rezvani (2015)) | Aggregation | Yes | No | Yes | No | No |
| (Ma <i>et al.</i> (2015)) | Aggregation | Yes | No | No | No | No |
| (Vamsi and Kant (2016)) | Aggregation | Yes | No | No | No | No |
| (Liu <i>et al.</i> (2016)) | Aggregation | Yes | No | No | No | No |
| (Gilbert <i>et al.</i> (2018)) | Aggregation | Yes | No | Yes | No | Yes |
| (Gao <i>et al.</i> (2018)) | Aggregation | No | No | Yes | Yes | No |
| (Ramalingam (2006)) | Collection | Yes | No | No | No | No |
| (Luo <i>et al.</i> (2009)) | Collection | Yes | No | Yes | No | No |
| (Gomez <i>et al.</i> (2011)) | Collection | Yes | Yes | No | No | No |
| (Whitehead (2016)) | Collection | Yes | Yes | No | No | No |
| (Chittibabu <i>et al.</i> (2018)) | Collection | Yes | No | No | No | No |
| (Puneeth <i>et al.</i> (2018)) | Collection | Yes | Yes | Yes | No | No |

Most of the works of trust-based data collection and trust-based data aggregation processes focus on minimizing energy consumption and delay. None of the works focus on the quality of data and quality of node before performing data collection and data aggregation operation to minimize energy consumption and delay. Only a few works like (Karthik and Ananthanarayana (2018a)), (Gilbert *et al.* (2018)), (Hossein *et al.* (2015)) use a trust for data reconstruction of faulty data and missing values. To address these issues, we propose a TDG which handles node misbehavior, selfish behavior, faulty data, and missing data in energy efficient and delay aware manner.

2.4 Ontology Matching

In this section, we discuss the recent works of ontology matching techniques. Ontology matching could supply a semantic connection between several ontologies for accessing and exchanging data semantically. In general, There are three types of ontology matching process: direct matching, indirect matching and hybrid matching. Direct matching process uses multiple ontology architecture to find the set of correspondence among concepts. In the indirect matching process, the global shared vocabulary is used as background knowledge for finding semantic correspondence among various concepts. The hybrid matching is the combination of direct and indirect ontology matching for establishing semantic correspondences among similar concepts of various Ontologies (Cerdeira (2014)). Notable methods of instance-based ontology matching techniques are reviewed and their future research directions are highlighted in (Abubakar *et al.* (2018)). It is found that similarity-based and machine learning based methods are renowned techniques in instance-based ontology matching (Abubakar *et al.* (2018)). A hybrid ontology matching technique is proposed in (Wang *et al.* (2012)), in which multiple matchers are used to find similarities between elements of the ontology. It uses hierarchical information to find weights of similarities. Finding the semantic similarity of Ontologies based on Word-Net and structure level is introduced in (He *et al.* (2011)). The combination of ontology driven and keyword matching system is introduced as a hybrid approach in (Ducatel *et al.* (2006)). Trust mechanism supplies a framework that infers a correct and wrong matching among entities of pervasive environments with trust metrics (Liu *et al.* (2013b)), (Wu *et al.* (2016)), (Xiong *et al.* (2017)), (Wang *et al.* (2015)). The comparison of recent instance based ontology matching techniques are given in Table 2.4. But none of them consider the trustiness of instances and entities for ontology matching, and there is no support for large scale matching (Jiang *et al.* (2016)).

Table 2.4: Comparison of instance-based ontology matching techniques

| Methods | Approaches | Support of Trustiness |
|----------------------------------|---------------------------|-----------------------|
| (Jean-Mary <i>et al.</i> (2009)) | Context-based matching | No |
| (Jiménez-Ruiz and Grau (2011)) | Context-based matching | No |
| (Nath <i>et al.</i> (2012)) | Similarity-based matching | No |
| (Faria <i>et al.</i> (2013)) | String-based matching | No |
| (Diallo (2014)) | Contextual-based matching | No |
| (Khiat <i>et al.</i> (2015)) | String-based matching | No |
| (Khiat and Benaissa (2015)) | String-based matching | No |

2.5 Upper Ontology

In this section, we discuss recent works on upper ontology. Upper ontology delineates universal conceptions that are independent of a specific problem or domain. It supplies a class of things, entities and their relationship for providing fundamental structure to domain and application ontology. We sum up the ontological commitments (Khan (2012)) of various upper ontologies like DOLCE (Guarino (2003)), BFO (B.Smith (2002)) and ONTONYM (Stevenson *et al.* (2009)) in Table 2.5.

Table 2.5: Comparison of Upper Ontologies

| Ontological Commitments | DOLCE | BFO | ONTONYM |
|--------------------------------|------------------------|-------------------------|------------------------|
| Representation of entities | In natural language | Represent as it is | Not clear |
| Support Instances | No | Yes | Yes |
| Multiple or single objects | Multiple objects | Only one object | Not clear |
| Real existence of objects | Yes | All object are real | Not clear |
| Past, present and future | Yes | Yes | Yes |
| Presence of entities | Both in space and time | Either in space or time | Both in space and time |
| Temporal Aspects | Provided | Not provided | Provided |
| Attributes and values | Included | No | No |
| Time and space modeling | No support | Supports modeling | Supports modeling |
| Layers (basic/ abstract/ core) | Basic | Basic | Not clear |
| Situations | Not clear | No | Not clear |
| Complex Event detection | Not clear | No | Not clear |

Most of the upper ontologies focus on temporal and spatial modeling; there is no support for complex event detection for accessing situations. None of the upper ontologies consider the trustiness of entities and instances for event detection and extracting the knowledge from situations.

2.6 Outcome of the Literature Survey

- In large scale applications, the centralized data model causes big overheads and consumes more network resources. From the recent literature, it is observed that in-network and local processing of sensor data reduces the energy consumption of communication (Puliafito *et al.* (2019)).

- Distributed data fault detection has advantages over centralized detection; however it does not guarantee the sufficient amount of data at the sink node for accurate event detection ([Chen et al. \(2006\)](#)).
- There is a need for a node to take a local decision like filtering the faulty data, selecting a secure node and ignoring the malicious, selfish node. The distributed approach works well at detecting the malicious and selfish node than the centralized scheme ([Yarinezhad and Hashemi \(2019\)](#)).
- State-of-the-art techniques use redundant information of a sensor node for reconstructing the data for data loss ([Gilbert et al. \(2018\)](#)).
- From recent works of event detection in the sensor network, it is noted that statistical category is suitable for event detection in WSN. In data transmitting, processing and detecting events, the centralized approach is appropriate for WSN ([Nasridinov et al. \(2014\)](#)).
- Existing event detection methods of MSN detect the medical event when two or more sensor data has abnormal value. The patient activity is also a reason for abnormal values of two or more sensors in PHS. So event detection methods in PHS should consider the patient activity information with vital sign information for identifying real medical event ([Karthik and Ananthanarayana \(2018a\)](#)).
- In most of the cases, the data collected from one sensor node is used for only one application and then ignored. All are application specific ([Chen et al. \(2009\)](#), [Cao et al. \(2016\)](#)).
- The generated data can be shared among different applications in pervasive environment for increasing the user comfortableness, reliability of the application and achieving the full potential of the application. The shared data plays a vital role in critical decision making ([Cao et al. \(2016\)](#)).
- To share and integrate data semantically, ontology matching technique establishes a semantic correspondence among various entities of pervasive application ontologies ([Abubakar et al. \(2018\)](#)).
- None of existing ontology matching techniques consider the trustiness of instances and entities for establishing a semantic correspondence between entities ([Karthik and Ananthanarayana \(2018b\)](#)).
- Research has to be carried out for dealing with a combination of direct and indirect ontology matching. High-quality upper ontology is required for better ontology matching ([Otero-Cerdeira et al. \(2015\)](#) & [Li et al. \(2019\)](#)).

2.6.1 Research Motivation

The outcome of the literature survey listed in section 2.6 leads to the following research motivation.

- Sensor-driven pervasive computing faces the challenge of how to model and reason on such massive amounts of data and how to facilitate sharing and interoperability across heterogeneous systems (Ye *et al.* (2015)).
- In most cases of pervasive application, data collected from one sensor are used only for one purpose and then discarded (Chen *et al.* (2009)). Pervasive applications attain their full potential only when data is shared among them. The absence of trust could affect the acceptance of sharing data in pervasive applications (Cao *et al.* (2016)).
- A significant amount of sensor data were affected by data faults in real time observing sensor-driven pervasive applications. For instance, i) 51% of data items of macroscope project (Tolle *et al.* (2005)) were found to be untrustworthy; ii) 3-60% of data items of great duck island experiment (Kamal *et al.* (2013)) were found to be untrustworthy; iii) In INTEL lab experiment (Sharma *et al.* (2010)), 20-25% of data items were untrustworthy and iv) In NAMOS experiment (Fang and Dobson (2013)) 15-35% data samples were untrustworthy.

2.6.2 Motivating Examples

- How can a smart home energy system meaningfully use traffic information, to predict a user arrival?
- How can a smart health care application trust the quality of data coming from health monitoring sensors and smart home sensors to suggest the treatment?
- How can an intelligent traffic control system effectively uses pollution data monitored in a city, to design a pollution free route?

2.6.3 Problem Definition

To design an efficient framework for finding trustworthiness and integrating context-aware sensor-driven pervasive applications through ontologies.

2.6.4 Research Objectives

With reference to the above research motivations, motivating examples and problem definition, the following research objectives are identified:

- Data modeling of low-level sensor data for trustworthiness.
- Finding the trustworthiness of sensor node and data for data gathering.
- Finding the trustworthiness of sensor node and data in monitoring single event.
- Finding the trustworthiness of sensor node and data in monitoring multiple events using contextual information.

- Construction of upper ontology and hybrid ontology matching technique for integrating trusted context-aware sensor-driven pervasive applications.

2.7 General Methodology

To accomplish the objectives of the research work, the trusted semantic framework is proposed for integrating context-aware pervasive applications and is shown in Figure 2.2. The following subsections explain the process of a trusted semantic framework

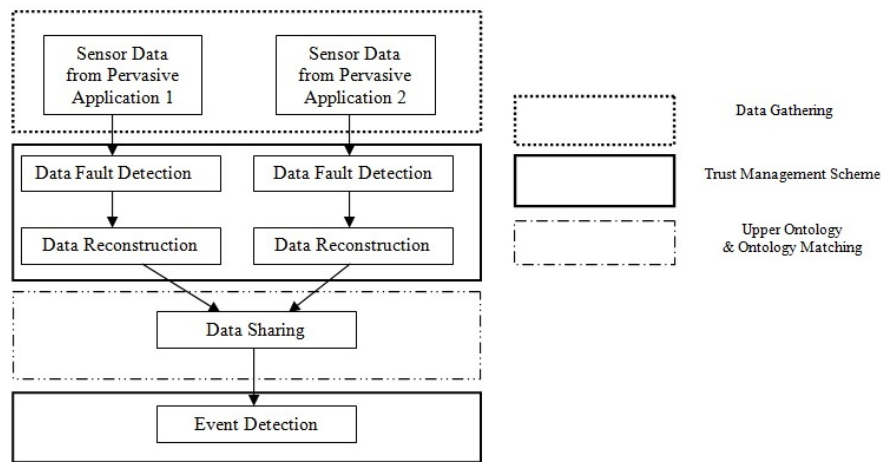


Figure 2.2: Trusted semantic framework for context-aware pervasive applications

for event detection in context-aware sensor-driven pervasive applications. Raw sensor data is gathered from pervasive applications with the help of trust-based data gathering process. Trust-based data fault detection is used to find the data fault in the gathered sensor data. Data faults and data losses can be reconstructed with the help of trust-based data reconstruction process (Karthik and Ananthanarayana (2017a) & Gilbert *et al.* (2018)). Upper ontology is developed for pervasive environments to establish semantic data sharing between pervasive applications (Karthik and Ananthanarayana (2018b)). The shared data plays an important role in event detection and decision-making process.

2.7.1 Data Gathering

In all pervasive applications, a gathering of sensor data from the environment is the main operation held in a sensor network, where sink node or base station gathers all generated data to do data analysis and decision making. The data generated by the sensor node in the pervasive environment should be transmitted to the sink node for event detection. Data gathering is primarily employed for gathering interesting sensor data from environments, finding the size of the network, deciding mean system load and

so on. Data gathering involves data collection without aggregation and data collection with aggregation known as data collection and data aggregation respectively (Karthik and Ananthanarayana (2019)).

2.7.2 Trust Management Scheme

TMS has been proposed to handle the node misbehavior, data related attacks, faulty data generation and data loss in the pervasive application. We identify untrustworthy node and data in sensor networks with the help of trust-based fault detection process, reconstruct the untrustworthy data and data losses with the help of trust based data reconstruction process and identify the events with the help of trust based event detection process (Karthik and Ananthanarayana (2017a)). Additionally, we have trust-based data collection, trust-based data aggregation and trust-based event detection are used to ease and ensure the trustworthy data exchange among trustworthy nodes for identifying the events (Gilbert *et al.* (2018)).

2.7.3 Upper ontology and ontology matching

An upper ontology talks about the general concepts in all domains. The main functionality of an upper level ontology is to offer the semantic interoperability between ontologies that are accessible through upper ontology. Ontology matching supplies a semantic correspondence between the entities of pervasive application ontologies for exchanging data semantically. We used four main concepts for upper ontology of pervasive environments. They are temporal properties, spatial properties, entities, and trust management to ensure trustworthy data exchange and reliable event detection. Hybrid ontology matching which combines direct and indirect matching is used to establish the semantic connection between different pervasive applications (Karthik and Ananthanarayana (2018b)).

2.8 Summary

This chapter provided a review of existing TMS in sensor networks, trust-based data collection, trust-based data aggregation, and ontology matching techniques. The research motivation, motivating examples, problem statement and research objectives were framed based on the outcome of the literature review. The proposed methodology and a short description of the research work were presented. The organization of this thesis with respect to five contribution chapters and five research objectives is shown in Figure 2.3.

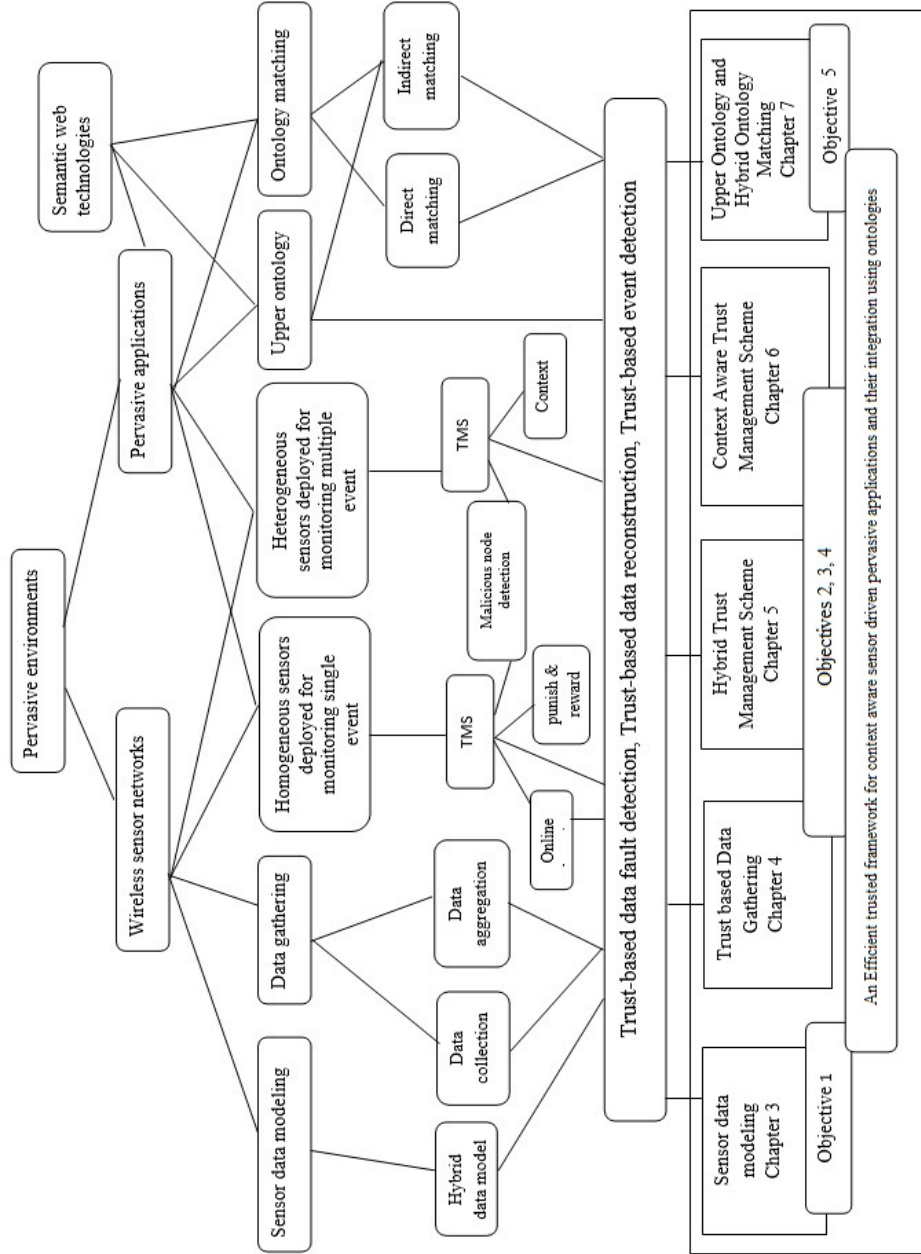


Figure 2.3: Organization of Thesis with respect to chapters and objectives

Chapter 3

Sensor Data Modeling for Data Trustworthiness

3.1 Preamble

In this chapter, we address the first research objective, data modeling of low level sensor data for trustworthiness. The nodes in the network are resource constrained in nature and face several challenges for producing the data from the unfriendly environment. A large amount of data is generated from WSN and suffers from data fault, inaccuracy and inconsistency. To increase the reliability of the application, several data trust management schemes are introduced to ensure the trustworthiness of data in the decision-making process. Apart from these schemes, in the absence of ground truth, sensor data models are used to find the trustiness of the sensor data. The data generated from the simulation of the data model is used as a metric to evaluate the degree of trustiness of sensor data. The existing sensor data models suffer from high energy consumption for data trustiness detection, and it becomes inaccurate when the data fault rate is high. To overcome this limitation, we are proposing an energy efficient sensor data model for evaluating the sensor data trustworthiness and reconstruct the sensor data in case of any data loss and data fault. The proposed sensor data model is hybrid in nature, which is also used to detect the events reliably.

The contributions of this chapter are follows:

1. A hybrid sensor data model is proposed for data fault detection, data reconstruction and event detection in sensor networks.
2. Analysis of energy consumption of sensor data model for detecting data faults in centralized, distributed, and the hybrid environment is given.

The rest of this chapter is organized as follows: The features of sensor data and assumptions are explained in subsection 3.2. In section 3.3, the proposed hybrid sensor data model and algorithms are introduced. Results and discussions are given in section 3.4. A summary is given at the end of this chapter.

3.2 Sensor Data Features

In this section, the various features of sensor data like data correlations and data provenance are highlighted. Motivations, assumptions and types of data loss in WSN for the

proposed data model are also explained.

3.2.1 Data correlations and Data Provenance

1. Temporal correlation: Sensor data generated at adjacent timestamps are almost similar.
2. Spatial Correlation: Sensor data generated from sensor nodes which are geographically nearer to each other are anticipated to be the same.
3. Attribute Correlation: Sensor data generated from different sensor of same events seems to be correlated with each other.
4. Data provenance: It gives information about the source node, the path is taken by the data item to reach the sink node in the multi-hop network, different versions of the data item and the undergone operations since its generation.

3.2.2 Motivation

In WSN applications, substantial quantities of sensor data were found as untrustworthy data. Due to frequent untrustworthy data sampling in real-world WSN applications, there is a need for an online hybrid sensor data model for in-network detection of untrustworthy data at the sensor node level, data reconstruction and event detection at the sink node.

3.2.3 Assumptions

We assume that sensor node location coordinates are available, which are critical for deciding spatial correlation. Battery state of sensor node is known because low battery power of sensor node may cause the sensor node to produce erroneous data. The behavior of the phenomenon should be defined by experts to get the expected rate of change. Here we are assuming that the data trust score ranges from -1 to +1 as in ([Karthik and Dhulipala \(2011\)](#)). The trust score from -1 to -0.4 denotes untrustworthy data item, the trust score ranges from -0.3 to +0.2 denotes uncertain data item and trust score ranges from +0.3 to +1 denotes trustworthy data item. If the number of sensed data item is less than 5, with this small sample size, the correlation estimation is extremely noisy. The bigger sample size is better.

3.2.4 Data loss in WSN

The different types of data loss in WSN are follows:

1. Element random loss: The simple pattern of data loss which occurs randomly due to the collision and noise in the network.
2. Block random loss: The neighbor node data were lost in a contiguous manner in the network. The congestion in the network is the main cause for this type of data loss.
3. Element frequent loss: The poor link quality between the nodes leads to frequent data loss. The untrustworthy link and sporadic data transmission are also reasons for element frequent loss.
4. Successive element loss: The battery depletion of the node is the main reason for this type of data loss pattern.

3.3 Proposed Sensor Model

In this section, the proposed hybrid sensor data modeling steps are explained and the hybrid sensor data model is shown as a flowchart in Figure 3.1. When data is generated at the source node, the localized data trustiness detection is used by source node to find its sensed data trustiness. If the data is trustworthy, then it is forwarded to trustworthy neighbor node. Node trustiness and data provenance are used to identify the trustworthy neighbors. Peer node data trustiness detection is used by the neighbor node to find trustiness of received data. Both neighbor and source nodes are deployed in the same region for monitoring the same event. The generated data is forwarded to the sink node via various intermediate nodes. The global data trustiness detection is used by the sink node to find the received data trustiness. Environmental model and disruption are used to increase the detection of data faults. If the received data is untrustworthy, the data reconstruction method is used to reconstruct the data faults and data loss. The event detection method is used to detect events reliably.

3.3.1 Localized data trustiness detection

The localized data trustiness detection includes the processing of checking the remaining battery power when it produces the data item. Then the source sensor node uses any one of the prediction models like moving average or auto regressive techniques to estimate the future data series. After estimating the future data, the original data is

compared with estimated data to evaluate the data trustiness. The error rate is calculated for all data items, and it is used for detection of data faults based on the application allowable threshold that is fixed during sensor network modeling.

3.3.2 Peer node data trustiness detection

The sensed data item is forwarded to the sink node through multi-hops. The neighbor node receives the data item from the source node and calculates the correlation coefficient between its data item and source node data item. The source node and neighbor nodes are deployed in the same region to monitor the same event. They are supposed to produce almost the same data. Due to this fact, the spatial correlation between the data items is used to find the data trustworthiness at the neighbor node. The proposed data model selects the neighbor node which has high trust value for data comparison.

3.3.3 Global data trustiness detection

At the sink node, the data from different sensory modules are collected. The correlation between sensor data from different sensor nodes is calculated. The temporal, spatial, attribute correlations and data provenance is used for data trustiness detection and data reconstruction. The untrustworthy data is reconstructed at the sink node with the help of temporal, spatial and attribute correlations. The reconstructed data is used for reliable event detection in the environment. To have efficient and reliable event detection, the sink node must ensure the minimum number of data received to reason over the environment. If the amount of data available at the sink node is not adequate for event detection, the remaining data can be reconstructed at the sink node with the help of the proposed data model.

3.3.4 Node trustiness and data provenance

The node trustiness in WSN denotes the node behavior in terms of communication capability, data generation about the event and the remaining battery level. The data provenance explains that the information includes a history of the data items starting from its generation. It gives the information about the source node of the data item and details about the intermediate nodes in which the data item is traveled to the sink node. In this proposed model, the data similarity and provenance similarity about the same event is modeled to find the trustiness of the data and data related attacks.

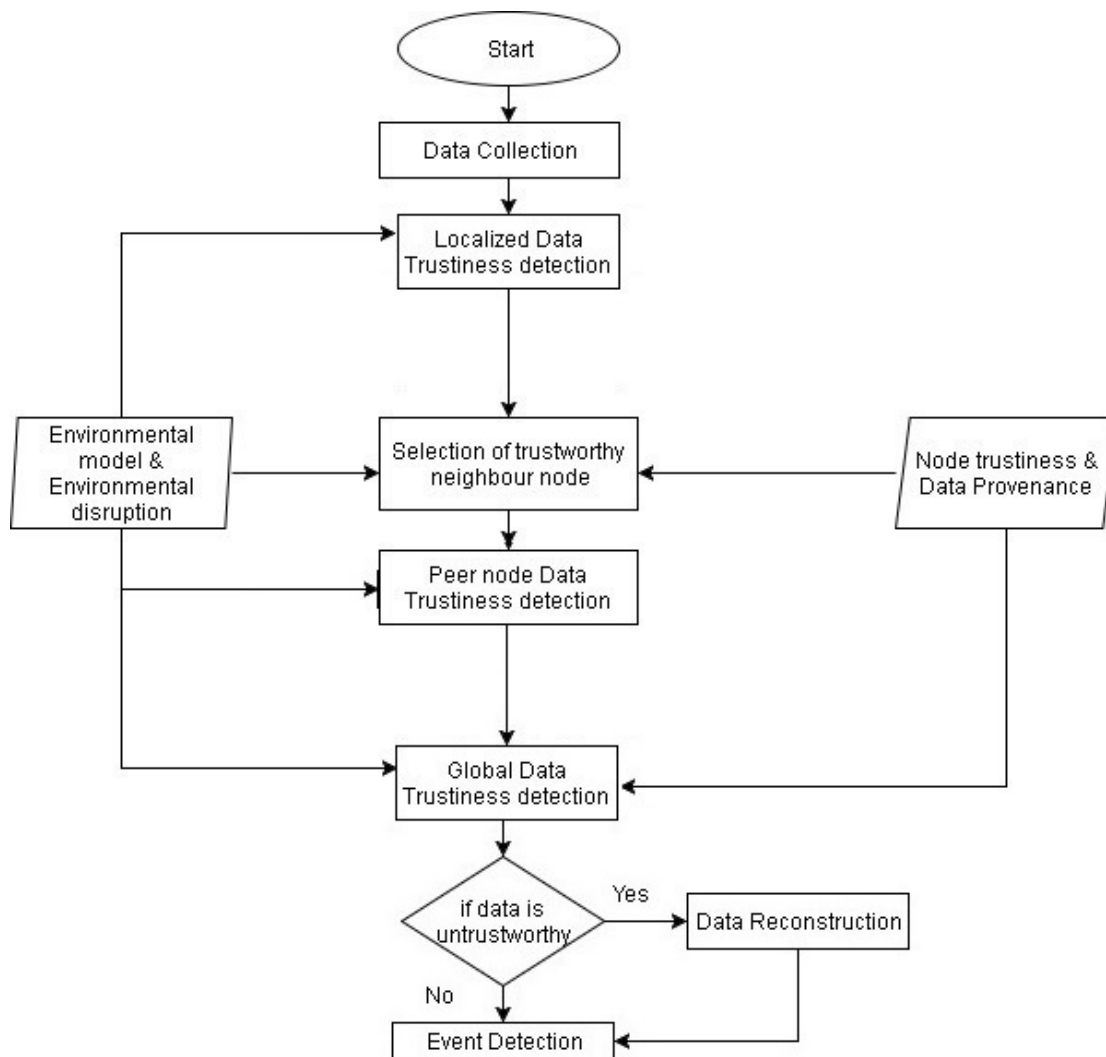


Figure 3.1: Proposed sensor data model

3.3.5 Data reconstruction

In real time experiments, the WSNs collect several attributes at a time. For example, a TelosB node gathers temperature, humidity and light data simultaneously. They have strong correlation among them. The temporal, spatial and attribute correlations are considered together to estimate the data with more accuracy for data construction. The sink node will construct the data with high accuracy by considering the data correlations from different nodes.

3.3.6 Environmental disruption & environmental model

Environmental disruption is utilized to increase the detection rate of data trustworthiness and event detection in WSN. For example, different patterns of weather and rain cause the sensor node to behave abnormally and produce abnormal data or untrustworthy data. This property is important when the sensor nodes are deployed in an outdoor harsh environment. By including the prior information about the environment disruption at the sink node for modeling, one can increase the performance of data model. The models of environment are essential in setting the anticipated doings and range of sensory systems. For example, the temperature of outdoor may not frequently change whereas the velocity of the wind will change frequently. So defining and fixing the anticipated rate of change about the sensory system is important in detecting the data trustworthiness and events.

3.3.7 Event detection

It is the process of gathering the sensor data without any data faults and data losses, recognizing the data pattern, mapping semantically for detecting the events.

3.3.8 Hybrid Sensor Data Modeling Algorithm

In this section, the hybrid sensor data model for data trustiness detection is presented in the form of an Algorithm. Algorithm 3.1 is used by the source node to find its data trustiness. Algorithm 3.2 is used by a neighbor node to find the data trustiness. Sink node uses Algorithm 3.3 and 3.4 to find the data trustiness and events.

Input: sensors id, location, time series model for prediction of sensor data, battery status of sensor nodes, environmental model, environmental disruption, data provenance, node trustiness, event boundary, battery threshold, error rate threshold and event threshold for application.

Output: Data Trustiness detection and event detection.

Algorithm 3.1: Localized data trustiness detection ()

- 1: if sensor node location is in event boundary then
- 2: if sensor battery status \geq battery threshold for the application then
- 3: if sensed data item lies between the range of restricted interval then
- 4: if sensed data item-predicted data item \leq error rate threshold then


```

5:         sensed data item = Trustworthy data item
6:         select the trustworthy neighbor node and forward it
7:     else
8:         sensed data item = Untrustworthy data item
9:         Drop the data item without forwarding
10:    end if
11: end if
12: end if
13: end if
14: Return data trustiness

```

Algorithm 3.2: Peer node detection()

```

1: if source node location & neighbor node location is in event boundary then
2:     if the neighbor node has minimum trust score then
3:         if the number of sensed data items  $\geq 5$  then
4:             if the correlation coefficient between data items  $\geq 0.3$  then source
                    node data item is trustworthy
5:                 forward the data item to next hop with a trust score
6:             else
7:                 source node data item is untrustworthy
8:                 drop the data item without forwarding
9:             end if
10:         else
11:             trust score =  $1 / (1 + |sourcenodedataitem - neighbornodedataitem|)$ 
12:         end if
13:         if trust score  $\leq 0.3$  then forward the data item to next hop with trust score
14:         end if
15:     end if
16: end if
17: Return data trustiness

```

Algorithm 3.3: Sink node detection()

```

1: Collect data items from various sensors
2: Calculate attribute correlation for all data items
3: if attribute correlation  $\geq$  application event threshold & freefromattacks() then

```

```

4:    data item is trustworthy
5:    else
6:    data item is untrustworthy
7: end if
8: if (n/2 data items) >= application event threshold & freefromattacks()= trustworthy
data item then
9:    Existence of event
10:   else
11:    No event
12: end if
13: Return data trustiness

```

Algorithm 3.4: freefromattacks()

```

1: if (data items are similar && data provenance are similar) then
2:    data item =uncertain data item
3: end if
4: if (data items are similar && data provenance are dissimilar) then
5:    data item= trustworthy data item
6: end if
7: if(data items are dissimilar && data provenance are similar) then
8:    data item=untrustworthy data item
9: end if
10: if(data items are dissimilar && data provenance are dissimilar) then
11:    data item=uncertain data item
12: end if
13: Return data trustiness and attack detection status

```

3.4 Results and Discussions

The goal of this section is to find the efficiency of the proposed hybrid data model in identifying the untrustworthy data items and events. It is an initial hybrid framework towards online detection of data faults and combines both centralized and decentralized schemes for real-time detection. The proposed scheme of hybrid nature applies a compounding of spatial-temporal and attributes analysis of data and data provenance to detect the untrustworthy data items. The scenario considered here is the indoor en-

environment where the sensor nodes (mica2dot) were deployed in INTEL Berkeley lab (Madden *et al.* (2004)) to monitor the temperature, humidity and light as shown in Figure 3.2. In this chapter, we considered only temperature and humidity readings gathered

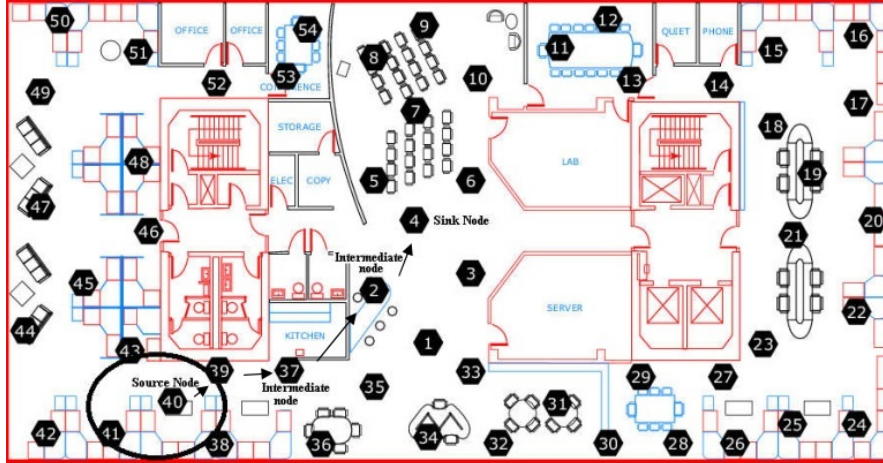


Figure 3.2: INTEL lab sensor deployment

by nodes 37, 38, 39, 40, 41, 43, 2, and 4. Here node 40 is the source node; node 4 is the sink node, nodes 38, 39, and 41 are neighbor nodes and nodes 37, and 2 are intermediate nodes. The data items collected from the INTEL Berkeley lab does not provide any fault annotations. To find the ground truth of data items, we followed two steps: First, we refer to (Sharma *et al.* (2010)) for identifying the data faults. Secondly, we manually scrutinize the dataset to identify the data faults and counter check to assure that the fault annotations are accurate as named in (Nguyen *et al.* (2013)). This direction of constructing the dataset ground truth is consistent like (Yao *et al.* (2010)) for data items with ground truth deficiency. The results obtained from the proposed model are cross verified with the manually annotated data set. Apart from these existing data faults in the data set, we inserted random data faults manually to verify the performance of the proposed sensor data model in identifying the data trustiness and event. The proposed data model detects the untrustworthy data and able to differentiate between events and untrustworthy data.

3.4.1 Detection Accuracy

Detection accuracy is an important metric to find the performance of sensor data modeling for data trustiness (Sharma *et al.* (2010)). We evaluate the detection accuracy as the ratio of the number of untrustworthy data item detected to the total number of untrust-

worthy data items. Simulations are performed for the scenario mentioned above of two intermediate nodes and one neighbor node to find the performance of proposed sensor data modeling. Figure 3.3 shows that LDTS (Li *et al.* (2013)) has less detection rate

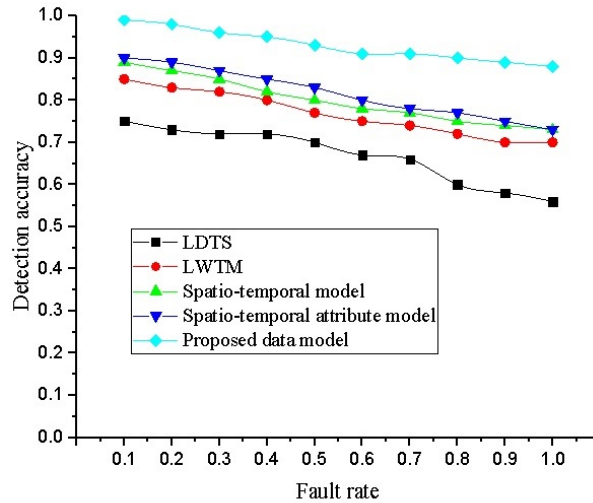


Figure 3.3: Detection Accuracy

than other methods since it considers only communication trust for detecting the trustiness and omits data fault. The consistency of data is considered in LWTM (Wang and Pang (2014)) with the remaining energy of the node. In data correlations approaches, the data related attack is not included while detecting the data trustiness. Intermediate nodes characteristics and data provenance are considered in the proposed model. It is shown in Figure 3.3 that when the fault rate is increasing; the performance of all existing approaches degrades gradually. But the proposed model maintains acceptable detection accuracy when the fault rate is high and outperforms all other existing models since it considers spatiotemporal and attribute correlations with data provenance techniques. During data trustiness detection, a certain amount of energy is consumed at the source node, intermediate nodes and sink node. The analysis of energy consumption for data fault detection in different environment is explained in the next subsection.

3.4.2 Energy Consumption Analysis for Data Trustiness Detection

For communication, processing, sensing in WSN, a particular quantity of energy is required by the sensor node. The remaining energy of the sensor node is very important in WSN applications. If the battery is depleted, then the sensor node is unable to perform its basic operations, and it becomes invalid from the network which has an impact on the reliability of the applications. According to (Li *et al.* (2013)), the energy consumed

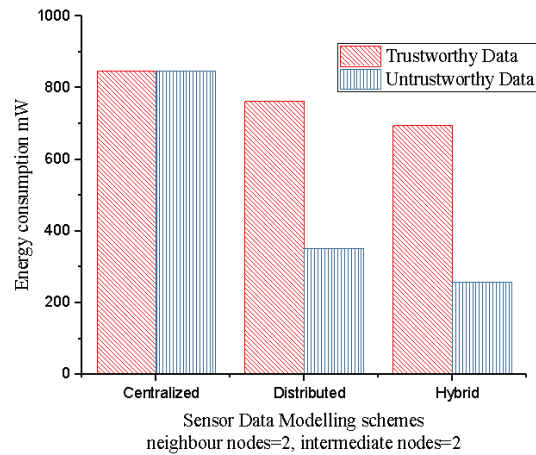


Figure 3.4: Energy consumption for case 1

by MICAz node which is operating at 7.3 MHz to perform listen, receive, compute and transmit operations are 68mW, 72mW, 26mW, 65mW respectively. In this section, the energy consumption analysis is done for different data trustiness detection techniques with different scenarios. Here we consider three different detection schemes. The centralized scheme uses centralized sink node for collecting all data items from sensor nodes to detect the data trustiness. In a distributed scheme, the source node and neighbor nodes are used to evaluate the data trustiness detection by exploiting the temporal and spatial features of sensor data. In the hybrid scheme, the source node, the trustworthy neighbor node and sink node are used to detect the data trustiness by utilizing the data correlations and data provenance. The number of neighbor nodes and intermediate nodes are chosen randomly and tested in the following cases.

Case 1: When neighbor nodes=2 and intermediate nodes =2, the centralized scheme consumes 846 mW for detecting trustworthy and untrustworthy data as shown in Figure 3.4. Irrespective of data nature, the sensed data items must be routed to the sink node for the data trustiness detection in a centralized scheme. The distributed scheme consumes 762 mW for trustworthy data and 351 mW for untrustworthy data item since it avoids the unnecessary communication of untrustworthy data item to sink node by localized detection and dropping the data item without forwarding to sink node. The proposed hybrid scheme consumes only 668 mW for a trustworthy data item and 257 mW for untrustworthy data item since it uses localized data trustiness, peer node data trustiness detection and global data trustiness detection.

Case 2: When neighbor nodes n=4, intermediate nodes=4, the centralized scheme consumes 1155 mW for untrustworthy data and trustworthy data detection. When the num-

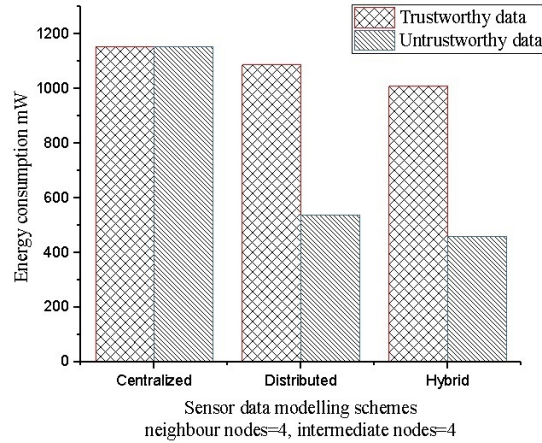


Figure 3.5: Energy consumption for case 2

ber of neighbor nodes and intermediate nodes increases, the energy consumption for data trustiness detection also increases as shown in Figure 3.5. The distributed scheme consumes 1087 mW for trustworthy data item detection and 510 mW for untrustworthy data item detection. The proposed hybrid scheme consumes only 461 mW for the untrustworthy data item detection and 1009 mW for trustworthy data detection.

Case 3: When neighbor nodes $n=5$, intermediate node $i=5$, the centralized scheme consumes 1386 mW for trustworthy data and untrustworthy data detection. The distributed scheme consumes 1318 mW for detecting trustworthy data item detection and 496 mW for untrustworthy data item detection. The proposed hybrid scheme consumes only 1206 mW for detecting trustworthy data item and 392 mW for untrustworthy data item detection as shown in Figure 3.6. Based on three different scenarios which we

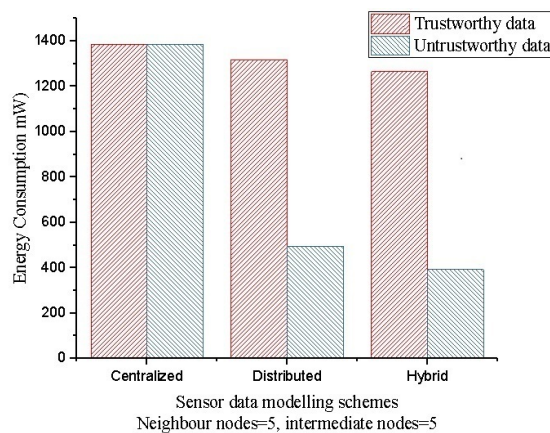


Figure 3.6: Energy consumption for case 3

took for energy consumption analysis; we can say that the proposed hybrid sensor data modeling scheme consumes less energy for data trustiness detection when compared

to other schemes. The proposed sensor data model is also used to reconstruct the data faults and losses at the sink node. The analysis of data reconstruction is given in the next subsection.

3.4.3 Analysis of Data Reconstruction

In this subsection, we analyze the performance of data reconstruction technique which considers temporal, spatial and attributes data correlations. The INTEL lab indoor environment data is used for analysis. The INTEL lab indoor environment consists of 54 MICAz nodes deployed as per Figure 3.2 to measure temperature, light, and humidity from February to April 2004. The sampling time is 31 seconds. Data loss in WSN applications is common due to poor quality link, hardware problem of the sensor node, battery depletion and noised which deeply degrade the accuracy of event detection in a critical application. According to (Kong *et al.* (2013)), the INTEL lab dataset suffers from 23 % of data loss in one month (84600 time slots) due to various reasons. These data losses should be reconstructed in order to identify the events. Root Mean Square Error (RMSE) is used as a metric to find the performance of data reconstruction algorithms.

Case 1: Element Random loss: The data loss of the nodes ranges from 5 to 50 with the increment of 5%. The number of element random data loss is represented in X-axis of the graph against the RMSE value in Y-axis. The performance of the proposed hybrid data model is better than other sensor data models as shown in Figure 3.7.

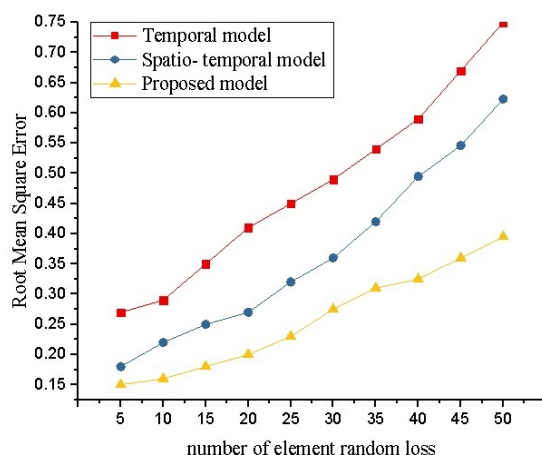


Figure 3.7: Element Random data loss

Case 2: Block Random data loss: In the second case, we are comparing the performance of the proposed hybrid model with a temporal and spatial-temporal model for block random data loss. The block random data loss of the nodes ranges from 5 to 50 with the increment of 5%. The number of nodes with block random loss is depicted in the X-axis of the graph against RMSE values in the Y axis. As shown in Figure 3.8, the performance of the hybrid proposed model is comparatively better than the temporal and spatial-temporal models.

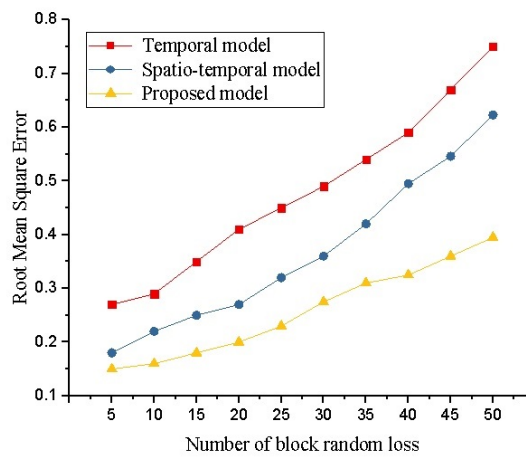


Figure 3.8: Block Random data loss

Case 3: Element Frequent loss: The functioning of the proposed hybrid data model is significantly outperforms other related data models like temporal and spatial-temporal models as shown in Figure 3.9. However the element frequent data loss increases, the RMSE value also increases gradually for all models.

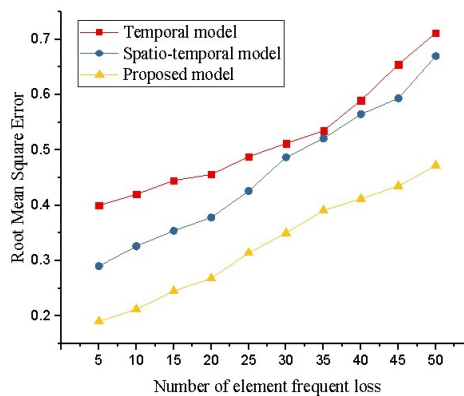


Figure 3.9: Element frequent data loss

Case 4: Successive element loss: The nodes with successive elements data loss ranges

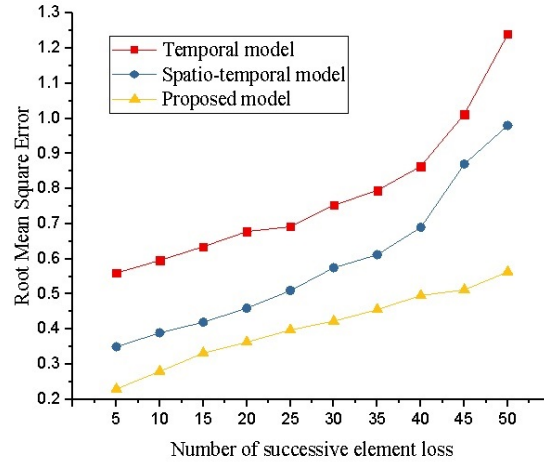


Figure 3.10: Successive element data loss

from 5 to 50 with 5 % increment is represented in the X-axis against the root mean square error in the Y-axis which depicts the accuracy of the data reconstruction algorithms. We can see from Figure 3.10 that the functioning of the proposed hybrid model is better than other models which utilize only temporal and spatiotemporal models.

After data gathering and data reconstruction process at the sink node, the data pattern is recognized and mapped semantically for detecting the events. The false positive rate of event detection analysis is given in the next sub section.

3.4.4 False Positive Rate for Event Detection

The proposed hybrid sensor model is also used to identify the events. Since the data outliers can be either faulty data or an event. Detection of events in WSN consists of data collection from sensor nodes which are free from data fault, data loss to identify the events reliably. The important requirement for the detection event in WSN is the low false positive rate. Here the False Positive Rate (FPR) is used as a metric to find the performance of the various event detection schemes. Low FPR gives good performance and high event detection rate. From Figure 3.11, we can say that the proposed hybrid data model has less FPR when compared to other related models for event detection since it includes temporal, spatial, attribute data correlations and data provenance techniques for the detection of events with trustworthy data items.

The FPR value for the proposed model is 2.2 %, and True Positive Rate (TPR) is 100%. The FPR value for spatiotemporal and Attribute model (STA)(Karthik and Ananthanarayana (2017a)) is 5%. The FPR value for Temporal and Attribute correlation model

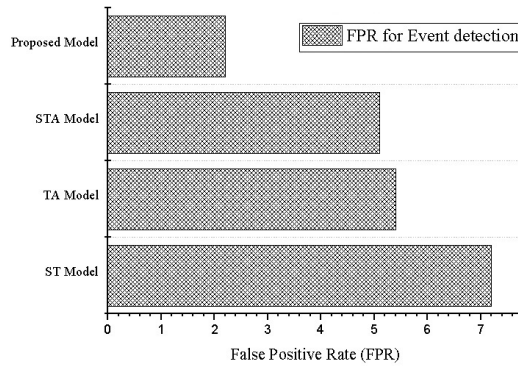


Figure 3.11: False Positive Rate for Event detection

(TA) (Illiano and Lupu (2015b)) is 5.4% and the FPR value for Spatio-temporal model (ST) (Yu *et al.* (2015)) is 7.2 %.

3.5 Summary

In this chapter, we proposed a sensor data model for evaluating the trustworthiness of data and event detection in WSN. Then the proposed sensor data model is tested with real-world sensor dataset. The result shows that the proposed sensor data model outperforms the existing data models in terms of detecting the data trustiness in an energy efficient way and detecting the events reliably by reconstructing the data faults.

Chapter 4

Trust-based Data Gathering in Wireless Sensor Network

4.1 Preamble

In this chapter, we address the second research objective, finding the trustworthiness of sensor node and data for data gathering. In all pervasive applications, a gathering of sensor data from the environment is the main operation held in a sensor network, where sink node or base station gathers all generated data to do data analysis and decision making. We strongly conceive that each process from perceiving the environment to decision making, demands trust based process to ease and ensure the trustworthy data exchange among trustworthy nodes. In this chapter, we propose a Trust-based Data Gathering (TDG) which focus on trust-based data collection, data aggregation, and data reconstruction to show that the absence of trust in a sensor-driven pervasive environment could affect the normal functionality of an application.

The primary contributions of this chapter are follows:

- 1) Trust-based Data Gathering is proposed to ensure trustworthy data sharing from the source node to the sink node in sensor networks. Furthermore, trust-based reconstruction is proposed to improve the reliability of the application.
- 2) Trust-based Data Collection and Trust-based Data Aggregation methods are proposed to protect the applications from node and data related attacks, selfish behavior, faulty and missing values.
- 3) Analysis of the effect of the malicious nodes, faulty data on data collection, and data collection with a trust mechanism, data aggregation and data aggregation with trust mechanisms are given.

The remainder of this chapter is prepared as follows: TDG is proposed in section 4.2. Section 4.3 gives the experimental setup and simulation environment details. Section 4.4 explains the performance of proposed work and comparison with state of the art techniques as results and discussions. A Summary is given in section 4.4.

4.2 Trust-based Data Gathering

In this section, a trust model is proposed for trust-based data gathering as shown in Figure 4.1.

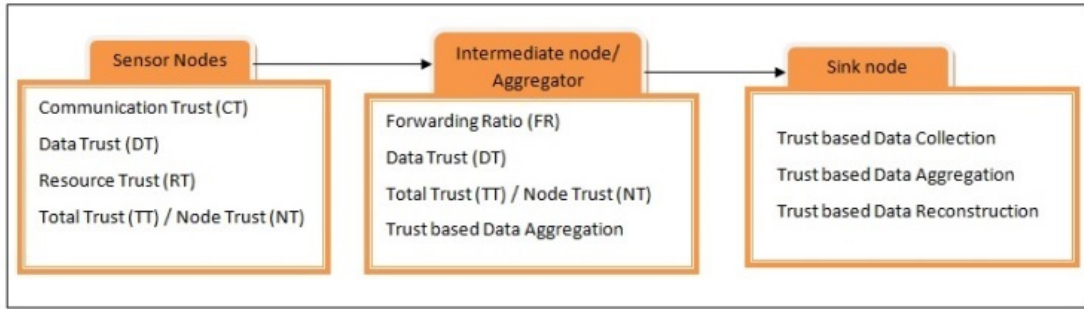


Figure 4.1: Proposed Trust Model

The proposed trust model uses Communication Trust (CT), Data Trust (DT), Resource Trust (RT), Forwarding Ratio (FR) for identifying malicious nodes, selfish nodes, faulty data, data loss, and uses Node Trust (NT)/Total Trust (TT) and data density based correlation for reconstructing the faulty data and data loss. In the proposed trust model, at sensor node, we calculate CT , DT , RT , and NT , at intermediate/aggregator node, we calculate FR , DT , NT and trust-based data aggregation. At sink node, we have trust-based data collection, data aggregation, and data reconstruction. To evaluate the trust value of the sensor node and data for data gathering, firstly, we use a trust model at the sensor node level. Secondly, we use trust-based data aggregation at the intermediate node and trust-based data collection, aggregation and data reconstruction at the sink node level. Trust in a sensor network is defined as a belief level or confidence level of a node that can have it on another node or sensor data. Trust value of sensor node reflects the node functionality, and the trust value of sensor data reflects the quality of data. In this proposed trust model, trust value ranges from -1 to +1 as in (Karthik and Dhulipala (2011)). -1 to -0.3 denotes the untrustworthy state or faulty state. -0.29 to +0.29 denotes uncertain state. +0.3 to +1 denotes trustworthy state.

4.2.1 Proposed Trust model for Data Gathering

In this trust model, trust evaluation consists of three parts: communication trust, data trust, and resource trust. The sensor node is either used for communicating its data or neighbor data to the next hop and sink node for the data analysis process. Therefore it is important to look at the communicating behavior of a sensor node before judging a sensor node like a normal or malicious one. So the Communication Trust (CT) of a sensor node is calculated using equation (4.1). Apart from communication, the sensor node can sense the environment and generate data about happenings in the environment.

Therefore evaluating the trustiness of sensor data is important in sensor-driven pervasive application. Data Trust (DT) of a sensor node is evaluated using equation (4.2). In equation (4.2), x denotes the data item of the source sensor node, and y denotes the data item of a neighbor node which monitors the same event. Evaluating the CT and DT of a sensor node does not reflect the total trust level of the node [34]. So we add Resource Trust (RT) to increase the reliability of trust evaluation. Equation (4.3) is used to calculate the RT of a sensor node. Total Trust (TT) of a sensor node is given in equation (4.4). TT is also called as Node Trust (NT). In some case, a sensor node can also act as a relay node. It forwards the sensor data to the next hop node or to sink node. Forwarding Ratio (FR) is used to calculate the selfish behavior of a sensor node which is given in equation (4.5).

$$CT = (S - U)/(S + U) \quad (4.1)$$

$$DT = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 (y_i - \bar{y})^2}} \quad (4.2)$$

$$RT = \begin{cases} -1 & \text{if } R_{resi} > Rth, \text{ untrustworthy} \\ 0 & \text{if } R_{resi} = Rth, \text{ uncertain} \\ +1 & \text{if } R_{resi} < Rth, \text{ trustworthy} \end{cases} \quad (4.3)$$

$$TT = w1CT + w2DT + w3RT \quad (4.4)$$

where $w1+w2+w3=1$

$$FR = \frac{\text{number of packets supposed to be relayed}}{\text{Total number of packets relayed}} \quad (4.5)$$

We borrowed CT evaluation model from (Shaikh *et al.* (2009)), (Dhulipala *et al.* (2013)). They used Successful (S) and Unsuccessful (U) transactions to calculate CT . We used a data correlation method to calculate the DT . For calculating DT , state-of-the-art techniques use all neighbor nodes data without considering data correlation degree (Gao *et al.* (2018)) and node trust. Due to the nature of WSN and harsh environment, the neighbor node data may be uncorrelated. So comparing source node data with uncor-

related neighbor node data leads to uncertain or untrustworthy state. To increase the reliability of data trust calculation, we choose neighbor node based on data correlation degree and its node trust value. Available resources play an important role in node collaboration and data forwarding. The resources reflect whether the node can perform its operations perfectly or not. Before collaborating or forwarding the data, nodes in WSN would check the resource availability of the target node. Resources like energy, bandwidth, and waiting queue, buffer size are an important metric for collaboration and data relay. Resource Trust (RT) is an important metric in WSN to decide about a node like normal or abnormal one. Its trust value is calculated using equation (4.3). R_{res} denotes the residual resources of the sensor node and R_{th} denotes the threshold value of node resources. For a normal node, the resource consumption will not vary much during monitoring, data generation, processing, and data forwarding. However, for the malicious node, its resource consumption varies frequently for executing its malicious attacks. The suitable weight values of CT , RT , and DT depends on the condition of the network. When there are less number of communication and data generation, RT gets more weightage than CT and DT . When there are enough communication and data generation from sensor node, the weight values of CT , RT and DT are equal. According to (Karthik and Ananthanarayana (2017b)), with smaller sample size and less number of interactions, the values of DT and CT are extremely noisy. When the number of data items and interactions are greater than or equal to five, then the value stabilizes. After five samples and interactions, the weight values of CT , DT and RT are equal. FR is used to find the trustiness of the intermediate node which does the data relaying operation.

4.2.2 Trust-based data collection and data aggregation

In this subsection, we propose the methods for trust-based data collection and data aggregation. The process of trust-based data collection is given as a flowchart in Figure 4.2. The process of trust-based data aggregation is given as a flowchart in Figure 4.3. In trust-based data collection (also called as data collection with trust mechanism), the NT is used to find the malicious node before collecting the data from the sensor node as shown in Figure 4.2. After collecting sensor data from nodes, their DT is calculated to check their data quality. If any untrustworthy data or data loss is found, then it is reconstructed at sink node using trust-based data reconstruction process. In the trust-

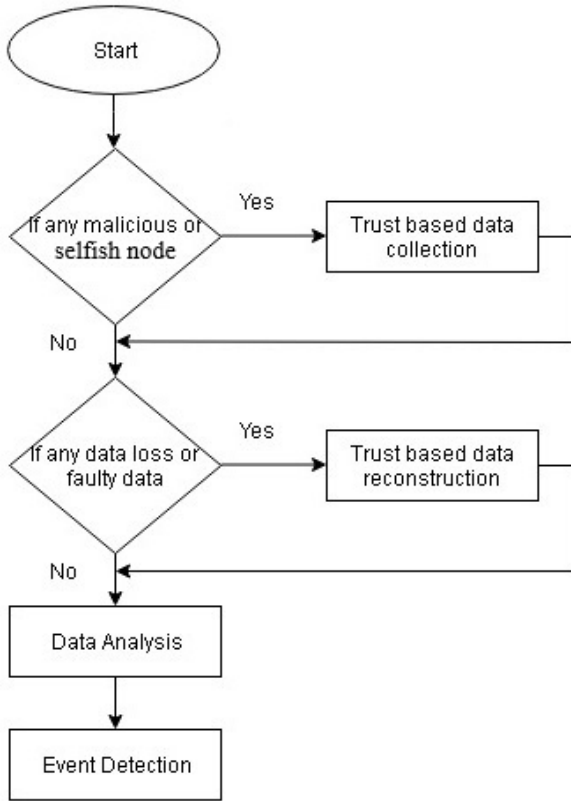


Figure 4.2: Data Collection with Trust mechanism

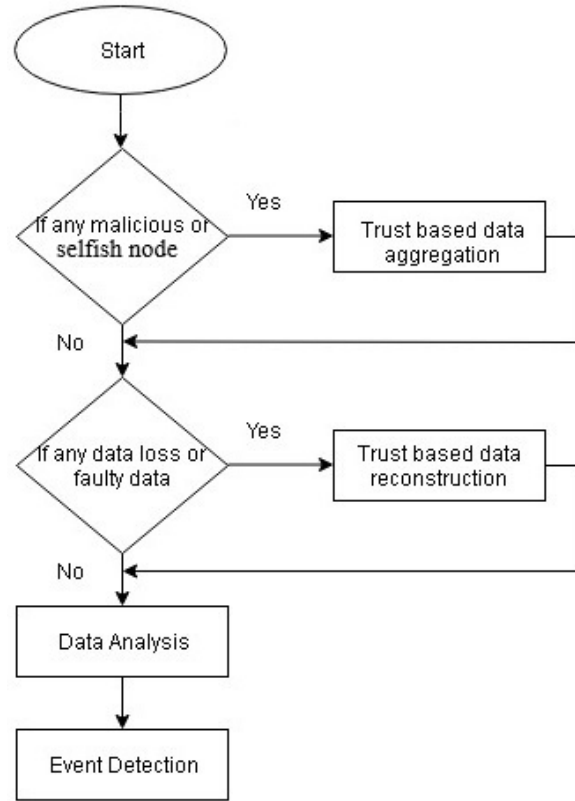


Figure 4.3: Data Aggregation with Trust mechanism

based reconstruction process, the node trust value and data density correlation degree (Gao *et al.* (2018)) are used to choose the trustworthy neighbor for the source node to reconstruct the faulty data or missing data. A sensor node is called trustworthy and data density correlated neighbour, if its data are close to the data of a certain number of its trustworthy neighbor nodes. After the reconstruction of the data item, the data analysis is performed at the sink node for event detection and decision making. In trust-based data aggregation process (also called as data aggregation with trust mechanism), the Node trust is used to find the trustiness of the node before aggregating the sensor data items as shown in Figure 4.3. The trustworthy aggregator is chosen based on NT for performing the data aggregation operation. After performing collecting data from sensor nodes, the aggregation operation is performed at the trustworthy aggregator, and aggregated data is forwarded to the sink node. After receiving the sensor aggregated data items through trustworthy intermediate nodes, sink node evaluates DT for all received data items. If any data fault or data loss is found then trust based reconstruction is used to reconstruct the data item. After data reconstruction at the sink node, data analysis is performed

to detect the event and decision making. The following Algorithms 4.1, 4.2 and 4.3 are used for trust-based collection, trust-based data aggregation and trust-based data reconstruction to have trustworthy data analysis and event detection in INTEL Berkeley lab dataset.

4.2.3 Algorithm 4.1: Trust-based Data Collection

Input: sensor node IDs, data items, resource status of the node, communication history, and sink node ID.

Output: Trust-based data collection

- 1: Collect data from nodes 31, 32, 33 & 34 of INTEL lab
- 2: Evaluate NT, DT and RT using equations (4.1), (4.2), & (4.3).
- 3: If all nodes are trustworthy then proceed to step 8.
- 4: Else ignore untrustworthy node, choose trusty neighbor and go to step 2.
- 5: End if
- 6: Evaluate FR of intermediate nodes using equation (4.5).
- 7: Forward the data items to sink node via node 1 and node 3.
- 8: Evaluate DT of received data item.
- 9: If data = untrustworthy then go to Algorithm 4.3.
 Else status= trustworthy data
- 10: End if
- 11: Data analysis and Decision making.

4.2.4 Algorithm 4.2: Trust-based Data Aggregation

Input: sensor node IDs, data items, resource status of the node, communication history, and sink node ID.

Output: Trust-based data aggregation

- 1: Aggregate data from nodes 31, 32, 33 & 34 of INTEL lab.
- 2: Choose the trustworthy aggregator which is common to all nodes.
- 3: Evaluate NT, DT and RT for all nodes using equations (4.1), (4.2), & (4.3).
- 4: If all nodes are trustworthy then forward the data items to aggregator
 Else choose the trusty neighbor and forwards its data item to the aggregator.
- 5: End if

- 6: Perform aggregation operation.
- 7: Choose a trustworthy intermediate node and forward the aggregated data to sink node
- 8: Check DT of received data items at the sink node.
- 9: If data is trustworthy then go to step 12
- 10: Else go to Algorithm 3.
- 11: End if
- 12: Data analysis and Decision making.

4.2.5 Algorithm 4.3: Trust-based Data Reconstruction

Input: sensor node IDs, data items, intermediate nodes, data losses

Output: reconstructed data for data faults and data losses

- 1: If data = faulty or data loss, then find its source node and go to step 4.
- 2: Else data item = trustworthy, exit.
- 3: End if
- 4: Find a list of neighbors for the source node
- 5: Find trustworthy and data density correlated neighbor as like ([Gao et al. \(2018\)](#)).
- 6: If trustworthy data density correlated neighbor is found, then go to step 10.
- 7: Else data item = uncertain
- 8: End if
- 9: Use data density correlated neighbor data to reconstruct faulty or data loss
- 10: Substitute reconstructed data for data analysis and decision making

4.3 Experimental setup and Simulation Environment

In this section, we present our experimental setup and simulation environment parameters to analysis the performance of TDG. We have used NS2 ([Issariyakul and Hossain \(2009\)](#)) and MATLAB simulations to implement the proposed TDG. We deployed 54 static sensor nodes randomly in the area of $100 * 100m^2$ which is similar to INTEL Berkeley research lab setup ([Madden et al. \(2004\)](#)). This sensor network consists of sink node, aggregators and intermediate nodes for data collection, data aggregation, and data forwarding respectively as shown in Figure 4.4. Apart from data collection process, the sink node is also used for data analysis and event detection. Sensor nodes are simulated in such a way that each node produces temperature, humidity and light

values together for every sampling period. The sampling period for this experiment is 31 seconds. We conducted the experiments for 100 minutes. INTEL lab dataset is used

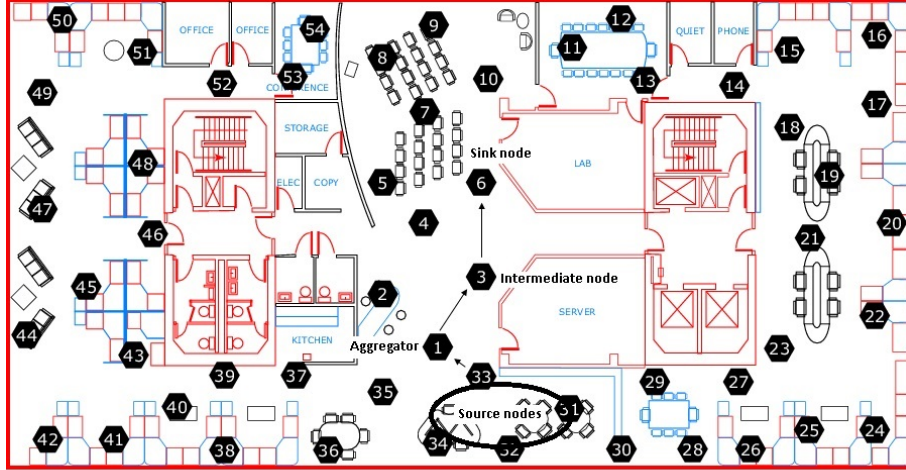


Figure 4.4: Sensor deployment in INTEL Lab Berkeley

to analyze the performance of the data reconstruction process and data fault detection schemes. Using NS2 simulations, CT and RT are evaluated as Trust metrics using equations (4.1) and (4.3) respectively. MATLAB simulation and INTEL lab dataset are used to evaluate the DT trust metric using equation (4.2). Malicious attacks are simulated in NS2 and MATLAB to find the performance of attack resiliency of TDG. Trust mechanism is included as CT , RT and attacks resiliency in NS2 simulations. Trust mechanism is included as DT in terms of data fault detection schemes and data reconstruction schemes like (Gilbert *et al.* (2018)) in MATLAB simulations. The proposed TDG is the compounding of CT , DT and RT metrics as given in equation (4.4). The weight values of CT , RT , and DT are set manually in simulation which depends on the condition of the network environment. The change in weight values would effect the trust value of node and data. Further, it might cause the application to misbehave.

4.4 Results and Discussions

This section aims to determine the performance of the proposed TDG scheme in terms of the energy consumption analysis, network delay analysis, data reconstruction analysis, data fault detection analysis and malicious node detection analysis.

4.4.1 Energy consumption analysis

In this subsection, the energy consumption analysis is carried out for the data collection process, data collection with a trust mechanism process, data aggregation, data aggregation with a trust mechanism process. For data transmission, data processing and sensing operations in the pervasive application, a certain amount of energy is needed by the sensor node. The unexpended energy of a wireless sensor node plays a vital role in the evaluation of resource trust. If the battery is exhausted, then the sensor node is not able to perform its basic operations like data transmission, processing, and sensing. Table 4.1 shows the amount of energy spent by the MICAz node which operates at 7.3 MHz to perform its basic operations.

Table 4.1: Amount of energy spent by MICAz node

| S.no | Operations | Energy consumption in mW |
|------|------------|--------------------------|
| 1. | Listen | 68 mW |
| 2. | Receive | 72 mW |
| 3. | Compute | 26 mW |
| 4. | Transmit | 65 mW |

Here we consider four nodes 31, 32, 33 and node 34 as source node which collects temperature data. The collected data should be forwarded to the sink node for data analysis and decision making. In the data collection process, all temperature data are forwarded to the sink node via intermediate node 1 and node 3. In case of any faulty data or data loss, the sink node would request the source node's neighbor to retransmit the data at a particular time slot. In data collection with a trust mechanism, before collecting data from the source node, their node trust evaluation is done. After collecting the data, their data trust is evaluated. All data are forwarded to the sink node via trustworthy intermediate nodes. If any data loss or faulty data is found in the collected data, it must be replaced by trust-based data reconstruction process. Figure 4.5 shows the energy consumption analysis of the Data Collection (DC) process and Data Collection process with Trust mechanism (DCT). We varied the number of faulty data to see its consequences in the DC and DCT processes. When there are no faulty data or data loss, the DC process works better than DCT. DCT consumes more energy than its counterpart when there is no faulty data or data loss since it evaluates NT and DT while collecting data. When we have more than 30% of faulty data, DCT works better than

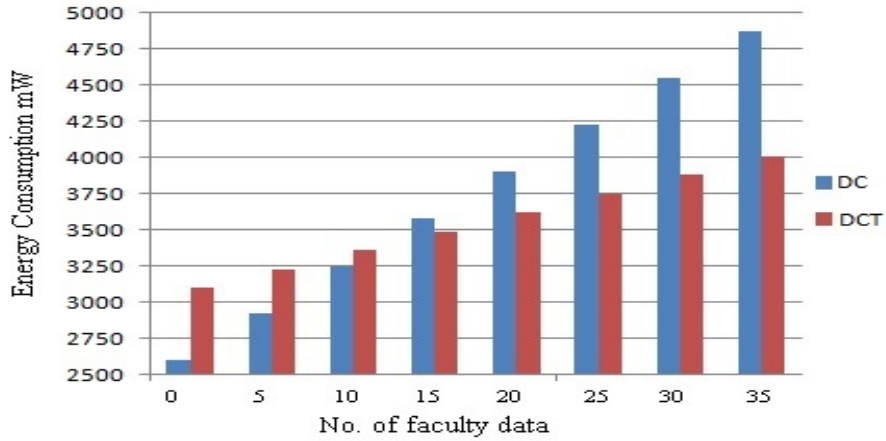


Figure 4.5: Energy consumption analysis of DC and DCT for faulty data

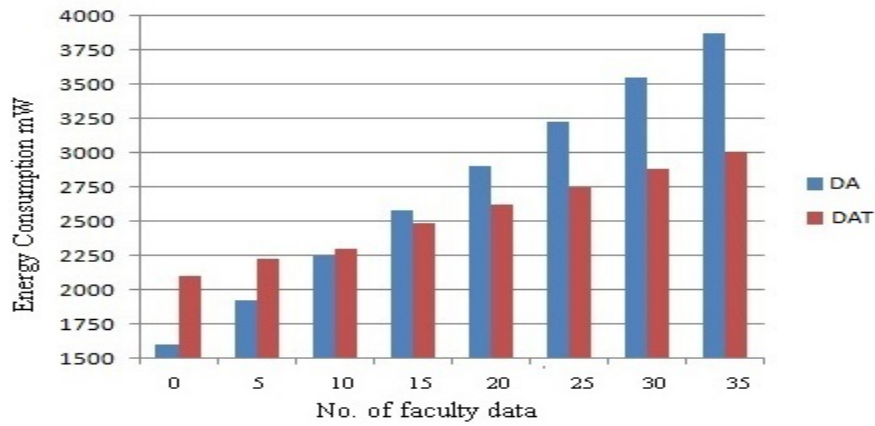


Figure 4.6: Energy consumption analysis of DA and DAT for faulty data

the DC process. In case of data faults or data loss, the sink node of the DC process would raise the request to the source node for retransmission of data at a particular time slot. Due to the retransmission of data from the source node to the sink node, the DC process consumes more energy than the DCT process. The DCT process reconstructs the faulty data or data loss without seeking retransmission which eventually reduces the energy consumption. The result shows that DCT works better than traditional DC process when there are more than 30% of data faults and data losses in collected data.

Figure 4.6 shows the energy consumption analysis of Data Aggregation (DA) process and Data Aggregation process with Trust mechanism (DAT). We varied the number of faulty data to see its consequences in the DA process. When there are no faulty data or data loss, DA process works better than DAT. DAT consumes more energy than its counterpart when there is no faulty data or data loss since it evaluates NT and DT while aggregating data. In case of data faults or data loss, the sink node would raise the re-

quest to the source node for retransmission of data at a particular time slot. Due to the retransmission of data from the source node to the sink node, the DA process consumes more energy than the DAT process. The DAT process reconstructs the faulty data or data loss without seeking retransmission which eventually reduces energy consumption. The result shows that the DAT process works better than the DA process when there are more than 30% of data faults and data losses in aggregated data.

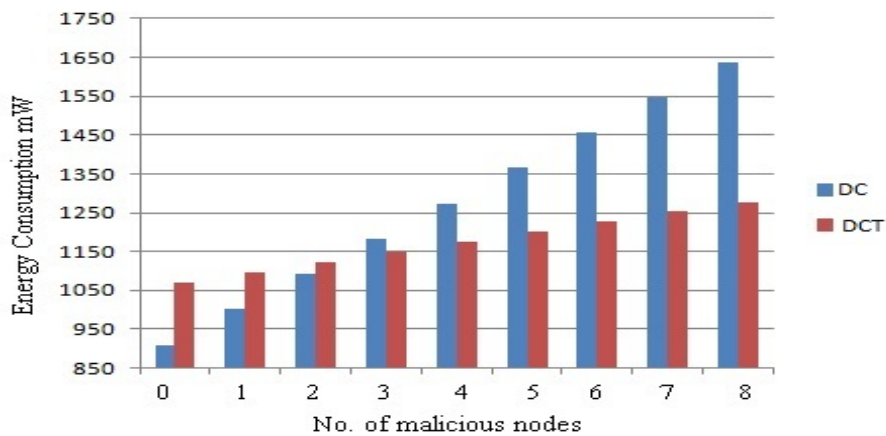


Figure 4.7: Energy consumption analysis of DC and DCT for malicious nodes

The energy consumption analysis between DC process and DCT process concerning malicious nodes is presented in Figure 4.7. We varied the number of malicious nodes from 0 to 8 with an increment of 11% in every cycle to observe its effect on energy consumption. When data is collected from normal nodes, DC process consumes less energy than DCT since it does not evaluate the NT and DT before collecting data. When there are malicious nodes in the network, the DC process consumes more energy than its counterpart, since the sink node has to find the trustworthy neighbor node to retrieve the data. When the number of malicious node increases more than 30%, the DCT process consumes less energy than the DC process as shown in Figure 4.7. The energy consumption analysis between the DA process and DAT process concerning malicious nodes is presented in Figure 4.8. We varied the number of malicious nodes from 0 to 8 with an increment of 11% in every cycle to observe its effect on energy consumption. When data is aggregated from normal nodes, traditional DA process without a trust mechanism consumes less energy than DAT since it does not evaluate the source node and neighbors' trustiness before aggregating data. When there are more than 30% of malicious nodes in the network, DA process consumes more energy than its counterpart, since the sink node has to request the trustworthy aggregator to retrieve

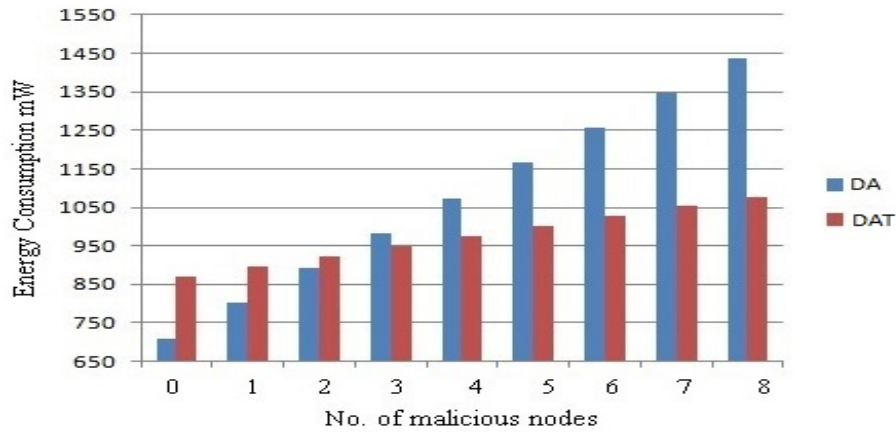


Figure 4.8: Energy consumption analysis of DA and DAT for malicious nodes

the data in traditional DA process. When the number of malicious node increases, the energy consumption for the DA process increases than the DAT process.

4.4.2 Network delay analysis

Network delay is defined as the time taken for the data item to reach the destination from the source. Network delay includes data transmission delay, data propagation delay, data processing delay and data queuing delay. Sensor nodes are used in delay sensitive and time-critical pervasive applications. In these applications, it is crucial to measure and maintain the delay for real-time control. It is also important to find the unnatural delay caused by faulty data and malicious nodes. When the source and destination nodes in the pervasive application have synched clocks, finding the network delay from source to the destination node is straight forward. When sending a data item, the source node fixes a time stamp, and while receiving the data item, the receiver fixes a time stamp. The difference between these two-time stamps of the data item is called network delay.

In the DC process, node 31, node 32, node 33 and node 34 are source nodes and node 6 is the sink node. Whenever the source node generates a data item, it fixes time stamp and forwards to next hop. Upon receiving the data item, the sink node fixes its time stamp. The sink node evaluates the network delay by finding the difference between two-time stamps. Figure 4.9 shows the network delay analysis between the DC process and DCT process. We varied the number of faulty data to see its consequences on network delay. When there are no faulty data, DCT takes extra time than the DC process to evaluate the trustiness of data. When the number of faulty data increases,

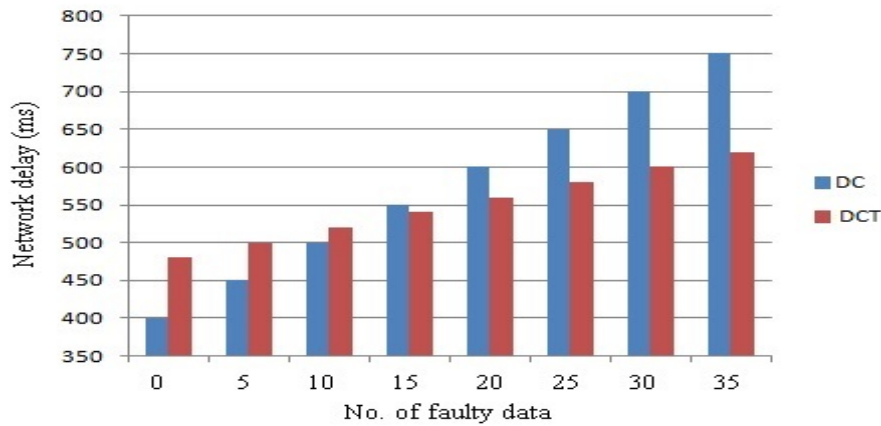


Figure 4.9: Network delay analysis of DC and DCT for faulty data

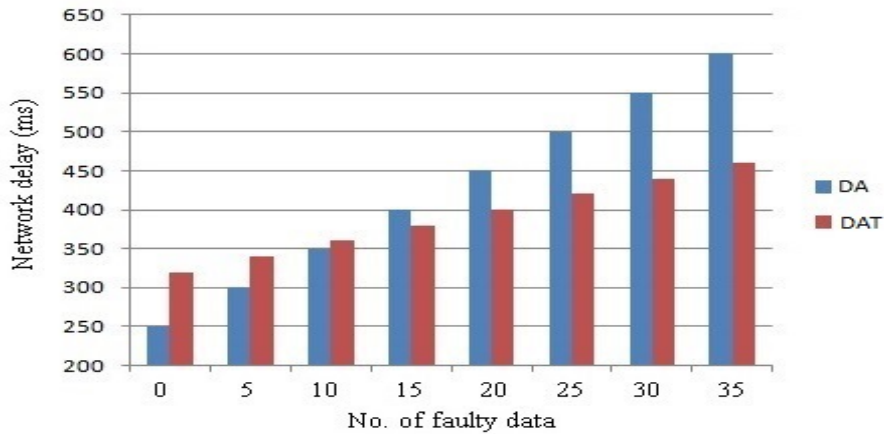


Figure 4.10: Network delay analysis of DA and DAT for faulty data

the DC process takes more time to collect the sensor data items since the sink node has to request source or its neighbors for retransmission. The result shows that the DCT process works better than the DC process when there are more than 30% of data faults and data losses in collected data.

In the DA process, node 31, node 32, node 33 and node 34 are considered as source nodes. Node 1 is considered as aggregator node. Node 3 is considered as an intermediate node for data forwarding. Node 6 is the sink node. Whenever the source node generates data item, it is forwarded to aggregator node 1 for data aggregation. After data aggregation, it is relayed to sink node via intermediate node 3. Figure 4.10 shows the network delay analysis between the DA process and DAT process. We varied the number of faulty data to see its consequences on network delay. When there are no faulty data, DAT takes extra time than the DA process to evaluate the trustiness of data. When the number of faulty data increases, the DA process takes more time to collect the sensor data items since the sink node has to request source or its neighbors for re-

transmission. The result shows that DAT works better than the DA process when there are more than 30% of data faults and data losses in aggregated data.

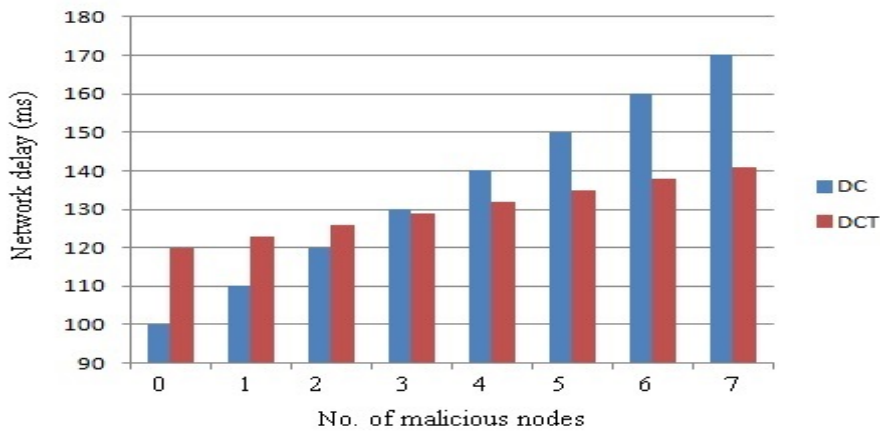


Figure 4.11: Network delay analysis of DC and DCT for malicious nodes

The network delay analysis between the DC process and DCT concerning the malicious node is depicted in Figure 4.11. We varied the number of malicious nodes to see its consequence in network delay. When data is collected from normal nodes, DC process takes less time than DCT since it evaluates the NT and DT before collecting data. When the number of malicious nodes increases, DC process takes more time to collect data from the node, since it involves retransmission process after finding node as a malicious one, whereas in DCT uses trust based reconstruction process for data recovery. The result shows that DCT works well than DC process when there are 30% malicious nodes in the network. The network delay analysis between the DA process and DAT

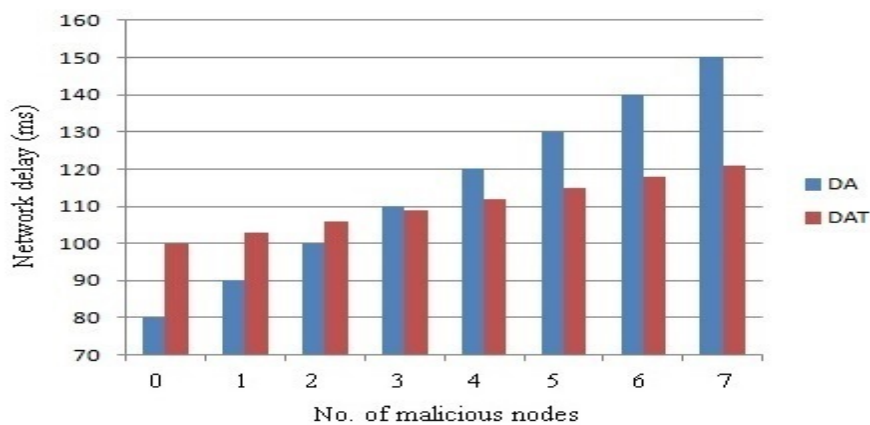


Figure 4.12: Network delay analysis of DA and DAT for malicious nodes

concerning the malicious node is depicted in Figure 4.12. We varied the number of malicious nodes to see its consequence in network delay. When data is aggregated from

normal nodes, DA process takes less time than DAT since it evaluates the NT and DT before aggregating data. When the number of malicious nodes increases, DA process takes more time to collect data from the node, since it involves retransmission process after finding node as a malicious one, whereas in DAT uses trust based reconstruction process for data recovery. The result shows that DAT works better than the DA process when there are more than 30% of malicious nodes in the network.

4.4.3 Data reconstruction analysis

In this subsection, we examine the performance of trust based data reconstruction technique which looks at data density correlation degree (Gao *et al.* (2018)) and NT for reconstructing the faulty data and data loss. INTEL lab dataset is used to find out the performance of the proposed data reconstruction method. According to (Dhulipala *et al.* (2013)), (Sharma *et al.* (2010)) Intel lab data set contains 23% of data loss due to several causes. Root Mean Square Error (RMSE) is used as a system of measurement to find the performance of various data reconstruction techniques. We had chosen 10% of element random loss, 10% of block random loss, 20% of element frequent loss and 20% of successive element data loss for data reconstruction experiments. Additionally, we inserted 10% synthetic data faults manually for experiments. We used a multi-attribute correlation based reconstruction method (Karthik and Ananthanarayana (2017a)) for comparison. We used two methods of multi-attribute correlation based reconstruction for data recovery. We used three neighbor nodes data for data reconstruction as a first method. In the second method, we used only two neighbor nodes data for data reconstruction. We varied the number of faulty and data losses from 5 to 35 with an increment of 5.

Data reconstruction based on 3 and 2 neighbor nodes data performs almost the same for few data faults and data losses. When we increase the number of data faults and data losses, the performance of all three data reconstruction methods gradually decreases. However, the proposed data reconstruction method outperforms the other two methods as shown in Figure 4.13, since it chooses the neighbor node based on data density correlation degree and its NT for data reconstruction. The data generated from the sensor node may be different from closely located neighbor node data due to various attacks, resource restriction, harsh and unfriendly environments. The result shows that the proposed data reconstruction method based on data density correlation and node

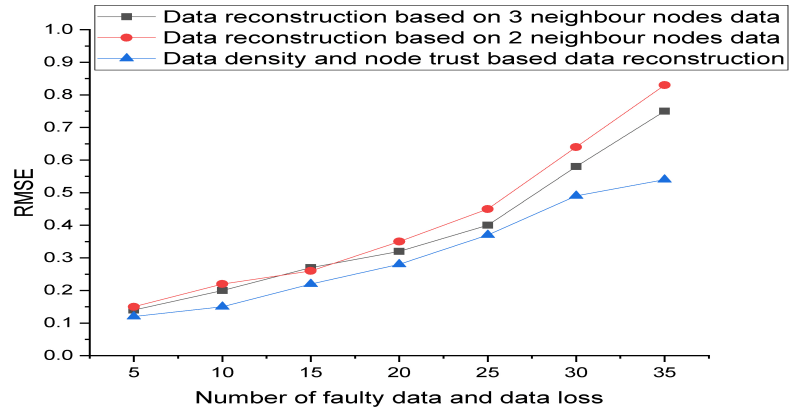


Figure 4.13: Data reconstruction analysis with data density and neighbor node correlation based methods

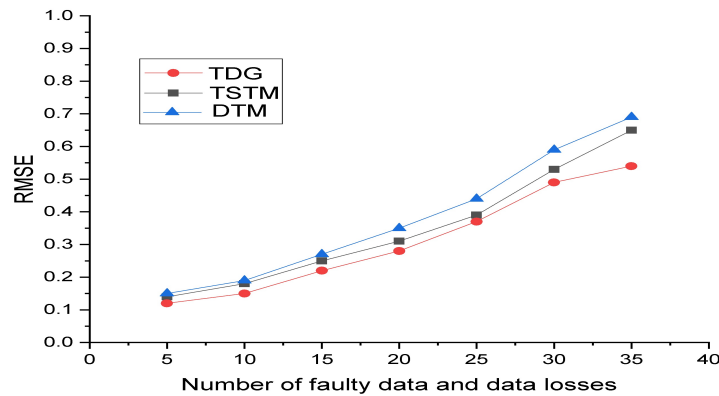


Figure 4.14: Comparison of various data reconstruction schemes

trust works well than multi attribute-based data reconstruction methods.

Figure 4.14 shows the performance of three data reconstruction schemes. We varied the number of faulty data and data losses from 5 to 35 with an increment of 5% in every cycle of the experiment. We compared TDG with TSTM (Gilbert *et al.* (2018)) and DTM based data reconstruction scheme (Karthik and Ananthanarayana (2017a)). Results show that the proposed method TDG achieves 2% to 5% of better performance than TSTM and DTM since it uses data density correlation based DT and NT for data reconstruction, whereas TSTM used time series analysis and compressed sensing method DT for data reconstruction. DTM uses multi-attribute based correlation for data reconstruction (Karthik and Ananthanarayana (2017a)).

4.4.4 Detection of data faults

This subsection aims to evaluate the performance of proposed TDG in finding the data faults. We used INTEL lab Berkeley dataset in this evaluation. In this work, we used temperature data gathered by node 33 from 23rd February 2004 to 2nd March 2004 for finding the efficiency of various data fault schemes. We have chosen node 33 because it contains data faults as stated in (Sharma *et al.* (2010)). The sampling time for node 33 to generate temperature data was 31 seconds. The dataset of INTEL lab Berkeley does not provide any fault annotation for temperature data gathered by node 33. We inspected

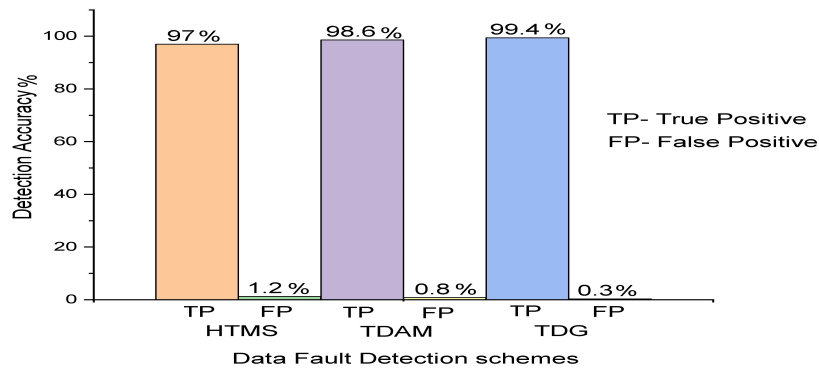


Figure 4.15: Comparison of data fault detection schemes

manually to discover the data faults in the temperature data of node 33, and we followed the steps given in (Sharma *et al.* (2010)) for discovering data faults and cross-checked to ensure the manual fault annotations. This type of fault annotation construction is similar and consistent as stated in (Nguyen *et al.* (2013)) for sensor data without ground truth. Our observation on node 33 temperature data shows that there are some faulty data items. For evaluating the performance of various data fault detection schemes, we used detection accuracy as a performance metric. It is defined as the ratio of a total number of true positives to the total number of actual data faults. There were 123 faulty data in node 33 of INTEL lab Berkeley according to our manual inspection. The proposed method TDG achieves 99.4% detection accuracy in finding data faults correctly as shown in Figure 4.15. It applies NT, RT, FR and data density correlation based methods for detecting the data faults. The TDAM (Gao *et al.* (2018)) achieves 98.6% detection accuracy with a false positive rate of 0.8% because it considers only data density correlation based DT without NT. The HTMS (Karthik and Ananthanarayana (2017b)) achieves 97% in rightly classifying data as faulty with false positive rate of

1.2%. HTMS is a combination of NT, DT and provenance-based trust but without data density correlation based trust. The proposed method TDG achieves better detection accuracy and outmatches recent data fault schemes like HTMS and TDAM.

4.4.5 Detection Accuracy of malicious nodes

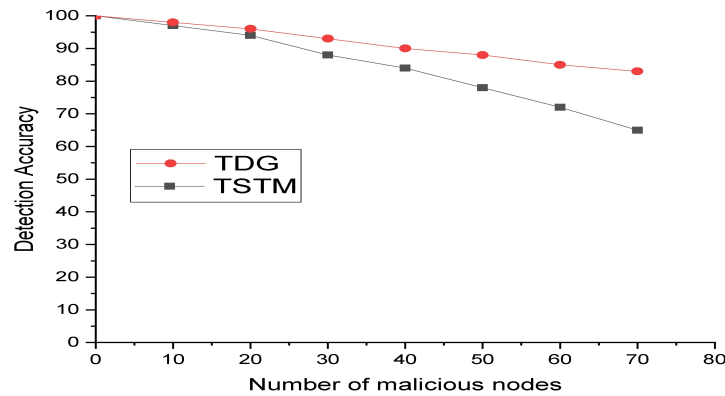


Figure 4.16: Comparison of detection accuracy of the malicious nodes

In this subsection, the proposed TDG is compared with existing TSTM (Gilbert *et al.* (2018)) for finding efficiency in detection of malicious nodes. The reason for preferring comparison model TSTM is the recent attack resistant trust-based model for data aggregation and data reconstruction. We simulated Sybil attack, bad mouthing attack, attacks on data, DOS attack and varied the number of malicious nodes from 0 to 80 with an increment of 10 for every cycle. Figure 4.16 depicts the detection accuracy of malicious nodes by TDG and TSTM. The proposed method TDG achieves 1% to 15% of better detection rate than TSTM because it considers NT, RT, FR and data density correlation based methods for detecting malicious nodes. TSTM considers energy trust, relative trust, data trust without data density correlation.

4.5 Summary

In this chapter, we proposed TDG in wireless sensor networks for trust based data collection, data aggregation, and data reconstruction. We used Intel lab dataset to show that the process from sensing to decision making demands trust-based process to ensure the trustworthy data exchange, data analysis, and decisiveness. We showed that DCT and DAT consume less energy and less network delay when we have more than 30% of data faults, data losses and malicious nodes in the network than traditional DC and DA process.

Chapter 5

Hybrid Trust Management Scheme for Wireless Sensor Networks

5.1 Preamble

In this chapter, we address the third research objective, finding the trustworthiness of sensor node and data in monitoring single event. The data collected from the WSN is used for making decisions. The condition for making critical decision is to assure the trustworthiness of the data generated from sensor nodes. However, the approaches for scoring the sensed data alone is not enough in WSN since there is an interdependency between node and data item. If the overall trust score of the network is based on one trust component, then the network might be misguided. In this chapter, we propose the hybrid approach to address the issue by assigning the trust score to data items and sensor nodes based on data quality and communication trust respectively. The proposed Hybrid Trust Management Scheme (HTMS) detects the untrustworthy data with the help of temporal and spatial correlations. The correlation metric and provenance data are used to score the sensed data. The data trust score is utilized for making a decision. The communication trust provenance data are used to compute the trust score of intermediate nodes and the source node. If the data item is reliable enough to make critical decisions, a reward is given by means of adding a trust score to the intermediate nodes and the source node. Punishment is given by reducing the trust score of the source and intermediate nodes if the data item is not reliable enough to make critical decisions.

The contributions of this chapter are follows:

1. Hybrid TMS for WSN is proposed, where a distributed approach is used at the node level and, the centralized approach is used at the base station for the malicious node, selfish node and untrustworthy data detection.
2. Punish, and reward mechanisms are introduced in TMS by considering the interdependency property and data trust score.
3. Attack resistant TMS is proposed by considering various attack models.

The rest of this chapter is organized as follows: Network assumptions, attack models and various types of trust and their definitions are elucidated in Section 5.2. Section 5.3

reports a sample WSN scenario for the proposed HTMS and algorithms for trust computation. The simulated and statistical results are represented in section 5.4. Finally, summary is given in section 5.5.

5.2 Network Assumptions, Attack Model and Various Types of Trust

In this section, the assumptions, attack model and various types of trust are described here.

5.2.1 Network Assumptions

WSN comprises of small static sensor nodes distributed in the terrain to monitor the environmental parameters. We presume that the sensor nodes have unique identification, information about their location and dimension property. All the sensor nodes in WSN are alike in processing capability, radio range, and battery power. The dimension property of the sensor node depicts the type of data measured by the node with its anticipated range.

5.2.2 Attack model

The trust model is utilized to discover malicious node and trustworthiness of data item. They are designed to work better with the protection of the application. However, during the trust evaluation process, not only node, data item but also the trust model may be attacked by adversaries. The probable attacks ([Han et al. \(2014\)](#)) against the trust management schemes, node and data item and their protection resiliency are elucidated here.

5.2.2.1 DoS Attack

The malicious node forwards much information to waste huge amount of resources in the environment. This attack can be handled by keep tracking the residual energy of node and comparing with others' energy in the network .

5.2.2.2 Bad Mouthing attack

The malicious node spread the wrong recommendation about neighbors in the network. This attack can be addressed by getting multiple recommendations from the nodes or having a direct transaction with target node instead of going for recommendation trust.

5.2.2.3 On-off Attack

The malicious node behaves well for some time and suddenly start to act abnormal in the network. This can be addressed by using trust decay factor where the trust score made long ago carries less weight than late trust scores. The use of adynamic sliding window also useful in detecting and overcoming this attack.

5.2.2.4 Conflicting behavior attack

The malicious node behaves differently to different nodes in the network. This makes other nodes to give conflicting recommendations about the single node. This attack can be addressed by direct sensing of nodes.

5.2.2.5 Attack on Data

The malicious node can alter, falsify and forward false data about the environment. This attack can be addressed by correlating the data with other sensors and by evaluating the trust score of data item.

5.2.2.6 Sybil Attack

The malicious nodes can produce many false ID and try to imitate as different nodes at different time in the network. This can be addressed by the identification of ID by powerful nodes like a base station or centralized server in the network.

5.2.2.7 Replication Attack

If an enemy seizes a node and pulls out its credentials, it is possible for an enemy to produce many numbers of replicas with the same identity and deploy at different locations. This is called replication attack. Like Sybil attack, this also can be handled by the base station.

5.2.2.8 Collusion Attack

Two or more malicious nodes are work together to give the wrong recommendation about nodes in the network. This attack is known to be the most destructive attack than above-said attacks. This can be handled by having direct observation of every node in the network.

5.2.3 Various Types of Trust

In literature, there are several definitions of trust. It is defined as a level of confidence or certainty that a sensor can have on another sensor for executing specific function based on past behaviors. In this proposed trust management scheme, we have used four types of trust, which are the following:

5.2.3.1 Direct Trust

It is evaluated by considering the direct collaboration between nodes. It can be calculated by taking account of successful and failed transactions between nodes ([Shaikh et al. \(2009\)](#)).

5.2.3.2 Indirect Trust

When there is no direct collaboration amongst two nodes, the indirect trust can be established between two nodes by getting recommendations from their neighbor nodes ([Shaikh et al. \(2009\)](#)).

5.2.3.3 Self-data trust

It is computed by the data source for the self-detection of its sensed data item by applying confidence interval method, battery power residual method and time series analysis method ([Karthik and Ananthanarayana \(2016\)](#)).

5.2.3.4 Peer data trust

It is computed by the neighbor node of the data source by equating its sensed data item with source node data item for finding the correlation coefficient ([Karthik and Ananthanarayana \(2016\)](#)).

5.2.3.5 Trust score

Usually, a trust score is represented as a numerical value. Here, we consider the trust score as an integer lying between -1 to +1 as suggested in ([Karthik and Dhulipala \(2011\)](#)).

5.2.3.6 Provenance data

It gives the knowledge about the data source, how the data item is generated, how it is transferred, how it is passed to sink node and the operations involved since its creation (Lim *et al.* (2010)).

5.2.3.7 Subject and object node

If a node needs to evaluate the trust score of another node, then the evaluating node is called a subject node and judged node is called an object node. If it is a multi-hop network, the subject node cannot evaluate the trust score of an object node directly. The trust score of the object node is evaluated by getting recommendations from intermediate nodes. The intermediate node is called as recommender. Here the transaction means the cooperation among the sensor nodes. The transaction is successful, if the subject node receives an acknowledgment from the object node and it ensures that the packet is routed towards sink node without any data alteration. The transaction is said to be an unsuccessful transaction, if the subject node does not receive any acknowledgment from the object node.

5.3 Proposed Trust Management Scheme

This section presents the trust evaluation of sensor node and the data item. Consider a WSN which consists of 11 sensor nodes as shown in Figure 5.1, which are used to monitor the events in the environment. As shown in Figure 5.1, Node 2 is the source node which monitors the environment and produces a data item and forwards to the next hop. The nodes 5, 9 and 11 are intermediate nodes which forward the data item to sink node. By using data provenance associated with the data item, the sink node evaluates the trust score of intermediate sensor nodes and data source.

The proposed HTMS works in three phases.

1. Trust assessment of data item.
2. Trust assessment of source node and intermediate nodes
3. Punish and reward

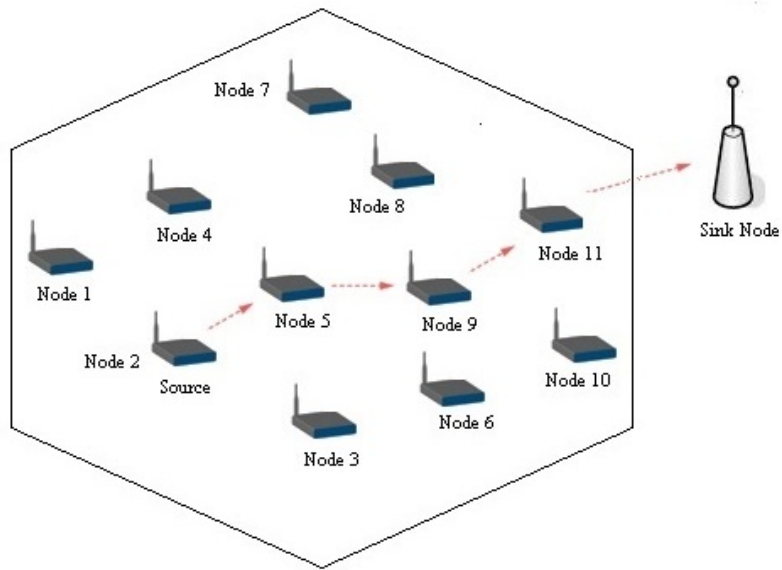


Figure 5.1: A simple WSN scenario

5.3.1 Structure of HTMS

In this subsection, we elucidate the structure of HTMS. Figure 5.2 expose the structure of proposed trust evaluation. The proposed Hybrid TMS consists of three main procedures: data trust evaluation, node trust evaluation, and trust score adjustments which comprise of six modules: self- data trust, peer data trust, direct trust, indirect trust, provenance based trust, punish and reward.

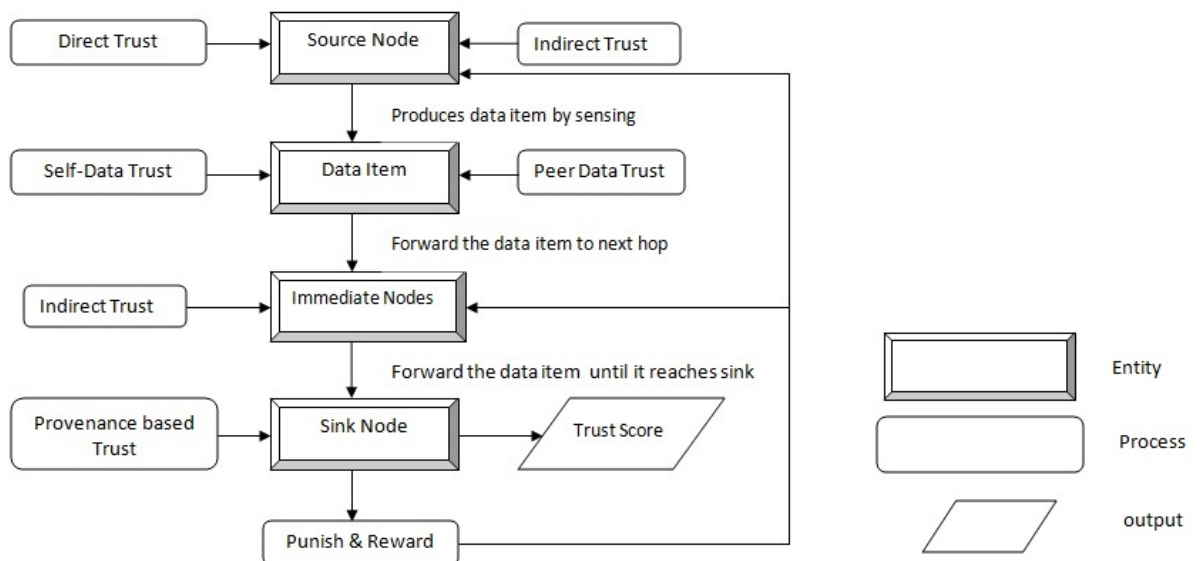


Figure 5.2: Structure of HTMS

Source node generates the data item by sensing the environment. The self-data trust is evaluated by a node for its data item. The data item is forwarded to next-hop with self-data trust. The next-hop neighbor node receives the data item and evaluates the peer-data trust. The data item is forwarded to the next hop until it reaches the sink node. The sink node uses data provenance to find the source and intermediate nodes. The direct trust and indirect trust of the source node and indirect trust of intermediate nodes are evaluated. The trust score adjustments are based on provenance based trust which is done by the sink node/ base station to find the final trust score. The punish/reward process is carried out at the sink node/base station by considering the final data trust score.

When we evaluate the trustworthiness of data item and node, there is a trust model between subject and object nodes. The proposed HTMS is applies to both single-hop and multi-hop networks. In a single-hop network, the direct trust evaluation is activated along with self-data trust and peer data trust evaluation procedures at the in-network node level. At the sink node level, the provenance based trust evaluation is activated to find the final trust score of node and data item. In the case of a multi-hop network, instead of the direct trust evaluation procedure, indirect trust evaluation procedure is activated along with other procedures.

In a single-hop network, we define threshold i as the minimum number of the transaction between subject and object nodes. If the transaction between subject and object node is higher than i , then direct trust evaluation is triggered. Otherwise, initial trust score or trust score of object node data item is considered. In multi-hop network, the subject node will select the set of trusted recommenders for evaluating the recommendation trust of the object node. We apply a localized distributed approach at node level to detect the faulty data item in online fashion. We apply a centralized approach at sink node level for replacing or leaving out faulty readings, malicious, selfish nodes by taking the decision from the application point of view.

5.3.2 Data Trust Evaluation

This subsection presents the data trust evaluation. Data trust is calculated in three steps: self-data trust, peer data trust and provenance-based trust.

5.3.2.1 Self-data trust

Self data trust score is calculated by using three methods: confidence interval method, battery power residual method and time series analysis method. In confidence interval method, the source node checks its data item with anticipated range of values. The dimension property is modeled with sensor nodes during the network and application initialization. If the sensed data item (D_i) falls within the expected range (D_{max}, D_{min}), then it gets good trust score. Otherwise, the data item is considered as either uncertain or untrustworthy as shown in Equation (5.1).

$$S_{dt1} = \begin{cases} -1, & \text{if } D_{min} > D_i > D_{max}, \text{ untrustworthy data item} \\ 0, & \text{if } D_i = D_{min} \text{ or } D_i = D_{max}, \text{ uncertain data item} \\ +1, & \text{others, trustworthy data item} \end{cases} \quad (5.1)$$

The second method deals with the inspection of residual battery power of sensor nodes while producing data item. Because the chance of generating faulty data item is more when the sensor node is running out of battery power [Fang and Dobson \(2013\)](#). The data item is considered as untrustworthy when the sensor battery residual power level is below the threshold value. Here we consider 5% as the threshold value. The appropriate threshold value is chosen based on the application. The generated data item is trustworthy when the battery residual power level is higher than the threshold level, and it is shown in the Equation (5.2).

$$S_{dt2} = \begin{cases} -1, & \text{if battery power level} < 5\%, \text{ untrustworthy data item} \\ 0, & \text{if battery power level} = 5\%, \text{ uncertain data item} \\ +1, & \text{if battery power level} > 5\%, \text{ trustworthy data item} \end{cases} \quad (5.2)$$

Time series data analysis is performed as the final step in self data trust evaluation. The sensor node uses time series prediction algorithm, auto regressive moving average method to forecast the upcoming data item based on previous data items. Then the sensor node equates the predicted data item (PD_i) with sensed data item (D_i). If the deviation between sensed data item and predicted data item goes beyond the threshold value ϕ then the sensor node assigns an untrustworthy score to the data item. If the prediction of the data item is not possible, then the sensor node assigns an uncertain

trust score as shown in Equation (5.3).

$$S_{dt3} = \begin{cases} -1, & \text{if } |D_i - PD_i| > \phi, \text{ untrustworthy data item} \\ 0, & \text{others, uncertain data item} \\ +1, & \text{if } |D_i - PD_i| < \phi, \text{ trustworthy data item} \end{cases} \quad (5.3)$$

5.3.2.2 Peer data trust

Sensor data items are not random. Usually, they are highly correlated with time and space. The data items coming from the same event and same region have a high correlation. In peer data trust evaluation, the neighbor node compares its data item x_i with source node data item y_i which monitors the same event to compute the correlation coefficient. The correlation coefficient between two data items x_i and y_i is computed by using equation (5.8).

The mean of the data item from node x is calculated by using this formula

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (5.4)$$

The mean of the data item from node y is calculated by using this formula

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i \quad (5.5)$$

The variance for the node x is calculated as

$$x^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (5.6)$$

The variance for the node y is calculated as

$$y^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2 \quad (5.7)$$

The Peer data trust P_{dt} can be calculated from the correlation coefficient between node x and y data items by using Equation (5.8).

$$P_{dt} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 (y_i - \bar{y})^2}} \quad (5.8)$$

The neighbor node forwards the data item with peer data trust score (P_{dt}) to base station through intermediate nodes. The base station use this trust score for evaluating the final data trust score (DT).

5.3.2.3 Provenance based data trust evaluation

In a multi-hop sensor network, the data item is forwarded to the sink node through several intermediate relay nodes. The data provenance (Lim *et al.* (2010)) is used to find the data source, history of data versions and path followed by the data items to reach the sink node. The aim of this trust evaluation is to allot a data trust score to every data item by looking its self data trust, peer data trust, and provenance data. Final data trust score is allotted by sink node by inspecting the data item similarity and provenance similarity. Data item similarity refers to the similar data items which are coming from different sensors of the same event. The dissimilar data provenance of similar data items of an event increases the trust score of the data item. When we have similar data provenances and dissimilar data items of the same event, then the data item trust score is decreased. For instance, let D_a, D_b be the data items and P_a, P_b be their provenances. Table 5.1 shows the data trust score assignment by the sink node for the compounding of data item similarity and data provenance similarity.

Table 5.1: Value and Provenance similarity

| Similarity | $P_a = P_b$ | $P_a \neq P_b$ |
|----------------|-------------|----------------|
| $D_a = D_b$ | No change | Reward |
| $D_a \neq D_b$ | Punish | No change |

We are adjusting the peer data trust score by considering the similarities of provenances and data items (Lim *et al.* (2010)). When the data items from two different sensor nodes are same, and their provenance are equal, there is no need to adjust the peer data trust score. When the data items from two different sensor nodes are same, and their provenances are different, then reward has been given by adding the trust score

of 0.05 to peer data trust as shown in the Equation (5.9). There is a large negative effect when we have different data items from different sensor nodes but their provenances are same. In this case, a punishment is given to peer data trust by reducing 0.05 from it. There is no change in peer data trust score when the data items of two sensor nodes and their provenance data are different. The reward and punish trust scores are chosen randomly, and it is application dependent.

$$DT = \begin{cases} P_{dt}, & \text{if}(P_{dt} > 0.95) \\ P_{dt}, & \text{if}(P_a = P_b) \& (D_a = D_b), \text{no change} \\ P_{dt} + 0.05, & \text{if}(P_a \neq P_b) \& (D_a = D_b), \text{reward} \\ P_{dt} - 0.05, & \text{if}(P_a = P_b) \& (D_a \neq D_b), \text{punish} \\ P_{dt}, & \text{if}(P_a \neq P_b) \& (D_a \neq D_b), \text{no change} \end{cases} \quad (5.9)$$

5.3.3 Node Trust Calculation

This subsection presents the node trust evaluation. Node trust is calculated in threefold steps: Direct, Indirect and provenance-based trust calculation.

5.3.3.1 Direct Trust calculation / Time based past interactions

Direct trust score is depending on the successful and failed transaction between subject and object nodes. Whenever the object node receives the packet from the subject node, the object node forwards to next hop or destination and then it will reply to the subject node with an acknowledgment. Whenever the packet is successfully forwarded to next hop or the destination and the source node receives an acknowledgment packet from object node, then the transaction is said to be a successful transaction. When the source did not receive any acknowledgment from a neighbor or fails to forward the packet to next hop or to destination, then it is termed as an unsuccessful transaction. If neighbor did not reply for the reception of data packet, then the source may assume either the communication is failed, and it tries to retransmit the same data packet or the neighbor node is the malicious one. If this is the case, then the source will give less trust score to the neighbor node.

For calculating the trust score of the node, the sliding window concept is used. The net-

work congestion, traffic should not have affect on trust evaluation [Shaikh et al. \(2009\)](#). Here we consider sliding window as suggested in [Shaikh et al. \(2009\)](#) for direct trust evaluation which does not provide any influence to real-time packet delivery. The timing window (T) can be utilized to measure the successful and unsuccessful transaction. The sliding window consists of several timing windows like T1, T2, T3, etc. and the timing window consists of several time units. In Figure 5.3, each window is considered

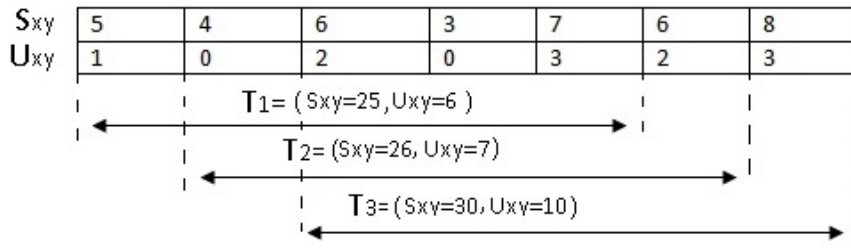


Figure 5.3: Sliding window

as 5 time elements. The window size selection is based on the application requirement. The window size can be fixed as shorter or longer based on network scenario ([He et al. \(2012\)](#)). During the first-time window T1, five time elements are considered. From the second-time element, the second timing window T2 is considered. As the time increases, the window T disremembers the understandings of the first time element and start processing the next five units. During first time unit of T1, the successful transactions and unsuccessful transactions are 5 and 1 correspondingly. During the first-time window T1, the total number of successful transactions and unsuccessful transactions are 25 and 6 respectively. The direct trust score T_{xy} of the node y at sensor x can be calculated by Equation (5.10).

$$T_{xy} = (S_{xy} - U_{xy}) / (S_{xy} + U_{xy}) \quad (5.10)$$

Where S_{xy} denotes the sum of successful transactions between sensors x and y in one-time window T and U_{xy} denotes the sum of failed transactions between sensors x and y in one-time window T. Figure 5.4 shows the depportment of direct trust score against successful and unsuccessful transactions. When we have all transactions as successful transaction, we get trust score as +1. The trust score reduces with the raise of unsuccessful transaction.

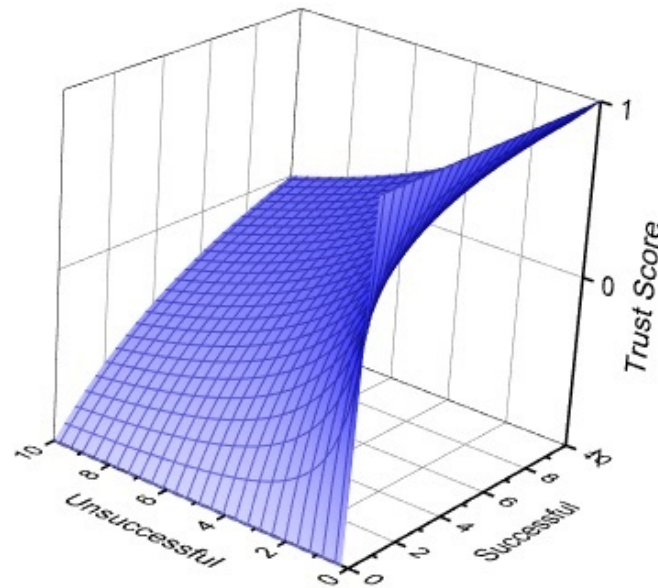


Figure 5.4: Direct trust evaluation

5.3.3.2 Indirect trust calculation

Indirect trust can be set up between two nodes that have not beforehand associated much because trust is transitive. The indirect trust score is calculated by two steps:

1. Find the trusted recommender between subject and object node.
2. Trust propagation of direct trust score from recommender on object node through trust chain ([Jiang et al. \(2015\)](#)).

Nodes may require indirect trust evaluation for certain reasons:

1. Lack of information about the behavior of a node, due to less communication amongst sensors.
2. To mix recommendations with direct trust score to get a complete trust score.

The node X wants to evaluate the trust score for node Z . since the node X has not interacted much with node Z , it cannot evaluate the trustworthiness of node Z directly. Hence node X triggers the indirect trust evaluation procedure as shown in Figure 5.5. The node X will broadcast the request for recommendations about node Z to its neighbors. Upon receiving the recommendation request, the node Y will give recommendation about node Z , since it has enough direct interactions with node Z . This process is called trust propagation.

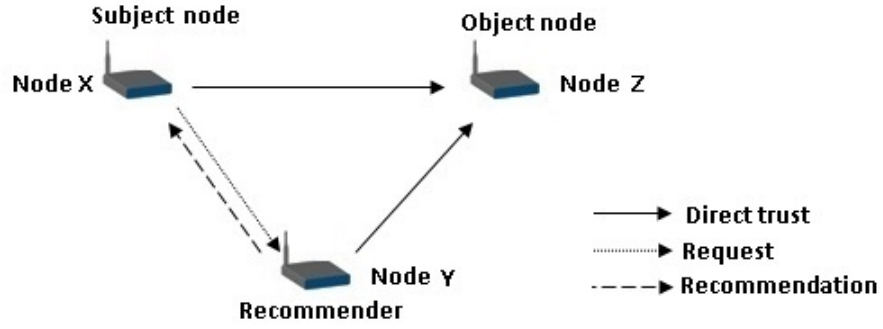


Figure 5.5: Indirect trust evaluation

$$IT_{xz} = \begin{cases} T_{xy} * T_{yz}, & \text{if } (T_{yz} < 0.3) \\ 0.3 + (T_{xy} - 0.3) * T_{yz}, & \text{else} \end{cases} \quad (5.11)$$

IT_{xz} is the indirect trust of node Z by node X, T_{xy} is the trust score of recommender node Y by node X, and T_{yz} is the recommendation value by node Y about object node Z. The indirect trust evaluation makes sure that the trust score of the object node should not go beyond the trust score of the recommender.

5.3.4 Provenance based node trust value evaluation

In a multi-hop sensor network, provenance data (Lim *et al.* (2010)) is used by the sink node to find out the source sensor node and forwarding path of the data item and history of version since its generation. The interdependency property is introduced in (Bertino (2014)), and it depends on the interdependency of sensor nodes and its sensed data items. The data item is continuously produced from the sensor nodes in WSN and it is being forwarded by intermediate sensor nodes to the sink node. The trust score of the data item should be evaluated continuously when it arrives at sink node. The sensor nodes in the WSN are evaluated continuously based on the sensed data item they produced and forwarded. The overall objective of this framework which is shown in Figure 5.6 is to assign a trust score [-1 to +1] to each data item and set of sensor nodes in WSN which participates in sensing process and forwarding the data item to the sink node. This framework maintains three types of trust score like source node trust, data item trust, and intermediate nodes trust score. The data correlation metric is used to score the sensed data. The data item trust score affects the trust score of the source sensor node and intermediate sensor in WSN. The trust score of the data source

node and intermediate sensor nodes affect the data item score. If the sensor reports a trustworthy reliable data, then the trust score of the source sensor node and intermediate sensor nodes increases. Alternatively, if the sensor node reports unreliable data, then the trust score of the sensor nodes in WSN decreases automatically.

The data trust score is used for making critical decisions in the application. If the trust

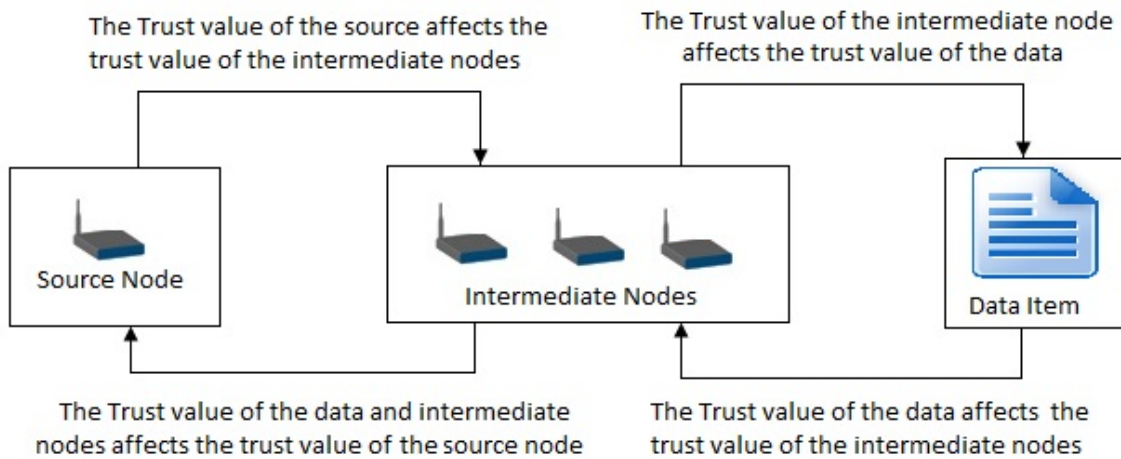


Figure 5.6: Interdependency property of data item and nodes

score of the data item is above the threshold value (ϕ) to make critical decisions, then the modifications of the trust scores for source node and intermediate nodes (which are responsible for producing and forwarding the data item) is performed by means of adding trust scores to them. If the data trust score is not reliable enough to make critical decisions, then the punishment is given to the source and intermediate nodes by reducing their trust scores. These modifications of trust scores of nodes as shown in Equation (5.12) are done by sink node or base station. The reward and punish trust scores are chosen randomly and it is application dependent. The adjustment of the trust score helps nodes to select the best path or best node for next transaction.

$$NT = \begin{cases} T_{xy}, & \text{if } (T_{xy} > 0.95) \& (DT) > \emptyset \\ T_{xy} + 0.05, & \text{if } (DT) > \emptyset, \text{ reward} \\ T_{xy} - 0.05, & \text{if } (DT) < \emptyset, \text{ punish} \\ T_{xy}, & \text{if } (\emptyset \text{ is uncertain}) \end{cases} \quad (5.12)$$

5.3.5 Algorithms for trust evaluation

From the above discourses, the data trust score and node trust score evaluation algorithms are presented in this subsection. Algorithm 5.1 is used to evaluate the self-data trust. It takes the sensed data item, minimum and maximum expected value, and energy level of the node as input. If the number of sensed data item is less than 5, then it checks first two procedures for evaluating the self-data trust score. According to (Schönbrodt and Perugini (2013)), with small sample size, the correlation estimation is extremely noisy. The bigger sample size is better, but we have to choose what we get. If the number of sensed data items is higher than or equal to 5, then all three checks have been performed to get the self-data trust score.

Algorithm 5.1: Self-data Trust

Input: Sensed data item X_i , Minimum and maximum value X_{min} , X_{max} , Energy level of the node, Time unit i

Output : Self data-trust S_{dt}

```
1: if  $i < 5$  then
2:   Compute  $S_{dt1}$ ,  $S_{dt2}$  by using Equations (5.1) & (5.2)
3:    $S_{dt} = (S_{dt1} + S_{dt2}) / 3$ 
4: else
5:   Compute  $S_{dt1}$ ,  $S_{dt2}$ ,  $S_{dt3}$  by using Equations (5.1), (5.2) & (5.3)
6:    $S_{dt} = (S_{dt1} + S_{dt2} + S_{dt3}) / 3$ 
7: endif
8: if ( $S_{dt} \leq -0.3$ ) then
9:   Drop the data item without forwarding
10: else
11:   Forward the data item with  $S_{dt}$ 
12: return  $S_{dt}$ 
13: endif
```

The Algorithm 5.2 is used to evaluate the peer-data trust. It takes the set of sensed data items of source and neighbor node as input and evaluates correlation coefficient between those data items and returns peer data trust.

Algorithm 5.2: Peer-Data Trust

Input: Source node readings: X_i , Neighbor node sensor readings: Y_i , S_{dt}

Output: Peer-Data Trust P_{dt}

```
1: if(i<5)then
2:   Compute  $P_{dt} = \sum_{i=1}^n \frac{1}{n} \frac{1}{(1+|D_i-N_i|)}$ 
3: endif
4: if(i>=5)&(  $S_{dt} >=0$ ) then
5:   Compute  $P_{dt}$  by using Equation (5.8)
6: else
7:    $P_{dt} = S_{dt}$ 
8: endif
9: if( $S_{dt} <=0$ )&(  $P_{dt} <=0$ ) then
10:  Drop the data item without forwarding
11: else
12:  Forward the data item with  $P_{dt}$ 
13: return  $P_{dt}$ 
14: endif
```

Provenance-based data trust score evaluation is presented in Algorithm 5.3. The sensed data items of the same event and their provenances are given as input for the algorithm. The algorithm rewards or punishes the peer data trust score based on data and provenance similarities to get final data trust score. The final data trust score is used to make a critical decision.

Algorithm 5.3: Provenance based Data Trust

Input: Sensor readings from different sensors V_a , V_b , Provenances for sensor readings P_a , P_b , reward trust score: 0.05, punish trust score: 0.05

Output: Data Trust score DT

```
1: Check for value and provenance similarity
2: if( $V_a = V_b$ )&(  $P_a = P_b$ ) then
3:   DT =  $P_{dt}$ 
4: endif
5: if( $V_a = V_b$ )&(  $P_a \neq P_b$ ) then
```

```

6:     DT=  $P_{dt}$ + reward trust score
7: endif
8: if( $V_a \neq V_b$ )&(  $P_a = P_b$ ) then
9:     DT=  $P_{dt}$ - punish trust score
10: endif
11: if( $V_a \neq V_b$ )&(  $P_a \neq P_b$ ) then
12:     DT=  $P_{dt}$ 
13: endif
14: return DT

```

To find the source node trust and intermediate node trust, Algorithm 5.4 is used. It takes data trust score threshold ϕ , initial trust score of the node, reward trust and punish trust score as input. It produces a final trust score in the WSN based on the threshold ϕ .

Algorithm 5.4: Node Trust

Input: DT, data trust score threshold ϕ , T_{xy} , reward trust score: 0.05, punish trust score: 0.05

Output: Node Trust score NT

```

1: if( $T_{xy} > 0.95$ )& (DT >  $\phi$ ) then
2:     NT=  $T_{xy}$ 
3: endif
4: if (DT >  $\phi$ ) then
5:     NT=  $T_{xy}$ + reward trust score
6: endif
7: if (DT <  $\phi$ ) then
8:     NT=  $T_{xy}$ - punish trust score
9: endif
10: if( $\phi$  is uncertain) then
11:     NT=  $T_{xy}$ 
12: endif
13: return NT

```

5.3.6 Steps involved in trust evaluation of HTMS

The main steps of trust evaluation include the following:

1. Sensor node collects data from the environment and calculates self-data trust by Algorithm 5.1.
2. The neighbor node receives the data item along self-data trust. Then it calculates peer data trust by Algorithm 5.2.
3. If the neighbor node receives more than one data item, then it chooses one data item which has high self-data trust.
4. The data item with peer data trust is forwarded through intermediate nodes until it reaches the sink node.
5. The sink node calculates the total trust score of the data item and node by Algorithms 5.3 and 5.4.

5.3.7 Attack Resistant Direct Trust Evaluation

The trust score update has two main steps. First one is trust score aging or decay factor in which the past trust score has less importance than present trust score. We allot various weightings to past and present trust scores to make the proposed TMS defend against on-off attacks. The second step is periodic update of trust score. Because of the dynamic nature of nodes in WSN, any sensor may connect with or disconnect from the network at any time. Therefore, the trust score of the sensor node should be updated periodically. Regular updates may consume lot of resources in the WSN. When the update interval is lengthy then, it cannot represent the present status of WSN. To address this problem, sliding window is used. The combination of trust aging/decay factor with sliding window technique works well to defend on-off attacks (He *et al.* (2012)). Then the trust score is calculated by

$$T_{xy} = \alpha((s_{xy} - U_{xy})/(s_{xy} + U_{xy})) \quad (5.13)$$

Where α is the aging factor, and it takes the value from $0 < \alpha < 1$, which describes that the trust score made long-ago must carry less importance than trust score made as of late. We are using exponential decrease method (Jiang *et al.* (2015)) to update α for giving less weight-age to old observations. The value of α is updated by $\alpha = \beta^{m-j}$, where $0 < \beta < 1$, m is the number of time units in sliding window and j is the current time

unit of the sliding window. Therefore, $\alpha_1 < \alpha_2 < \alpha_3 < \dots < \alpha_m$ and $\sum_{j=1}^m \alpha_j = 1$.

When the object node dynamically changes its behavior, then the subject node uses aging factor to catch and reduce the effect of the dynamic nature of object node. It should be calculated in such a way that the observations made a long time ago cannot be used for predicting the nature of the object node and bad behavior is considered for longer time than good behavior.

The length of the sliding window can be made longer or shorter based on the frequency of on-off attack. During on-state, the sensor node would drop more than 80% of packets which leads to more number of unsuccessful transactions. In off-state, the sensor node successfully interacts with its neighbors without dropping much packets. The malicious node may take advantage of behaving bad and good alternatively to keep its good trust score (He *et al.* (2012)). To overcome this on-off attack, the evaluation of trust score between x and y nodes, T_{xy} can be modified as:

$$T_{xy} = \begin{cases} -1, & \text{if}(f = 1) \\ (s - u)/(s + u), & \text{if}(f = 0) \\ \text{window size}(), & \text{if}(0 < f > 1) \end{cases} \quad (5.14)$$

Where s and u represents the number of successful and unsuccessful transactions and f is the frequency of behavior changes by a node and it is calculated by

$$f = \text{onstate}/(\text{onstate} + \text{offstate}) \quad (5.15)$$

$$\text{window size} = \begin{cases} n = n - 1, & \text{if}(f = 0.4) \\ n = n - 2, & \text{if}(f = 0.6) \\ n = n - 3, & \text{if}(f = 0.8) \end{cases} \quad (5.16)$$

Equation (5.16) is applies to the scenario with the window size $n=5$. By using equations (5.14), (5.15) and (5.16), we can overcome the on-off attacks in direct trust evaluation.

5.3.8 Attack Resistant Indirect Trust Evaluation

If the subject node receives multiple recommendations about object node, then it checks for consistency among the recommendations as follows: the subject node evaluates the

mean(m) and standard deviation(sd) for recommendations.

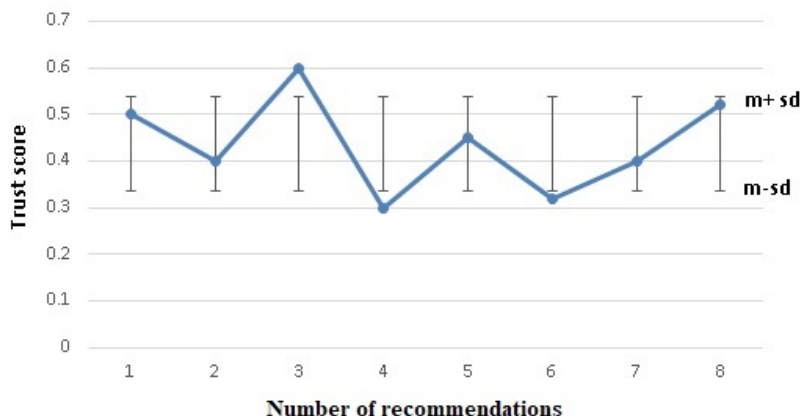


Figure 5.7: Filtering of recommendations

The recommendations within ($m-sd$, $m+sd$) are considered as good recommendations. Figure 5.7 shows the filtering of recommendations from various nodes to defend against the bad-mouthing attack.

From Figure 5.7, we can say that the subject node ignores the 3, 4 and 6 recommendations as they are falling outside the range ($m-sd$, $m+sd$). If the subject node receives a recommendation from the recommender, then it checks for trust score of the recommender is higher than the threshold value to overcome the bad-mouthing attack for single recommendation.

5.4 Results and Discussions

In this section, we report our performance of proposed HTMS. First, we explain about simulation environment, and then we elucidate our experimental analysis and results. We have implemented the proposed HTMS using MATLAB and NS2 simulations. We deployed 8 sensors randomly in the area of $100*100m^2$. Static nodes are considered for simulations, and they are organized in random topology. This network comprises of one sink node/base station which is located in the terrain.

5.4.1 Trust evaluation comparison

To the best of our knowledge, Distance-Based Trustworthiness Assessment for sensors (DBTA) ([Won and Bertino \(2015\)](#)) is the recent method for trust evaluation which consider the interdependency property. Hence, DBTA is chosen as the comparison method

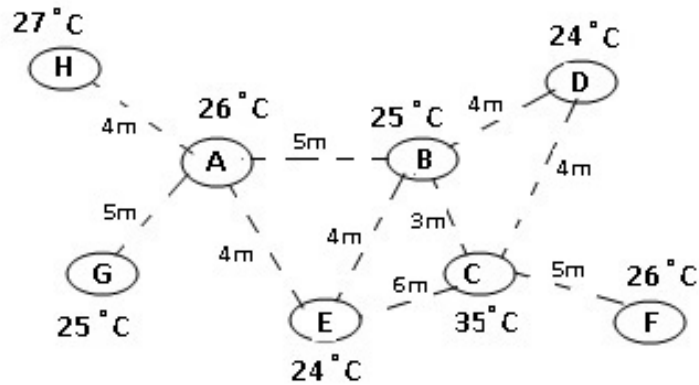


Figure 5.8: Sample scenario for trust evaluation

for the evaluation of trust scores. In DBTA, the data item trust score is calculated by considering the time t , location information and previous trust score of the sensor. The trust score of sensor node is calculated by considering the previous trust score of node and data item. The trust score in DBTA ranges from 0 to 1 and trust score 0.5 is the trustworthy state. Figure 5.8 shows the sample scenario which is adapted from DBTA (Won and Bertino (2015)) for trust evaluation.



Figure 5.9: Data and node trust score using DBTA

The initial trust score of all nodes in DBTA is 0.5. We assume that sensed data items of sensor nodes in the scenario do not change throughout the trust evaluation process. During the trust evaluation at time t , the trust score of node A, B and C are 0.58, 0.25 and 0.09 correspondingly. Sensor node A data item gets high trust score since it is consistent with neighbor nodes data items. The data item of B gets a low trust score even though it is working well because of its closest neighbor node C sensed data item. The node C data item gets very less trust score since it is abnormal and it is not consistent with other neighbors in the network. At the time $t+3$, the trust score of data items of node A, B and C change to 0.58, 0.55 and 0.09 correspondingly. The trust score

of the data item of node B becomes high as the trust score of C becomes low as time goes on. Figure 5.9 shows the trust score of node B and its data item over the period using DBTA. The data item might be vanished by the time DBTA recognizes that there are data fault in the gathered data. Dealing with such a problem would need an online data evaluation process and the facility to rapidly achieve data recovery and remedial process. In the proposed method HTMS, the trust score of the data item is calculated by

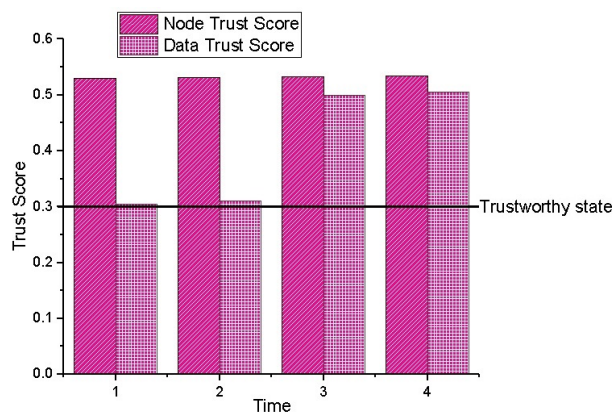


Figure 5.10: Data and node trust score using proposed method

self-data trust, peer-data trust, and provenance based trust procedures. The trust score in HTMS ranges from -1 to 1 and trust score 0.3 is the trustworthy state. For real-time detection of trustworthiness of data item, the self-data trust procedure is triggered to find out the trust scores of data items by restricted interval method and energy depletion method. The time series data analysis is also conducted with above two methods when we have a set of sensor data items. For the sample scenario in Figure 5.8, the data trust score of node B at time t is 0.53. The trust score of data item and node evaluated using HTMS is shown in Figure 5.10.

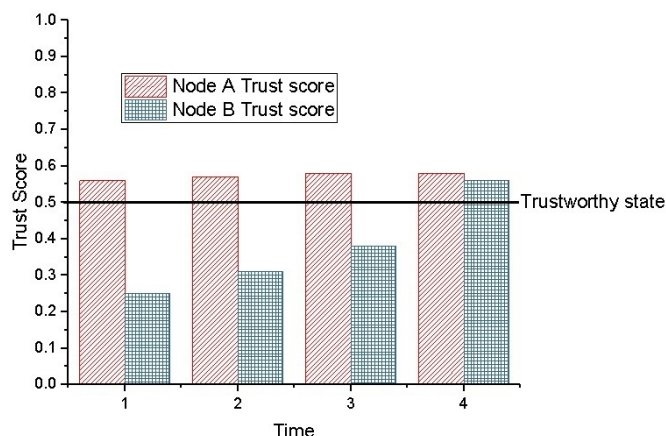


Figure 5.11: Node trust score evaluation by DBTA

According to DBTA trust evaluation, if the trust score of the data item is less, then the node will get less trust score because the trust score of a node is based on the trust score of its data item. Only one trust component (data trust score) is considered in DBTA (Won and Bertino (2015)) for scoring both data item and node in WSN. Considering only one component for trust evaluation might misguide the network.

Figure 5.11 shows the trust scores of nodes A and B by DBTA. In DBTA, node B gets an untrustworthy score for the first few iterations since the trust evaluation of node B depends on neighbor node C. In proposed method HTMS, the communication trust

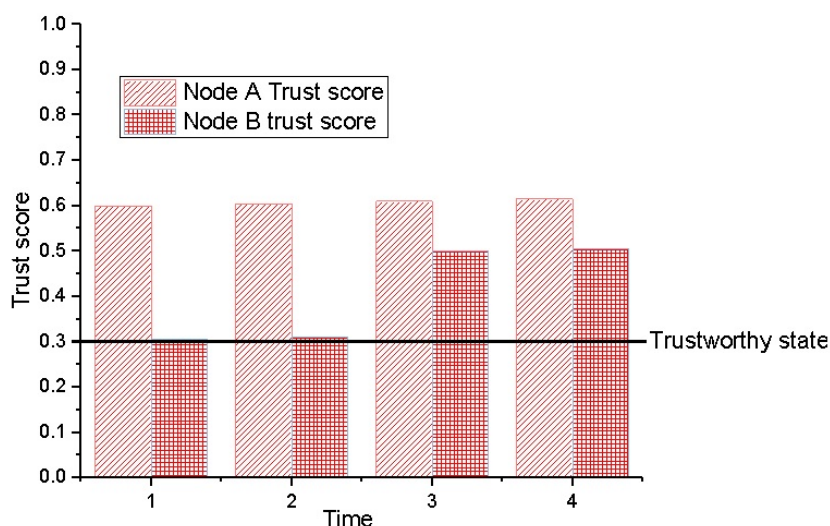


Figure 5.12: Node trust score evaluation by HTMS

is considered for scoring the node. And data trust score is considered for reward and punishment procedures. The node B gets good trust score and termed as a trustworthy node in the first iteration itself which is shown in Figure 5.12.

Table 5.2 shows that the DBTA method works well when detecting the data trust of untrustworthy node C with four trustworthy neighbors.

Table 5.2: Case 1: Untrustworthy node C with four trustworthy neighbor

| Trust Evaluation/Iterations | Data Trust | |
|-----------------------------|----------------------|----------------------|
| | DBTA | HTMS |
| At Time t | Untrustworthy | Untrustworthy |

Table 5.3 revealed the fact that the DBTA is not suitable for data trust evaluation of node F, because it has only one untrustworthy node.

Table 5.3: Case 2: Trustworthy node F with one untrustworthy node

| Trust Evaluation/Iterations | Data Trust | |
|-----------------------------|---------------|--------------------|
| | DBTA | HTMS |
| At Time t | Untrustworthy | Trustworthy |
| At Time t+1 | Untrustworthy | Trustworthy |

The DBTA can find the final data trust score of node B after three iterations when it is surrounded by one untrustworthy neighbor and three trustworthy neighbors, and it is represented in Table 5.4

Table 5.4: Case 3: Trustworthy node B with one untrustworthy neighbor and three trustworthy neighbors

| Trust Evaluation/Iterations | Data Trust | |
|-----------------------------|--------------------|--------------------|
| | DBTA | HTMS |
| At Time t | Untrustworthy | Trustworthy |
| At Time t+1 | Untrustworthy | Trustworthy |
| At Time t+2 | Untrustworthy | Trustworthy |
| At Time t+3 | Trustworthy | Trustworthy |

Even though the trustworthy node A is surrounded by trustworthy neighbors, the DBTA is detecting the final trust score of node A in the second iteration as shown in Table 5.5. When the number of neighbors are increasing, the performance of DBTA decreases because it is considering all neighbors for trust score evaluation.

Table 5.5: Case 4: Trustworthy node A with more number of trusted neighbors (<3)

| Trust Evaluation/Iterations | Data Trust | |
|-----------------------------|--------------------|--------------------|
| | DBTA | HTMS |
| At Time t | Untrustworthy | Trustworthy |
| At Time t+1 | Trustworthy | Trustworthy |

Table 5.6 and 5.7 shows that the DBTA and HTMS works well in data trust score evaluation when trustworthy node has only one trusted neighbor.

Table 5.8 and 5.9 show that the DBTA is taking time to evaluate the final trust score of a trustworthy node when it is surrounded by trustworthy and untrustworthy neighbors. In all cases from 1 to 8, the proposed HTMS evaluates the final data trust score in the first iteration itself.

Table 5.6: Case 5: Trustworthy node H with only one trusted neighbor

| Trust Evaluation/Iterations | Data Trust | |
|-----------------------------|--------------------|--------------------|
| | DBTA | HTMS |
| At Time t | Trustworthy | Trustworthy |

Table 5.7: Case 6: Trustworthy node G with only one trusted neighbor

| Trust Evaluation/Iterations | Data Trust | |
|-----------------------------|--------------------|--------------------|
| | DBTA | HTMS |
| At Time t | Trustworthy | Trustworthy |

Table 5.8: Case 7: Trustworthy node E with one untrustworthy neighbor and two trusted neighbors

| Trust Evaluation/Iterations | Data Trust | |
|-----------------------------|--------------------|--------------------|
| | DBTA | HTMS |
| At Time t | Untrustworthy | Trustworthy |
| At Time t+1 | Untrustworthy | Trustworthy |
| At Time t+2 | Untrustworthy | Trustworthy |
| At Time t+3 | Untrustworthy | Trustworthy |
| At Time t+4 | Trustworthy | Trustworthy |

Table 5.9: Case 8: Trustworthy node D with one trustworthy neighbor and one untrustworthy neighbor

| Trust Evaluation/Iterations | Data Trust | |
|-----------------------------|--------------------|--------------------|
| | DBTA | HTMS |
| At Time t | Untrustworthy | Trustworthy |
| At Time t+1 | Untrustworthy | Trustworthy |
| At Time t+2 | Untrustworthy | Trustworthy |
| At Time t+3 | Untrustworthy | Trustworthy |
| At Time t+4 | Trustworthy | Trustworthy |

5.4.2 Detection of untrustworthy data item

The goal of this subsection is to find the efficiency of proposed HTMS in identifying the untrustworthy data items. we first explain the real-time scenario in which the proposed HTMS is used to find the untrustworthy data item with the help of their trust scores. We use sensorscope ([Barrenetxea et al. \(2007\)](#)) project where sensors were deployed between Switzerland and Italy in 2007 to monitor the temperature, humidity, soil moisture, rain and wind speed. We use the temperature readings gathered by node 2. we have selected node 2 because it contains various data faults according to ([Nguyen et al. \(2013\)](#)). The sampling period for temperature data was two minutes. The data items collected from sensorscope project ([Barrenetxea et al. \(2007\)](#)) does not provide any annotation of faults. To find the ground truth of data items, we followed two steps: First, we refer to ([Nguyen et al. \(2013\)](#)) for identifying the data faults. Secondly, we manually scrutinize the dataset to identify the data faults and counter check to assure that

the fault annotations are accurate as named in (Nguyen *et al.* (2013)). This direction of constructing the dataset ground truth is consistent like (Yao *et al.* (2010)) for data items with ground truth deficiency. Our observances depict that there are data faults in node 2 temperature data items. There are some random, bias and drift faults (Nguyen *et al.* (2013)) in the dataset.

We calculate the efficiency of our proposed HTMS using this dataset and data faults identified with the help of (Nguyen *et al.* (2013)) and manual inspection. We use detection accuracy rate as our metric. For evaluating the detection rate, we identify the number of true positives out of total number of actual data faults in the dataset.

$$\text{Detection accuracy} = \frac{\text{Total number of true positives}}{\text{Total number of actual data faults}} \quad (5.17)$$

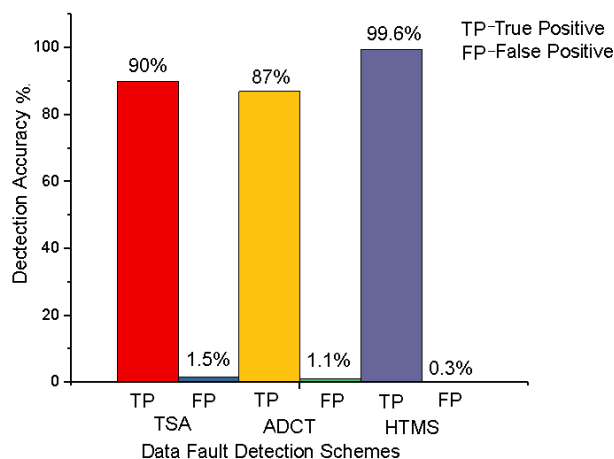


Figure 5.13: Detection accuracy of untrustworthy data items

There are 262 data faults in node 2 temperature readings in sensorscope project according to (Gilbert *et al.* (2018)) and our manual inspection. Our proposed method HTMS can identify 261 data faults out of 262 data faults and achieve 99.6 % detection rate accuracy. It is an initial hybrid framework towards online detection of data faults with the help of their trust scores and combines both centralized and decentralized schemes for real-time detection. The proposed scheme HTMS applies a compounding of Time Series Analysis (TSA), spatial analysis of data and data provenance to detect the untrustworthy data items. Figure 5.13 depicts the detection accuracy of TSA, ACDT (Spatial data analysis) (Talbi *et al.* (2017)) and proposed method HTMS. The TSA achieves 90% detection accuracy with false positive rate of 1.5% whereas the existing hybrid method ACDT achieves 87% detection accuracy with false positive rate

of 1.1%. The proposed hybrid method HTMS achieves 99.6% detection accuracy with 0.3% false positive rate. Figure 5.14 reveals the fact that when the data fault rate is

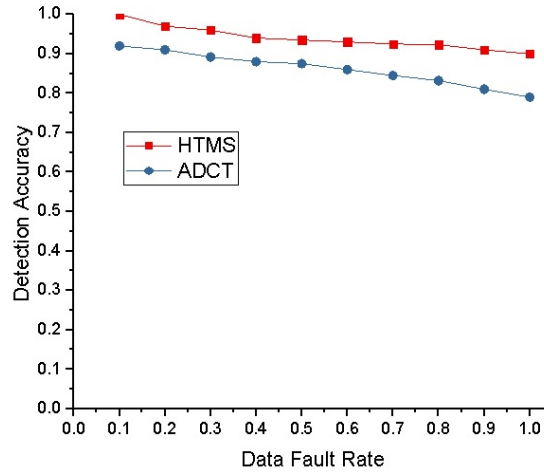


Figure 5.14: Detection accuracy of untrustworthy data items

increasing, the performance of existing schemes degrades gradually. But the proposed method HTMS maintains acceptable detection accuracy when the data fault rate is high and outperforms existing hybrid ACDT since the latter method considers only spatial data correlations.

5.4.3 Detection of malicious node

The energy parameter of node in WSN denotes the residual energy of the sensor node. For sensing, processing of sensed data and communication process, a certain amount of energy is consumed by the sensor node. If the residual energy of the node is less than the threshold value, it is not possible for the node to do its basic functionalities. To defend against the Hello flood attack and DOS attacks, the power aware model is considered in the proposed method. The sensor node specification and energy consumption rate for communication is adapted from (De Meulenaer *et al.* (2008)). State of Charge (*SoC*) of a battery is the proportion of its whole energy capability of battery that is still accessible to discharge. Depth of Discharge (*DoD*) is an amount of power discharge from the battery. When the capacity of battery is full, then *DoD* is 0%. The battery capacity decreases as the rate of discharge current increases.

We provide a method to find suspicious and unwanted communication of node and subsequent detection of malicious nodes. In hello flood attack (Pires *et al.* (2004)), a node *X* may try to send a HELLO packet with high power to make the other nodes to believe that node *X* is neighbor node to them. If the node receives a HELLO packet from node

X , then they believe that node X is a neighbor to it. When compared to other nodes in WSN, the malicious node may involve in unwanted communication which results in high energy depletion from its battery. In case of normal nodes, the energy depletion is normal and consistent with other nodes in WSN even though it involves different number of computation and listen states due to the number of neighbor nodes. The energy model is constructed in such a way that the energy consumed for computation step for all the nodes in WSN remains constant with the type of microcode operation carried out (De Meulenaer *et al.* (2008)). The rate of a particular computation can be evaluated by knowing every cycle of average energy consumption and the entire number of cycles (De Meulenaer *et al.* (2008)). To calculate, how long a battery will last long at a given rate of discharge or given load is given by c-rate. The c-rate is calculated by using peukart's capacity (Vervaeet and Baert (2002)).

$$T = \frac{C}{I} \quad (5.18)$$

where C is the given capacity of the battery, and I is the discharge current. It can be verified by calculating the State of Charge

$$SoC(t) = \frac{Q(t)}{Q_n} \quad (5.19)$$

where $Q(t)$ is the current capacity and Q_n is the nominal capacity.

Without looking the running efficiency and battery aging, the time-changing SoC can be showed in terms of DoD .

$$SoC(t) = 100\% - DoD(t) \quad (5.20)$$

In the sample scenario given in Figure 5.8, node B has to undergo listen, receive, computation and transmit operation for trust evaluation. According to (De Meulenaer *et al.* (2008)), the energy required to do listen, receive, compute and transmit operation are 68mW, 72mW, 26mW, 65mW respectively for MICAz node. Since it is a static network, the number of neighbor nodes will not change throughout the network lifetime. The node B has to do 4 listen operation, 1 receive operation, 1 computation and 1 transmit operation. So the total energy required to do one communication operation is 435 mW. The number of communication and compute operation depends on the number of

neighbor nodes. In the proposed method, the node has to listen from n , the number of neighbor nodes, receive from $(n-(n-1))$ nodes, perform $(n-(n-1))$ nodes computation and transmit the $(n-(n-1))$ nodes data to next hop or sink node. The selection of the neighbor node depends on the data trust score. The node which produces trustworthy data with high trust score is selected amongst neighbors. The malicious node which executes HELLO flood attack and involves in unwanted communication resulting in high energy depletion of battery when compared to normal nodes. We are considering the homogeneous nodes in WSN, by keep tracking the SoC of nodes, we can easily identify the malicious nodes launching the HELLO packet flood and DOS attacks.

5.4.4 Detection of selfish nodes

To defend against selective forwarding attack, the forwarding ratio (Yao *et al.* (2006)) of intermediate nodes and energy level of nodes in WSN are considered. The selfish node is a kind of node, which involves in the process of dropping the incoming packets without being relaying to its next hop. According to (De Meulenaer *et al.* (2008)), receiving a packet from its neighbor consumes 72mW and transmitting a packet consumes 65mW. The selfish node would receive the packet; without doing computation, it may drop the packet. Sometimes, it will listen, receive and do computation on data item but to save its energy it may drop the packet without forwarding to next hop or sink node. So, energy consumption would be less when compared to a normal node. By following the SoC rate of nodes over time, we can identify the selfish node in the network.

5.4.5 Memory requirement analysis of HTMS

A Trust table is kept at each node for direct trust evaluation in WSN as shown in Table 5.10. Each record size in the trust table of sensor node is $3+2w$ bytes, where w is the window size. Thus the total memory necessity for HTMS at every sensor node is $(n-(n-1))(3+2w)$ bytes, where n is the number of nodes in WSN. The trust database size is based on the size of the sliding window w .

Table 5.10: Memory requirement for direct trust evaluation at sensor node

| Node ID | Sliding window based past interaction | | Trust score |
|---------|---------------------------------------|---------------------------|-------------|
| | Successful interactions | Unsuccessful interactions | |
| 2 bytes | 1 byte | 1 byte | 1 byte |

A trust table is maintained by node, which wants to evaluate the indirect trust of nodes in WSN as shown in Table 5.11. The record size in the trust table is $4n$ bytes, where n is the number of recommendations from recommenders.

Table 5.11: Memory requirement for indirect trust evaluation at sensor node

| Node ID | Peer recommendation | Trust score |
|---------|---------------------|-------------|
| 2 byte | 1 byte | 1 byte |

5.4.6 Detection rate comparison

In this section, the existing hybrid trust model ACDT (Talbi *et al.* (2017)) and an efficient distributed trust model for WSN (EDTM) Lim *et al.* (2010) are compared with proposed method HTMS for finding the efficiency in detecting the malicious nodes and selfish nodes. The reason for choosing comparing model ACDT is the recent efficient hybrid model for the detection of malicious nodes. The EDTM is the recent distributed trust model for WSN. Therefore, it is chosen as comparing method in detection rate. The malicious attacks like DOS attack, bad mouthing attack, on-off attack, attack on information, selective forwarding attack, replication attack, Sybil attack, and collusion attacks are simulated. We alter the number of malicious nodes from 1 to 10 with 10% increment at every second. Figure 5.15 shows that the detection rate of malicious nodes by the proposed method is better than EDTM and ACDT because the HTMS considers communication data trust, interdependency property and data provenance for trust evaluation.

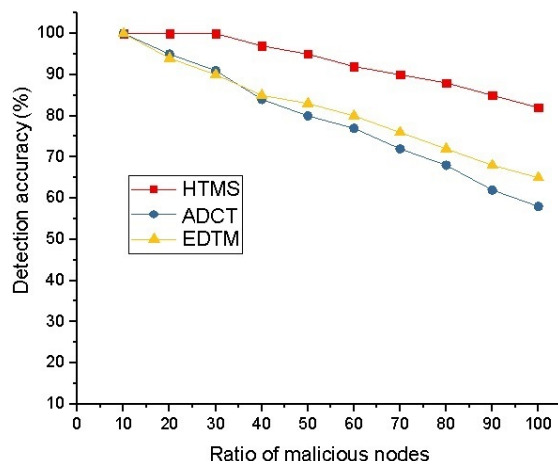


Figure 5.15: Detection accuracy of malicious nodes

5.4.7 Energy consumption comparison

Here we equate the energy usage of EDTM (Jiang *et al.* (2015)) and proposed method for collecting the trust evidences and trust evaluation. The EDTM is a distributed trust

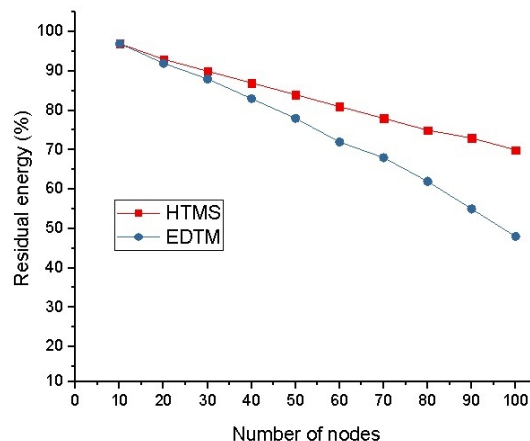


Figure 5.16: Comparison of the energy consumption

model which collect the trust evidence from all neighboring nodes since it is fully distributed in nature. The proposed method is a localized distributed method which collects the evidence from only one trusted neighbor node. Figure 5.16 depicts that the proposed method is more energy efficient than EDTM. We vary the number of nodes from 1 to 100 with 10% increment for every second and the number of neighbor nodes are less than 4. The HTMS is energy efficient, and also it takes less memory space for trust evaluation because it considers only one neighbor node.

5.5 Summary

In this chapter, a hybrid TMS for WSN is proposed, in which the data source computes its sensed data item trustworthiness by data correlation technique. The sensed data item is forwarded with self data trust to next hop neighbor. The neighbor node receives the data item and evaluates the peer data trust by spatial correlation technique. The sensed data item is forwarded with peer data trust to sink node through intermediate nodes. The sink node utilizes data provenance, interdependency property, and communication capability to evaluate the final trust score of data item, intermediate nodes, and the source node. The proposed attack resisted HTMS decreases the effects of untrustworthy data, malicious and selfish node by punish and reward mechanisms. The experimental result depicts that the HTMS outperforms existing methods by 9% in detecting and discarding the untrustworthy data, malicious and selfish nodes.

Chapter 6

Context-Aware Trust Management Scheme for Pervasive Healthcare

6.1 Preamble

In this chapter, we address the fourth research objective, finding the trustworthiness of sensor node and data in monitoring multiple events using contextual information. Medical Sensor Nodes (MSN) are used in pervasive healthcare applications like remote patient monitoring, elderly care to collect patient's vital signs for identifying the medical emergency. These resource restricted sensor nodes are prone to various malicious attacks, data faults, and data losses. Presence of faulty data, data loss in collected patient data may lead to incorrect analysis of the patient condition, which decreases the reliability of pervasive healthcare system. This chapter aims is to alert the caregiver and raise the alarm only when the patient enters into medical emergency. The proposed scheme also reduces the false alarms and alerts caused by data fault and misbehaving sensor nodes. To achieve this, we introduce a context-aware trust management scheme for data fault detection, data reconstruction and event detection in pervasive healthcare systems. It employs heuristic functions, data correlation and contextual information based algorithms to identify data faults and events. It also reconstructs the data faults and data loss for identifying the patient condition. Performance of this approach is evaluated with the help of real data samples and compared with normal TMS without context.

The primary contributions of this chapter are summed up as follows:

1. To the best of our knowledge, we are the first to introduce the Context-Aware Trust Management Scheme (CATMS) for pervasive healthcare considering contextual information, heuristic functions and data correlation methods to identify the medical emergencies.
2. We extended the sensor data validation process to have trust enabled data fault detection, trust enabled data reconstruction process and event detection by considering data correlations, contextual information, present and past evidence of medical sensor data.
3. The proposed TMS can alert the caregiver and raise the alarm only when the patient

enters into medical emergency.

Rest of the chapter is organized as follows: The taxonomy of trustworthiness in pervasive healthcare, data fault types and different types of data loss patterns are introduced in section 6.2. The proposed method is discussed in Section 6.3. Results and discussions are explained in Section 6.4. Section 6.5 concludes our chapter.

6.2 Taxonomy of Trustiness in Pervasive Healthcare

In this section, we illustrate the taxonomy of trustworthiness of pervasive healthcare, types of data faults and data loss patterns.

6.2.1 Taxonomy of Trustiness in Pervasive Healthcare

Trustworthiness in pervasive healthcare is defined as possibility of assuring the rightness and high quality of data gathered from medical sensor nodes. The objective of TMS is to assure the medical data is free from data fault, data loss and gathered from a trustworthy medical sensor node to detect the medical events. The taxonomy of trustworthiness in pervasive healthcare is shown in Figure 6.1. Construction of taxonomy of trustworthiness is based on the existing works ([Li and Zhu \(2014\)](#)), ([Bui *et al.* \(2011\)](#)), ([Haron *et al.* \(2017\)](#)), characteristics of MSN and methods used in proposed TMS for assuring trustiness in pervasive healthcare. The taxonomy aims to show the various trust evidences, trust attributes are used for evaluating the trustiness in pervasive healthcare. The following subsections explain the various aspects of taxonomy in detail.

6.2.1.1 Data source

The MSN are data sources in PHS. Direct trust is computed by considering successful and unsuccessful interaction between subject and object node. Indirect trust is evaluated by collecting the opinions from neighboring nodes when there is no direct interaction between subject and object node. We cannot compute direct and indirect trust directly soon after network set up and initialization because for the first few interactions; the direct and indirect trust score are extremely noisy. In those cases, resource level trust is computed by considering the amount of available resources of sensor node such as battery level, buffer level, sensor age, bandwidth and length of the waiting queue. Contextual information contains information about location, time of data generation and nearby entities.

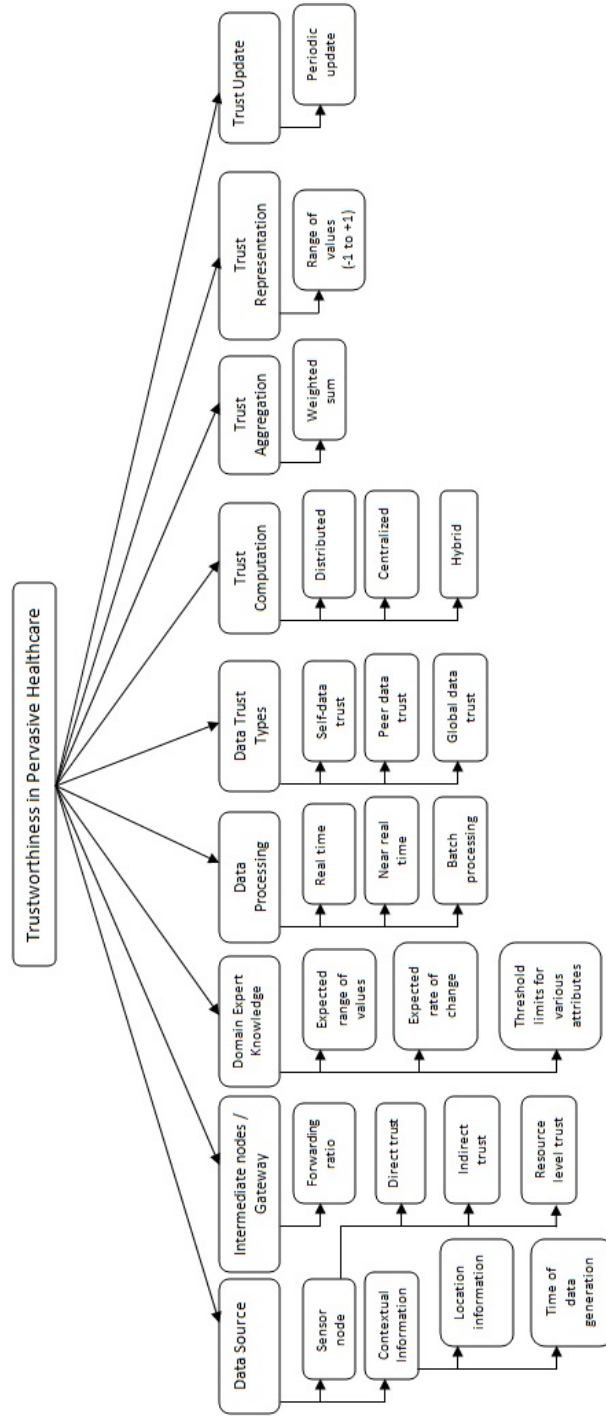


Figure 6.1: Taxonomy of Trustworthiness in Pervasive Healthcare

6.2.1.2 Intermediate node/Gateway

The intermediate node is used as a relay node for forwarding the medical information. The trust value of the intermediate node is calculated by forwarding ratio (Li and Zhu (2014)). It is the ratio of a total number of data items forwarded to the total number of data items supposed to be forwarded.

6.2.1.3 Domain Expert knowledge

Medical domain expert knowledge is used for framing the heuristic rules like an expected range of values, expected rate of change of all physiological data, minimum number of interactions, buffer size threshold, bandwidth threshold, node waiting queue threshold, battery level threshold and buffer size threshold for trustiness evaluation (Hossein *et al.* (2015)).

6.2.1.4 Data processing

There are three types of data processing in pervasive healthcare environment. Data processing pertains to the time when the data is evaluated for finding the trustiness of pervasive healthcare. If the data is processed immediately at the source node, then it is referred to real-time data processing. It enables online detection of data fault and quick decision making in gathered data. If the data is processed at the neighbor node or next hop gateway is referred to near real-time data processing. At the sink node, set of data is collected and processed in a periodic time interval. This type of processing is called batch data processing (Haron *et al.* (2017)).

6.2.1.5 Data trust types

Data trust is classified into three types. The source node evaluates its data for data fault detection. A trust score is assigned for data by the source node based on its confidence level. This type of trust is called self-data trust, and this process is referred to self-validation. Peer- node validation is the process of evaluating the confidence level of the source node data by neighbor node or next hop gateway node. The outcome of this process is peer data trust. The evaluation of trustiness of data by sink node from an application point of view is called global data trust (Karthik and Ananthanarayana (2017b)).

6.2.1.6 Trust computation

Trust computation is carried out at three different locations. A centralized trusted server does the trust computation to find out the trustiness of data and node is called centralized trust computation. If the individual sensor node evaluates the trustiness of data and trustiness of node themselves is called distributed trust computation. Sometimes it is also known as in-network detection or online detection. Hybrid trust computation is a combination of distributed and centralized trust computation ([Karthik and Ananthanarayana \(2017b\)](#)).

6.2.1.7 Trust Aggregation

Multiple trust evidences are used to evaluate the trustiness of node and data in pervasive healthcare. The trust evidence is collected at different locations such as source node, neighbor node, relay node, and sink node. The process of aggregating the multiple evidences from various nodes is called trust aggregation. A weighted sum is basic method of trust aggregation process where the trust evidence are multiplied with specific weight depends on its significance.

6.2.1.8 Trust representation

The trust score is used to represent the trustiness state of node and data. It ranges from -1 to +1, where trust score from -1 to -0.29 represents untrustworthiness, uncertain is represented from -0.3 to +0.29 and trustworthy state corresponds to the trust score from 0.3 to 1 ([Dhulipala et al. \(2013\)](#)).

6.2.1.9 Trust update

The trust score of the node and data must be updated regularly. We cannot represent the current status of data and node if the time interval for the update is too lengthy. So, the time interval for trust update should be periodic and should represent the current status of data and node without consuming many resource in the network ([Cho et al. \(2011\)](#)).

6.2.2 Data Faults and Data Loss

We study the following data related vulnerabilities to design the context-aware trust management scheme in pervasive healthcare.

Data faults: The data faults in MSN might occur due to battery depletion of the node,

untrustworthy sensor node, unreliable link and malicious attacks (Sharma *et al.* (2010)).

6.2.2.1 Types of Data faults

1. Out-of-range faults: Sensor data items that divert from the anticipated range of values (Sharma *et al.* (2010)), (Yu *et al.* (2015)).
2. Constant or struck-at faults: Sensor data items that remain constant or showing very little diversion for a certain period than anticipated (Yu *et al.* (2015)).
3. Data outliers: Sensor data items that divert from other data items but lies within the anticipated range (Sharma *et al.* (2010)).
4. Spike faults: The rate of change of data item over a certain period is higher than anticipated (Sharma *et al.* (2010)).

The data faults are detected with the help of data trust value ranges from -1 to +1. We refer to (Karthik and Ananthanarayana (2016)) for identifying the trust value for data faults, uncertain data, and normal data.

Data Loss: The data loss in MSN might happen by collision, untrustworthy link and malicious attacks (Kong *et al.* (2013)).

6.2.2.2 Patterns of data loss in sensor networks

State-of-the-art techniques like (Li *et al.* (2011)), (Zhu *et al.* (2009)) presume that the data loss in sensor network adopts a random distribution. This idea of data loss pattern does not apply to the medical sensor networks. We have four main types of data loss patterns (Kong *et al.* (2013)) in pervasive healthcare applications as shown in Figure 6.2, where 0 represents the data loss. They are:

1. Element random loss: Data values are neglected randomly. This is the simplest form of data loss pattern. The reason for this type of data loss is collision and noise [55].
2. Block random loss: Data values from neighboring nodes in contiguous time slot are neglected randomly. Congestion is the main reason for this type of data loss [55].
3. Element frequent loss in row: Data values in particular rows are dropped randomly. The untrustworthy link is the root cause for element frequent data loss in row [55].
4. Successive element loss in row: Data values of sensor node starts dropping from a particular time slot. This loss pattern happens when node is running out of battery power or physically damaged [55].

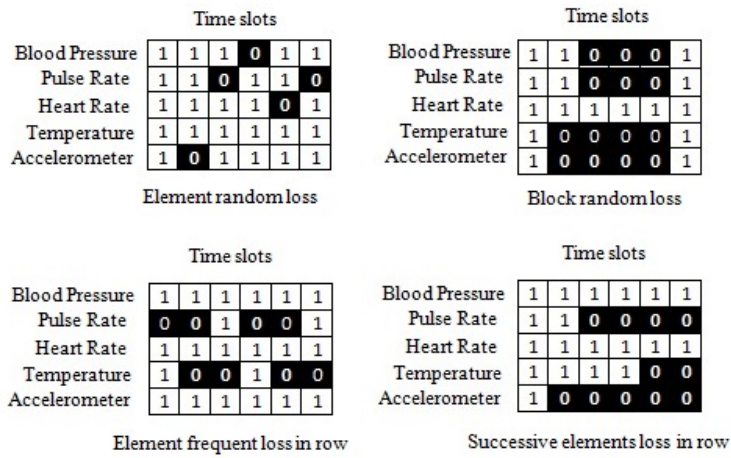


Figure 6.2: Data loss patterns

6.3 Proposed Trust Management Scheme for Pervasive Healthcare

This section introduces the proposed CATMS for pervasive healthcare. Consider an MSN which consists of five different physiological sensors deployed on human body to observe medical events. As shown in Figure 6.3, the gathered physiological data

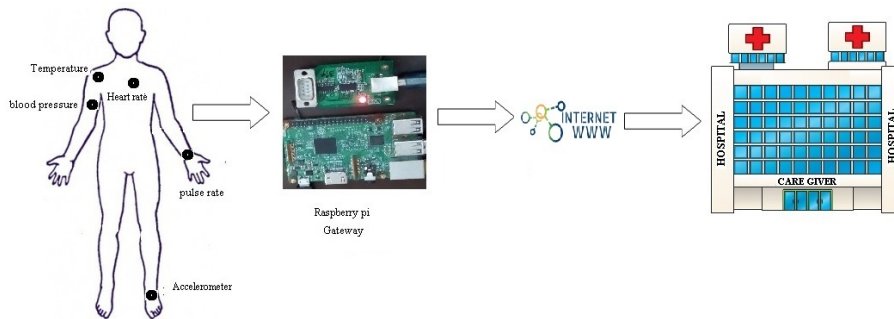


Figure 6.3: Pervasive healthcare system

are sent to a gateway which has internet connectivity for forwarding it to the hospital or caregiver for necessary action. Here Raspberry Pi is the intermediate node/gateway node for relaying the physiological data to the hospital using internet connectivity. The computer system in hospital is the sink node which normally does the data analysis process to detect medical event and for providing treatment to patient. Data collection is performed with the help of a medical sensor network prototype shown in Figure 6.4. It consists of heart rate sensor, pulse rate sensor, blood pressure sensor, body temperature, accelerometer and GPS. 250 users (215 patients and 35 healthy adults) are considered for the experiments. The medical events are Fever, Asthma, TB, malaria, cold and headache.

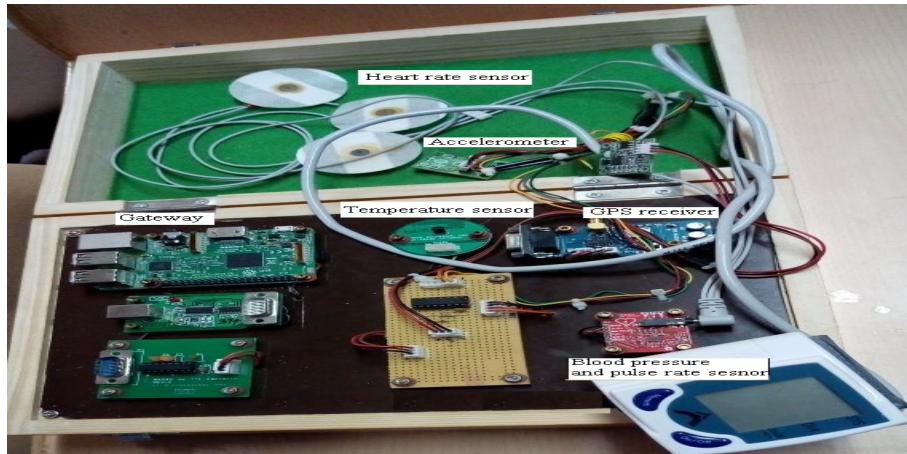


Figure 6.4: Medical sensor network prototype

6.3.1 Structure of CATMS

In this subsection, we explain the structure of CATMS. The proposed CATMS consists of three main procedures: Data fault detection, Data reconstruction, and Event detection as shown in Figure 6.5. It comprises of four modules: Node Trust Evaluation, Sensor self-validation, Peer node validation and Sink node validation. It also comprises of four entities namely: medical sensor node, physiological data, intermediate or gateway node and sink node. Medical sensor node generates physiological data, and it is forwarded to the sink node via an intermediate node or gateway.

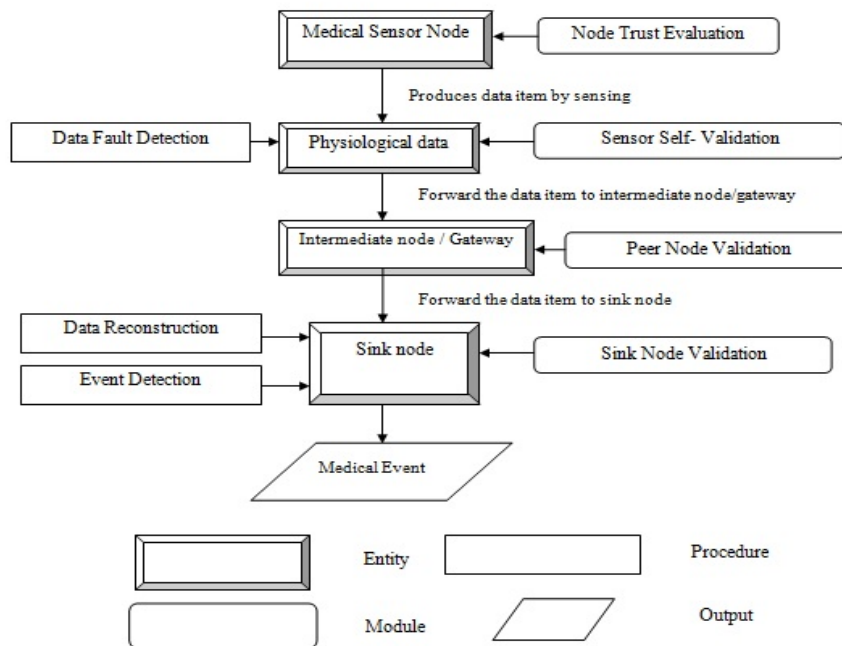


Figure 6.5: Structure of CATMS

6.3.1.1 Data Fault Detection

It is the process of detecting data faults in the gathered physiological data using heuristic functions, data correlation methods, and contextual information. Heuristic functions include investigating medical sensor node age, medical sensor node battery level, expected range of physiological data and expected rate of change of physiological data. To save the battery power of the medical sensor node and increase the accuracy of data fault detection method, hybrid method of data fault detection is followed which combines distributed and centralized detection. For distributed detection of data faults, node trust evaluation, sensor self-validation, and peer node validation modules are used. For centralized detection of data faults, sink node validation module is used. Peer node validation is actuated at an intermediate node/gateway node when gathered physiological data are spatially correlated to each other like heart rate and pulse rate. Node trust evaluation module is actuated at a medical sensor node only when the data trust score is extremely noisy or when the number of physiological data samples are not enough for determining data trust score.

6.3.1.2 Data Reconstruction

It is the process of predicting data and replacing faulty data, missing data with predicted data. Data fault and missing data should be identified and isolated before data analysis and medical diagnosis process. This process is employed at the sink node and it uses auto correlation, spatial correlation, multi-attribute correlation and contextual information for data prediction. Based on RMSE values from data prediction methods, sink node chooses the best data reconstruction scheme for replacing missing data and faulty data.

6.3.1.3 Event detection

It is the process of collecting data from MSN, replacing faulty and missing data with data fault detection and data reconstruction methods, recognizing the data pattern, mapping semantically with medical domain knowledge to detect events in pervasive health-care. Event detection is done at the sink node. When two or more physiological data shows abnormal value, then contextual information and accelerometer data are used for patient activity recognition. We use methods from ([Zhang *et al.* \(2015\)](#)), ([Mannini *et al.* \(2013\)](#)) for activity recognition using accelerometer data and contextual information. If

the activity of the patient is detected, and two or more physiological attributes have abnormal values, then medical attention or raising alarm for medical emergency is not required. If the patient is at rest and no activities are detected and two or more physiological attributes have abnormal values, then there is a need for medical attention or raising the alarm for a medical emergency. Based on the detected event, alarm/alert is given to caregiver or remote medical diagnosis is carried out.

6.3.1.4 Node trust evaluation

Node trust score is calculated from any one of three methods: direct trust; indirect trust and resource level trust.

Direct Trust (DT): Sliding window is used to calculate direct trust between the subject node (x) and object node (y). It represents the number of successful interactions (Sxy) and unsuccessful interactions (Uxy). DT shows communication capability of node and cooperativeness with other nodes. Direct trust score is calculated by using the equation (6.1).

$$DT = \frac{(Sxy - Uxy)}{(Sxy + Uxy)} \quad (6.1)$$

Indirect Trust (IDT): It is calculated by using equation (6.2) when there is no communication among subject node and object node or to combine opinions with direct trust score to find out total trust score.

$$IDT = (Tsz * Tzo) \quad (6.2)$$

Where Tsz is the trust score of Z by subject node S and Tzo is the opinion score about object node O by node Z .

Resource level Trust (RLT): RLT is required when there are no interactions between subject and object nodes or when direct and indirect trust score are extremely noisy. RLT score is calculated by using equation (6.3) by considering the available node resources such as Battery level (B), Buffer Size (BS), BandWidth (BW) and Waiting Queue (WQ) as suggested in (Hossein *et al.* (2015)). Threshold limits (Bth , $BWth$,

$BSth$, and $WQth$) are defined by domain experts.

$$RLT = \begin{cases} -1, & \text{data fault, if } (B < Bth \text{ or } BS < BSth \text{ or } BW < BWth \text{ or } WQ < WQth) \\ 0, & \text{uncertain, if } (B = Bth \text{ or } BS = BSth \text{ or } BW = BWth \text{ or } WQ = WQth) \\ +1, & \text{trustworthy data, if } (B > Bth \text{ or } BS > BSth \text{ or } BW > BWth \text{ or } WQ > WQth) \end{cases} \quad (6.3)$$

6.3.1.5 Sensor self- validation

Heuristic functions like checking the expected range of data values, checking the expected rate of change, checking the sensor battery level, inspecting the sensor node trust score and examining of sensor age are employed in the source node to check the trustiness of data. In addition to heuristic functions, temporal correlation-based data fault detection is employed in sensor node to detect the data fault locally. Outcome of this module is self-data trust. If physiological data is trustworthy, self-data trust score is embedded with data and is forwarded to the next hop or gateway. Otherwise, the faulty data is discarded without forwarding to next hop or gateway node.

Heuristic-based trust evaluation: The heuristic based data fault detection includes the process of examining of sensor age A , checking the battery level of sensor node B , checking the expected range of data items $(Xmin, Xmax)$ and examination of expected Rate of Change (RC) .

The age of sensor plays an important role in determining reliability of the node. Elements of the sensor node can be anticipated to degrade over time (Jiang *et al.* (2015)). For instance, the pulse rate sensor might produce untrustworthy data when it wears out over time. The data item is considered as a data fault when the sensor age is below threshold Ath as described in equation (6.4). The threshold value is selected based on the application and defined by experts.

$$HT1 = \begin{cases} -1, & \text{if sensor age} > Ath, \text{ data fault} \\ 0, & \text{if sensor age} = Ath, \text{ uncertain} \\ +1, & \text{if sensor age} < Ath, \text{ trustworthy data} \end{cases} \quad (6.4)$$

The probability of providing untrustworthy data is high when the sensor node battery is depleted (Jiang *et al.* (2015)). The data item is viewed as data fault when battery level of sensor node crosses threshold value Bth . The threshold value is chosen based on application. It is represented in equation (6.5).

$$HT2 = \begin{cases} -1, & \text{if battery level} > Bth, \text{ data fault} \\ 0, & \text{if battery level} = Bth, \text{ uncertain} \\ +1, & \text{if battery level} < Ath, \text{ trustworthy data} \end{cases} \quad (6.5)$$

The source node checks its generated data item with an expected range of values ($Xmin$, $Xmax$) (Jiang *et al.* (2015)). If the data item falls within the expected range of values, then it is trustworthy data item. Otherwise it is data fault as shown in equation (6.6). Domain experts define the expected range of values for all sensors. For example, the expected range of values for oxygen saturation ratio sensor are ($Xmin = 0\%$, $Xmax = 100\%$).

$$HT3 = \begin{cases} -1, & \text{if } Xmin > X > Xmax, \text{ data fault} \\ 0, & \text{if } Xi = Xmin \text{ or } Xi = Xmax, \text{ uncertain} \\ +1, & \text{if others, trustworthy data} \end{cases} \quad (6.6)$$

The source node checks its generated data with previous set of data items (Karthik and Ananthanarayana (2016)). If the difference between previous data item and current data item goes beyond expected rate of change threshold $RCth$ which is defined for all physiological sensors by domain experts, then it is data fault as depicted in equation (6.7). For example, the expected rate of change of body temperature would be two to four degree Celsius (Kelly (2006)), whereas the expected rate of change of blood pressure would be high if the person involved in activities like walking and jogging [59] [60].

$$HT4 = \begin{cases} -1, & \text{if expected rate of change} < RCth > \text{expected rate of change, data fault} \\ 0, & RCth \text{ is unavailable, uncertain} \\ +1, & \text{if expected rate of change} = RCth, \text{ trustworthy data} \end{cases} \quad (6.7)$$

Temporal correlation based trust evaluation: The sensor node employs time series model to predict the forthcoming data using the previous set of data items (Karthik and Anan-

thanarayana (2016)). After prediction of the data item, the sensor node compares the predicted data item PXi with actual generated data item Xi . If the difference of two data items moves over threshold value $TCth$, then the generated data item is considered as a data fault. The chosen threshold value varies from one application to other and also depends on observed phenomena. Temporal Correlation based Trust score calculation (TCT) is shown in equation (6.8).

$$TCT = \begin{cases} -1, & \text{if } |Xi - PXi| > TCth, \text{ data fault} \\ 0, & \text{if } |Xi - PXi| = TCth, \text{ uncertain} \\ +1, & \text{if } |Xi - PXi| < TCth, \text{ trustworthy data} \end{cases} \quad (6.8)$$

Data trust score is extremely noisy soon after network set up, initialization and when the number of gathered data items is less than 5, then sensor node checks the trust score of sensor data by evaluating the Node Trust (NT) as shown in equation (6.9). The trust score of sensor node is calculated from any one of the node trust evaluation methods: Direct Trust (DT) or Indirect Trust (IDT) or Resource Level Trust (RLT) using equations (6.1), (6.2) and (6.3). The correlation coefficient is very noisy with a limited number of data samples (Karthik and Ananthanarayana (2017b)). The trust score of a sensor ranges from -1 to +1, where -1 to -0.30 represents data fault, -0.29 to +0.29 represents uncertain state and trustworthiness of data represented by the score ranges from +0.30 to +1 (Dhulipala *et al.* (2013)).

$$NT = \begin{cases} -1, & \text{if trust score of sensor} < -0.3, \text{ data fault} \\ 0, & \text{if } -0.3 < \text{trust score of sensor} < 0.3, \text{ uncertain} \\ +1, & \text{if trust score of sensor} > 0.3, \text{ trustworthy data} \end{cases} \quad (6.9)$$

6.3.1.6 Peer node validation

In case of spatially correlated sensors like heart rate sensor and pulse rate sensor, the spatial correlation based data fault detection is used in the peer node validation module. The spatial correlation coefficient is used to predict the upcoming data and it is used for detecting the data fault. This spatial correlation based trust evaluation method is employed at next hop intermediate node or gateway node. The outcome of this module

is peer trust score. If physiological data is trustworthy, the peer trust score is embedded with it and forwarded to the sink node. Otherwise, the faulty data is discarded without forwarding to sink.

Spatial correlation based trust evaluation: In peer node validation, the peer node equates its data item with a data item of source node which observes the same phenomena to calculate the correlation coefficient (Karthik and Ananthanarayana (2016)). For instance, the data items from the pulse rate sensor and data items from the heart rate sensor are highly correlated even though they are deployed in different regions (Salem *et al.* (2014)). The correlation coefficient of data items from two different sensors SC is calculated by using equation (6.10). The data item PXi is predicted using a spatial correlation coefficient and compared with actual data item Xi . If the difference of two data items moves over threshold value $SCth$, then the generated data item is conceived as a data fault as shown in equation (6.11). The chosen threshold value varies from one application to other and also depends on observed phenomena. When the number of data items is less than 5, then the data trust score is calculated by Initial Trust (IT) using equations (6.12) and (6.13) as suggested in (Karthik and Ananthanarayana (2016)), since the limited number of samples give highly noisy correlation coefficient.

$$SC = \frac{\sum_{i=1}^n (Xi - \bar{X})(Yi - \bar{Y})}{\sqrt{\sum_{i=1}^n (Xi - \bar{X})^2 (Yi - \bar{Y})^2}} \quad (6.10)$$

$$SCT = \begin{cases} -1, & \text{if } |Xi - PXi| > SCth, \text{ data fault} \\ 0, & \text{if } |Xi - PXi| = SCth, \text{ uncertain} \\ +1, & \text{if } |Xi - PXi| < SCth, \text{ trustworthy data} \end{cases} \quad (6.11)$$

$$IT = \sum_{i=1}^n \frac{1}{n} \frac{1}{1 + |Xi - Yi|} \quad (6.12)$$

$$SCT = \begin{cases} -1, & \text{if } IT < -0.3, \text{ data fault} \\ 0, & \text{if } -0.3 < IT < 0.3, \text{ uncertain} \\ +1, & \text{if } IT > 0.3, \text{ trustworthy data} \end{cases} \quad (6.13)$$

6.3.1.7 Sink node validation

Sink node uses multi-attribute correlation and contextual information to detect data fault, data reconstruction, and event detection. It receives physiological data with self data trust and peer data trust. Sink node replaces the faulty data and missing data with predicted data using multi-attribute correlation based on RMSE value. Outcome of this module is detection of data fault and medical event.

Multi-attribute correlation based trust evaluation: Sink node uses temporal correlation, spatial correlation and multi-attribute correlation-based data fault detection methods to have global view about observed physiological sensor data for detecting the data fault. Multi-Attribute Correlation (*MAC*) among several physiological data is calculated. The data item PXi is predicted using multi-attribute correlation coefficient and compared with actual data item Xi . If the difference of two data items moves over threshold value $MACth$, then the generated data item is viewed as a data fault as shown in equation (6.15). For instance, the multi-attribute correlation coefficient of three variables (xyz) is calculated by using equation (6.14) where r_{xy} represents the correlation between two variables.

$$MAC = \frac{\sqrt{r_{xz}^2 + r_{yz}^2 - 2r_{xz}r_{yz}r_{xy}}}{1 - r_{xy}^2} \quad (6.14)$$

$$MACT = \begin{cases} -1, & \text{if } |Xi - PXi| > MACth, \text{ data fault} \\ 0, & \text{if } |Xi - PXi| = MACth, \text{ uncertain} \\ +1, & \text{if } |Xi - PXi| < MACth, \text{ trustworthy data} \end{cases} \quad (6.15)$$

The centralized anomaly detection ([Wittenburg et al. \(2012\)](#)) is used for medical Event Detection (*ED*) by observing the changes in medical data. This procedure is actuated

when there is a change in data pattern in two or more physiological sensor data. We use methods from (Zhang *et al.* (2015)), (Mannini *et al.* (2013)) for activity recognition using accelerometer data and contextual information. Medical event is detected by observing the changes in data pattern and patient activities. Patient activity results in change of location (dynamic) and accelerometer (motion) data. In some cases, the patient might involve in basic strenuous exercises (Oh *et al.* (2016)) without changing location (static) also results in activity. If the patient is involved in activity which results in changes of data pattern of physiological attributes, then there is no need for Medical Attention (*MA*). If there is no activity and there is a change of data pattern of two or more physiological attributes, then there would be an event as shown in equation (6.16), and sink node raises an emergency alarm as shown in equation (6.17) .

$$ED = \begin{cases} \text{medical event, if}(\text{location} = \text{static and no motion is detected}) \\ \text{patient activity, if}(\text{location} = \text{dynamic or motion is detected}) \end{cases} \quad (6.16)$$

$$MA = \begin{cases} \text{no alarm, if}(ED = \text{patient activity}) \\ \text{alarm, if}(ED = \text{medical event}) \end{cases} \quad (6.17)$$

6.3.2 Algorithms for Trust Evaluation

In this subsection, the algorithms for trust evaluation are presented. The heuristic functions based trust evaluation is introduced in Algorithm 6.1.

Algorithm 6.1: Heuristic approach based trust evaluation

Input: physiological data of patient, battery level of sensor node B, expected range of change RC, expected range of data Xmin, Xmax, sensor age A, sensor buffer size BS, node bandwidth BW, waiting queue status WQ, battery threshold value Bth, sensor age threshold value Ath, number of interactions between nodes i, buffer size threshold value BSth, node bandwidth threshold value BWth, Waiting queue threshold value WQth.

Output: condition of physiological data

- 1: Compute HT1 using equation (6.4)
- 2: if HT1 > 0 then

```

3:     compute HT2 using equation (6.5)
4:     else condition of physiological data= data fault
5:     exit
6: end if
7: if HT2>0 then
8:     compute HT3 using equation (6.6)
9:     else condition of physiological data= data fault
10:    exit
11: end if
12: if HT3>0 then
13:    compute HT4 using equation (6.7)
14:    else condition of physiological data= data fault
15:    exit
16: end if
17: if HT4>0 then
18:    condition of physiological data= trustworthy data
19:    else condition of physiological data= data fault
20: end if
21: return condition of physiological data

```

In Algorithm 6.2, the temporal correlation based trust evaluation is introduced.

Algorithm 6.2: Temporal correlation based trust evaluation

Input: physiological data, number of data items i

Output: condition of physiological data

```

1: if( $i \leq 5$ ) then
2:     compute NT using equation (6.9)
3:     if ( $NT > 0$ ) then
4:         condition of physiological data= trustworthy data
5:         else condition of physiological data= data fault
6:     end if

```

```

7: end if
8: if(i>5) then
9:     compute TCT using equation (6.8)
10:    if(TCT>0) then
11:        condition of physiological data= trustworthy data
12:    else condition of physiological data= data fault
13:    end if
14: end if
15: return condition of physiological data

```

The spatial correlation based trust evaluation is presented in Algorithm 6.3.

Algorithm 6.3: Spatial correlation based trust evaluation

Input: spatially correlated physiological data (pulse rate and heart rate data), number of data items i

Output: condition of physiological data

```

1: if (i<=5) then
2:     compute IT using equation (6.12)
3:     if(IT>0) then
4:         condition of physiological data= trustworthy data
5:     else condition of physiological data= data fault
6:     end if
7: end if
8: if (i>5) then
9:     compute SCT using equation (6.11)
10:    if(SCT>0) then
11:        condition of physiological data= trustworthy data
12:    else condition of physiological data= data fault
13:    end if
14: end if
15: return condition of physiological data

```

According to (Ravichandran and Arulappan (2013)), no single algorithm in data fault detection is ideal for detecting all kinds of data faults and also recommends that two or more algorithms can be employed in succession to improve the accuracy of data fault detection. Therefore we combine Algorithms 6.2 and 6.3 with multi-attribute correlation method to form Algorithm 6.4. We combine Algorithms 6.1, 6.2, 6.3 and 6.4 with contextual information to form Algorithm 6.5 to improve the accuracy of data fault detection.

Algorithm 6.4 is the sequence of methods from Algorithms 6.2, 6.3 and multi attribute correlation based trust evaluation are used for data fault detection.

Algorithm 6.4: Multi-attribute correlation based trust evaluation (combination of Algorithms 6.2, 6.3 and multi-attribute correlation)

Input: physiological data, number of data items i , $\text{count1}=\text{count2}=0$.

Output: condition of physiological data, medical attention status

```
// Step 1: Temporal correlation based trust evaluation
1: if( $i \leq 5$ ) then
2:     compute NT using equation (6.9)
3:     if ( $NT > 0$ ) then
4:         condition of physiological data= trustworthy data
5:         else condition of physiological data= data fault
6:         count1++
7:     end if
8: end if
9: if( $i > 5$ ) then
10:    compute TCT using equation (6.8)
11:    if( $TCT > 0$ ) then
12:        condition of physiological data= trustworthy data
13:        else condition of physiological data= data fault
14:    count1++
15:    end if
```

```

16: end if
// Step 2: Spatial correlation based trust evaluation
17: if (i<=5) then
18:     compute IT using equation (6.12)
19:     if(IT>0) then
20:         condition of physiological data= trustworthy data
21:     else condition of physiological data= data fault
22:         count2++
23:     end if
24: end if
25: if (i>5) then
26:     compute SCT using equation (6.11)
27:     if(SCT>0) then
28:         condition of physiological data= trustworthy data
29:     else condition of physiological data= data fault
30:         count2++
31:     end if
32: end if
// Step 3: Attribute correlation based trust evaluation
33: compute MCC using equation (6.14)
34: compute MACT using equation (6.15)
35: if(MACT>0) then
36:     condition of physiological data= trustworthy data
37: else condition of physiological data= data fault
38:     count2++
39: end if
40: if count1&count2>=2 then
41:     medical event is detected
42:     medical attention status= emergency alarm
43: end if
44: return condition of physiological data and medical attention status

```

The Algorithm 6.5 is the sequence of Algorithm 6.1, 6.2, 6.3, 6.4 and contextual information based data fault detection.

Algorithm 6.5: Proposed Algorithm (combination of algorithms 6.1, 6.2, 6.3 and 6.4 with contextual information based detection)

Input: physiological data, battery status of sensor node B, expected rate of change RC, expected range of data Xmin, Xmax, sensor trust value S, sensor age A, sensor buffer size BS, node bandwidth BW, waiting queue status WQ, battery threshold value Bth, sensor age threshold value Ath, number of interactions between nodes i, buffer size threshold value BSth, node bandwidth threshold value BWth, Waiting queue threshold value WQth, contextual information, count1=count2=count3=0

Output: condition of physiological data, medical attention status

//Step 1: Heuristic approach

```
1: Compute HT1 using equation (6.4)
2: if HT1>0 then
3:     compute HT2 using equation (6.5)
4:     else condition of physiological data= data fault
5:     count1++
6: end if
7: if HT2>0 then
8:     compute HT3 using equation (6.6)
9:     else condition of physiological data= data fault
10    count1++
11: end if
12: if HT3>0 then
13:     compute HT4 using equation (6.7)
14:     else condition of physiological data= data fault
15:     count1++
16: end if
17: if HT>0 then
```

```

18:     condition of physiological data= trustworthy data
19:     else condition of physiological data= data fault
20:     count1++
21: end if
22: return condition of physiological data
// Step 2: Temporal correlation based trust evaluation
23: if(i<=5) then
24:     compute NT using equation (6.9)
25:     if (NT>0) then
26:         condition of physiological data= trustworthy data
27:         else condition of physiological data= data fault
28:         count2++
29:     end if
30: end if
31: if(i>5) then
32:     compute TCT using equation (6.8)
33:     if(TCT>0) then
34:         condition of physiological data= trustworthy data
35:         else condition of physiological data= data fault
36:         count2++
37:     end if
38: end if
// Step 3: Spatial correlation based trust evaluation
39: if (i<=5) then
40:     compute IT using equation (6.12)
41: if(IT>0) then
42:     condition of physiological data= trustworthy data
43:     else condition of physiological data= data fault
44:     count3++
45:     end if
46: end if
47: if (i>5) then
48:     compute SCT using equation (6.11)

```

```

49:     if(SCT>0) then
50:         condition of physiological data= trustworthy data
51:     else condition of physiological data= data fault
52:         count3++
53:     end if
54: end if
// Step 4: Attribute correlation based trust evaluation
55: compute MCC using equation (6.14)
56: compute MACT using equation (6.15)
57: if(MACT>0) then
58:     condition of physiological data= trustworthy data
59: else condition of physiological data= data fault
60:     count3++
61: end if
// Step 5: Contextual information based detection
62: if count1&count2&count3>=2 then
63:     if(location information= changing or accelerometer= motion)then
64:         patient activity is detected
65:         medical attention status= not required
66:     end if
67:     if(location information=constant and accelerometer= motion) then
68:         patient activity is detected
69:         medical attention status= not required
70:     end if
71: else medical event is detected
72: medical attention status= emergency alarm
73: end if
74: return condition of physiological data and medical attention status

```

6.3.3 Steps required for evaluation of trust in CATMS

The following steps are required for the evaluation of trust in CATMS:

1. Medical sensor node gathers data from the patient and evaluates the trustiness of collected data using Algorithm 6.1 and 6.2.

2. The gateway or intermediate node receives data item and evaluates the trustiness of collected data using Algorithm 6.3, only if the data items are spatially correlated to each other (for instance, heart rate and pulse rate). Otherwise, it forwards the data item to the sink node.
3. Sink node receives the data item and evaluates the trustiness of data item using Algorithm 6.4 and 6.5.

6.4 Results and Discussions

This section discusses the results of CATMS for pervasive healthcare. We have analyzed proposed method on three categories: data fault detection; data reconstruction; and event detection.

6.4.1 Data fault detection analysis

Heuristic functions, temporal correlation, spatial correlation, multi-attribute correlation, and context-aware data fault detection methods are employed for data fault detection.

Algorithm 6.1 is employed at low-end sensor nodes for sensor self validation. Heuristic functions are devised for battery status checking of medical sensor nodes, expected range of values for sensors, expected rate of change of vital signs and sensor age to check data fault. The sensor age threshold, expected range of values and expected rate of change of medical sensor values are defined by domain experts.

Algorithm 6.2 is employed at low-end sensor nodes for sensor self-validation. Temporal correlation/ autocorrelation is utilized in Algorithm 6.2 to find data fault. Future physiological sensor data is predicted based on historical data series. If the difference between predicted value and actual observed value is greater than the threshold limit, then the particular data item is likely data fault. The threshold value is defined by domain experts (doctors). Wrong threshold will affect the overall performance of the system.

Algorithm 6.3 is employed at the source sensor node and neighbor node for sensor self-validation where data faults are detected by using spatial correlation algorithm. According to (Salem *et al.* (2014)), (Salem *et al.* (2013)), heart rate sensor and pulse rate sensor are different sensor but they are producing the same value. Always they are correlated to each other. Let X_i and Y_j be data items reported by heart rate and pulse rate sensors respectively. Estimate heart rate data PX_{i+1} based on Y_{j+1} data observed by pulse rate sensor. If the difference between the predicted value and actual

observed value is greater than the threshold limit which is defined by a domain expert, then the data item is likely to be faulty.

Multi-attribute correlation is used in Algorithm 6.4 to detect data fault. It is a combination of Algorithm 6.2 and 6.3 used by sink node for sink node validation and detection of data fault. Apart from temporal and spatial correlation, multi-attribute correlation is used in this algorithm by sink node to have a global view about observed physiological sensor data and to detect the data fault. From (Salem *et al.* (2014)), we know that measured attributes are correlated to each other. When there is an event, it affects the physiological sensor values equally, and when there is a data fault, it won't follow that trend of correlation between sensor data. By using the correlation coefficient among multi-attributes, we can detect the data fault. If more than two physiological sensor values are not following the trend and seasonality, then there would be an event.

In Algorithm 6.5, the sequence of heuristic functions, temporal correlation, spatial correlation, multi-attribute correlation and contextual information of patient are applied to detect data faults. The contextual information and accelerometer values are used to identify patient activity (Mannini *et al.* (2013)). There is a possibility that patient physiological sensors show abnormal values when patient involved in activity like jogging, running or walking (Oh *et al.* (2016)). Initially, it checks for abnormal values of sensors. If two or more sensors show abnormal value, then algorithm checks for the activity of patient using accelerometer and location information. If the patient involved in any activity, then medical attention is not required. Otherwise, medical attention is required. Algorithms are evaluated using real physiological sensor data set (25000 data samples) of pervasive healthcare prototype which is shown in Figure 6.4 for remote patient monitoring injected with various faulty data like constant, outliers, and spike and out-of-range faults. 250 users (215 patients and 35 healthy adults) are considered for the experiments. The medical events are Fever, Asthma, TB, malaria, cold and headache. The expected range of values for physiological sensors, expected rate of change, the threshold limit for various node attributes, temporal, spatial and multi-attribute correlations algorithms are defined by domain experts (doctors). We manually injected the data faults for evaluating the algorithms with the following cases. Performance of proposed approach is evaluated with the help of real data samples and compared with normal TMS without context. Detection accuracy is used as metric for finding the performance of various algorithms.

1. Case 1: Dataset with 20% of constant faults
2. Case 2: Dataset with 30% of out-of-range faults
3. Case 3: Dataset with 40% outliers
4. Case 4: Dataset with 30% spike faults
5. Case 5: Dataset with 10% constant faults, 10% out-of-range faults, 10% outliers, 10% spike faults

Detection accuracy is the ratio of the number of detected data faults to the total number of actual data faults. Figure 6.6 shows the performance of Algorithm 6.1 from case 1 to case 5. From Figure 6.6, we can observe that Algorithm 6.1 works well in detecting out-of-range data faults and spike faults. Heuristic functions are not suitable for detecting constant faults and outliers.

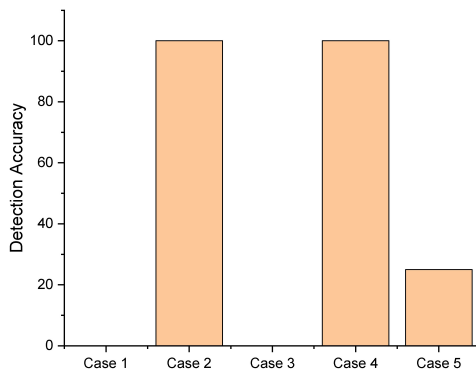


Figure 6.6: Performance of Algorithm 6.1

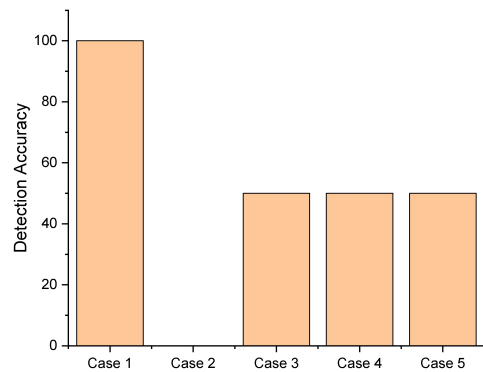


Figure 6.7: Performance of Algorithm 6.2

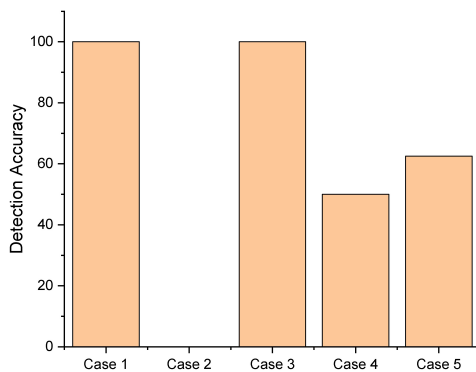


Figure 6.8: Performance of Algorithm 6.3

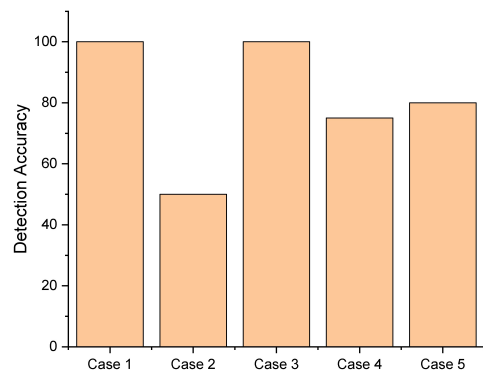


Figure 6.9: Performance of Algorithm 6.4

Figure 6.7 shows the performance of Algorithm 6.2 from case 1 to case 5. The result exhibits that detection accuracy for case 1 is 100%, case 2 is 0%, case 3 is 50%, case

4 is 50% and case 5 is 50%. Algorithm 6.2 detects constant fault and partially detects outliers and spike faults.

The performance of Algorithm 6.3 is depicted in Figure 6.8 from case 1 to case 5. Figure 6.8 show that Algorithm 6.3 is effective to defect the constant faults, and outliers. The detection accuracy for case 1 is 100%, case 2 is 0%, case 3 is 100%, case 4 is 50% and case 5 is 62.5%.

Algorithm 6.4 is introduced with a combination of Algorithms 6.2 and 6.3 with multi-attribute correlation-based data fault detection. Figure 6.9 shows the performance of Algorithm 6.4 for all cases. Detection accuracy of Algorithm 6.4 is 100% in case 1, 50% in case 2, 100% in case 3, 75% in case 4 and 80% in case 5. To overcome the disadvantages of other Algorithms 6.1, 6.2, 6.3, and 6.4, Algorithm 6.5 is proposed for data fault detection. It is the combination of Algorithms 6.1, 6.2, 6.3 and 6.4 with contextual information for data fault detection. The performance of Algorithm 6.5 is shown in Figure 6.10. It works well in detecting all types of faults and 100% detection accuracy is achieved for all cases. The sequence of multiple methods is employed in Algorithm 6.5. Because of multiple methods for detection of faulty data, the cost for computation is very high. However it is compensated with the cost of communication. Communication of data faults consumes more energy than computation. Due to early detection of data faults, the communication of data faults is avoided.

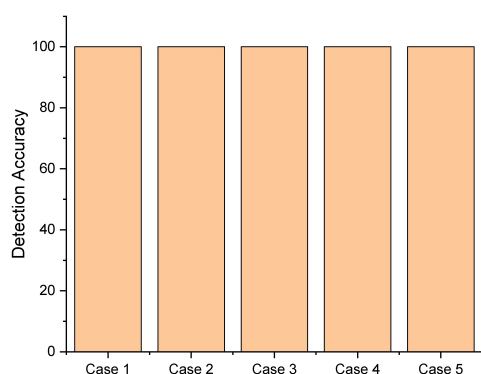


Figure 6.10: Performance of Algorithm 6.5

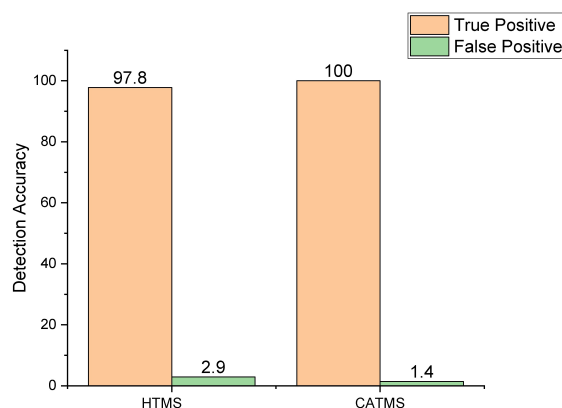


Figure 6.11: Comparison of TMS

The performance of the proposed CATMS is compared with HTMS (Karthik and Ananthanarayana (2017b)). HTMS is chosen for comparison purpose because it is one of the recent methods for detecting data faults using time series analysis, spatial analysis, and data provenance methods. HTMS achieves 97.8% detection accuracy in rightly classi-

ifying the data as faulty data and 2.9% of normal data is wrongly classified as data fault as shown in Figure 6.11. The proposed method CATMS achieves 100% detection accuracy in rightly classifying the data as faulty data and 1.4% of normal data is wrongly classified as a data fault. The combination of heuristic methods, correlation-based detection methods with contextual information are used in proposed CATMS to improve the detection accuracy and reduce the False Positive Rate (FRR). The proposed method CATMS maintains acceptable FPR and outperforms existing HTMS convincingly.

6.4.2 Data reconstruction analysis

This subsection discusses the results of the data reconstruction scheme in MSN based on multi-attribute correlation and contextual information to improve the accuracy of event detection in pervasive healthcare. We consider a scenario of physiological sensors deployed over the human body for measuring Blood Pressure (BP), Heart Rate (HR), Pulse Rate (PR), Body Temperature (BT) and accelerometer. We also consider that all measured parameters are free from data faults to expect missing data. To evaluate the proposed data reconstruction algorithm based on multi-attribute correlation and contextual information, Root Mean Square Error (RMSE) is used (Karthik and Ananthanarayana (2016)), (Kong *et al.* (2013)). RMSE is the difference between the esti-

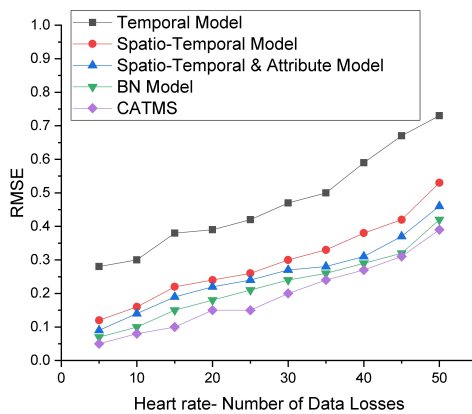


Figure 6.12: Data reconstruction of HR

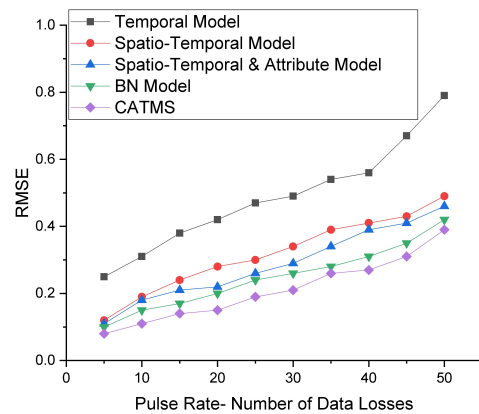


Figure 6.13: Data reconstruction of PR

mated value and actual observed value. Physiological sensor dataset (25000 samples) of pervasive healthcare prototype for remote patient monitoring is utilized for performance evaluation. Since the original dataset does not contain any type of data losses, we manually introduce various types of data loss in the dataset. A better data reconstruction for data loss should have a smaller RMSE value.

The proposed algorithm for data reconstruction undergoes six steps: 1) we first calculate auto correlation coefficient or serial correlation coefficient of physiological data and estimate the data value for data loss using correlation coefficient. Then we calculate the RMSE value. 2) We calculate spatio- temporal correlation coefficient for spatially correlated values like heart rate and pulse rate. Using spatio-temporal correlation coefficient [68], estimate the missing data. Then calculate the RMSE value. 3) We calculate spatio-temporal and attribute correlation coefficient of physiological data and estimate the missing data. RMSE value for estimated data is calculated. 4) we use bayesian network based method (Zhang *et al.* (2018)) to reconstruct data and calculate RMSE value. 5) Use proposed method CATMS to estimate the missing data and calculate RMSE. 6) Finally, we compare the error rates from five different steps. Choose the lowest RMSE for better data reconstruction scheme. The random data loss is chosen for basic comparison of different data reconstruction schemes. For comparison, the RMSE value is plotted in Y axis and X axis represents data loss from 5% to 50% with the increment of 5%. Figure 6.12, 6.13, 6.14 and 6.15 shows the performance of

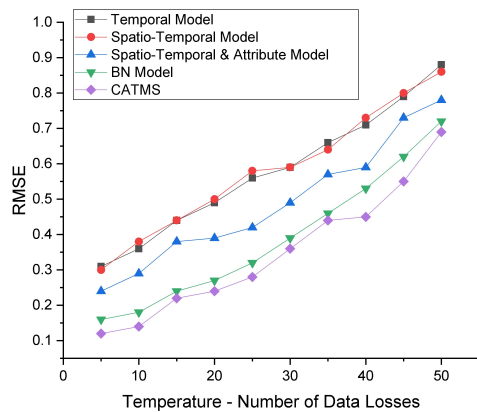


Figure 6.14: Data reconstruction of BT

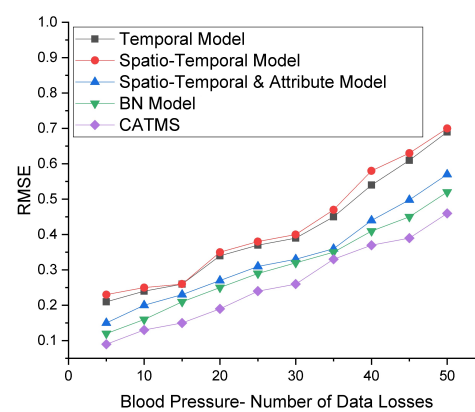


Figure 6.15: Data reconstruction of BP

various data reconstruction schemes for physiological sensor data. When there is an increase in data loss rate, there is an increase in error rate in all schemes. However, the proposed schemes achieve better RMSE value for physiological data loss ranges from 5% to 50% with an increment of 5%. When we increase the number of data loss, the RMSE value is linearly increasing for all attributes.

6.4.2.1 Data reconstruction for data loss patterns

In this subsection, the performance of data reconstruction for data loss patterns are given. Figure 6.16 illustrates the histogram comparison of four algorithms of data reconstruction schemes in Element Random Loss (ERL). The total data loss is 50%. We vary the data loss pattern from 0% to 15% for different attributes. Result shows

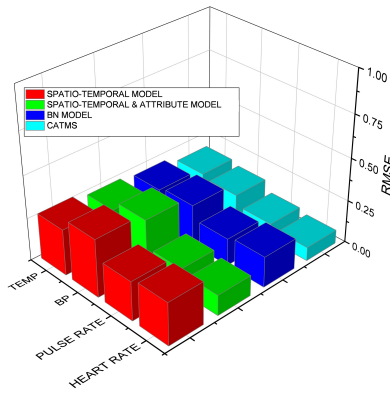


Figure 6.16: Error rate of ERL

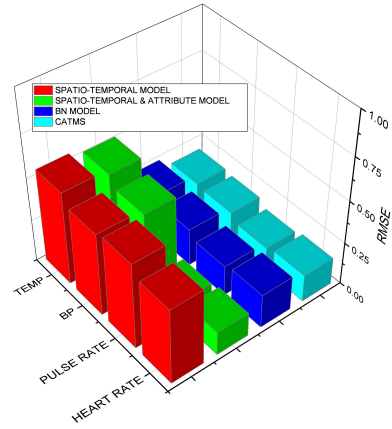


Figure 6.17: Error rate of EFL in row

that the proposed CATMS achieves better RMSE value when compared to Spatio-Temporal (ST) model ([Zhang *et al.* \(2016b\)](#)), Spatio-Temporal& Attribute (STA) model ([Karthik and Ananthanarayana \(2017c\)](#)) and Bayesian Network (BN) model ([Zhang *et al.* \(2018\)](#)). In Element Frequent Loss (EFL) in row pattern, the total data loss is 40%. The rows are chosen randomly for attributes, and we vary the data loss pattern from 0 to 15% for different attributes. BN model and proposed model CATMS works well for this type of data loss. However, proposed model outperforms other existing models in terms of RMSE value as shown in Figure 6.17. Figure 6.18 shows the performance of data reconstruction schemes for Block Random Loss (BRL). We vary the data loss pattern from 0% to 10 % for different attributes. The total data loss in block random loss is 40%. The proposed method CATMS has better RMSE value when compared to other models. The performance of data reconstruction schemes for Successive Element Loss(SEL) in row is depicted in Figure 6.19. The total data loss is 60%. We vary the data loss pattern from 0% to 15% for different attributes. Even in successive element loss, proposed model works better than other models.

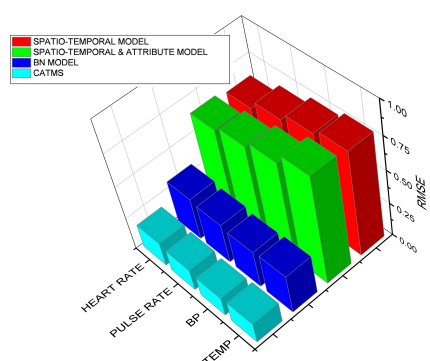


Figure 6.18: Error rate of BRL

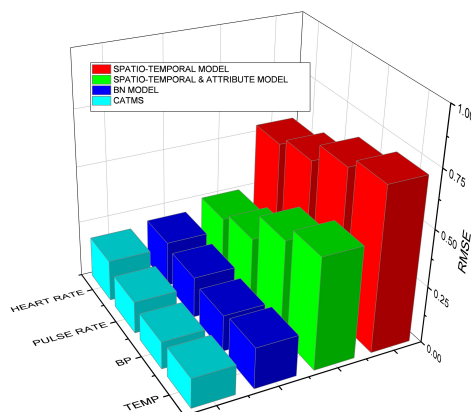


Figure 6.19: Error rate of SEL

6.4.3 Event detection analysis

In this subsection, we discuss the performance of the different event detection algorithm of remote patient monitoring in pervasive healthcare. We used detection accuracy, false positive as metrics to find the performance of various algorithms in event detection (Wittenburg *et al.* (2012)). Detection accuracy is the ratio of a total number of detected events to the total number of actual events. False positive rate (FPR) is the ratio of a total number of negative events classified wrongly as positive to the total number of actual negative events.

We consider a scenario of physiological sensors deployed over the human body for

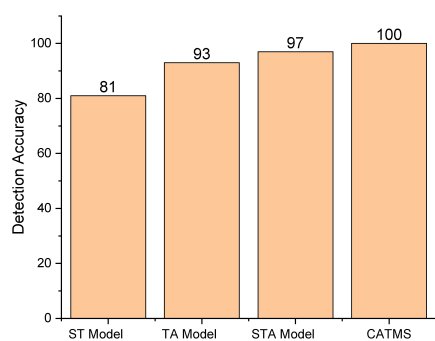


Figure 6.20: Event Detection accuracy

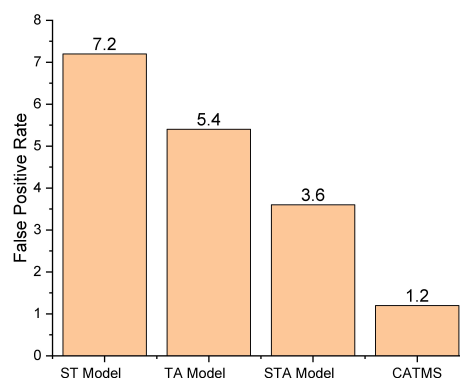


Figure 6.21: FPR – Event detection

measuring blood pressure, heart rate, pulse rate, body temperature and accelerometer. We manually inserted 20 medical events in dataset. We also consider that all measured parameters are free from data faults and missing data. For performance evaluation,

we compared proposed CATMS with Spatio-Temporal model (ST Model) (Karthik and Ananthanarayana (2017a)), Temporal-Attribute (TA Model) (Karthik and Ananthanarayana (2017c)) and Spatio-Temporal multi-Attribute models (STA Model) (Wang *et al.* (2017)) as shown in Figure 6.20. Results show that the ST model achieves 81% accuracy, TA model achieves 93% detection accuracy, STA model achieves 97% detection accuracy and proposed model CATMS achieves 100% detection accuracy. The use of contextual information, accelerometer data with multi-attribute correlation in CATMS is the root cause for achieving better results than other methods. In false positive rate, ST model has 7.2%, TA model gets 5.4%, STA model has 3.6%, and proposed model achieves as low as 1.2% FPR as shown in Figure 6.21. From the results of event detection accuracy and false positive rate, we conclude that the proposed model outperforms other existing models convincingly.

6.5 Summary

The proposed context-aware trust management scheme integrates heuristic functions, data correlation and contextual information based algorithms for identifying real medical emergencies. The proposed approach works with data correlation and contextual information based algorithms for identifying data faults, for data reconstruction of data faults and data losses and for distinguishing faulty data and clinical emergencies. We evaluated our proposed scheme on real data samples injected with various synthetic data faults and data losses. The proposed scheme is capable of online detection of data fault and reconstruction of data faults and data losses for reliable event detection. Experimental results prove that the effectiveness of context-aware trust management scheme in detecting data faults, and differentiating medical emergencies from sensor data faults and node misbehavior. The proposed CATMS for pervasive healthcare is mainly depends on rule-based heuristic methods. Domain knowledge is required to frame the rules and also it requires threshold and parameter settings. A bit of human involvement is required during system initialization to achieve effective results. However the proposed solution can be applied pervasively after system initialization. Statistical techniques are used in the data reconstruction methods without considering the underlay distribution of the input data. Still we achieved reasonable accuracy in data reconstruction techniques and outperformed state-of-the-art techniques. A test bed is implemented to test the proposed approach practicality and we achieved better results.

Chapter 7

Upper Ontology and Hybrid Ontology Matching for Pervasive Applications

7.1 Preamble

In this chapter, we address the fifth research objective, construction of upper ontology and hybrid ontology matching technique for integrating trusted context-aware sensor-driven pervasive applications. Pervasive environments include sensors, actuators, hand-held devices, set of protocols and services. The specialty of this environment is its power to manage with any device at any time anywhere and work autonomously for providing customized services to the user. The different entities of the pervasive environment collaborate with each other to accomplish an objective by sharing data among them. It raises an interesting problem called semantic heterogeneity. To address this problem, a hybrid ontology matching technique which combines direct and indirect matching techniques is proposed. To share and integrate data semantically, ontology matching technique establishes a semantic correspondence among various entities of pervasive application ontologies. To find the efficiency of the proposed approach, we carried out set of experiments with real-world pervasive applications. Experimental results prove that the proposed approach shows excellent performance in hybrid ontology matching. Results also proved that the use of background knowledge has influences over the performance of ontology matching technique.

The main contributions of this chapter include: i) we propose an upper ontology for pervasive environments with trust mechanism to deal with faulty data, missing data of various entities; ii) we propose a hybrid ontology matching technique which combines context, instance and upper ontology with trust mechanism for ontology alignment and iii) we tested our approach with four different ontology matching tasks of pervasive environments.

Rest of this chapter is organized as follows: The upper ontology for pervasive environments with trust mechanism is introduced in section 7.2. Section 7.3 explains direct and indirect ontology matching. The proposed hybrid ontology matching is discussed in section 7.4. Results and discussions are given in section 7.5. Section 7.6 gives a

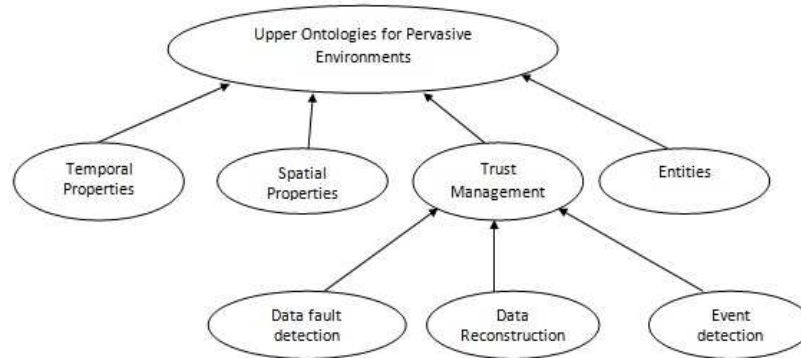


Figure 7.1: Upper Ontology for Pervasive Environments

summary about this chapter.

7.2 Upper Ontology for Pervasive Environments

In this section, we represent the upper ontology of pervasive environments. In general, upper ontology describes high-level concepts which are not restricted to single application or domain. Usually, it contains the core concepts and set of requirements which are independent of any pervasive domains. We listed out four main concepts for upper ontology of pervasive environments. They are temporal properties, spatial properties, entities, and trust management as shown in Figure 7.1. There are five temporal properties of data which are generated from pervasive environments. They are rate of change of data, data generation time, data validity time, sampling time and temporal dimension. Rate of change of data describes the rate at which dynamic data changes per time. Data generation time describes the time at which the data is generated from the source. Data validity time explains about the validity of data. The rate at which the data is collected or sampled from the environment is called sampling time. The generated data may represent either past, current or future state of environment is referred to temporal dimension.

There are two ways to represent spatial information in pervasive environment. They are physical location coordinate representation and symbolic location representation. Physical location coordinates are generated by GPS either in 2D or 3D. The symbolic locations are human-friendly location names like conference room and seminar hall. Entities in pervasive environment refer to data sources (sensors, users) and actuators. Trust management in pervasive environment refers to the process of finding the trustiness of data, devices, and user. It also includes trust based data fault detection, trust based data

reconstruction and trust-based event detection (Karthik and Ananthanarayana (2017a)). The temporal and spatial properties of data are used in trust-based data fault detection, data reconstruction and event detection. We borrow methods from (Dhulipala *et al.* (2013)), (Karthik and Ananthanarayana (2017b)) to find the trustiness of instance and entities of ontologies.

7.3 Direct and Indirect Ontology Matching

The growth of information technology has led to the use of data sharing among applications, devices, or different systems. Each system has different use of terms to symbolize the same information. The likeness in these terms is required, so that the data can be incorporated. Ontology matching is a way to balance the use of terms in data sources. In the context of pervasive environment, applications reach its full potential when data is collected by multiple sources and shared among them. For example, a pervasive google map application reaches its full potential when it collects data from environmental and traffic system to predict the traffic and pollution free route. The increase of knowledge representation is needed in the process of data exchange between machines (or applications). There are mainly two types of ontology matching for data sharing: direct and indirect ontology matching. In this section we discuss about two different types of ontology matching for data sharing. Direct matching process uses multiple ontology architecture to find the set of correspondence among concepts. In the indirect matching process, the global shared vocabulary is used as background knowledge for finding semantic correspondence among various concepts.

Direct ontology matching is restricted to matching named entities (i.e., entities, instances) between ontologies. It is also called one-to-one ontology matching because, in case of matching two ontologies, it matches one named entity from the source ontology to one named entity from the target ontology. There are many cases in which there is one same entity (which represents the real world) but it is described with different vocabulary and data formats. This often leads to conflict in the process of data merging or integration. Direct ontology matching usually employs string-based and structured-based similarity measures. String-based similarity compute the string similarity of entities' label in the two input ontologies. Structure-based similarity exploit the structure of the ontologies to determine relations of two input ontologies. The limitation of these two measures is that mapping cannot be found without any lexical or

structural similarity. In this situation, using background knowledge will help to discover the matching. This process is called Indirect ontology matching (Husein *et al.* (2016)).

7.4 Hybrid Ontology Matching

In this section, the hybrid ontology matching is introduced as shown in Figure 7.2. A hybrid ontology matching technique is required in pervasive environment, in which multiple matchers are used to find similarities between elements of ontology for attaining its full potential. The hybrid matching is the combination of direct and indirect ontology matching for establishing semantic correspondences among similar concepts of various Ontologies (Cerdeira (2014)). The proposed ontology matcher takes two schemas, namely, source and target ontology as input. Ontology matching is referred to the process of detecting similar entities between source and target Ontologies and establishing communication among them for data sharing and exchange. It produces an ontology alignment as an output. In addition to source and target schemas, the matcher takes some parameters and resources as input to support the process of ontology alignment. Usually, the parameter includes minimum trust values of various entities, instances and some heuristic rules for establishing the communication between various entities. The upper ontology acts as a background ontology resource in on-

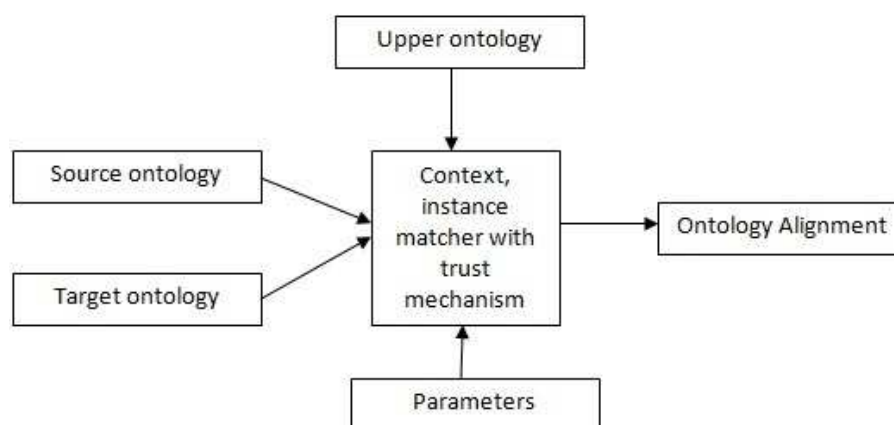


Figure 7.2: Hybrid Ontology Matching

ontology matching process. The context-based matcher collects information about entity id, entity name, entity neighbors and does the matching operation between source and target Ontologies with respect to collected information. Instance ontology matching equates two sets of instances or individuals of source and target ontologies for establishing connections. Instance matcher is similar to record linkage technique in databases

(Abubakar *et al.* (2018)). It matches the entities with value similarity and heuristic rule-based methods (Abubakar *et al.* (2018)). Existing instance does not handle missing instance value in ontology matching process. The main advantage of the proposed method is reconstructing the missing instance value and faulty instance value with the help of trust based data reconstruction method (Abubakar *et al.* (2018)). Moreover, the trustiness of instance value is checked before the ontology alignment is made between various entities. If the trust value of instance is below threshold, then the mapping is ignored with particular entity in target ontology. The alignment between entities of Ontologies can be represented in quadruple format $\langle aid, es, et, t \rangle$ where aid is the alignment id, es and et are entities of source and target schemas and t is the trust value which holds the alignment between entities.

7.5 Results and Discussions

We carried out the set of experiments on pervasive real-world examples with three metrics from the field of information retrieval: precision, recall and Fmeasure. We evaluated the performance of the proposed approach against direct and indirect matching. We run our proposed approach four times on each source and target schemas pairs of each application. In the first run, we considered only direct matching. During the second run, we did with indirect matching where upper ontology is considered as background knowledge. In the third set of experiments, we considered hybrid ontology matching (without trust mechanisms). During the final run, hybrid ontology matching with trust mechanism is considered. List of Ontologies used for experiments is shown in Table

Table 7.1: List of Ontologies used for experiments

| Ontologies | Concepts | Properties |
|---------------------|----------|------------|
| Smart Home (Os) | 10 | 18 |
| Healthcare (Oh) | 12 | 22 |
| Traffic System (Ot) | 13 | 24 |
| Environment (Oe) | 12 | 20 |
| Upper Ontology (Ou) | 24 | 68 |

7.1. Table 7.2 shows the list of applications and reference alignments. Let M be the number of total matches found by domain expert. The number of right matches done by the proposed approach is represented as C. Let W be the number of wrong matches done by the proposed approach. The precision is calculated by $P=(C/(C+W))*100$

and recall is calculated by $R=(C/M)*100$. The F-measure is evaluated by using $F=(2PR/(P+R))*100$.

Table 7.2: List of applications and reference alignments

| Applications | Ontologies | Reference alignments (M) |
|-------------------------|------------|--------------------------|
| Thermostat control | Os and Ot | 24 |
| Medical event detection | Oh and Os | 16 |
| Route design | Ot and Os | 18 |
| Lighting system | Oe and Os | 20 |

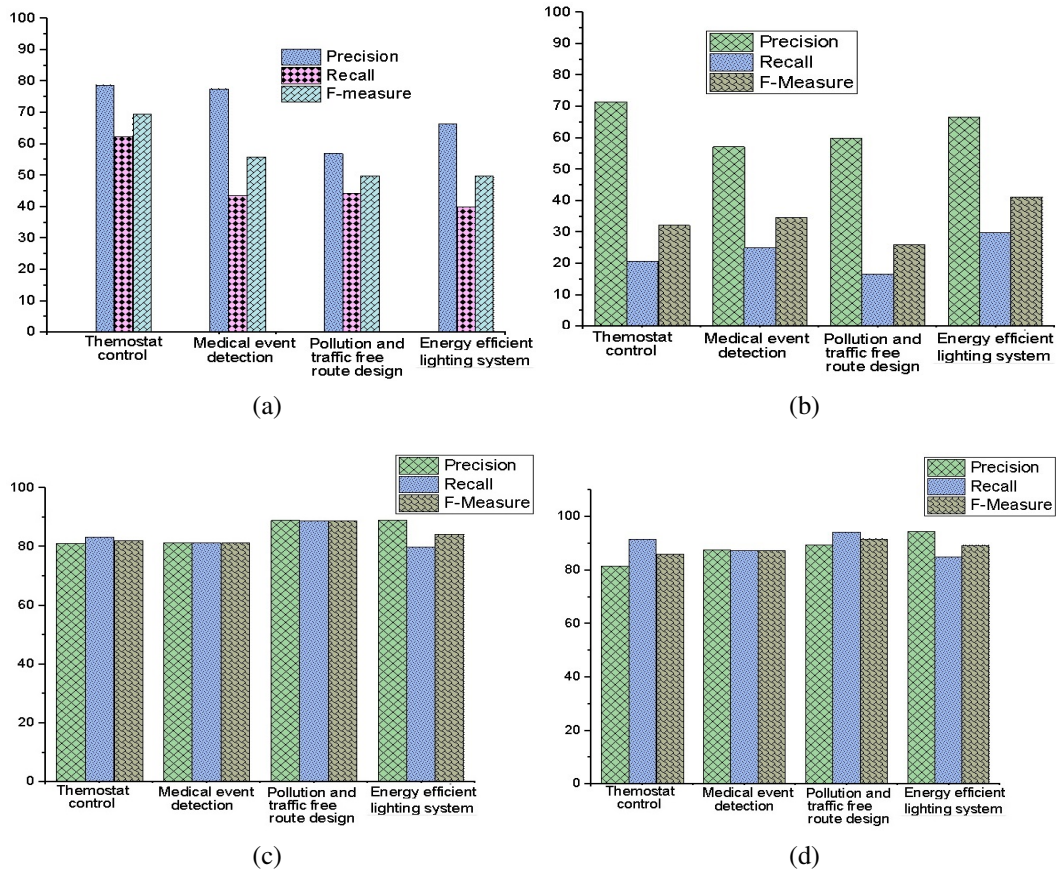


Figure 7.3: (a) Direct Ontology Matching (b) Indirect Ontology Matching (c) Hybrid Ontology Matching without Trust (d) Hybrid Ontology Matching with Trust

For thermostat control application, smart home ontology Os interacts with traffic system ontology Ot to predict the user arrival for switching on the thermostat. In health-care application, healthcare ontology Oh interacts with smart home ontology Os to find the activities of the patient and medical event detection. For designing pollution and traffic free route, traffic system ontology Ot interacts with environment ontology Oe. To minimize the energy consumption of electrical equipment, smart home ontology

interacts with environment ontology. Based on the results presented in Figures 7.3a, 7.3b,7.3c we conclude that the hybrid ontology matching achieves 5% to 20% better performances in ontology matching when compared to direct and indirect matching techniques. However, the performance of matching technique will gradually increase only if we have quality upper ontology as background knowledge. We manually inserted 5% of untrustworthy data and simulated a few entities to behave abnormally in all four applications to check the impact of trust mechanism in ontology matching. Based on the results presented in Figure 7.3d, we can say that the hybrid ontology matching with trust mechanism gives 1% to 5% better performance than hybrid ontology matching without trust mechanism.

7.6 Summary

We proposed an upper ontology for pervasive environments for hybrid ontology matching. Ontology matching establishes the semantic correspondence between matching entities of various application ontologies for data sharing using background knowledge as upper ontology. Our experimental results show that upper ontology plays a key factor in the performance of ontology matching. Even though the experiment is restricted to the pervasive environment, we did evaluations with several heterogeneous applications within a pervasive environment for ontology matching.

Chapter 8

Conclusions and Future Work

The sharing and exchange of data in pervasive environments are beneficial for accurate event detection in many pervasive applications. Ensuring the trustiness of data and node is the prime pre-process before detecting the events in harsh and unfriendly pervasive environments. The success of pervasive application relies on trustworthy data exchange among trustworthy nodes for data analysis and decision making. Hence, the effective mechanisms are required for the trustworthy data exchange among various entities of pervasive applications. The research work in this thesis is directed towards the design and development of a trusted semantic framework for context-aware sensor-driven pervasive environments to enable trustworthy data generation and exchange among applications for event detection.

We proposed a sensor data model for evaluating the trustworthiness of data and event detection in WSN. Firstly, we looked into various characteristics of sensor data in WSN applications and their associated data faults, inaccuracies and inconsistencies. Then the proposed sensor data model is tested with real-world sensor dataset. The result shows that the proposed sensor data model outperforms the existing data models in terms of detecting the data trustiness and events in reliable fashion and reconstruction of data in an energy efficient way. In hybrid sensor data modeling, the result shows that the proposed model consumes 5% to 20% lesser energy than the existing data models for detecting the data trustiness.

We proposed TDG in wireless sensor networks for trust-based data collection, data aggregation, and data reconstruction. We used Intel lab dataset to show that the process from sensing to decision making demands the trust-based process to ensure trustworthy data exchange, data analysis, and decisiveness. We showed that DCT and DAT consume less energy and less network delay when we have more than 30% of data faults, data losses and malicious nodes in the network than traditional DC and DA process. We also showed that the absence of trust from the data collection process to decision making the process in a sensor-driven pervasive application in deployed harsh environment could affect the normal functionality of the application.

A hybrid TMS for WSN is proposed, in which the data source computes its sensed data

item trustworthiness by seeing the data consistency with the assistance of data correlation technique. The sensed data item is forwarded with self data trust to next hop neighbor. The neighbor node receives the data item and evaluates the peer data trust by spatial correlation technique. The sensed data item is forwarded with peer data trust to sink node through intermediate nodes. The sink node utilizes data provenance, interdependency property and communication capability to evaluate the final trust score of the data item, intermediate nodes, and the source node. The proposed HTMS decreases the effects of untrustworthy data, malicious and selfish node. The result depicts the effectiveness of HTMS in detecting and discarding the untrustworthy data, malicious and selfish nodes. The experimental result depicts that the HTMS outperforms existing methods by 9% in detecting and discarding the untrustworthy data, malicious and selfish nodes.

The proposed CATMS integrates heuristic functions, data correlation and contextual information-based algorithms for identifying real medical emergencies. The proposed approach works with data correlation and contextual information based algorithms for identifying data faults, for data reconstruction of data faults and data losses and distinguishing faulty data and clinical emergencies. We evaluated our proposed scheme on real data samples injected with various synthetic data faults and data losses. The proposed scheme is capable of online detection of data fault and reconstruction of data faults and data losses for reliable event detection. Experimental results prove that the effectiveness of context-aware trust management scheme in detecting data faults and detecting medical emergencies.

We proposed an upper ontology for pervasive environments for hybrid ontology matching. Ontology matching establishes the semantic correspondence between matching entities of various application ontologies for data sharing using background knowledge as upper ontology. Our experimental results show that upper ontology plays a key factor in the performance of hybrid ontology matching. Even though the experiment is restricted to the pervasive environment, we did evaluations with several heterogeneous applications within a pervasive environment for ontology matching.

The trusted semantic framework is used to detect the data faults, malicious behavior, reconstruct the faulty data, data losses and detect the events in reliably. The proposed framework is effectively used to establish the semantic correspondence between pervasive applications for trustworthy data sharing and exchange to detect the events. How-

ever, the following issues need to be investigated in the future.

1. A data model can be constructed to work with text data (non-numeric data) by including data semantics with the help of ontologies and semantic rules.
2. A Trust model can be developed for the prediction of future trust score of the sensor node based on past scores instead of the history of communication.
3. Domain knowledge is required to frame the rules, and also it requires threshold and parameter settings. A bit of human involvement is required during system initialization to achieve effective results. This can be automated in the future.
4. A Trust-based data reduction method can be constructed in future using data correlation technique, which preserves the battery power of sensors and maximizes the network lifetime.

References

- Abubakar, M., H. Hamdan, N. Mustapha, and T. N. M. Aris, Instance-based ontology matching: A literature review. *In International Conference on Soft Computing and Data Mining*. Springer, 2018.
- Aivaloglou, E. and S. Gritzalis (2010). Hybrid trust and reputation management for sensor networks. *Wireless Networks*, 16(5), 1493–1510.
- Akyildiz, I. F., W. Su, Y. Sankarasubramaniam, and E. Cayirci (2002). Wireless sensor networks: a survey. *Computer networks*, 38(4), 393–422.
- Barrenetxea, G., M. Bystranowski, O. Couach, H. Dubois-Ferriere, S. Dufey, M. Krichane, J. Mezzo, S. Mortier, M. Parlange, G. Schaefer, *et al.*, Demoabstract: Sensorscope, an urban environmental monitoring network. *In 4th European conference on wireless sensor networks (EWSN 2007), Delft, Netherlands*. 2007.
- Berners-Lee, T., J. Hendler, O. Lassila, *et al.* (2001). The semantic web. *Scientific american*, 284(5), 28–37.
- Bertino, E., Data trustworthiness—approaches and research challenges. *In Data privacy management, autonomous spontaneous security, and security assurance*. Springer, 2014, 17–25.
- Boukerche, A. and Y. Ren (2009). A secure mobile healthcare system using trust-based multicast scheme. *IEEE Journal on Selected Areas in Communications*, 27(4), 387–399.
- B.Smith (2002). Bfo. <https://basic-formal-ontology.org>.
- Bui, V., R. Verhoeven, J. Lukkien, and R. Kocielnik, A trust evaluation framework for sensor readings in body area sensor networks. *In Proceedings of the 8th International Conference on Body Area Networks*. ICST, 2013.
- Bui, V. T., A trust management model for body sensor networks. *In 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*. IEEE, 2011.
- Bui, V. T., J. J. Lukkien, and R. Verhoeven, Toward a trust management model for a configurable body sensor platform. *In Proceedings of the 6th International Conference on Body Area Networks*. ICST, 2011.
- Cao, Q. H., I. Khan, R. Farahbakhsh, G. Madhusudan, G. M. Lee, and N. Crespi, A trust model for data sharing in smart cities. *In 2016 IEEE International Conference on Communications (ICC)*. IEEE, 2016.
- Cerdeira, L. O. (2014). *Study and application of new methods for ontology matching*. Ph.D. thesis, Universidade de Vigo.
- Chen, H. (2009). Task-based trust management for wireless sensor networks. *International Journal of Security and its applications*, 3(2), 21–26.

- Chen, H., H. Wu, J. Hu, and C. Gao, Event-based trust framework model in wireless sensor networks. *In 2008 International Conference on Networking, Architecture, and Storage*. IEEE, 2008.
- Chen, J., S. Kher, and A. Somani, Distributed fault detection of wireless sensor networks. *In Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks*. 2006.
- Chen, L., C. Nugent, M. Mulvenna, D. Finlay, and X. Hong, Semantic smart homes: towards knowledge rich assisted living environments. *In Intelligent Patient Management*. Springer, 2009, 279–296.
- Chitra, K. G., A, Fault aware trust determination algorithm for wireless body sensor network (wbsn). *In Proceedings of first international conference on smart system, innovations and computing*. Springer, 2018.
- Chittibabu, Y., C. Anuradha, and S. R. C. P. Murty (2018). Fuzzy trust based energy aware multipath secure data collection in wireless sensor network. *Journal of Computational and Theoretical Nanoscience*, 16(2), 669–675.
- Cho, J.-H., A. Swami, and R. Chen (2010). A survey on trust management for mobile ad hoc networks. *IEEE Communications Surveys & Tutorials*, 13(4), 562–583.
- Cho, J.-H., A. Swami, and R. Chen (2011). A survey on trust management for mobile ad hoc networks. *IEEE Communications Surveys & Tutorials*, 13(4), 562–583.
- Cover, T. M., P. E. Hart, *et al.* (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1), 21–27.
- De Meulenaer, G., F. Gosset, F.-X. Standaert, and O. Pereira, On the energy cost of communication and cryptography in wireless sensor networks. *In 2008 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*. IEEE, 2008.
- Dereszynski, E. W. and T. G. Dietterich (2011). Spatiotemporal models for data-anomaly detection in dynamic environmental monitoring campaigns. *ACM Transactions on Sensor Networks (TOSN)*, 8(1), 3.
- Dhulipala, V. S. and N. Karthik (2017). Trust management technique in wireless sensor networks: Challenges and issues for reliable communication: A review. *CSI Transactions on ICT*, 5(3), 281–294.
- Dhulipala, V. S., N. Karthik, and R. Chandrasekaran (2013). A novel heuristic approach based trust worthy architecture for wireless sensor networks. *Wireless personal communications*, 70(1), 189–205.
- Diallo, G. (2014). An effective method of large scale ontology matching. *Journal of biomedical semantics*, 5(1), 44.

Dogan, G. and K. Avincan (2017). Multiprotu: A kalman filtering based trust architecture for two-hop wireless sensor networks. *Peer-to-Peer Networking and Applications*, 10(1), 278–291.

Dondio, P., E. Manzo, and S. Barrett, Applied computational trust in utilities management: a case study on the town council of cava de'tirreni. *In IFIP international conference on trust management*. Springer, 2007.

Duarte, M. F. and Y. H. Hu (2004). Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 64(7), 826–838.

Ducatel, G., Z. Cui, and B. Azvine, Hybrid ontology and keyword matching indexing system. *In Proc. of IntraWebs Workshop at WWW*. 2006.

Fang, L. and S. Dobson, In-network sensor data modelling methods for fault detection. *In International joint conference on ambient intelligence*. Springer, 2013.

Faria, D., C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, and F. M. Couto, The agreementmakerlight ontology matching system. *In OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*. Springer, 2013.

Fasolo, E., M. Rossi, J. Widmer, and M. Zorzi (2007). In-network aggregation techniques for wireless sensor networks: a survey. *IEEE Wireless Communications*, 14(2), 70–87.

Gao, Y., X. Li, J. Li, and Y. Gao, A trustworthy data aggregation model based on context and data density correlation degree. *In Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. ACM, 2018.

Gao, Y. and W. Liu (2014). Betrusted: a dynamic trust model based on bayesian inference and tsallis entropy for medical sensor networks. *Journal of Sensors*, 2014.

Gao, Y. and W. Liu, A security routing model based on trust for medical sensor networks. *In 2015 IEEE international conference on communication software and networks (ICCSN)*. IEEE, 2015.

Ghasemzadeh, H., V. Loseu, and R. Jafari, Collaborative signal processing for action recognition in body sensor networks: a distributed classification algorithm using motion transcripts. *In Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*. ACM, 2010.

Gilbert, E. P. K., B. Kaliaperumal, E. B. Rajsingh, and M. Lydia (2018). Trust based data prediction, aggregation and reconstruction using compressed sensing for clustered wireless sensor networks. *Computers & Electrical Engineering*, 72, 894–909.

Gomez, L., R. De Mol, P. Hogewerf, and A. Ipema, Trustworthiness assessment of cow behaviour data collected in a wireless sensor network. *In 5th European Conference on Precision Livestock Farming, Prague Czech Republic, 11-14 July, 2011*. 2011.

- Govindan, K. and P. Mohapatra (2011). Trust computations and trust dynamics in mobile adhoc networks: A survey. *IEEE Communications Surveys & Tutorials*, 14(2), 279–298.
- Gruber, T., *Ontology*. Springer, US, 2009.
- Guarino, N. (2003). Dolce. <http://http://www.loa.istc.cnr.it/old/DOLCE.html>.
- Guestrin, C., P. Bodik, R. Thibaux, M. Paskin, and S. Madden, Distributed regression: an efficient framework for modeling sensor network data. *In Proceedings of the 3rd international symposium on Information processing in sensor networks*. ACM, 2004.
- Guo, J., J. Fang, and X. Chen, Survey on secure data aggregation for wireless sensor networks. *In Proceedings of 2011 IEEE International Conference on Service Operations, Logistics and Informatics*. IEEE, 2011.
- Gwadera, R., M. Riahi, and K. Aberer, Pattern-wise trust assessment of sensor data. *In 2014 IEEE 15th International Conference on Mobile Data Management* volume1. IEEE, 2014.
- Han, G., J. Jiang, L. Shu, J. Niu, and H.-C. Chao (2014). Management and applications of trust in wireless sensor networks: A survey. *Journal of Computer and System Sciences*, 80(3), 602–617.
- Haron, N., J. Jaafar, I. A. Aziz, M. H. Hassan, and M. I. Shapiyai, Data trustworthiness in internet of things: A taxonomy and future directions. *In 2017 IEEE conference on big data and analytics (ICBDA)*. IEEE, 2017.
- He, D., C. Chen, S. Chan, J. Bu, and A. V. Vasilakos (2012). Retruster: Attack-resistant and lightweight trust management for medical sensor networks. *IEEE transactions on information technology in biomedicine*, 16(4), 623–632.
- He, W., X. Yang, and D. Huang, A hybrid approach for measuring semantic similarity between ontologies based on wordnet. *In International Conference on Knowledge Science, Engineering and Management*. Springer, 2011.
- Hosseini, J., R. Mohammad, *et al.* (2015). A fuzzy fully distributed trust management system in wireless sensor networks. *International Journal of Electronics and Communications*, 9(17), 1–10.
- Hunkeler, U. (2013). Distributed sensor data models and their impact on energy consumption of wireless sensor networks. Technical report, EPFL.
- Hur, J., Y. Lee, H. Youn, D. Choi, and S. Jin, Trust evaluation model for wireless sensor networks. *In The 7th International Conference on Advanced Communication Technology, 2005, ICACT 2005*. volume1. IEEE, 2005.
- Husein, I. G., S. Akbar, B. Sitohang, and F. N. Azizah, Review of ontology matching with background knowledge. *In 2016 International Conference on Data and Software Engineering (ICoDSE)*. IEEE, 2016.

- Illiano, V. P. and E. C. Lupu (2015a). Detecting malicious data injections in event detection wireless sensor networks. *IEEE Transactions on Network and Service Management*, 12(3), 496–510.
- Illiano, V. P. and E. C. Lupu (2015b). Detecting malicious data injections in wireless sensor networks: A survey. *ACM Computing Surveys (CSUR)*, 48(2), 24.
- Issariyakul, T. and E. Hossain, Introduction to network simulator 2 (ns2). *In Introduction to network simulator NS2*. Springer, 2009, 1–18.
- Jean-Mary, Y. R., E. P. Shironoshita, and M. R. Kabuka (2009). Ontology matching with semantic verification. *Journal of Web Semantics*, 7(3), 235–251.
- Jesus, P., C. Baquero, and P. S. Almeida (2015). A survey of distributed data aggregation algorithms. *IEEE Communications Surveys & Tutorials*, 17(1), 381–404.
- Ji, S., J. S. He, and Z. Cai, *Data Gathering in Wireless Sensor Networks*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. ISBN 978-3-642-40009-4, 535–565. URL https://doi.org/10.1007/978-3-642-40009-4_16.
- Jiang, C., S. Liu, Z. Lin, G. Zhao, R. Duan, and K. Liang (2016). Domain-aware trust network extraction for trust propagation in large-scale heterogeneous trust networks. *Knowledge-Based Systems*, 111, 237–247.
- Jiang, J., G. Han, F. Wang, L. Shu, and M. Guizani (2015). An efficient distributed trust model for wireless sensor networks. *IEEE transactions on parallel and distributed systems*, 26(5), 1228–1237.
- Jiménez-Ruiz, E. and B. C. Grau, Logmap: Logic-based and scalable ontology matching. *In International Semantic Web Conference*. Springer, 2011.
- Kamal, A. R. M., C. Bleakley, and S. Dobson (2013). Packet-level attestation (pla): A framework for in-network sensor data reliability. *ACM Transactions on Sensor Networks (TOSN)*, 9(2), 19.
- Kanaga, G., Trustworthy architecture for wireless body sensor network. *In Wearable Technologies: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2018, 333–362.
- Karthik, N. and V. Ananthanarayana, Data trustworthiness in wireless sensor networks. *In 2016 IEEE Trustcom/BigDataSE/ISPA*. IEEE, 2016.
- Karthik, N. and V. Ananthanarayana, Data trust model for event detection in wireless sensor networks using data correlation techniques. *In 2017 fourth international conference on signal processing, communication and networking (ICSCN)*. IEEE, 2017a.
- Karthik, N. and V. Ananthanarayana (2017b). A hybrid trust management scheme for wireless sensor networks. *Wireless Personal Communications*, 97(4), 5137–5170.
- Karthik, N. and V. Ananthanarayana, Sensor data modeling for data trustworthiness. *In 2017 IEEE Trustcom/BigDataSE/ICCESS*. IEEE, 2017c.

- Karthik, N. and V. Ananthanarayana (2018a). Context aware trust management scheme for pervasive healthcare. *Wireless Personal Communications*, 1–39.
- Karthik, N. and V. Ananthanarayana, Towards an upper ontology and hybrid ontology matching for pervasive environments. *In International Conference on Intelligent Systems Design and Applications*. Springer, 2018b.
- Karthik, N. and V. Ananthanarayana (2019). Trust based data gathering in wireless sensor network. *Wireless Personal Communications*, 108(3), 1697–1717.
- Karthik, N. and V. S. Dhulipala, Trust calculation in wireless sensor networks. *In 2011 3rd International Conference on Electronics Computer Technology* volume4. IEEE, 2011.
- Kelly, G. (2006). Body temperature variability (part 1): a review of the history of body temperature and its variability due to site selection, biological rhythms, fitness, and aging. *Alternative medicine review*, 11(4).
- Khalid, O., S. U. Khan, S. A. Madani, K. Hayat, M. I. Khan, N. Min-Allah, J. Kolodziej, L. Wang, S. Zeadally, and D. Chen (2013). Comparative study of trust and reputation systems for wireless sensor networks. *Security and Communication Networks*, 6(6), 669–688.
- Khan, Z. C. (2012). Ontological commitments. *Web page*=<http://www.thezfiles.co.za/ROMULUS/ontologicalCommitments.html>.
 URL <http://www.thezfiles.co.za/ROMULUS/ontologicalCommitments.html>.
- Khiat, A. and M. Benaissa, Insmt+ results for oaei 2015 instance matching. *In OM*. 2015.
- Khiat, A., M. Benaissa, and M. A. Belfedhal, Strim results for oaei 2015 instance matching evaluation. *In OM*. 2015.
- Kong, L., M. Xia, X.-Y. Liu, M.-Y. Wu, and X. Liu, Data loss and reconstruction in sensor networks. *In 2013 Proceedings IEEE INFOCOM*. IEEE, 2013.
- Li, H., Z. Dragisic, D. Faria, V. Ivanova, E. Jiménez-Ruiz, P. Lambrix, and C. Pesquita (2019). User validation in ontology alignment: functional assessment and impact. *The Knowledge Engineering Review*, 34.
- Li, W. and X. Zhu, Recommendation-based trust management in body area networks for mobile healthcare. *In 2014 IEEE 11th International conference on mobile ad hoc and sensor systems*. IEEE, 2014.
- Li, X., F. Zhou, and J. Du (2013). Ldts: A lightweight and dependable trust system for clustered wireless sensor networks. *IEEE transactions on information forensics and security*, 8(6), 924–935.

- Li, Z., Y. Zhu, H. Zhu, and M. Li, Compressive sensing approach to urban traffic sensing. *In 2011 31st International Conference on Distributed Computing Systems*. IEEE, 2011.
- Lim, H.-S., Y.-S. Moon, and E. Bertino, Provenance-based trustworthiness assessment in sensor networks. *In Proceedings of the Seventh International Workshop on Data Management for Sensor Networks*. ACM, 2010.
- Liu, C.-x., Y. Liu, and Z.-j. Zhang (2013a). Improved reliable trust-based and energy-efficient data aggregation for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 9(5), 652495.
- Liu, X., Y. Wang, S. Zhu, and H. Lin (2013b). Combating web spam through trust-distrust propagation with confidence. *Pattern Recognition Letters*, 34(13), 1462–1469.
- Liu, Y., C.-x. Liu, and Q.-A. Zeng (2016). Improved trust management based on the strength of ties for secure data aggregation in wireless sensor networks. *Telecommunication Systems*, 62(2), 319–325.
- Luo, H., J. Tao, and Y. Sun, Entropy-based trust management for data collection in wireless sensor networks. *In 2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*. IEEE, 2009.
- Ma, T., Y. Liu, and Z.-j. Zhang (2015). An energy-efficient reliable trust-based data aggregation protocol for wireless sensor networks. *International Journal of Control and Automation*, 8(3), 305–318.
- Madden, S. *et al.* (2004). Intel lab data. *Web page*=<http://db.csail.mit.edu/labdata/labdata.html>.
- Mannini, A., S. S. Intille, M. Rosenberger, A. M. Sabatini, and W. Haskell (2013). Activity recognition using a single accelerometer placed at the wrist or ankle. *Medicine and science in sports and exercise*, 45(11), 2193.
- Momani, M., S. Challa, and R. Alhmouz (2010). Bayesian fusion algorithm for inferring trust in wireless sensor networks. *Journal of networks*, 5(7), 815.
- Nasridinov, A., S.-Y. Ihm, Y.-S. Jeong, and Y.-H. Park, Event detection in wireless sensor networks: Survey and challenges. *In Mobile, Ubiquitous, and Intelligent Computing*. Springer, 2014, 585–590.
- Nath, R. P. D., H. Seddiqui, and M. Aono, Resolving scalability issue to ontology instance matching in semantic web. *In 2012 15th International Conference on Computer and Information Technology (ICCIT)*. IEEE, 2012.
- Nguyen, T. A., D. Bucur, M. Aiello, and K. Tei, Applying time series analysis and neighbourhood voting in a decentralised approach for fault detection and classification in wsns. *In Proceedings of the Fourth Symposium on Information and Communication Technology*. ACM, 2013.

- Ni, K., N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava (2009). Sensor network data fault types. *ACM Transactions on Sensor Networks (TOSN)*, 5(3), 25.
- Oh, D.-J., H.-O. Hong, and B.-A. Lee (2016). The effects of strenuous exercises on resting heart rate, blood pressure, and maximal oxygen uptake. *Journal of exercise rehabilitation*, 12(1), 42.
- Osman, A. Guerassimov, A. Mehaoua, A. Marcus, and B. Furht, Sensor fault and patient anomaly detection and classification in medical wireless sensor networks. *In 2013 IEEE international conference on communications (ICC)*. IEEE, 2013.
- Otero-Cerdeira, L., F. J. Rodríguez-Martínez, and A. Gómez-Rodríguez (2015). Ontology matching: A literature review. *Expert Systems with Applications*, 42(2), 949–971.
- Paschalidis, I. C. and Y. Chen (2010). Statistical anomaly detection with sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 7(2), 17.
- Pires, W., T. H. de Paula Figueiredo, H. C. Wong, and A. A. F. Loureiro, Malicious node detection in wireless sensor networks. *In 18th International Parallel and Distributed Processing Symposium, 2004. Proceedings.. IEEE, 2004.*
- Puliafito, C., E. Mingozzi, F. Longo, A. Puliafito, and O. Rana (2019). Fog computing for the internet of things: A survey. *ACM Transactions on Internet Technology (TOIT)*, 19(2), 1–41.
- Puneeth, D., N. Joshi, P. K. Atrey, and M. Kulkarni (2018). Energy-efficient and reliable data collection in wireless sensor networks. *Turkish Journal of Electrical Engineering & Computer Sciences*, 26(1), 138–149.
- Rajasegarar, S., C. Leckie, M. Palaniswami, and J. C. Bezdek, Distributed anomaly detection in wireless sensor networks. *In 2006 10th IEEE Singapore international conference on communication systems*. IEEE, 2006.
- Ramalingam, L. (2006). An efficient data collection scheme based on trust evaluation in large scale wireless sensor networks. *Arpn Journals*.
- Ravichandran, J. and A. I. Arulappan (2013). Data validation algorithm for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 9(12), 634278.
- Reddy, V. B., S. Venkataraman, and A. Negi (2017). Communication and data trust for wireless sensor networks using d-s theory. *IEEE Sensors Journal*, 17(12), 3921–3929.
- Rezvani, M. (2015). *Trust-Based Data Aggregation for WSNs in the Presence of Faults and Collusion Attacks..* Ph.D. thesis, University of New South Wales, Sydney, Australia.
- Salem, O., Y. Liu, and A. Mehaoua (2013). Anomaly detection in medical wireless sensor networks. *Journal of Computing Science and Engineering*, 7(4), 272–284.

- Salem, O., Y. Liu, A. Mehaoua, and R. Boutaba (2014). Online anomaly detection in wireless body area networks for reliable healthcare monitoring. *IEEE journal of biomedical and health informatics*, 18(5), 1541–1551.
- Sang, Y., H. Shen, Y. Inoguchi, Y. Tan, and N. Xiong, Secure data aggregation in wireless sensor networks: A survey. In *2006 Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'06)*. IEEE, 2006.
- Schönbrodt, F. D. and M. Perugini (2013). At what sample size do correlations stabilize? *Journal of Research in Personality*, 47(5), 609–612.
- Shadbolt, N., T. Berners-Lee, and W. Hall (2006). The semantic web revisited. *IEEE intelligent systems*, 21(3), 96–101.
- Shaikh, R. A., H. Jameel, B. J. d'Auriol, H. Lee, S. Lee, and Y.-J. Song (2009). Group-based trust management scheme for clustered wireless sensor networks. *IEEE transactions on parallel and distributed systems*, 20(11), 1698–1712.
- Sharma, A. B., L. Golubchik, and R. Govindan (2010). Sensor faults: Detection methods and prevalence in real-world datasets. *ACM Transactions on Sensor Networks (TOSN)*, 6(3), 23.
- Shvaiko, P. and J. Euzenat (2013). Ontology matching: state of the art and future challenges. *IEEE Transactions on knowledge and data engineering*, 25(1), 158–176.
- Stevenson, G., S. Knox, S. Dobson, and P. Nixon, Ontonym: a collection of upper ontologies for developing pervasive systems. In *Proceedings of the 1st Workshop on Context, Information and Ontologies*. ACM, 2009.
- Sun, Y., H. Luo, and S. K. Das (2012). A trust-based framework for fault-tolerant data aggregation in wireless multimedia sensor networks. *IEEE Transactions on Dependable and Secure Computing*, 9(6), 785–797.
- Taghikhaki, Z., N. Meratnia, and P. J. Havinga, Energy-efficient trust-based aggregation in wireless sensor networks. In *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2011.
- Talbi, S., M. Koudil, A. Bouabdallah, and K. Benatchba (2017). Adaptive and dual data-communication trust scheme for clustered wireless sensor networks. *Telecommunication Systems*, 65(4), 605–619.
- Tolle, G., J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, *et al.*, A macroscope in the redwoods. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*. ACM, 2005.
- Vamsi, P. R. and K. Kant (2016). Trust aware data aggregation and intrusion detection system for wireless sensor networks. *International Journal on Smart Sensing and Intelligent Systems*, 9(2), 537–562.

- Vervaet, A. and D. Baert (2002). The lead acid battery: semiconducting properties and peukert's law. *Electrochimica acta*, 47(20), 3297–3302.
- Vijayalakshmi, P. *et al.* (2013). Trust based data aggregation in wireless sensor networks. *International Journal of Computer Applications*, 73(22).
- Wälchli, M., P. Skoczylas, M. Meer, and T. Braun, Distributed event localization and tracking with wireless sensors. *In International Conference on Wired/Wireless Internet Communications*. Springer, 2007.
- Wang, G. and J. Wu (2011). Multi-dimensional evidence-based trust management with multi-trusted paths. *Future Generation Computer Systems*, 27(5), 529–538.
- Wang, M., A. Xue, and H. Xia (2017). Abnormal event detection in wireless sensor networks based on multiattribute correlation. *Journal of Electrical and Computer Engineering*, 2017.
- Wang, N. and Y. Pang (2014). An improved light-weight trust model in wsn. *Computer Modeling & New Technologies*, 18(4), 57–61.
- Wang, X., J. Su, B. Wang, G. Wang, and H.-F. Leung (2015). Trust description and propagation system: semantics and axiomatization. *Knowledge-Based Systems*, 90, 81–91.
- Wang, Z., R. Bie, and M. Zhou, Hybrid ontology matching for solving the heterogeneous problem of the iot. *In 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE, 2012.
- Whitehead, J. R. (2016). Cluster-based trust proliferation and energy efficient data collection in unattended wireless sensor networks with mobile sinks.
- Wittenburg, G., N. Dziengel, S. Adler, Z. Kasmi, M. Ziegert, and J. Schiller (2012). Cooperative event detection in wireless sensor networks. *IEEE Communications Magazine*, 50(12), 124–131.
- Won, J. and E. Bertino, Distance-based trustworthiness assessment for sensors in wireless sensor networks. *In International Conference on Network and System Security*. Springer, 2015.
- Wu, G. W., Z. S. Liu, and P. Pirozmand, A fuzzy trust model for public key distribution in body area networks. *In Advanced materials research* volume989. Trans Tech Publ, 2014a.
- Wu, G. W., Z. S. Liu, and P. Pirozmand, A fuzzy trust model for public key distribution in body area networks. *In Advanced materials research* volume989. Trans Tech Publ, 2014b.
- Wu, J., R. Xiong, and F. Chiclana (2016). Uninorm trust propagation and aggregation methods for group decision making in social network with four tuple information. *Knowledge-Based Systems*, 96, 29–39.

- Xiao, X.-Y., W.-C. Peng, C.-C. Hung, and W.-C. Lee, Using sensor ranks for in-network detection of faulty readings in wireless sensor networks. *In Proceedings of the 6th ACM international workshop on Data engineering for wireless and mobile access*. ACM, 2007.
- Xiong, F., Y. Liu, and J. Cheng (2017). Modeling and predicting opinion formation with trust propagation in online social networks. *Communications in Nonlinear Science and Numerical Simulation*, 44, 513–524.
- Yao, Y., A. Sharma, L. Golubchik, and R. Govindan (2010). Online anomaly detection for sensor systems: A simple and efficient approach. *Performance Evaluation*, 67(11), 1059–1075.
- Yao, Z., D. Kim, and Y. Doh, Plus: Parameterized and localized trust management scheme for sensor networks security. *In 2006 IEEE International Conference on Mobile Ad Hoc and Sensor Systems*. IEEE, 2006.
- Yarinezhad, R. and S. N. Hashemi (2019). Distributed faulty node detection and recovery scheme for wireless sensor networks using cellular learning automata. *Wireless Networks*, 25(5), 2901–2917.
- Ye, J., S. Dasiopoulou, G. Stevenson, G. Meditskos, E. Kontopoulos, I. Kompatsiaris, and S. Dobson (2015). Semantic web technologies in pervasive computing: A survey and research roadmap. *Pervasive and Mobile Computing*, 23, 1–25.
- Ye, J., G. Stevenson, and S. Dobson (2016). Detecting abnormal events on binary sensors in smart home environments. *Pervasive and Mobile Computing*, 33, 32–49.
- Yu, H., Z. Shen, and C. Leung (2010). Towards trust-aware health monitoring body area sensor networks. *International Journal of Information Technology*, 16(2), 1–20.
- Yu, T., A. M. Akhtar, X. Wang, and A. Shami, Temporal and spatial correlation based distributed fault detection in wireless sensor networks. *In 2015 IEEE 28th Canadian conference on electrical and computer engineering (CCECE)*. IEEE, 2015.
- Zahariadis, T., H. C. Leligou, P. Trakadas, and S. Voliotis (2010). Trust management in wireless sensor networks. *European Transactions on Telecommunications*, 21(4), 386–395.
- Zhang, B., Z. Huang, and Y. Xiang (2014). A novel multiple-level trust management framework for wireless sensor networks. *Computer Networks*, 72, 45–61.
- Zhang, H., J. Liu, and A.-C. Pang (2018). A bayesian network model for data losses and faults in medical body sensor networks. *Computer Networks*, 143, 166–175.
- Zhang, H., J. Liu, A.-C. Pang, and R. Li, A data reconstruction model addressing loss and faults in medical body sensor networks. *In 2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2016a.

Zhang, J., R. Shankaran, A. O. Mehmet, V. Varadharajan, and A. Sattar, A trust management architecture for hierarchical wireless sensor networks. *In IEEE Local Computer Network Conference*. IEEE, 2010a.

Zhang, L., X. Wu, and D. Luo, Improving activity recognition with context information. *In 2015 IEEE international conference on mechatronics and automation (ICMA)*. IEEE, 2015.

Zhang, Y., H. Cheng, and D. Chen, Data reconstruction with spatial and temporal correlation in wireless sensor networks. *In Proceedings of the 3rd ACM Workshop on Mobile Sensing, Computing and Communication*. ACM, 2016b.

Zhang, Y., N. Meratnia, and P. J. Havinga (2010b). Outlier detection techniques for wireless sensor networks: A survey. *IEEE Communications Surveys and Tutorials*, 12(2), 159–170.

Zheng, J. and A. Jamalipour, *Wireless sensor networks: a networking perspective*. John Wiley & Sons, 2009.

Zhu, H., Y. Zhu, M. Li, and L. M. Ni, Seer: metropolitan-scale traffic perception based on lossy sensory data. *In IEEE INFOCOM 2009*. IEEE, 2009.

Publications

Journal Papers

1. Karthik N., and V. S. Ananthanarayana, "A Hybrid Trust Management Scheme for Wireless Sensor Networks." *Wireless Personal Communications* (2017): 1-34. Springer. DOI: <https://doi.org/10.1007/s11277-017-4772-4>.
2. Karthik N., and V. S. Ananthanarayana, "Context Aware Trust Management Scheme for Pervasive Healthcare", *Wireless Personal Communications* (2018): 1-39, Springer. DOI: <https://doi.org/10.1007/s11277-018-6091-9>.
3. Karthik N., and V. S. Ananthanarayana, "Trust based Data gathering in Wireless Sensor Networks." *Wireless Personal Communications* (2019) Springer. DOI: [10.1007/s11277-019-06491-y](https://doi.org/10.1007/s11277-019-06491-y).

Conference Papers

1. Karthik N., and Ananthanarayana V.S., "Data Trustworthiness in Wireless Sensor Networks" 15th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TRUSTCOM-2016), Tianjin University, Tianjin, China. (CORE A)
2. Karthik N, and Ananthanarayana V.S. , "Data Trust Model for Event Detection in Wireless Sensor Networks using Data Correlation Techniques", 4th IEEE International Conference on Signal Processing, Communications and Networking, ICSCN-2017, MIT Chennai, India.
3. Karthik N, and Ananthanarayana V.S. , "Sensor Data Modelling for Data Trustworthiness", 16th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (Trustcom/BigDataSE/ICCESS-2017), UTS, Sydney, Australia. (CORE A)
4. Karthik, N., and V. S. Ananthanarayana. "An Ontology Based Trust Framework for Sensor-Driven Pervasive Environment", 11th IEEE Asia International Conference on Mathematical Modelling and Computer Simulation (AMS), 2017, UMS, Sabah, Malaysia. (CORE C)
5. Karthik N., and Ananthanarayana V S, "Trust Based Semantic Architecture for Pervasive Environments", 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT-2018), MNIT, Jaipur, India.
6. Karthik, N., and V. S. Ananthanarayana, "A Trust Model for Lightweight Semantic Annotation of Sensor Data in Pervasive Environment", 17th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2018), Singapore. (CORE C)
7. Karthik, N., and V. S. Ananthanarayana, "Towards an Upper Ontology and Hybrid Ontology Matching for Pervasive Environments", 18th International Conference on Intelligent Systems Design and Applications (ISDA 2018), VIT Vellore, India. (CORE C)

Curriculum Vitae

Mr. Karthik N

Full-Time Research Scholar
Department of Information Technology
National Institute of Technology Karnataka
P.O. Srinivasanagar, Surathkal
Mangalore-575 025

Permanent Address

Karthik N
Door No. 146, Pillaiyar Koil Street
Kalavai Puthur, Kalavai -632506
Vellore District, Tamilnadu, India.
Email: nkarthikapce@gmail.com
Mobile: +917904820693.

Academic Records

1. M.E. in Pervasive Computing Technology from Anna University, Trichy, Tamilnadu, India, 2011.
2. B.E. in Computer Science and Engineering from Adhiparasakthi College of Engineering, Vellore, Tamilnadu, India, 2008.

Research Interests

Pervasive Computing
Semantic Web
Fog Computing