Muhammad Usman
Vallipuram Muthukkumarasamy
Xin-Wen Wu
Surraya Khanum

Mobile Agent-Based Anomaly Detection and Verification System for Smart Home Sensor Networks

Springer

Mobile Agent-Based Anomaly Detection and Verification System for Smart Home Sensor Networks

Muhammad Usman
Vallipuram Muthukkumarasamy
Xin-Wen Wu · Surraya Khanum

Mobile Agent-Based Anomaly Detection and Verification System for Smart Home Sensor Networks

Springer

Muhammad Usman
Department of Computer Sciences
Quaid-I-Azam University
Islamabad
Pakistan

Vallipuram Muthukkumarasamy
School of Information and Communication Technology
Griffith University
Gold Coast, QLD
Australia

Xin-Wen Wu
School of Information and Communication Technology
Griffith University
Gold Coast, QLD
Australia

Surraya Khanum
Department of Computer Sciences
Quaid-I-Azam University
Islamabad
Pakistan

ISBN 978-981-10-7466-0 ISBN 978-981-10-7467-7 (eBook)
https://doi.org/10.1007/978-981-10-7467-7

Library of Congress Control Number: 2017962994

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Dedicated to the research community...

Preface

The rapid technological developments in microelectronics and associated technologies have realized contemporary networking and computing paradigm, viz. shared sensor networks. This paradigm primarily relies on tiny sensor nodes, as key building blocks, to form a number of applications such as smart transport system, smart home, smart cities, smart irrigation system, and infrastructure and environment monitoring. The tiny sensor nodes, in the above-cited application domains, are vulnerable to in situ attacks, errors, and faults. On the similar account, the data sent by tiny nodes in the form of sensor readings is susceptible to transit attacks and errors. A multi-aspect and comprehensive anomaly detection and verification system is, therefore, desired to aptly identify anomalies (or abnormalities) and convey this information to a central node. The system is known as abnormality identification and confirmation system in the subsequent discussion. The contemporary abnormality identification systems are unable to accurately detect the causes of abnormalities. The solitary focus of existing systems is on the identification of abnormalities. To determine the root causes of abnormalities is imperative to remove them.

This book has elucidated an on-the-spot confirmation service for sensor networks, which leverages from the mobile agent technology to ascertain the root cause of abnormalities. A detailed system, which is not only able to detect abnormalities but can also identify the root cause of abnormalities, is introduced for smart home sensor networks. The system empowers mobile agents to employ data which is received through a synchronized resource management technique to carry out the in situ analysis of susceptible nodes. The synchronized resource management technique allows tiny nodes to share statuses of their resources with related cluster leader nodes for better network resource administration. Moreover, the key proposition of the work presented in this book is to use the information received through the synchronized resource management technique to identify numerous kinds of resource-consumption status-related abnormalities. Another key aim of the presented system is to maximize the usage of received synchronized resource management technique-based observations for abnormality identification. In this account, the statistical relationships between varied features of interest are exploited to identify abnormalities which occur due to faults on nodes and exhaustion of

resource and denial-of-sleep attacks. The system employs the data received from synchronized resource management technique-based observations using mobile agents to verify the root causes of abnormalities.

The frequent transmissions of mobile agents cannot be performed due to the fact that transmission is an energy-expensive operation as compared to processing operation. To solve this problem, two methods, namely weighted-sum optimization and 2-sigma, are presented. The nature of the proposed effective mobile agent transmission methods is generic. The proposed methods, therefore, can also be employed by other mobile agent-enabled applications for wireless sensor networks. This book has also introduced a mobile agent-enabled method that performs abnormality identification and confirmation using cross-layer features. It employs fuzzy logic and cross-layer optimization techniques to identify cross-layer abnormalities and optimize mobile agent transmission. A regions computation technique is presented, which employs statistical methods to facilitate decision making about mobile agent transmission and abnormality identification. A cross-layer rule-base, based on the fuzzy logic, is presented along with algorithmic specifications to identify abnormalities and to facilitate transmissions of mobile agents after taking into account the communication link states.

A non-validated system design may adversely affect the resources of a wireless sensor network or even it may go into a standstill state. Therefore, this study extends the theory of Petri net to the formal characterization and investigation of an abnormality identification and confirmation system which employs mobile agent technology in tiny resource-constrained sensor networks. Formal definitions, of the presented system, using standard Petri net, are elucidated to formalize and verify the behavioral characteristics and also flow of the work of the presented methods. A Generalized Stochastic Petri net (GSPN) model is formulated to study the time-based conduct of the presented system in an immensely non-deterministic communication environment of wireless sensor networks. The formal behavior is then verified by experiments that are carried out on a real test bed. The performance of the proposed methods is thoroughly analyzed through theoretical analyses, experimentation on a real test bed, extensive simulations, and comparisons with related schemes. The results indicate the abilities of the proposed methods to detect different nature of abnormalities with high accuracies and increase network lifetime by optimizing mobile agent transmission in addition to effectively identifying the sources of abnormalities.

This book has focused on a single node mobile agent itinerary model. In future work, the proposed work could be extended to a multi-node mobile agent itinerary model in large-scale networks. Another possible extension could be the exploitation of higher-order joins to detect more complex natures of abnormalities.

Islamabad, Pakistan — Muhammad Usman
Gold Coast, QLD, Australia — Vallipuram Muthukkumarasamy
Gold Coast, QLD, Australia — Xin-Wen Wu
Islamabad, Pakistan — Surraya Khanum

Acknowledgements

The work elucidated in this text is based on the study carried out by the lead author under the supervision of co-authors Prof. Vallipuram Muthukkumarasamy and Dr. Xin-Wen Wu. Ms. Surraya Khanum has been involved in refining and improvement in certain aspects of the research work.

Authors are deeply indebted to Associate Professor Farooq Ahmed, a former colleague of the lead author, for introducing him to a rich body of knowledge, namely Petri net theory, which ultimately stimulated authors to explore structural and system-specific behavioral properties of the elucidated methods.

Authors acknowledge the generous financial support provided by Griffith University to fully fund this research study and to provide several travel grants to present and publish findings in multiple conferences and also publish in several top-tier journals. Authors would also like to thank Mrs. Robynee Barnes and others of GELI for their help in improving the presentation of this book.

Authors are also indebted to current and past members of the Network Security Research Group, Griffith University, for their criticism and suggestions which have helped authors to improve the quality of the work presented in this text.

Contents

About the Authors

Dr. Muhammad Usman received his Ph.D. from the School of Information and Communication Technology, Griffith University, Australia. He has obtained Juniper Networks, USA, certifications as an Internet specialist and Internet associate in enterprise routing and switching. He is a member of the Network Security Research Group and the Institute for Integrated and Intelligent Systems (IIIS), Griffith University, Australia. He is also a member of the Computer Science Teacher Association endorsed by the Association for Computing Machinery (ACM), USA. He is currently associated with the Department of Computer Sciences, Quaid-I-Azam University, Pakistan, as an Assistant Professor. His current research interests are security and privacy, cloud computing, Internet of things, distributed systems, and modeling and analysis. He has published over twenty-five research papers for international journals and conferences including prestigious journals such as IEEE Transactions on Consumer Electronics. He has been a recipient of several honors, awards, and grants throughout his industrial and academic career.

Dr. Vallipuram Muthukkumarasamy obtained his B.Sc. in Engineering from the University of Peradeniya, Sri Lanka, and his Ph.D. from Cambridge University, England. He is currently attached to the School of Information and Communication Technology, Griffith University, Australia, as an Associate Professor. His current research areas include the investigation of security issues in wireless networks, sensor networks, trust management in mobile ad hoc networks (MANETs), key establishment protocols and medical sensors. He currently heads the Network Security Research Group at the Institute for Integrated and Intelligent Systems at Griffith University. Also providing leadership with regard to innovative learning and teaching practices, he has received a number of best teacher awards.

Dr. Xin-Wen Wu received his Ph.D. from the Chinese Academy of Sciences, Beijing. He has worked in the University of California, San Diego (as a postdoctoral researcher), the Chinese Academy of Sciences, and the University of Melbourne (as a research fellow). He was also affiliated with the University of Ballarat, Australia. He joined Griffith University, Australia, in 2010 as a faculty

member at the School of Information and Communication Technology. His research interests include cyber security and data privacy, applied cryptography, coding techniques, and information theory and its applications. He has published extensively in these areas, including 3 books and over 80 research papers in leading journals of IEEE, Springer, and Elsevier, in addition to proceedings of international conferences. He is a senior member of IEEE.

Ms. Surraya Khanum received her M.S. (Computer Science) from International Islamic University, Islamabad, Pakistan. She has been associated with Griffith University, Australia, as a visiting research associate. She has also worked as a Lecturer at King Khalid University, Saudi Arabia, and as a Visiting Lecturer at Department of Computer Sciences, Quaid-I-Azam University, Pakistan. She has published numerous research papers, predominantly in the domains of mobile agent-based distributed systems and intrusion detection systems.

Acronyms

AA	Anomaly verification Agent
ADM	Anomaly Detection Module
ADVM	Anomaly Detection and Verification Module
AIS	Artificial Immune System
ARIMA	Auto Regressive Integrated Moving Average
ART	Adaptive Resonance Theory
AU	Aggregation Unit
BS	Base Station
BY	BatterY status
CAP	Contention Access Period
CFP	Contention Free Period
CLN	Cluster Leader Node
CRC	Cyclic Redundancy Check
CRM	Coordinated Resource Management
CU	Coordination Unit
DoS	Denial of Service
DTQ	Data Transmission Quality
DWT	Discrete Wavelet Transform
ECG	ElectroCardioGram
EEPROM	Electrically Erasable Programmable Read-Only Memory
EM	Expectation Maximization
GA	Genetic Algorithm
GEP	Gene Expression Programming
GPS	Global Positioning System
GSPN	Generalized Stochastic Petri Net
GTS	Guaranteed Time Slots
HMM	Hidden Markov Model
IDS	Intrusion Detection System
LEACH	Low-Energy Adaptive Clustering Hierarchy
LIFO	Last In First Out

LQI	Link Quality Indicator
MA	Mobile Agent
MAC	Medium Access Control
MAS	Mobile Agent Server
MAW	Mobile Agent Watermarking
NA	Nodal Agent
PAN	Personal Area Network
PCA	Principal Component Analysis
PER	Packet Error Rate
PHY	PHYsical layer
PN	Petri Net
RAM	Random Access Memory
RERR	Route ERRor
RF	Radio Frequency
ROC	Receiver Operating Characteristic
ROM	Read-Only Memory
RSSI	Received Signal Strength Indicator
RTS	Ready To Send
SA	Static Agent
S-MAC	Sensor-Medium Access Control
SR	Sensor Reading
SSH	Secure SHell protocol
SVM	Support Vector Machine
TDMA	Time Division Multiple Access
UoD	Universe of Discourse
VNL	Victim Node List
WSN	Wireless Sensor Network

Notations

Table 1 Notations and their definitions

Notation	Definition
a to f	User set adjustment parameters
a^* to f^*	Variables to compute domains of fuzzy numbers
A	Anomalous fuzzy number
A_r^l and A_r^r	Left and right bounds of the anomalous region
AA	Anomaly verification agent
AO	Anomalous observations
AS	Action set
A_{rep}	Application repository
A_{data}	Application data
A_{unt}	Aggregation unit
B_c^l	Battery current level
B_t^l	Battery threshold level
BA	Anomalous behavior
BT_η	Tolerated category 1 behavior
BT_δ	Tolerated category 2 behavior
BT_ζ	Tolerated category 3 behavior
Beh	msn_q behavior
CU	Coordination unit
cln_q	q_th cluster leader node
d^ag	Aggregated sensed data
d^al	Anomaly alarm
d_i^ag	i_th Aggregated sensed data
d_i^al	i_th anomaly alarm
E	Edges denoting communication links

(continued)

(continued)

Notation	Definition
f	Received packet count
f_i	i_th feature
F	Set of arcs
a to f	User-defined adjustment variables
a to f	User-defined adjustment variables

Table 2 Notations and their definitions

Notation	Definition
F_q	Collection of values of features of q_th node
FS	Features of interest
FS_1	Features λ, ϕ, and v
FS_2	Features $\imath$ and f
G	A graph denoting a smart home sensor network
h	Number of historical observations used to compute agent transmission score
$H(.)$	Inhibition function
$I(.)$	Input function
msn_q	q_th cluster member node
m_{fx}^{fq}	Value of a fixed value feature
m_{rg}^{fq}	Value of a continuous random variable feature
M_j	j_th marking state
$M_j^{'}$	A marking state other than j
M_0	Initial marking state
N	Normal fuzzy number
$N(\lambda, \imath)$	First-order join for in situ fault or attack
$N(\imath, v)$	First-order join for resource exhaustion attack
$N(\phi, v)$	First-order join for fault on node and attack on resource node
$N(\phi, \imath)$	First-order join for faulty node
$N(f, \imath)$	First-order join for denial-of-sleep attack and faulty node
$O^+(.)$	Output function
O_j	j_th observation
p_i	i_th place
P	Set of places
P_{fx}^{fq}	Normal profile value of a fixed value feature
P_{rg}^{lb}	Lower bound of a continuous random variable feature
P_{rg}^{ub}	Upper bound of a continuous random variable feature
Prf_q	Normal profile bound of q_th node
R	In situ verification result

(continued)

(continued)

Notation	Definition
RP	Repository
RS	Resource status
$RM(M_0)$	Reachable marking from initial state
s^l and s^r	Left- and right-side standard deviation values
S^t_{AA}	Anomaly verification agent transmission score
S_{msn_q}	Historical observation score to transmit anomaly verification agent
S_msn_q	Segment of the stack memory of msn_q
S_agnt_q	Segment of the stack memory of $agnt_q$
t_i	i_th transition
T	Tolerance fuzzy number
T	Set of transitions
T_i^{lb}	Start time of a timeslot
T_i^{ub}	Finish time of a timeslot
T_{ar}^{Fq}	Function to compute time of arrival of Fq
T_{ar}^{WR}	Time of arrival of a watermarked result
TR	Trust value
$T_{ar}(F_q)$	Function to compute time of arrival of Fq
V	Vertices denoting nodes
V_1	The laptop-class node (top-level node)
V_2	Cluster leader node (intermediate-level node)
V_3	Cluster member node (leaf-level node)
WR	Watermark
$W(.)$	Rate or weight for timed or immediate transitions
$\Pi(.)$	Priority function
α_1	Weighting factor for tolerated instance of f_i
α_2	Weighting factor for anomalous instance of f_i
σ	Firing sequence

Table 3 Notations and their definitions

Notation	Definition
λ	Minimum to maximum bounds to sensor reading
ι	Time interval
ϕ	Values of entitled actions by cluster member node
v	Resource status of cluster member node
κ	Anomaly detection action
τ	Tuning action
$\psi, \zeta, \delta, \eta$	Thresholds for agent transmission

Chapter 1
Introduction

A fundamental feature of resource-limited wireless sensor networks (WSNs) which distinguishes them from traditional networking paradigms is their data-centric nature. Reliable data reception is of paramount importance in order to accomplish application-specific tasks in WSNs. The classical approach for abnormality identification is to detect abnormalities in received data at a central or distributed locations in a network. This approach is, however, unable to identify the root cause of abnormalities, that is, if abnormalities are occurred on-the-spot or in transmission. The work elucidated in this book has introduced an innovative approach to confirm the root cause of abnormalities by employing mobile agents (or merely agents for brevity). The detailed abnormality identification and confirmation system along with its underlying methods is designed, evaluated, and analyzed throughout this book.

1.1 Overview

1.1.1 Wireless Sensor Networks

Wireless sensor networks (sensor networks for brevity) are formed of numerous tiny nodes (or motes) [1]. A mote typically has one or more sensors (i.e., transducers) and analog-to-digital converters. The transducers periodically or continuously sense a physical phenomenon (depending upon application requirements) and convert physical or environmental parameters into electrical signals. This signal is then passed through a conditioning phase which performs various functions such as filtering and analog-to-digital conversion. The digital signal is then used by a tiny processor for application-specific processing. Sensed data is typically forwarded to a base station, a central authority, through a multihop or single-hop communication link. A user (or a

M. Usman et al., *Mobile Agent-Based Anomaly Detection and Verification System for Smart Home Sensor Networks*,
https://doi.org/10.1007/978-981-10-7467-7_1

system administrator) can then retrieve and analyze received sensed data through a user application.

Recent advancements in the domain of information technology have enhanced the use of sensor networks in several application domains such as smart irrigation systems, health monitoring, smart home, smart cities, smart transport systems, and industrial monitoring and process control [2]. These applications are briefly discussed below.

Smart home: Tiny sensors are used for smoke detection, automated doors and windows, motion detection, automatic air condition switches, and many such purposes in a smart home environment. These sensing devices form a smart home senor network which is managed locally or remotely through a smart phone or computing device application.

Built infrastructure monitoring: Frequent physical inspections and monitoring of a built infrastructure are not possible due to the growth in the numbers of construction of buildings, bridges, and other mechanical or civil structures in recent years. Sensors are, therefore, deployed for structural health monitoring, damage detection, and even for an automated control of structures.

Health monitoring: Advancements in healthcare facilities including medicines have caused a rapid decrease in morbidity rates in recent years. This has resulted in an increase in an aging population. The implanted, wearable, and ambient monitoring sensors form a class of sensor networks, namely body area networks, to offer health monitoring services for the elderly.

Industrial monitoring and process control: Increasing a productivity and safety of manufacturing plants and staff are growing requirements for many large-scale organizations. Tiny sensors are used for industrial automation, process control, and manufacturing monitoring to meet productivity and safety requirements.

Smart cities: Sensor networks are one of the essential constituent technologies of smart cities. Sensors can be used to monitor traffic, street lights, environment, transportation, water consumption, electricity consumption, and gas consumption in smart cities.

Sensor networks offer a unique and coherent amalgamation of distributed sensing, processing, and communication capabilities. This characteristic distinguishes sensor networks from the rest of the networks. Some other distinguishing features of sensor networks are discussed below. Mote, fundamental building blocks of sensor networks, has low battery power, small memory, low computational capacity, and limited communication range. Sensor networks have different deployment types such as flat or cluster-based and fixed or mobile deployment. The nature of motes with respect to resources in sensor networks is either a homogeneous or heterogeneous. Motes interact with their environment in order to sense and report a physical phenomenon to a cluster leader or base station. The data traffic pattern of sensor networks is, therefore, different from traditional networks. Sensor networks may have bursty data traffic patterns in some applications, that is, motes detect an event such as fire

and then continuously track and report it to a central node [3]. Alternatively, in some applications such as built infrastructure monitoring, motes may periodically report the status of a structure to a central node [4]. Other distinguishing features of sensor networks include an application-orientated nature, the absence of network-wide global identities, and the data-centric nature of networks. These unique characteristics demand a carefully articulated design of the sensor network application in order to effectively achieve application-specific goals.

1.1.2 Agents in Sensor Networks

A software agent is typically composed of data and code [5]. It can roam among different motes in order to perform its designated task. An agent works autonomously, and it may communicate with other agents and entities such as other motes, cluster leaders, and a base station in a network. There are two types of agents, namely *weak* and *strong* with respect to mobility [6]. In the case of weak mobility, only states of code and data are maintained, that is, a new execution of an agent starts on the next mote. On the other hand, in the case of strong mobility, the state of the execution of an agent is halted at a random point in time on a mote and then continues it on the next mote. The execution of agent resumes from the state it was stopped on the previous mote. Note that in the case of strong mobility, data, execution state (i.e., status of stack and program counter), and code are all transmitted from one mote to another mote.

Sensor networks have been traditionally based on a client-server computing model. In this computing model, motes gather sensed data from a physical phenomenon and transmit it to a corresponding cluster leader or base station depending upon the network topology. Sensor networks, however, typically have a low network bandwidth, which can cause congestion or a bottleneck problem at cluster leader or base station motes. In order to solve such problems, Qi et al. [7] suggested the use of agents for multiresolutional data integration in resource-limited networks.

Over the years, agents have been successfully employed in sensor networks in order to perform a number of tasks, for instance, parallelism, code and data dissemination, localization, fusion of distributed data, collaborative signal and information processing, and security services [5, 7]. A few prominent benefits of agents are given below [8]. (i) They can take computing operations to the sources of data instead of anticipating transfer of raw data over unreliable links. This can decrease the costs of energy which are otherwise incurred. (ii) Agents reduce network latency. (iii) Agents are autonomous entities, as they are able to perform asynchronous operations. This feature enables the initiator motes of agents to perform other operations. (iv) Finally, agents add the properties of fault tolerance and robustness in a system.

Despite several advantages, there are a few challenges which should be considered before deploying agents in sensor networks. The key challenges are controlling communication cost, the size of the agent, and the inter-mote movement of the agents. Transmission is an expensive operation as compared to a processing operation. A

transmission of a single bit costs as much energy as is consumed by an execution of 800–1000 instructions [9]. The size of agents may affect both energy and memory resources. Agents, therefore, should be efficiently programmed in order to minimize their impact on the memory of a sensor mote. Memory techniques such as virtual memory, stack, and last in, first out (LIFO) may be adopted in order to accommodate a large-sized agent or multiple agents on a single mote. An inter-mote agent movement may pose security threats. Therefore, an appropriate security mechanism must be incorporated in order to secure agents against antagonist motes and legitimate motes against adversary agents.

1.2 Motivation

Over the course of the last decade, sensing technology has appeared as a prominent research domain in information technology, largely due to its popularity among numerous applications. This fact has motivated the research community to design innovative solutions for sensor networks. The work carried out in this study is an attempt to design, evaluate, and analyze a robust system with its underlying methods for sensor networks. The nature of the presented system is generic, so that it can be deployed on most of the periodically sensed data transmitting applications. The proposed system and underlying methods are, however, tailor-made for the two applications, namely *built environment monitoring* [10] and *smart home sensor network* [11].

In a built environment monitoring application, motes are deployed on multiple buildings in order to monitor usages of water, electricity, gas, and also emissions of carbon dioxide (CO2). Motes transmit sensor readings to their respective cluster leaders after a specified interval of time. Cluster leaders then forward aggregated data to a base station. A user can make decisions by performing an analysis on received data. On the other hand, as discussed in the previous section, motes such as gas sensors and temperature sensors sense their environment and transmit readings to a central node in a sensor network. Sensed data, at a central node, is then used to know the current state of a sensed phenomenon and to take appropriate action(s), if required. A typical smart home sensor network is depicted in Fig. 1.1.

In both of the above-stated practical applications, low resource motes are designated to perform a sensing task and then to report that sensed data to respective resource-rich cluster leaders or a base station mote. The desired performance of both motivating applications is largely reliant on correctness of values of the obtained readings at a central supervisory mote. Motes along with their transmitted sensor readings are, however, susceptible to both types of abnormalities, that is, in situ (on-the-spot) and in transit (in transmission). These abnormalities can be caused by numerous factors such as errors, attacks, and faults. Application-specific goals of a sensor network cannot be achieved without timely detecting inconsistencies in received sensed data. This study, therefore, has presented an abnormality identification and confirmation system which not only can detect abnormalities with high accuracy, but also can ver-

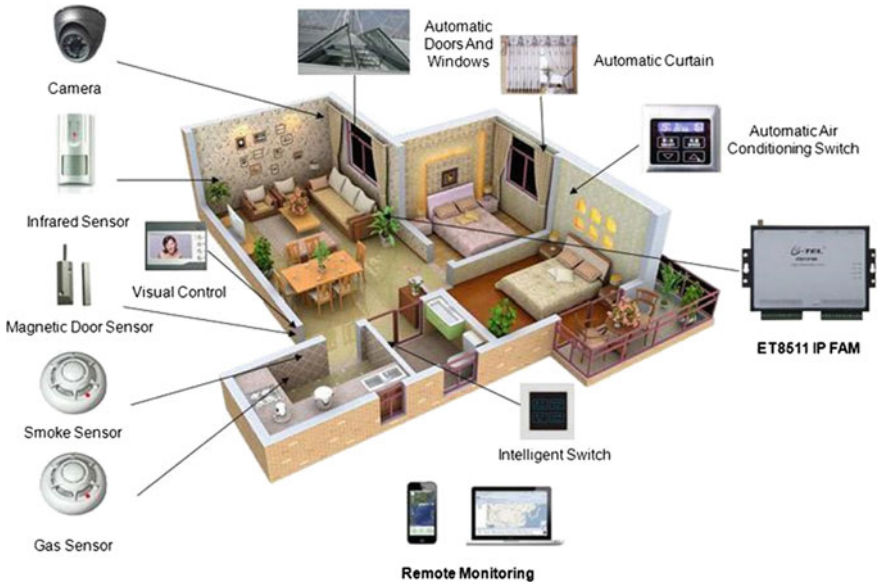

Fig. 1.1 Smart home sensor network

ify the source of abnormalities to effectively counter them. The proposed system has exploited a heterogeneous nature of devices in sensor networks. The system is set up that most of the abnormality identification and on-the-spot confirmation processing are performed by resource-rich motes.

1.3 Problem Domain

Abnormality identification systems are typically deployed to find abnormal and infrequent patterns in network data [12]. Over recent years, the research domain of the abnormality identification in resource-constrained networks has fascinated the research community [13–16], mainly because of its significance to provide key services such as secure working of a network, event reporting, and data reliability. This study, however, has focused on the two main services, namely *fault detection* and *intrusion detection*.

A well-equipped and well-designed abnormality identification and confirmation system must be capable of identifying the root causes of occurrence of abnormalities after their detection in order to effectively mitigate them. Nevertheless, the existing schemes are only designed to detect abnormalities [13–16], which severely limits their effectiveness. The identification of sources of abnormalities can be carried out through on-the-spot confirmation of a susceptible mote. The on-the-spot confirmation of a susceptible mote can be typically carried out by the physical confirmation of

a malicious mote. However, recurrent physical accesses to motes are not always possible, especially when they are positioned on hazardous terrains or within the premises of privately owned buildings and infrastructures. The approach of a physical diagnosis is also time-consuming.

This study has, therefore, introduced an automated and efficient service to perform on-the-spot confirmation of the mistrustful behavior of a mote using agents. A detailed system, with underlying methods, is designed. The system uses information received through a synchronized resource management scheme to detect different natures of abnormalities. The synchronized resource management scheme facilitates the sharing of resource status by motes with their cluster leaders for better resource management. An operating system for sensor networks, namely TinyOS, facilitates the synchronized resource management mechanism, also known as the Coordinated Resource Management (CRM) mechanism, by means of low-level interfaces in order to administer and share state of the hardware of a mote over the network [17, 18]. This study has employed synchronized resource management scheme-based information to detect different nature of abnormalities such as those triggered by anomalous sensor reading values and memory and battery statuses of motes. The statistical relationship among features of the synchronized resource management scheme-based observations is investigated to identify more complex nature of abnormalities such as produced by resource exhaustion and denial-of-sleep attacks, and mote faults. The synchronized resource management scheme-based observations are further used by agents to confirm the cause of abnormalities.

The cautious transmission of agents is desired in sensor networks because of the resource-limited characteristics of tiny motes (as discussed in previous section). In order to carefully but effectively transmit agents, the two methods, namely *2-sigma* and *weighted-sum optimization*, are elucidated. This study has also presented a cross-layer scheme which can discover cross-layer abnormalities and can also effectively transmit agents after consulting the current state of a communication link. An underlying regions computation method is presented in order to define different regions over a cross-layer feature space to facilitate the processes of agent transmission and detection of abnormalities. A corresponding cross-layer fuzzy logic rule-base is formulated, and algorithmic specifications are presented for agent transmission and detection of cross-layer abnormalities.

The completeness and correctness are key properties that need to be verified before the deployment of a system. A system, without its validation, may go into a halt state or it may not be able to satisfy the user requirements. Formal methods and formal modeling languages, such as Petri nets, specify formal definitions and facilitate the verification of large and complex systems [19]. This study, therefore, extends the Petri net theory to address and formally verify the algorithmic properties of the system. Furthermore, not considering the effect of temporal behavior of the time-sensitive abnormality identification and identification systems may reduce their utility after their deployment. Therefore, the Petri net model verification is performed to formulate a Generalized Stochastic Petri net (GSPN) model to examine time-based conduct of the presented system. This process has also helped in enabling the system to detect temporal abnormalities which are caused because of late arrivals

of data packets containing on-the-spot confirmation results and values of features. The theoretical analyses, simulations, experiments on a real test bed, and comparison with related schemes are performed to thoroughly examine the performance of the methods in different settings.

1.4 Book Organization

This chapter has dealt with an overview of sensor networks, agents, motivating applications, and the problem domain for the study. The book is structured as described below. Chapter 2 overviews related work, identifies limitations in works presented in the related literature, and highlights the context of this study. Chapter 3 presents the network model, an architecture of abnormality identification and confirmation module, and algorithmic specifications of the presented system. Chapter 3 also details the formal specifications, formal characterization and analysis, temporal behavior modeling, and temporal behavior validation of the system through implementation on a real test bed. Chapter 4 explains the methods for the first-order abnormalities and agent transmission optimization. The formal specifications and verification of the methods, simulation and implementation results, and findings of a comparative study are also discussed. Chapter 5 elaborates on the fuzzy logic-based cross-layer system along with its theoretical and implementation results. Chapter 6 concludes the key findings and outlines the future directions of this study.

References

1. F. Li, J. Luo, W. Wang, Y. He, Autonomous deployment for load balancing k-surface coverage in sensor networks. IEEE Trans. Wirel. Commun. **14**(1), 279–293 (2015)
2. B. Fateh, M. Govindarasu, Joint scheduling of tasks and messages for energy minimization in interference-aware real-time sensor networks. IEEE Trans. Mob. Comput. **14**(1), 86–98 (2015)
3. M.A. Serna, A. Bermudez, R. Casado, Hull-based approximation to forest fires with distributed wireless sensor networks, Melbourne, Australia (2013), pp. 265–270
4. L. Lin, C. Yang, K.J. Wong, H. Yan, J. Shen, S.J. Phee, An energy efficient mac protocol for multi-hop swallowable body sensor networks. Sensors **14**(10), 19457–19476 (2014)
5. M. Dong, K. Ota, L.T. Yang, S. Chang, H. Zhu, Z. Zhou, Mobile agent-based energy-aware and user-centric data collection in wireless sensor networks. Comput. Netw. **74**, 58–70 (2014)
6. A. Fuggetta, G.P. Picco, G. Vigna, Understanding code mobility. IEEE Trans. Softw. Eng. **24**(5), 342–361 (1998)
7. H. Qi, S.S. Iyengar, K. Chakrabarty, Multiresolution data integration using mobile agents in distributed sensor networks. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **31**(3), 383–391 (2001)
8. D.B. Lange, M. Oshima, Dispatch your agents; shut off your machine. Commun. ACM **42**(3), 88–89 (1999)
9. S. Khan, A.-S.K. Pathan, N.A. Alrajeh, *Wireless Sensor Networks: Current Status and Future Trends*, 1st edn. (CRC Press Inc., Boca Raton, FL, USA, 2012)
10. E.U. Gaura, J. Brusey, R. Wilins, Barnham J., Wireless sensing for the built environment: enabling innovation towards greener, healthier homes, United Kingdom (2011), pp. 1–6

11. Y. Xue, X. Chang, S. Zhong, Y. Zhuang, An efficient energy hole alleviating algorithm for wireless sensor networks. IEEE Trans. Consum. Electron. **60**(3), 347–355 (2014)
12. S. Han, M. Xie, B. Tian, S. Parvin, Anomaly detection in wireless sensor networks: a survey. J. Netw. Comput. Appl. **34**(4), 1302–1325 (2011)
13. C. O'Reilly, A. Gluhak, M.A. Imran, S. Rajasegarar, Anomaly detection in wireless sensor networks in a non-stationary environment. IEEE Commun. Surv. Tutor. **16**(3), 1413–1432 (2014)
14. M. Moshtaghi, C. Leckie, S. Karunasekera, S. Rajasegarar, An adaptive elliptical anomaly detection model for wireless sensor networks. Comput. Netw. **64**, 195–207 (2014a)
15. H. Kumarage, I. Khalil, Z. Tari, A. Zomaya, Distributed anomaly detection for industrial wireless sensor networks based on fuzzy data modelling. J. Parallel Distrib. Comput. **73**(6), 790–806 (2013a)
16. M. Xie, J. Hu, S. Han, H.-H. Chen, Scalable hypergrid k-nn-based online anomaly detection in wireless sensor networks. IEEE Trans. Parallel Distrib. Syst. **24**(8), 1661–1670 (2013a)
17. J. Waterman, G.W. Challen, M. Welsh, Peloton: Coordinated resource management for sensor networks, Switzerland (2009), pp. 1–5
18. P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, D. Culler, Tinyos: an operating system for sensor networks, in *Ambient Intelligence*, ed. by W. Weber, J.M. Rabaey, E. Aarts (Springer, Berlin, 2005), pp. 115–148
19. F. Ahmad, I. Fakhir, S.A. Khan, Y.D. Khan, Petri net-based modeling and control of the multi-elevator systems. Neural Comput. Appl. **24**(7–8), 1601–1612 (2014)

Chapter 2
Background

A number of novel threats have been developed for low resource networks because of advancements in technology and innovations in attack techniques in the recent years. Prevention-based methods alone, therefore, may not offer a comprehensive security and fault detection solution for sensor networks. Aptly formed detection-based methods can supplement prevention-based methods to provide more robust security mechanisms. Abnormality identification is a detection-based method that is employed for providing a security service, namely intrusion identification. It may also be employed in other application domains such as object tracking and fault detection [1]. A key focus of this study is to design and analyze a system which can detect abnormalities that are typically originated due to faults and intrusions on tiny nodes and the data transmitted by them.

This chapter reviews related previous studies in order to underline their limitations and highlight the research context of this study. Attack types and sensor network security mechanisms are briefly reviewed first to highlight the position of detection of abnormality mechanisms in the sensor network defense. The abnormality identification preliminaries and the types of abnormalities in sensor networks are then discussed. An objective of this study is to design, evaluate, and analyze a system to detect and verify abnormalities in sensor networks. Different abnormality identification schemes for sensor networks are, therefore, reviewed and critically analyzed. The related previous studies which employ agents in abnormality identification applications for sensor networks are also reviewed. A critical review on the related research on the cross-layer abnormality identification, security of agents, and formal modeling and verification is also presented toward the end of this chapter in order to set the research context of this study.

M. Usman et al., *Mobile Agent-Based Anomaly Detection and Verification System for Smart Home Sensor Networks*,
https://doi.org/10.1007/978-981-10-7467-7_2

2.1 Sensor Network Security

As discussed in the preceding chapter, the nature of sensor networks is different from that of traditional computer networks. One of the foremost distinguishing characteristics is the limited resources of sensor networks. Abnormality identification is a well-studied domain in traditional wireless and wired networks, but because of the distinctive nature of sensor networks, these methods may not be suitable for resource-limited networks. Another key characteristic which needs to be considered while articulating an abnormality identification system for sensor networks is their wireless communication channels. Sensor networks typically operate on a broadcast communication mode, which makes them susceptible to a variety of attacks. Sensor networks may also adopt a point-to-point communication mode based on application requirements. Continuous surveillance of sensor nodes in applications where they are deployed on difficult terrains is not feasible in order to protect them from physical attacks. Sensor nodes are typically deployed in a large scale. As a consequence, it is not adequate to design a centralized security or fault detection mechanism. A decentralized and collaborative security mechanism is, therefore, more suitable for sensor networks in such cases.

Sensor networks may suffer from abnormalities due to attacks, faults, and errors [2, 3]. The attack-based threats for sensor networks are classified into three categories, namely *Insider versus outsider*, *Active versus passive*, and *Laptop-class versus mote-class attacks* [4].

Insider versus outsider attacks: The former type of attacks are performed by legitimate nodes, whereas latter attacks are initiated by external nodes.

Active versus passive attacks: In active attacks, an adversary node modifies or fabricates a network traffic, whereas in passive attacks it just monitors or eavesdrops transmitting data packets.

Laptop-class versus mote-class attacks: In laptop-class attacks, an antagonist uses a more powerful device such as a laptop-class device to carry out attacks, whereas nodes with similar capabilities to victim nodes are used to launch mote-class attacks.

In another threat classification, Zia and Zomaya [5] introduced four threat classes, namely *interruption*, *interception*, *modification*, and *fabrication* of Fleeger [6] in sensor networks.

Interruption: A communication link between two or more nodes is lost in an interruption threat. Examples can be node capture and jamming attacks.

Interception: A whole network or a part of a network compromises by an antagonist node. For example, node capture, denial-of-sleep, resource exhaustion, black hole, and sinkhole attacks.

Modification: An antagonist node captures and modifies data in a modification threat. Example attacks are denial of service (DoS) and flooding attacks.

Fabrication: In this class of threats, an adversary node injects false data in a network traffic. A simple example of fabrication threats can be a sybil attack.

There are a number of measures which can be taken to secure sensor networks from vulnerabilities and attacks. Some of the key security services are secure local-

ization and data aggregation, secure auditing, cryptography, key management, secure routing, access control, and abnormality identification. A choice of a security service mainly emerges from security requirements and available resources of network. Ngai [7] classifies security measures in three broad categories, namely *protect*, *detect*, and *react*.

Protect: This network defense approach is based on biometrics, firewalls, and cryptographic techniques.

Detect: The security services like intrusion and abnormality identification belong to the detect class of defense mechanisms. The study focuses on identification and confirmation of sources of abnormalities which are caused by attacks and faults on sensor nodes and their transmitted data.

React: This type of sensor network defense mechanism usually functions after detection of attacks or faults in a network. It can be based on action(s) such as decrement in trust count of a node, announcement of a node as a malicious or faulty, minimizing communication with suspicious nodes, and disconnecting a node from rest of the network.

Intrusions are activities which are carried out by antagonists to distract the normal functionality of a node or network. A software system, namely intrusion detection system (IDS), is employed to identify abnormalities. The prevalent IDS systems can be classified on the basis of *detection methodology*, *detection approach*, and *detection location*. The IDS can be categorized into three broad classes with respect to the detection methodologies, namely *protocol state analysis*, *misuse detection*, and *abnormality identification* [8].

Stateful protocol analysis: The priori profiles of the normal activities of the entities of a system are compared with that of real events in the process of stateful protocol analysis to identify abnormalities. This IDS methodology is not commonly used in sensor networks.

Misuse detection: This methodology works on the basis of priori assumptions, experiences, knowledge, and information. It is also called rule-based IDS. In a misuse detection method, signatures of known intrusions are matched with the current events of the system to detect intrusion activities. This method is simple and useful for detecting known attacks, but it cannot identify novel attacks.

Abnormality Identification: This methodology creates a normal profile of a subject on the basis of its usual activities. After that, the normal profile is compared with the current activities of the system. Significant deviation from normal activities is considered as an intrusion. This method can detect novel attacks, but it is complex and resource hungry. One of the focuses of this study is to detect abnormalities by optimally utilizing the resources of sensor networks. There are two types of abnormality identification profiles, namely *dynamic* and *static*. The dynamic one is automatically updated with the observance of new incidents. On the contrary, the static one is typically established by a user and has constant behavior definition. Defining a comprehensive profile that has all normal activities is not a trivial job in sensor networks. In numerous applications, normal behavior constantly evolves. It is, thus, a challenging task to define a usual behavior notion of a system. The

specifications of an application should be considered while designing an abnormality identification system.

The IDS can be deployed either at a *host* or *network* level [9]. A host-based IDS focuses on detecting misuse or abnormal use of the resources of node such as battery, processor, and memory. On the contrary, a network-focused IDS focuses on detecting interferences and abnormalities in network traffic. Sometimes both deployment types are used in conjunction with each other and known as *hybrid* IDS. The detection location of an IDS can be either *centralized* or *distributed*. In a case of a centralized detection location, an abnormality identification module (AIM) is installed on a central node, whereas ADMs are installed on several distributed nodes in the case of a distributed detection type. The work presented in this study employs static detection methodology, a hybrid detection approach, and a distributed location to identify different nature of abnormalities. The problem domain of this study, as a research map, is depicted in Fig. 2.1.

Data, *network*, and *node* are three key types of abnormalities that are prevalent in sensor networks [10]. The data abnormalities are associated with data which are collected by a single or numerous sensors in a sensor mode. Sensor readings, which are accumulated within a certain location, typically show similarity and consistency. A discrepancy in the accumulated data is considered as data abnormality. There may be numerous causes for data abnormalities for instance diverse nature of attacks,

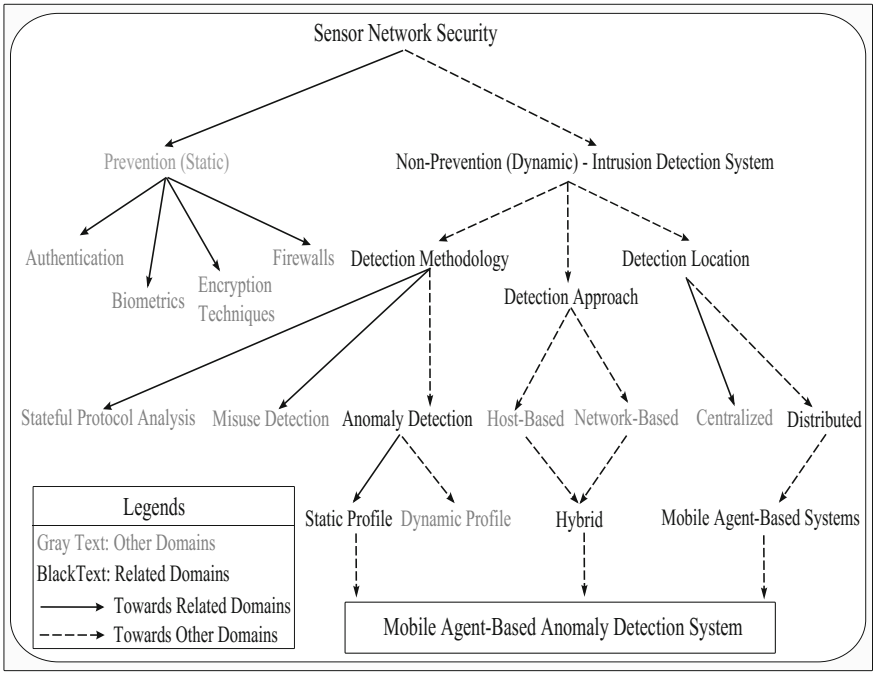

Fig. 2.1 Research map

Table 2.1 Abnormalities and their potential reasons: Republished with permission of Taylor and Francis Group LLC Books, from Security for Multihop Wireless Networks, Muhammad Usman, Vallipuram Muthukkumarasamy, Xin-Wen Wu, and Surraya Khanum, edited by Shafiullah Khan and Jaime Lloret Mauri, 2014; permission conveyed through Copyright Clearance Center, Inc.

Abnormality	Potential reasons
Data	Different nature of attacks, errors, abrupt mote restarts, abnormal events, non-synchronizations, and faults of hardware/software
Node	Degradation of resources, mote restarts and failure, malfunctioning, and physical threats
Network	Connectivity loss, recurrent connectivity, loops in routing, network elements failures, and transmission storms

errors, mote restart, non-synchronization, and software or hardware errors. Data abnormalities are classified into *temporal*, *spatial*, and *spatiotemporal* abnormalities. Physical attacks, node failures, and node malfunctioning are usual causes of node abnormalities. Motes are imitated by cloning in physical attacks. Node abnormalities include resource degradation, node failures, and node resets. Network abnormalities are instigated because of the jamming attack, that is, interference or non-availability in the communication frequency, faults in network-wide elements such as gateways, sensor mote failures. Network abnormalities occur due to exploiting limitations in routing protocols [9]. A complete connectivity loss, loop in routing, recurrent connectivity, and transmission storm abnormalities are also known as network abnormalities. The summary of the types of abnormalities and their potential reasons is given in Table 2.1 [10].

2.2 Abnormality Identification

The research community has examined and employed numerous techniques and methods from several disciplines such as statistics, artificial intelligence, machine learning, data mining, and also from other domains, to design and develop schemes to identify abnormalities. The literature of abnormality identification is, therefore, classified into four classes, namely *statistical*, *artificial intelligence and agent-based*, *learning*, and *other* schemes in this section for a critical review purpose. The representative schemes of each class are discussed in this section. It is pertinent to mention that some schemes may belong to more than one category. They are, however, classified into those classes which are more relevant. Figure 2.2 [11] illustrates a high-level abnormality identification taxonomy in sensor networks.

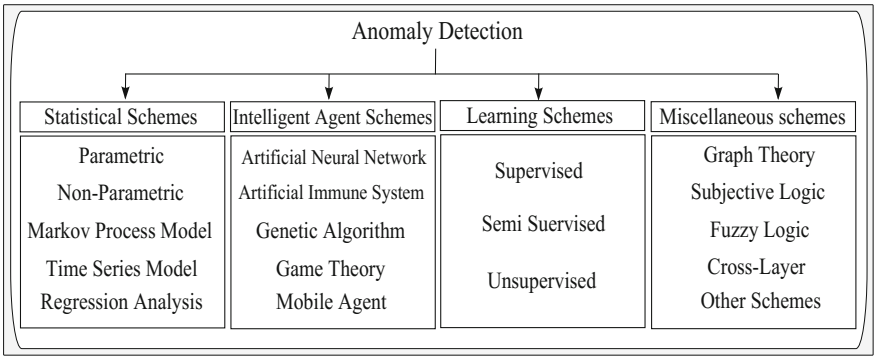

Fig. 2.2 A taxonomy: Republished with permission of Taylor and Francis Group LLC Books, from Security for Multihop Wireless Networks, Muhammad Usman, Vallipuram Muthukkumarasamy, Xin-Wen Wu, and Surraya Khanum, edited by Shafiullah Khan and Jaime Lloret Mauri, 2014; permission conveyed through Copyright Clearance Center, Inc.

2.2.1 Statistical Schemes

Statistical abnormality identification schemes are classical techniques. In a statistical model, relationships among variables are characterized through mathematical relations. This kind of model should hold either of two properties, namely *randomness* or *systematic variation*. A classical division of statistical techniques is *parametric* versus *nonparametric* [1].

2.2.1.1 Parametric Schemes

This kind of techniques relies on a supposition that the underlying data distribution information is known. Another imperative assumption about parametric schemes is normally distributed data. This kind of techniques requires change in parameters after certain period of time. This class may be divided as: *Gaussian* and *non-Gaussian*.

Gaussian Schemes: Gaussian (bell shape or normal) distribution of variables is presumed in the model. An ecological application-focused abnormality identification method is presented to identify incidents and deduce absent readings in data [12]. The fundamental concept of the technique is to identify spatiotemporal association among sensed readings. The technique employs previous knowledge of sensed data to identify abnormalities, as each mote matches its present and past observations with adjoining motes. A fundamental shortcoming of the technique is its sole focus on uni-dimensional abnormalities which occurs because of spatiotemporal data abnormalities.

A scheme presented by Wu et al. [13] determines abnormalities in two dimensions. It first discovers anomalous sensors and then identifies the boundaries of events. The scheme recognizes spatially associated abnormalities by matching median value with

each observation that is obtained from the neighbor motes. The fundamental concept is to deploy algorithm on every mote to identify abnormal readings along the boundaries of events. Abnormal signals and readings, showing occurrence of an incident, may not be differentiated in the scheme. Furthermore, mote which observes same type of incidents should spatially be associated. Therefore, the authors presented two algorithms to identify anomalous motes and detect boundaries of events.

In former algorithm, every mote computes differences between own readings and their median with that of neighbor motes. A mote is declared as abnormal if the deviation is large. The latter algorithm computes two zones for each sensor node. The deviations are then individually computed by employing the former algorithm. If median value of single zone is ominously diverse than other mote, the mote is announced as abnormal. The scheme employs a receiver operating characteristic (ROC) analysis to adaptively calculate the values of thresholds. The ROC is a graphical plot which is typically used to show the performance of a binary classifier [14]. The computation of ROC values is an energy consuming process. Therefore, this approach is not useful for resource-limited motes.

Non-Gaussian Schemes: The distribution of data is not normal in such schemes. Jun and colleagues studied a non-Gaussian abnormality identification scheme which can detect spatiotemporal abnormalities [15]. Abnormalities are presumed as non-correlated with respect to space and time. Abnormalities are characterized as per impulsive noise behavior having distribution, namely Symmetric α-Stable (SαS). Each cluster member mote carries out local abnormality identification and corrects temporally abnormal data which are then transmitted to the respective cluster chief nodes. The cluster chief node then receives that data and further discovers and corrects spatial abnormalities in data. This abnormality identification scheme offers two benefits: (i) a communication cost decrease and (ii) an increase in the quality of accumulated data. In spite of these benefits, the distribution, namely SαS might not be suitable for networks because of their dynamic nature of topology.

2.2.1.2 Nonparametric Schemes

These schemes have no presumptions about data distribution [16]. These schemes are, therefore, more suited to sensor networks with dynamic topology. Nonparametric schemes characteristically use one of two methods, namely *Histogram* and *Kernel functions* [16]. The former provides a probability distribution within a specific range [17]. A histogram is typically composed of rectangular bars. The tallness of a bar is directly proportional to data frequency in a defined range. Histogram represents continuous type of data, not like bar graphs which plot categorical data. A scheme was presented by Xie et al. for hierarchical networks [18]. Their scheme constructs histograms in both an online and a distributed fashion. The new histogram estimate errors are theoretically studied and optimum parameter values are derived for abnormality identification. A main flaw of the scheme is, however, its consideration of only univariate data.

Alternatively, kernel function-based approaches have been studied in the non-parametric class of statistical abnormality identification in the literature. Kernel, a weighting function, is employed for the estimation of the probability density functions of random variables for abnormality identification [19]. A number of kernel function-based detection techniques have been studied over the recent years. Palpanas et al. [20] presented a technique which needs no advance distribution information of sensed data. Every mote locally identifies abnormalities by using an estimator, namely kernel density. The sensed data is abnormal if it is outside user-defined limits on a node in question. A basic limitation of this technique is its only suitability for univariate data.

Subramaniam and colleagues [21] furthered Palpanas et al. [20] work. Their scheme is meant for abnormality identification in multidimensional data. The scheme has two approaches to identify global abnormalities. In the former approach, every mote identifies local abnormalities, as suggested by Palpanas et al. [20]. An abnormality report is then forwarded by a sensor node to its parent node in order to detect further abnormalities. This process continues till the sink mote identifies global abnormalities. On the other hand, in the latter approach, each mote identifies global abnormalities by employing estimator at global level. The empirical evaluation of the scheme demonstrates that it can identify abnormalities with high accuracy. However, this scheme is not able to identify spatial abnormalities.

2.2.1.3 Markov Process Model

The *Markov process* model is one such approach for abnormality identification in which a model of data can be either parametric or nonparametric. The kind of model treats incidents as state variables and uses a matrix, namely transition matrix to characterize the transition frequencies that occurs among states [22]. The abnormal conduct is then identified by matching input and output among two consecutive states. This method is essentially suitable for sequential schemes. A type of the Markov process model, namely *Hidden Markov Model* (HMM) is also employed in statistical abnormality identification. In HMM, transitions and associated states are veiled and merely the productions are observable [23]. A typical HMM is illustrated in Fig. 2.3 [11].

Paschalidis et al. [24] have presented an identification method which can discover both time-based and spatial abnormalities. The method uses tree indexed Markov chains in order to formalize a spatial architecture. The temporal values are replaced with integers on the vertices of the trees in the Markov chains. Large deviations from previous abnormality free traces are then considered as abnormalities. The method uses decision rules in order to differentiate between usual and abnormal activity. This estimate takes to the discovery of both temporal and spatial abnormalities. However, this is not feasible in some applications to define a comprehensive usual behavior of motes because of the vibrant nature of the network behavior.

Gao and colleagues [25] presented an indirect association-based abnormality identification method. Their work relies on a theory that an indirect association among

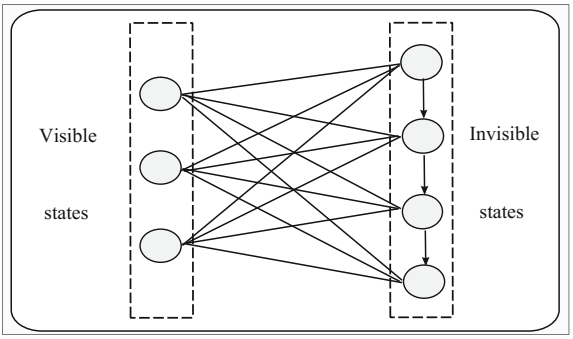

Fig. 2.3 A typical HMM: Republished with permission of Taylor and Francis Group LLC Books, from Security for Multihop Wireless Networks, Muhammad Usman, Vallipuram Muthukkumarasamy, Xin-Wen Wu, and Surraya Khanum, edited by Shafiullah Khan and Jaime Lloret Mauri, 2014; permission conveyed through Copyright Clearance Center, Inc.

several attributes of data may be used to formulate the routine behavior of a node. The Markov chain is then employed for calculating a state-transition probability matrix which is employed to identify abnormalities. This method is proficient for simultaneous identification of multiple types of abnormalities, but it causes a substantial computational overhead, which is not appropriate for resource-constrained networks.

Zheng et al. [26] presented a sequential technique for mobile nodes. The technique relies on a multiscale examination of the network traffic. The authors claim that the time duration length of capturing data traffic may affect the abnormality identification outcomes. The examination of data in various timescales is, therefore, important. A method, namely discrete wavelet transform (DWT), is used for multiscale data study. It may also be employed for the decorrelation of probabilistic procedures. A stochastic model is designed to enhance the detection rate. An algorithm, namely expectation maximization (EM), has been employed to compute the parameter values. The algorithm also identifies variations in the predicted model by examining the score of variation. The score of variation is calculated as entropy among current and past values. The abnormalities are then identified by discovering changes in the score of variation. This procedure is appropriate for discovering timescale abnormalities in the data, but this is energy-rich approach due to the large amount of processing involved in the abnormality identification process.

2.2.1.4 Time Series Model

This is a non-random ordered arrangement of data. Dissimilar to other statistical techniques, the time series model works on basic supposition that the consecutive measurements are collected after equal breaks of time [27]. The time series study has typically two aims, namely (i) identifying the conduct of a model that is based on

the arrangement of measurements and (ii) predicting the future conduct of a model. Singh and colleagues [28] discussed a scheme which works on an assumption that the variables used for abnormality identification should be integers that are nonnegative. The model is employed for forecasting static time series. The values of mean and variance remain static over the passage of time in the model.

The model has two algorithms. The initial algorithm amends the abnormal data of every mote at the base station or sink mote by employing predicted values. The final algorithm is then used for abnormality identification in mote data having 95% value of the confidence interval. A high-level pseudocode of the algorithms is given in Algorithms 2.1 and 2.2. In this scheme, the abnormality identification is performed in a noncooperative manner. This technique has two fundamental weaknesses. First, the forecast of the stationary behavior over the time series is not suitable for sensor networks, as typically they have dynamic topology. Second, the temporal and contextual internode relationship is not considered, which may make the abnormality identification procedure more strong.

Chuah and Fu [29] presented an abnormality identification scheme by employing time series examination in an ECG application. In the method, physiological motes are positioned to observe numerous physiological actions like heartbeats and pulse rate of senior citizens. Sensor nodes transmit data periodically through wireless link with a computer which employs an adaptive window-based discord discovery (AWDD) technique for detecting anomalous pulse rates and heartbeats. The anomalous data is then transmitted to a distant station where a doctor can identify diseases in order to take adequate remedial actions, if required. The method, however, does not consider synchronization between transmitted and received sensed data.

Algorithm 2.1 Determining reasonable model and forecasting

1: Find prevalent structure of autocorrelation
2: Calculate $AR(p)$, order p is determined using AIC criterion
3: Calculate step 2 residual $MA(q)$ through mandate of q
4: Carry out examination of residual
5: Forecast upcoming measurements and remove abnormalities

Algorithm 2.2 Abnormality identification

1: Identify 95% of the confidence interval ($\mu \pm 1.96\sigma$) where μ is predicted and σ denotes standard error values
2: Confirm null hypothesis
3: Accept/ ignore values using step 2 //Rejection shows abnormal value

2.2.1.5 Regression Analysis

This is a statistical procedure which is used to find the relationship among multiple variables. A regression analysis is typically performed to identify the underlying effect of single (dependent) variable on next (independent) variable(s) [30]. These variables may be either associated or non-associated. Curiac et al. presented an auto regression-based malicious node detection scheme [31]. The scheme deploys detectors which are positioned at the central node to screen observations of member nodes. The present and past observations are then matched by using autoregressive predictors for abnormality identification. The difference of the present value from a user-defined bound is treated as abnormal that results in the initiation of a block, namely decision block. The block then handles the malicious mote. This scheme is beneficial in predicting association between diverse variables, but the collection of an appropriate forecast measure is not a trivial task.

Kim and colleagues [32] presented a nonparametric regression abnormality identification scheme for heterogeneous sensor networks. The office room is deliberated as a use case, wherein abnormal everyday incidents are forecasted by using the combination of regression analysis and Bayesian network. The scheme learns the usual conduct of motes and assesses the extent of abnormality. The analysis of the proposed scheme was performed in an office room. The experiment location was furnished with motion and light detectors in order to accumulate sensed data with no or small advance information. The abnormalities were then discovered on the collected data by using the bound value which was derived during the learning stage. This method is modest yet useful for only smaller networks.

2.2.2 Artificial Intelligence and Agent-Based Schemes

Artificial intelligence is a discipline of designing and studying intelligent agents. Intelligent agents are entities which perceive events from their surroundings and take appropriate action(s), if required, in order to enhance the success likelihood of a system. The objective of the intelligent agent-based schemes is to mimic human intellect in tiny motes to perform numerous tasks [33]. Intelligent agent-based abnormality identification schemes have been mainly proposed in the domains, namely *Artificial Neural Network*, *Artificial Immune System*, *Genetic Algorithm*, *Game Theory*, and *Agent*.

2.2.2.1 Artificial Neural Networks

Adaptive resonance theory (ART) learns new knowledge without forgetting previously obtained knowledge [34]. ART has a chronological learning ability. ART is typically capable of updating existing labels, or it can create a novel class for fresh observations if they do not fit to the prevalent data classes. A typical ART is made

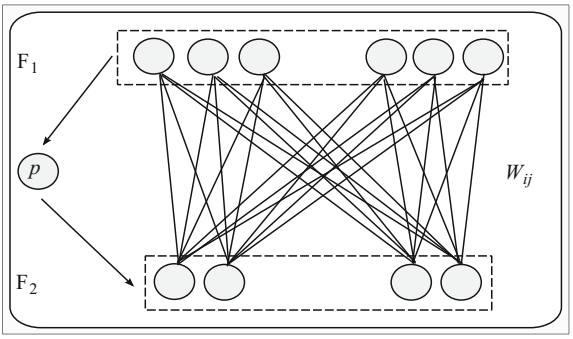

Fig. 2.4 An ART neural network: Republished with permission of Taylor and Francis Group LLC Books, from Security for Multihop Wireless Networks, Muhammad Usman, Vallipuram Muthukkumarasamy, Xin-Wen Wu, and Surraya Khanum, edited by Shafiullah Khan and Jaime Lloret Mauri, 2014; permission conveyed through Copyright Clearance Center, Inc.

up of two fields, namely *comparison*, denoted by F_1, and *recognition*, represented by F_2, having n and m neurons, correspondingly, where n denotes features and m represents categories. The links between the layers F_1 and F_2 have variable weights which are denoted by W_{ij}. A sensitivity threshold or vigilance aspect ρ assesses the similarity between the provided input and learned classes. A classical generalized high-level structure of ART is drawn in Fig. 2.4 [11].

Walchli et al. [35] proposed an abnormality identification technique for discovering abnormalities in signals using the Fuzzy ART neural network, viz. ART neural. The technique employs fuzzy ART to compute, categorize, and compress the sensed observations by employing time series examination. This technique is able to update prevalent label classes, though it is resource-extensive, particularly processing and memory expensive. Yuan et al. [36] designed and analyzed a fuzzy ART-based method in order to predict missing observations in sensor networks. Their proposed imputation technique considers the spatiotemporal information of network. The scheme is reliant on a supposition that the nodes in network are highly linked with respect to space and time. Pearson correlation coefficients and R-squared values are then employed for confirmation of the space association. An improved algorithm is employed to predict the absent data. The scheme is simple yet effective, but only focus on spatial and temporal abnormalities. Moreover, the usage of processing-intensive fuzzy ART only discovers single kind of abnormalities that might not be appropriate resource-constrained networks.

2.2.2.2 Artificial Immune System

The artificial immune system (AIS) is an adaptive technique which is inspired by theoretical immunology [37]. The usage of AIS for the abnormality identification in sensor networks is a relatively fresh method. Fu et al. [38] presented a biologically

inspired abnormality identification model for layered networks. The model combines the benefits of fuzzy and AIS theory. The method has three components, namely *co-simulation*, *global recognition*, and *sensing of local danger*. In the situation of an antagonistic mote launching a hazard, the motes in the locality of enemy mote sense the hazard to send an indication to the coordinator of personal area network (PAN) which works as a choice maker. The choice maker broadcasts a danger area according to the obtained danger signals. The decision maker begins and preserves a pool of receptors. The pool of receptors produces antibodies on the basis of the antigens. The principle of negative selection is then employed to discover abnormalities in the traffic. The technique employs parameters of PHY and MAC layers for abnormality identification. The model is adaptable and flexible, but it is resource-expensive and it might not be able of differentiating among abnormalities occur due to faults and attacks.

Salmon et al. [39] designed a danger theory immune-inspired technique for abnormality identification. The fundamental concept of the work is to employ Dendritic Cell Algorithm (DCA) for abnormality identification. This scheme has four phases, namely *collection*, *analysis*, *decision*, and *reaction*. The researchers associated multiple computational elements into their immune-stimulated counterparts. Mapping between the computational and biological elements is given in Table 2.2 [11]. In the initial phase, antigens and signals are saved. An investigation is then carried out in the second phase to produce output signals which discover the development state of the Dendritic Cells (DC). DCs perform the classification as usual or abnormal and identify their grade of abnormality in the next phase. In the terminal phase, T and B cells are activated in order to tackle the abnormal mote. This technique mimics immune system of human beings, and it has the ability of self-protection, but the confirmation of the qualities and quantities of antigens and antibodies is further required. More work is also needed to assess the performance of the scheme against multiple natures of abnormalities.

Table 2.2 Mapping between computational and biological elements: Republished with permission of Taylor and Francis Group LLC Books, from Security for Multihop Wireless Networks, Muhammad Usman, Vallipuram Muthukkumarasamy, Xin-Wen Wu, and Surraya Khanum, edited by Shafiullah Khan and Jaime Lloret Mauri, 2014; permission conveyed through Copyright Clearance Center, Inc.

Computational elements	Biological elements
Attack	Pathogen
Sensor mote	Tissue
coverage of mote	Danger area
Abnormality identity knowledge	Antigens
Decision manager element in sensor lymph	lymph node
Abnormality identification manager and context manager element	Dendritic cells
Abnormality countermeasure element	T and B cells
Countermeasure initiated by network protection system	Antibody

2.2.2.3 Genetic Algorithm

Genetic algorithms (GAs) were designed to imitate the natural evolution processes in order to solve computational problems [40]. A GA-based abnormality identification technique was presented by Khanna et al. [41]. The technique uses information of packet, status of battery, utilization of data, and quality of service (QoS) orthodoxy parameters for abnormality identification. Sensor nodes are categorized as cluster head, inter-cluster routers, common sensor nodes, and inactive nodes (sleep state). The leader mote uses a GA-based competing fitness function for optimally selecting cluster heads or inter-cluster routers that works as local observing motes. The native observing motes observe the actions of their neighbor motes using diverse parameters, namely *modification or dropping of packets*, *value of signal strength* , *rate of packet transmission*, *delay in response*, and *fake transmissions from conceded motes*. The technique optimizes previous abnormality identification methodologies in terms of detection capability. Nevertheless, the scalability of the technique necessitates the further validation.

Gene expression programming (GEP) mimics organic progression for programming. A GEP-oriented abnormality identification technique, in which a GEP forecast model for data traffic was used for time series study of the usual data traffic was presented by Gao et al. [42]. The abnormality identification technique is composed of 5 tuples: (P_m, F_s, M, F, O_p), where P_m denotes the parameter set, F_s represents the set of function, M shows the set of variables, F illustrates the function for fitness, and O_p is a set for operation. The set of parameter is reliant on the size of population, P_{size}. The gene can possess either tail T or head H if the following relation is fulfilled.

$$Gene = \begin{cases} H, \\ T, \\ H = h, \\ T = h - 1 \end{cases} \tag{2.1}$$

In the above equation, H is a component quantity of h and T is a component quantity of $h + 1$. The function set, F_s, has simple operators, i.e., $-$, $+$, $*$, and $\div$. The set for variable, M, has two factors, namely traffic length, i, and time, t. The fitness can be obtained from the resultant relation.

$$F_n = R^2 = 1 - SSE/SST \tag{2.2}$$

where the symbols *SST* and *SSE* are total of square and total of square of error of the overall variable set, correspondingly. The fitness function, F, relies on R-Squared that yields the square value of the coefficient of the Pearson product correlation. A GEP-based abnormality identification algorithm is given in Algorithm 2.3. This technique enhances the efficacy of conventional time series methods by removing the requirement of advance information about data traffic characteristics. However,

the depletion of energy of such a rigorous procedure might have an adversarial effect on the low resources of motes.

Algorithm 2.3 Gene expression programming

Input: Dataset M_t, here the data length is i
Output: correlation coefficient, relative error, and Time series mode
1: On the basis of i, convert data into $i + 1$ time series data
2: Initialize population using Genes which are progressed by abnormality identification procedure
3: Assess fitness of individuals
4: Stop as soon as maximum threshold is obtained, else continue
5: Select best
6: Carry out each operation
7: Goto Step 3

2.2.2.4 Game Theory

The abnormality identification procedure is modeled as a game among an abnormality identification module and an adversary in the game theory. One such model was studied by Agha et al. [43]. The factors for risk such as previous conduct of adversary and types of recognized attacks are known in this technique, and abnormality identification is carried out using these factors. An opponent may have three choices, namely attack on group, no attack on group, and attack on additional group in a fixed cluster scenario. On the contrary, the identification module has two response types, viz. protect group or protect a different group. This produces a $2*3$ matrix among two players. Increasing the turnover of every player, that is, obtaining the Nash equilibrium in the game, is a stimulating. It is also not constantly likely to enlist all possible positions of a system because of the active type of network topology.

Reddy [44] designed a mechanism which employs a zero-sum game for abnormality identification in transmitting data trail. Consider a network group having seven motes in a transmitting trail of two motes. Further assume that two out of those seven nodes are malicious. Furthermore, there is a non-malicious node among two abnormal motes. Let σ be the proportion of motes which are arbitrarily selected as examination points and $\hbar$ is the dedicated acknowledgment facts in trail of packets. The likelihood of identifying an antagonist mote, P_d, can then be computed as

$$P_d = P_y = 1 - P_{ack} \tag{2.3}$$

where P_y is the probability of packets that are dropped from acknowledgment facts and P_{ack} is the probability of acknowledgments at a foundation mote from acknowledgment facts. The P_{ack} can be computed as

$$P_{ack} = \sum_{i=1}^{n} P_{em}(i) \tag{2.4}$$

where $P_{em}(i)$ is anomalous motes packets probability. The packets probability among selected check points may be computed from the following relations.

$$P_{(m,n)} = P_{ack} - P_{ack}(n) \tag{2.5}$$

$$P_{(m,n)} = \sum_{j=1}^{n} P_{ack}(j) - \sum_{j=1}^{m} P_{ack}(j) \tag{2.6}$$

Employing 2.3 to 2.6, the number of abnormal motes in a particular path may be computed from the following relation.

$$xC_{P_{ack}} = (x!)/((P_{ack}!) * (x - P_{ack})!) = P_d \tag{2.7}$$

This technique is useful in the determination of a doubtful mote which is origin of abnormalities in network traffic. The technique is reliant on a game, namely cooperative zero-sum. An adequate choice and localization of a non-malicious mote among two or more antagonist motes is, however, not a trivial task without advance knowledge of the present network state. Furthermore, due to the cooperative approach, the inclusion of even only one opponent may spoil the entire procedure of the identification of an abnormal mote.

2.2.2.5 Agent

Agents are intelligent software entities which roam over the network to perform their designated jobs. The works carried out by Ketel [45], Pugliese and colleagues [46], Eludiora and colleagues [47], and Khanum and colleagues [48] have been chosen in this study to compare the multi-aspect performance of studied abnormality identification and confirmation system with these schemes. The rationale of choosing these schemes is similarity of the research domain, that is, employing the agent technology for abnormality/ intrusion/ attack detection in sensor networks. However, the work presented in this study has not only extended the role of agents to fine-grained description of the confirmation of abnormalities, but has also detected different types of abnormalities with high accuracies. The details of the working of the related schemes are provided below.

Krugel and Toth [49] conducted one of the pioneer studies which employed agents for abnormality identification. In their scheme, agents are dispatched as guards to carry out arbitrary sampling. If abnormality is found in an arbitrary inspection of a component of a network, then a complete identification is started. This work decreases the cost related to the transmission and receipt of agents to every mote, but in non-appearance of guards, motes are susceptible to attacks.

Ketel [45] presented an architecture which employs the agent technology for a distributed abnormality identification. The technique uses several static and agents, namely *Agents (MA)*, *Static Agents (SA)*, and *Nodal Agent (NA)* to detect abnormalities. The structure has *Mobile Agent Server (MAS)* and *Victim Node List (VNL)*. SAs are positioned on every cluster leader node. An SA transmits a message to an MAS after detecting an abnormal activity on a mote. MAS transmits an MA to monitor a node after detecting a threat. MAs are classified as *thin* and *thick* agents. The former and latter are associated with the resource-constrained and resource-rich motes, correspondingly. NAs are located on the observing motes in order to identify local abnormalities. Lastly, VNL has a list of victim motes at every cluster leader mote. VNL is accountable for issuing the route map of an MA. The configuration of several elements or agents on motes enhances the abnormality identification system cost.

Pugliese and colleagues [46] presented a rule-based technique for identifying network-layer abnormalities by employing agents. The essence of weak process model (WPM) is used to decrease the reachability rules. The attack model is categorized into high and low attack classes. The researchers have, however, merely reported initial implementation outcomes. A comprehensive performance is needed in order to confirm its utility for resource-limited sensor networks.

Eludiora and colleagues [47] presented an identification scheme in which motes directly interact with leader node rather than cluster leaders. The agents are used by leader nodes to interact with each other. A designated role of the agents is to wander between multiple motes and leader nodes to carry out the job of abnormality identification. The scheme is based on two algorithms, one each for data analysis and abnormality identification. The first algorithm detects denial of service (DoS) threats and thus appraises the status of a mote. On the contrary, the next algorithm computes the failure probability of a leader node to discover the abnormality. The scheme is designed for the networks where motes and base stations have only one hop distance. The one hop communication decreases consumption of energy, but it can cause a communication bottleneck on leader node. This tactic is not suitable for large networks and it only focuses on the DoS attacks which are other major limitations of the scheme.

A hierarchical abnormality identification system for sensor network has been designed by Khanum and colleagues [48]. The scheme employs three agents, namely *Management*, *Analyzer*, and *Coordinating*. The first is mobile, where second and third are static agents in nature. Motes are positioned in a clustered topology. Abnormality identification architectures are installed on each cluster leader node. Abnormalities are discovered at two levels, i.e., network and node. The cluster leaders detect network-wide abnormalities, whereas node-wide abnormalities are discovered by the Analyzer agents. Khanum et al. [48] has presented an abnormality identification architecture at a high level, since they have not provided any internal details about the abnormality identification technique. The cost that is related with the wandering of agents is not analyzed.

2.2.3 Learning Schemes

Learning algorithms enable computer and associated devices to learn from past observations to make decisions [50]. The learning process is typically based on extracting valuable patterns of data and forecasting on fresh data on the basis of the previously obtained knowledge [51]. Sensor nodes may be skilled by employing one among three algorithms, namely *supervised*, *semi-supervised*, and *unsupervised* learning to set normal behavior at deployment time. The selection of an algorithm can be made due to a number of factors such as availability of resources, network size, and abnormality identification method.

2.2.3.1 Supervised Learning

This techniques tags a training set using a pair (feature, label), represented as (a_i, b_i),...,(a_n, b_n), here $i = 1$ to n. The goal of the approach is to tag each instance of features of fresh input data. This class can be considered as a *regression function* or *classifier*. The supervised learning is a regression problem if $c \in \mathbb{R}$ and if c is based on "real" values, whereas it is a classification problem if c is based on "whole" values [52]. The likelihood of data classification is high and provides more accurate outcomes, but this is a time taking and energy extensive approach. Moreover, this is not always likely to receive a tagged dataset.

A key model from this class is known as support vector machine (SVM), which is a non-probabilistic binary and linear classifier which generates a hyperplane which divides data in dual classes by having a maximum likely distance. The vectors which lie near to the separating lines are the main point of interests. Figure 2.5 [11] represents a linear SVM graph which classifies a data set into two categories in a two-dimensional plane. SVM-based distributed abnormality identification scheme was presented by Rajasegarar and colleagues [53]. In the technique, local quarter spheres are calculated by each child mote and then they are transmitted to the respective parent motes. The respective parents are accountable for the local abnormality identification. Parent motes compute the universal radius using the obtained value of native radii. The parent motes then transmit the universal radius to all offspring in order to identify the universal abnormalities in their vicinities. This technique is able to discover both native and universal abnormalities, but the calculation of only quarter sphere might error some key information that may be employed for abnormality identification.

Xiao et al. [54] designed a Nave Bayes algorithm to discover abnormal motes in a network. The classifier, viz., Bayes presumes the occurrence of particular parameters from a class (usual or abnormal), which is unconnected to the occurrence of other parameters in a particular class. Abnormal conduct of a mote is discovered by employing numerous parameters such as energy consumption, interaction. The experiment results demonstrate that the Nave Bayes classifier provides a higher identification

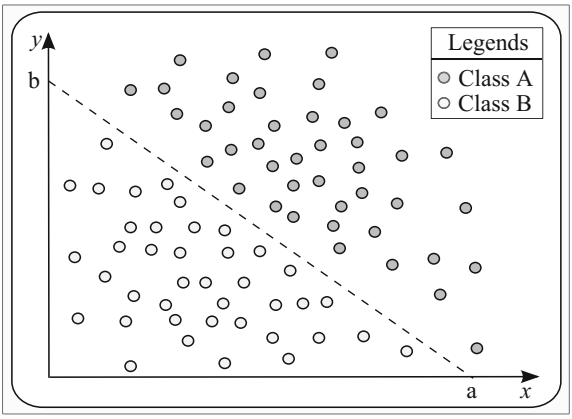

Fig. 2.5 SVM in two-dimensional horizon: Republished with permission of Taylor and Francis Group LLC Books, from Security for Multihop Wireless Networks, Muhammad Usman, Vallipuram Muthukkumarasamy, Xin-Wen Wu, and Surraya Khanum, edited by Shafiullah Khan and Jaime Lloret Mauri, 2014; permission conveyed through Copyright Clearance Center, Inc.

rate and less false positive rate. Nonetheless, the researchers have not analyzed the characteristic shortcomings of the methodology in this technique.

2.2.3.2 Semi-supervised Learning

This learning technique employs large unlabeled data along with small labeled data in order to form a robust classifier [55]. This technique is particularly beneficial in health and bioengineering domains where labeled data is not easily available because of high labeling cost of data. This cost is high because of the involvement of the human expertise. This technique is not typically employed in sensor networks for abnormality identification due to trouble in getting training data which encompasses all likelihoods of usual and abnormal conduct of a mote or network in advance.

2.2.3.3 Unsupervised Learning

These schemes use a non-labeled data set with features (a_i,...,a_n) for building classifiers. This scheme needs a threshold or certain criterian to identify abnormalities in a data set. The choice of an adequate threshold is a difficult task particularly in dynamic sensor network applications where a state of the usual conduct of the system keeps changing. A approach method of this class is clustering of measurements and then identification of abnormalities using distance from boundary or centroid of the cluster. This technique is apt for such applications where training data is not available.

In *K-means clustering* k-means of every cluster is computed [56]. These centroids are then linked with a training set. Next, every individual datum is linked with the nearest mean. As a consequence, each individual point is associated with one of the centroids. An abnormality identification technique, using k-means clustering, was

presented by Rajasegara et al. [57]. In the technique, every mote gathers local data in order to create a normal profile. These normal profiles are then transmitted to associated cluster leader. The cluster leaders collect local data and create a universal normal profile which is transmitted to all sensor nodes in their proximities. The process of abnormality identification is carried out using universal profile. The k-means technique is employed to improve the process of clustering. This technique involves a noteworthy communication burden, as cluster leader motes transmit universal profile to every member mote.

Xie and colleagues [58] presented an abnormality identification technique which employs an unsupervised learning method, namely *Principal Component Analysis (PCA)* which is a mathematical procedure which changes the associated variables into non-correlated variables using orthogonal transformations. The transformed variables are called as *principal components (PC)*. The technique employs a distance-oriented method for decreasing feature size from multivariate to univariate in order to enhance the abnormality identification procedure. An error coefficient is presented to consider the information loss that is caused during the conversion process. This technique has significantly reduced the training burden, but it suffers from the additional conversion cost.

An ellipsoidal neighborhood outlier (abnormality) factor for distributed abnormality identification was designed by Rajasegarar and colleagues [59]. The scheme presents a distributed abnormality identification architecture which employs several hyperellipsoidal clusters to model the data of each sensor node in the network. A method for the computation of a difference score among internode hyperellipsoidal models is presented. The experimental results demonstrate a reduction in the communication overhead as compared to the centralized schemes. In another study, Rajasegarar et al. [60] presented a distributed hyperspherical cluster-based algorithm. The algorithm minimizes the communication overhead by merging different clusters and transmitting a compact description to other nodes for abnormality identification. The objective of the scheme is to detect global abnormalities at node level. These schemes are, however, susceptible to a collusion attack, in which antagonist nodes can collude to compromise the reliability of the computed hyperellipsoidal and hyperspherical models which are used for the abnormality identification.

2.2.4 Other Schemes

There are several other abnormality identification schemes in the literature which may not belong to any of the above-mentioned classes. This section reviews such few related schemes.

2.2.4.1 Graph Theory

Graph theory has also been employed for abnormality identification in addition to other purposes such as aggregation of sensor readings as suggested by Bokareva et al. [61]. Ngai et al. [62, 63] presented novel algorithms for the detection of an abnormal mote by employing the graph theory. The scheme has two steps. Initially, a base station collects a list of doubtful motes. Finally, it set the precise location of the abnormal mote by using a graph, namely network flow. This scheme is helpful for the identification of the conspiring motes. A voting scheme is employed in order to announce a suspicious mote as abnormal or usual because of their past conduct. This tactic is beneficial in identifying node abnormalities, but it is a resource-expensive approach, especially in the worst case scenario, i.e., when an abnormality stays at the width or breadth end of a graph.

Ho and colleagues [64] designed a detection method which incorporates group information aspect before the positioning of a network. The identification is then performed at mote level after they receive a demand from their adjoining sensor to transmit a message. Every group is recognized using its exclusive deployment location. A mote, in every group, is positioned on the (a, b) location that may be computed from the following relation.

$$f(a, b) = 1/(2\pi\sigma^2)e^{((a-a_g)^2+(b-b_g)^2)/2\sigma^2} \tag{2.8}$$

In the above relation, (a, b) is the location of a sensor mote in a group and (a_g, b_g) represents the placement position of g. The notation σ denotes the standard deviation. If i mote, a group g_i member, gets a demand from an adjoining mote j, then i confirms that the distance among the group and adjoining mote is less than a pre-specified distance. If the distance is less, then j is treated as a genuine mote. Else, it is considered as an abnormal mote. This technique is beneficial in the identification of abnormalities that occurs due to replica motes. The efficiency of this technique is, however, highly dependent on the correct positioning and advance knowledge of the conduct of the network that might not always be possible.

2.2.4.2 Subjective Logic

Yuan et al. [65] modeled a subjective logic grounded abnormality identification framework. It is appropriate for those cases where ambiguity and partial knowledge is required. The framework has two algorithms. The initial algorithm combines the adjoining motes judgment in order to select the fate of a sensor mote as either usual or abnormal. However, this technique has three shortcomings. (i) The decision-making procedure may also involve abnormal motes, which may disturb the abnormality identification procedure. (ii) This procedure cannot distinguish between abnormal and usual data. (iii) A bound for the locality is fixed as 0.5 that might not be appropriate in every situation. To address these limitations, the researchers extended the

concept and suggested the next algorithm that has the following refinements in the abnormality identification procedure. (i) The elimination of the judgment of a doubtful mote. (ii) The spatial correlation between data is used to discriminate between the abnormal and usual data. (iii) The past information is taken into account to weigh the belief of adjoining motes. In spite of enhancement in the abnormality identification procedure, the consideration of past data may need more memory space. Moreover, the deliberation of the time-based correlation may further enhance the strength of the framework.

2.2.4.3 Fuzzy Logic

Fuzzy logic is a class of many-valued logic which can handle imprecise data for the approximate reasoning unlike the traditional crisp logic which is meant for fixed reasoning. The received sensor readings at a cluster leader node or a base station are often imprecise, even if they are located within close proximity to each other. This may affect the accuracy of the decisions which are made by cluster leaders or a base station. Fuzzy logic is, therefore, employed in abnormality identification applications in sensor networks to improve their performance.

Chi and Cho [66] designed a fuzzy logic-oriented abnormality identification technique in order to integrate a security mechanism into a routing protocol, namely directed diffusion. Several routing parameters like energy level of mote, error rate, rate of message transmission, and neighbor motes list are used to construct the rules for abnormality identification. This scheme, however, only focuses on the detection and prevention of the attacks. Furthermore, it is only suitable for directed diffusion-based senor networks.

PonoMarchuk and Seo [67] proposed a technique which is based on two levels of detection. First, an abnormality identification module monitors and identifies abnormalities on the packet inter-arrival time and packet reception rate, based on the user-defined threshold values. Second, composite rules, which consider both features for abnormality identification purposes, are employed for abnormality identification. The experimental results, based on a simulation study, show the capability of the technique to detect abnormalities with a high detection rate. This technique, however, needs the user knowledge in order to set the values of the thresholds in the first stage and the values of the parameters in the second stage. Furthermore, unified optimization of the threshold and parameter values are also not investigated.

Linda and colleagues [68] examined the role of fuzzy logic in abnormality identification for the security of embedded sensor networks. The scheme presents an algorithm which formulates a rule-base on the basis of fuzzy logic to model the normal behavior. The rules are generated from an incoming stream of data packets by employing a clustering algorithm. This authors, however, have not analyzed computational and memory overheads of the presented method. This does not establish its suitability for extremely low resource embedded sensor networks.

In another study, Kumarage et al. [69] presented an abnormality identification scheme which employs a fuzzy data modeling approach. The scheme performs par-

titioning, by employing the fuzzy c-means clustering, of the sensed data which is transmitted by the industrial sensor nodes. The abnormality identification is then carried out in a nonparametric and non-probabilistic fashion by the means of fuzzy membership functions. Adaptive thresholds are computed for the abnormality identification. The scheme achieves high accuracy as compared to competing schemes. An iterative process of the computation of a threshold value, however, is not an efficient approach for resource-constrained sensor networks.

A more sophisticated abnormality identification model was designed and analyzed by Moshtaghi et al. [70]. The model introduces a new mechanism to estimate the parameters of Takagi-Sugeno fuzzy logic methodology. An incremental approach is adopted for the computation of an inverse of the covariance matrix and a weighted sample mean in order to construct the fuzzy rule-base which evolves with the passage of time. The constructed rule-base is then employed for the abnormality identification process. This scheme is, however, not suitable for those sensor networks which have a dynamic behavior.

2.2.4.4 Cross-Layer Abnormality Identification

Two or more features from different layers of an Open System Interconnection (OSI) model are employed for the abnormality identification in the cross-layer abnormality identification models. Over the years, the community has presented a number of such abnormality identification methods for sensor networks [71–75]. One such approach was presented by Onat and colleagues [67] where every mote preserves a profile of its adjoining motes. The profile has two features, namely *rate of packet arrival* and *received power average*. This technique is, however, not capable of detecting more sophisticated abnormalities due to the trivial nature of the underlying detection method.

Bhuse [72], in his PhD dissertation, investigated the use of multiple layer features, namely *physical (PHY)*, *Media Access Control (MAC)*, *network*, and *application* for the abnormality identification. In PHY layer, the Received Signal Strength Indicator (RSSI) value is employed for abnormality identification. The neighboring motes RSSI value is computed and a noteworthy nonconformity from that base value is treated as an abnormality. In MAC layer, S-MAC and TDMA protocols are employed for the abnormality identification. In these protocols, slots of time are assigned to motes for communication. The interaction outside an allocated slot indicates an intrusion activity. In network layer, a protocol is proposed which uses forwarding tables that are created by the protocols for the abnormality identification procedure. This procedure needs the incorporation of high-level knowledge in tables in order to formulate abnormality detection tables (ADTs). Abnormalities are then identified using ADTs. In application layer, a time for round trip is treated as an abnormality identification feature. Boubiche et al. [73] presented a technique for layer-wise abnormality identification. A common feature of the above-mentioned schemes is their capability to detect the layer-wise abnormalities in the respective layers.

Becker and colleagues [74] employed multiple learning algorithms such as k-Nearest Neighbors, Support Vector Machines, Bayes Classifier, Neural Network using multiple features from different layers. An experiment study illustrates that the naive approaches, for example, Bayes classifier and decision tree serve improved performance in contrast with the other techniques. In the experimentation setup, several features are employed for the abnormality identification. The performance of this abnormality identification technique is, however, heavily reliant on the selection of appropriate features. A fundamental method of using several features might not be adequate for resource-limited motes.

More recently, Dai and colleagues [75] presented a multivariate classification method for the identification of abnormal motes in large networks. The detection process is based on the multivariate classification technique. The multivariate technique excerpts the preferences of a mote related to the malicious behavior. It then creates a sample space for all motes which are the network constituents. This is followed by the detection of anomalous motes using given criteria. The experimental outcomes show that a false detection rate is less than 0.5%. In spite of having low rate of false detection, the effectiveness of this technique depends on identification criteria. The choice of appropriate identification criteria is not a trivial job in certain sensor network applications, particularly where motes have a vibrant behavior.

2.2.4.5 Other Schemes

Li et al. [76] presented a quantitative approach for the identification of anomalous motes. The approach is based on the data transmission quality (*DTQ*) function. A sensor network is separated into multiple groups. Every mote keeps a DTQ table of its adjoining motes. The *DTQ* function is provided in the subsequent relation.

$$DTQ = kD/E \times (STB())/(P()) \tag{2.9}$$

In the above relation, k is an integer which is greater than 0 and D/E represents the packets count that are communicated in a unit energy. The notation $STB()$ represents a factor for the stability of data dispatch, and $P()$ is the likelihood of effectively communicated data packets. The *DTQ* function value stays static or varies efficiently for usual motes and changes for anomalous motes. The concluding fate of a mote as usual or abnormal is defined using voting between members of the group. The voting technique is, however, vulnerable to the conspiring threats.

Krontiris and colleagues [77] designed a novice abnormality identification approach. In the method, selected motes, namely *watchdog motes*, carries out the job of observing the neighboring motes. The watchdog motes are selected as per the following criteria. Assume a situation, where mote A has a communication connection with mote B. Then mote A and other motes which are located inside the juncture of the radio range of mote A and mote B may work as regulator. A regulator mote observes adjoining motes as per the following mechanism. (i) If a specific mote drops n data packets in t time, a triggered is then triggered by a regulator

mote. (ii) In a particular location, if regulator motes trigger alarms, then the doubtful mote is broadcasted as an abnormal mote. This method is simple, but vulnerable to high false positive rate. Moreover, voting requirement by half of the motes in order to announce a doubtful mote as an abnormal mote might not be appropriate for few sensor network applications, especially where high rate of detection is required.

Krishnamachari and colleagues [78] presented a distribution algorithm for the detection of environmental events of interest. The algorithm explicitly takes into account the possibility of the faults in sensor measurements for detecting these faults. The performance of the proposed algorithm has been analyzed through simulation study and analysis. The outcomes indicate that the method can detect 85 to 95% faults. Lazarevice et al. [79] has conducted a detailed comparative study to identify different kind of network intrusions. A number of abnormality identification techniques and their extended works have been evaluated on the DAPRA data set. The results indicate superior performance of some schemes as compared to other schemes.

This section has first discussed the statistical abnormality identification schemes. A number of related schemes, which use artificial intelligence, agent technology, machine learning, fuzzy logic, and several other methods for abnormality identification, are then critically reviewed. The security of agent will be reviewed in the next section, as our proposed abnormality identification and confirmation system rely on the agent technology for the confirmation of abnormalities.

2.3 Security of Agents

A key challenge in agent-enabled applications is to secure agents from antagonist nodes. This is a non-trivial task due to the complete control of a host node on an agent during its execution.

2.3.1 Securing Agents on Middleware

A networking system, in general, and sensor network, in particular, can be categorized into three layers, namely *application*, *middleware*, and *hardware* from the viewpoint of layered architecture. The security on these systems can be deployed on any of these layers [80]. A number of middleware architectures have been designed and studied in the literature which integrated security at middleware of agent applications [80–84]. These architectures facilitate easy modification or even complete removal of the applications without interfering in the security mechanisms of the tiny sensor nodes.

2.3.2 Other Approaches

Over the years, some studies have also been conducted in other types of networks in order to secure agents from antagonists. Some of the few prominent schemes from the literature are reviewed below [85–89].

One of the pioneer works, about the identification of the manipulation attack against the agents, was carried out by Vigna [85]. In the work, agents can grab traces of the execution of the instructions which are executed by a malicious node during the execution of an agent on that node. The traces are logs of the actions which are performed by agents. Each node saves traces of the execution of agents and then transmits the hash of those traces in order to save the network bandwidth. A parent node of an agent can then verify the integrity of the execution of an agent by re-executing it and matching the traces. This solution, however, demands high computational and memory resources to compute hash values and store traces. This solution is, therefore, not feasible for low resource sensor networks.

Zhang et al. [86] designed a secure integrated scheme for agents and Web services. The scheme offers an authentication protocol which does not require a username–password pair for authentication, due to the fact that this kind of authentication is infeasible for agents. The presented protocol, therefore, relies on an identity-based public key management algorithm. An analysis carried out by the authors shows that the scheme can simplify the key management process. This scheme is, however, only focused on Web services.

A study on securing agents with designated hosts was conducted by Mu et al. [87]. In the presented model, a parent node of an agent selects a destination node for that agent and then performs an authentication with the node. This approach avoids the misuse and non-repudiation problems. The model is theoretically verified in the electronic commerce settings. This scheme, however, needs a full scale experimental analysis to be deemed as effective for the intended purpose.

In another work, Malik et al. [88] designed a secure transfer procedure for agents among agencies (i.e., agent-host nodes). Mobile-C is a multi-agent platform which supports C and C++ static and agents. In the proposal, all agents are authenticated by a trusted third party, a system administrator in this case. The design of the framework is inspired from the Secure Shell (SHH) protocol. The transmitter and receiver nodes authenticate each other using public key cryptography before the transmission and reception of an agent. The receiver node also verifies the integrity of the received agent. A turnaround time for agents is evaluated in different scenarios. The authors, however, have not performed any attack-analysis in order to show the usefulness of the cryptographic methods to secure agents.

More recently, Esparza and colleagues [89] employed watermarking and fingerprinting approaches for the detection of an agent manipulation attack. The presented approach, namely *agent watermarking* (MAW), enables an agent parent node to confirm the execution integrity of the agent on the basis of the inserted watermark. The authors further presented a technique to detect and punish an antagonist node by using a trusted third party. The usability of the schemes is demonstrated through a

proof-of-concept and an extensive performance evaluation. The target of the MAW is traditional networks, but we have extended this approach in networks because the size of a watermark can be varied in order to enable MAW to work in low resource sensor networks.

2.4 Formal Modeling and Analysis

A badly designed agent-enabled system can go into a static state, or it may even badly affect the restricted resources of motes by not performing its intended functionality. Formal modeling is, therefore, typically employed in order to address formal descriptions and verification of the correctness of large systems [90]. The process of formal modeling and analysis also removes inconsistencies from the overall system design. It also ensures the completeness of a system design by the identification of missing or invalid requirements. The missing requirements can then be included in the system design or invalid requirements can be improved to obtain a better overall design. Over the course of recent years, formal modeling has been a well-studied procedure of the validation of the system behavior prior to its implementation. In the last decade, the community of researchers has carried out numerous studies to formally verify wireless communication systems [91–93]. Similarly, some attempts have also been carried out to formally characterize protocols [94–96].

Verification of algorithms, protocols, or even systems can also be useful in creating their best designs for resource-constrained sensor networks. Numerous studies have performed to formally prove the security procedures for sensor networks along similar lines. Law and colleagues [97] formally modeled and analyzed a distributed key management architecture for sensor networks, which has two interconnected supervised and unsupervised security kingdoms. A tool, namely CoProve is used to confirm the correctness of the presented protocols. The specifications of protocols are fed as input. A working example situation of the system is then studied to assess the performance of the protocols. In a PhD work, Werner [98] applied formalism for checking of energy consumption models by motes in already defined settings. This method, however, may not be adequate for stochastic systems.

The Automated Validation of Internet Security Protocols and Applications (AVISPA) is also studied in the literature [99]. It presents a formalism to provide the specification of security protocols. Over the years, the tools like Co- Prove, AVISPA, and brute-force algorithms have been used for validation of security protocols. However, the Petri net theory is a mathematical modeling language. The mathematical foundations of Petri net theory give sturdy guarantee on descriptions, modeling, and verifications of qualitative and quantitative properties of systems [100]. Despite having enriched properties for modeling and verification, the Petri net theory has not been extensively studied in sensor network security literature in general, and abnormality identification schemes in particular. Over the course of the last decade, a few notable security solutions have been formally modeled and verified using the

Petri net theory in sensor networks [101–104]. These studies, however, have several limitations which are discussed below.

In 2009, He et al. [101] employed Petri nets to design and formally verify an enhanced secure localization scheme for sensor networks. Both localization and attack driven models are constructed using the Petri net theory. A state equation technique is then employed to perform a reachability analysis. A state equation-based analysis demonstrates that only the secure states are reachable in the system design. The authors have, however, not conducted a thorough behavior analysis of the security protocol. Furthermore, the formal models have not been validated through simulation or real implementation.

Rodriguez et al. [102] exploited the dynamic characteristics of the Petri net theory, namely *synchronization* and *concurrency* in order to model and formally verify an encryption scheme for sensor networks. The communication system is modeled through standard Petri nets. An elliptic curve-based cryptography protocol is then modeled and illustrated using place and transition invariants. The authors have only derived the reachability set to analyze the behavior of the presented cryptographic protocol. Furthermore, the constructed model is trivial and it is incapable of estimating temporal behavior of the protocol, which is an important performance metric for any security protocol.

Tseng et al. [103] verified a robust self certificate-based user authentication scheme for sensor networks by using the Petri net theory. A security analysis of an authentication scheme is performed after the construction of a standard Petri net model. The Petri net model is simulated on HPSim in order to verify the reachability of different states from the root state. An extensive analysis of the behavioral properties is, however, not conducted to estimate the aptness of the authentication scheme.

More recently, Sbai and Escheikh [104] employed Petri nets for the verification of an encryption scheme in sensor networks. A Petri net model of the encryption scheme is constructed through Promela, a modeling language. The correctness properties are then formulated through linear temporal logic. A dedicated model checker tool, SPIN, is used to validate the correctness of the encryption scheme. Like other schemes, this work is also focused on only limited aspects of the behavior analysis of an encryption scheme.

2.5 Limitations

A well-designed abnormality identification mechanism can detect an inconsistent, a malfunctioning, or a troublesome node which may disrupt the usual working of a sensor network. Designing an adequate abnormality identification system is a tough task considering the unique characteristics of sensor networks. One key design consideration, while articulating an abnormality identification mechanism for sensor networks, is the placement of abnormality identification modules in the network.

Traditionally, centralized or distributed design choices have been used for the positioning of abnormality identification modules.

Another key design choice is to discover abnormalities with minimum energy cost. The main focus of the existing schemes is, therefore, on articulating the light weight abnormality identification methods along with an acceptable rate of detection. This method might extend the lifetime of a network. However, this might not be capable to identify the sophisticated nature of abnormalities in a timely manner. The failure of identifying such abnormalities might have the worst effect on the functionality and lifetime of the network. Thus, there must be a balance among making an abnormality identification mechanism lightweight and its ability to detect sophisticated abnormalities.

A literature survey, performed throughout this chapter, has shown that the majority of the prevalent abnormality identification schemes have merely focused on the identification of abnormalities [10, 12, 15, 18–21, 105]. It is, however, imperative to discover the origin of abnormalities for their effective mitigation, which is to our knowledge not considered in the existing literature. This book has, therefore, presented an agent-enabled abnormality identification system that not only identifies different types of abnormalities, but also offers a service for on-the-spot confirmation of abnormal motes by using agents.

The survey of literature also reveals that agents are employed in different roles in existing abnormality identification schemes [45, 47, 48]. However, a common limitation of these techniques is their non-consideration of infrastructure and communication costs which are linked with agents. The work presented in this study has, therefore, considered these factors and deployed abnormality identification modules on resource-rich cluster leader nodes and also presented methods for the optimization of agent transmission.

Securing agents from antagonists have not been studied extensively in the sensor network literature. This is, in fact, explored more in the literature related to traditional networks. The focus of the schemes in traditional networks is, however, only on a security performance. These schemes may not be readily applied because of their limited resources. Therefore, this study has extended the MAW approach [89] to sensor networks by making necessary amendments to secure agents from agent execution integrity attacks.

A few schemes in the literature have considered the detection of abnormalities at different layers of an OSI model [72, 75]. These schemes, however, have not considered a cross-layer optimization in a Zigbee-IEEE 802.15.4 standard to detect the different nature of abnormalities. This study has considered a cross-layer approach, not only for the abnormality identification, but also for the effective transmission of agents after taking into account the current state of the communication link. Fuzzy logic is further used to improve the performance of the proposed method.

The review of related works also shows that many schemes are merely designed and their detailed analyses have not been performed in order to validate their aptness. Absence of the detailed analyses highlights a key point, that is, whether those schemes are adequate for resource-limited networks. In this view, the performance

of the methods presented in this study is thoroughly investigated through theoretical analyses, experiments on a real test bed, and extensive simulations.

2.6 Summary

This chapter has first provided a background on the sensor network security in order to highlight the problem domain of this study. Schemes from the abnormality identification literature are then critically reviewed. This is followed by a discussion on agent security, where a middleware architecture is presented to integrate the security in agent-based resource-constrained network applications in a middleware layer. A brief overview of the literature on the formal modeling and analysis is also presented and limitations in the existing related schemes have been highlighted to set the research context of this study.

In the next chapter, the abnormality identification and confirmation system has been elucidated.

2.7 Bibliographic Notes

The types of anomalies, discussed in Sect. 2.1, were first introduced by [10] and subsequently discussed with associated concepts in [11]. The taxonomy of schemes, focused on abnormality identification, and subsequent discussion on these schemes and their limitations were first provided in [11].

References

1. Y. Zhang, N. Meratnia, P. Havinga, Outlier detection techniques for wireless sensor networks: a survey. IEEE Commun. Surv. Tutor. **12**(2), 159–170 (2010)
2. A. Perrig, J. Stankovic, D. Wagner, Security in wireless sensor networks. Wirel. Sensor Netw. Commun. ACM **47**(6), 53–57 (2004)
3. K. Marzullo, Failures of continuous-valued sensors. ACM Trans. Comput. Syst. (TOCS) **8**(4), 284–304 (1990)
4. T.G. Roosta, Attacks and defenses of ubiquitous sensor networks. Ph.D. thesis, University of California, Berkeley, 2008
5. T. Zia, A. Zomaya, Security issues in wireless sensor networks, in *Proceedings of the International Conference on Systems and Networks Communication* (2006), p. 40
6. C.P. fleeger, *Security in Computing* (Prentice Hall Professional Technical Reference, Prentice Hall PTR, New Jersey, 2003)
7. E.C.H. Ngai, Intrusion detection for wireless sensor networks, in *PhD Term 2 Paper* (Department of computer Science and Engineering, The Chinese University of Hong Kong, 2005), pp. 6–20

8. K. Scarfone, P. Mell, Guide to intrusion detection and prevention systems (idps) (Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, 2007), pp. 1–127
9. I. Butun, S.D. Morgera, R. Sankar, A survey of intrusion detection systems in wireless sensor networks. IEEE Commun. Surv. Tutor. **16**(1), 266–282 (2014)
10. R. Jurdak, X.R. Wang, O.r Obst, P. Valencia, Wireless sensor network anomalies: diagnosis and detection strategies, in *Intelligence-Based Systems Engineering*, vol. 10, Intelligent Systems Reference Library (Springer, Berlin, 2011), pp. 309–325
11. M. Usman, V. Muthukkumarsamy, X.-W. Wu, S. Khanum, Anomaly detection in wireless sensor network: challenges and future trends, in *Security for Multihop Wireless Networks Edition* (Auerbach publications Taylor and Francis Group, USA, 2014a)
12. L.M.A. Bettencourt, A.A. Hagberg, L.B. Larkey, Separating the wheat from the chaff: practical anomaly detection schemes in ecological applications of distributed sensor networks, in *Distributed Computing in Sensor Systems* (Springer, Berlin, 2007), pp. 223–239
13. W. Wu, X. Cheng, M. Ding, K. Xing, F. Liu, P. Deng, Localized outlying and boundary data detection in sensor networks. IEEE Trans. Knowl. Data Eng. **19**(8), 1145–1157 (2007)
14. H.A. Guvenir, M. Kurtcephe, Ranking instances by maximizing the area under roc curve. IEEE Trans. Knowl. Data Eng. **25**(10), 2356–2366 (2013)
15. M.C. Jun, H. Jeong, C.-C.J. Kuo, Distributed spatio-temporal outlier detection in sensor networks, in *Proceedings of the SPIE*, vol. 5819 (2005), pp. 273–284
16. V. Chandola, A.M Banerjee, V. Kumar, Anomaly detection: a survey. ACM Comput. Surv. **41**(3), 1–58 (2009)
17. A. Gibson, *Exposure and Understanding the Histogram* (Peachpit press, Prentice Hall Professional Technical Reference, Berkeley, 2011)
18. M. Xie, J. Hu, B. Tian, Histogram-based online anomaly detection in hierarchical wireless sensor networks, in *IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)* (2012), pp. 751–759
19. B. Miladinovic, *Kernel Density Estimation of Reliability with Applications to Extreme Value Distribution* (BiblioBazaar, 2011)
20. T. Palpanas, D. Papadopoulos, V. Kalogeraki, D. Gunopulos, Distributed deviation detection in sensor networks. SIGMOD Rec. **32**(4), 77–82 (2003)
21. S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, D. Gunopulos, Online outlier detection in sensor data using non-parametric models, in *Proceedings of the 32nd International Conference on Very Large Data Bases* (2006), pp. 187–198
22. N. Marchenko, C. Bettstetter, Cooperative arq with relay selection: an analytical framework using semi-markov processes. IEEE Trans. Veh. Technol. **63**(1), 178–190 (2014)
23. Y. Xie, J. Hu, Y. Xiang, S. Yu, S. Tang, Y. Wang, Modeling oscillation behavior of network traffic by nested hidden markov model with variable state-duration. IEEE Trans. Parallel Distrib. Syst. **24**(9), 1807–1817 (2013b)
24. I.C. Paschalidis, Y. Chen, Anomaly detection in sensor networks based on large deviations of markov chain models, in *47th IEEE Conference on Decision and Control* (2008), pp. 2338–2343
25. Y. Gao, C. Chen, J. Bu, W. Dong, D. He, Icad: indirect correlation based anomaly detection in dynamic wsns, in *IEEE Wireless Communications and Networking Conference* (2011), pp. 647–652
26. S. Zheng, J.S.A. Baras, S. Zheng, J.S. Baras, Sequential anomaly detection in wireless sensor networks and effects of long-range dependent data. Spec. IWSM Issue Seq. Anal. **31**(1), 458–480 (2012)
27. W. Kim, T. He, D. Wang, C. Cao, S. Liang, Assessment of long-term sensor radiometric degradation using time series analysis. IEEE Trans. Geosci. Remote Sens. **52**(5), 2960–2976 (2014)
28. A.K. Singh, B. Giridhar, P.S. Mandal, Fixing data anomalies with prediction based algorithm in wireless sensor networks, in *7th IEEE Conference on Wireless Communication and Sensor Networks, Panna, India, 5–9 Dec* (2011), p. 6

29. M.C. Chuah, F. Fu, Ecg anomaly detection via time series analysis, in *Frontiers of High Performance Computing and Networking ISPA 2007 Workshops*. Lecture Notes in Computer Science, vol. 4743 (Springer, Berlin, 2007), pp. 123–135
30. A.K.M.E. Saleh, M. Arashi, S.M.M. Tabatabaey, *Statistical Inference for Models with Multivariate t-Distributed Errors* (Wiley, USA, 2014)
31. D.-I. Curiac, O. Banias, F. Dragan, C. Volosencu, O. Dranga, Malicious node detection in wireless sensor networks using an autoregression technique, in *Third International Conference on Networking and Services* (2007), pp. 83–83
32. S.Y. Kim, M. Imada, M. Ohta, Detecting anomalous events in ubiquitous sensor environments using bayesian networks and nonparametric regression, in *21st International Conference on Advanced Information Networking and Applications* (2007), pp. 236–243
33. S.J. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd edn. (Pearson Education, USA, 2009)
34. T. Trappenberg, *Artificial Intelligence: A Modern Approach*, 2nd edn. (Oxford University Press, Great Britain, 2010)
35. M. Walchli, T. Braun, Efficient signal processing and anomaly detection in wireless sensor networks, in *Applications of Evolutionary Computing*. Lecture Notes in Computer Science (Springer, Berlin Heidelberg, 2009), pp. 81–86
36. Y.Y. Li, L.E. Parker, Classification with missing data in a wireless sensor network, in *IEEE Southeastcon* (2008), pp. 533–538
37. J. Timmis, P. Andrews, E. Hart, On artificial immune systems and swarm intelligence. Swarm Intell. **4**(4), 247–273 (2010)
38. R. Fu, K. Zheng, T. LU, D. Zhang, Y. Yang, Biologically inspired anomaly detection for hierarchical wireless sensor networks. J. Netw. **7**(8), 1214–1219 (2012)
39. H. Salmon, C.M.D. Farias, P. Loureiro, L. Pirmez, S. Rossetto, P.D.A. Rodrigues, R. Pirmez, F. Delicato, L.F.R. Carmo, Intrusion detection system for wireless sensor networks using danger theory immune-inspired techniques. Int. J. Wireless Inf. Netw. **20**(1), 39–66 (2013)
40. O. Roeva, *Genetic Algorithms in Applications*, 1st edn. (InTech, Croatia, 2012)
41. R. Khanna, H. Liu, H.-H. Chen, Reduced complexity intrusion detection in sensor networks using genetic algorithm, in *IEEE International Conference on Communications* (2009), pp. 1–5
42. H.L. Gao, G. Chen, W. Guo, A gep-based anomaly detection scheme in wireless sensor networks, in *International Conference on Computational Science and Engineering, CSE 2009*, vol. 2 (2009), pp. 817–822
43. A. Agah, S.K. Das, K. Basu, M. Asadi, Intrusion detection in sensor networks: a non-cooperative game approach, in *Proceedings of the Third IEEE International Symposium on Network Computing and Applications, NCA* (2004), pp. 343–346
44. Y.B. Reddy, A game theory approach to detect malicious nodes in wireless sensor networks, in *Third International Conference on Sensor Technologies and Applications* (2009), pp. 462–468
45. M. Ketel, Applying the mobile agent paradigm to distributed intrusion detection in wireless sensor networks, in *40th Southeastern Symposium on System Theory* (2008), pp. 74–78
46. M. Pugliese, A. Giani, F. Santucci, Weak process models for attack detection in a clustered sensor network using mobile agents, in *Sensor Systems and Software*, vol. 24, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, ed. by S. Hailes, S. Sicari, G. Roussos (Springer, Berlin, 2010), pp. 33–50
47. S.I. Eludiora, O.O. Abiona, A.O. Oluwatope, S.A. Bello, M.L. Sanni, D.O. Ayanda, C.E. Onime, E.R. Adagunodo, L.O. Kehinde, A distributed intrusion detection scheme for wireless sensor networks, in *IEEE International Conference on Electro/Information Technology (EIT)* (2011), pp. 1–5
48. S. Khanum, M. Usman, A. Alwabel, Mobile agent based hierarchical intrusion detection system in wireless sensor networks. Int. J. Comput. Sci. Issues (IJCSI) **9**(3), 101–108 (2012)
49. C. Krugel, T. Toth, Applying mobile agent technology to intrusion detection, in *ICSE Workshop on Software Engineering and Mobility* (2001), pp. 1–5

50. T. Bokareva, N. Bulusu, S. Jha, Learning sensor data characteristics in unknown environments, in *Proceedings of the Third Annual International Conference on Mobile and Ubiquitous Systems: Networking Services* (2006), pp. 1–8
51. K.P. Murphy, *Machine Learning: A Probabilistic Perspective*, 1st edn. (MIT Press, USA, 2012)
52. X. Zhu, Semi-supervised learning with graphs. Ph.D. thesis, Carnegie Mellon University, 2005
53. S. Rajasegarar, C. Leckie, M. Palaniswami, J.C. Bezdek, Quarter sphere based distributed anomaly detection in wireless sensor networks, in *IEEE International Conference on Communications* (2007), pp. 3864–3869
54. Z. Xiao, C. Liu, C. Chen, An anomaly detection scheme based on machine learning for wsn, in *1st International Conference on Information Science and Engineering* (2009), pp. 3959–3962
55. T.H. Lim, Detecting anomalies in wireless sensor networks. Ph.D. thesis, Department of Computer Science, University of York, 2010
56. C. Boutsidis, M. Magdon-Ismail, Deterministic feature selection for k-means clustering. IEEE Trans. Inf. Theory **59**(9), 6099–6110 (2013)
57. S. Rajasegarar, C. Leckie, M. Palaniswami, J.C. Bezdek, Distributed anomaly detection in wireless sensor networks, in *10th IEEE Singapore International Conference on Communication systems* (2006), pp. 1–5
58. M. Xie, S. Han, B. Tian, Highly efficient distance-based anomaly detection through univariate with pca in wireless sensor networks, in *10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)* (2011), pp. 564–571
59. S. Rajasegarar, A. Gluhak, M.A. Imran, M. Nati, M. Moshtaghi, C. Leckie, M. Palaniswami, Ellipsoidal neighbourhood outlier factor for distributed anomaly detection in resource constrained networks. Pattern Recogn. **47**(9), 2867–2879 (2014a)
60. S. Rajasegarar, C. Leckie, M. Palaniswami, Hyperspherical cluster based distributed anomaly detection in wireless sensor networks. J. Parallel Distrib. Comput. **74**(1), 1833–1847 (2014b)
61. T. Bokareva, N. Bulusu, S. Jha, Graph theory based aggregation of sensor readings in wireless sensor networks, in *Proceedings of the 33rd IEEE Conference on Local Computer Networks (LCN)* (2008), pp. 514–515
62. E.C.-H. Ngai, J. Liu, M.R. Lyu, On the intruder detection for sinkhole attack in wireless sensor networks, in *IEEE International Conference on Communications*, vol. 8 (2006), pp. 3383–3389
63. E.C.-H. Ngai, J. Liu, M.R. Lyu, An efficient intruder detection algorithm against sinkhole attacks in wireless sensor networks. Comput. Commun. **30**(11–12), 2353–2364 (2007)
64. J.-W. Ho, D. Liu, M. Wright, S.K. Das, Distributed detection of replica node attacks with group deployment knowledge in wireless sensor networks. Ad Hoc Netw. **7**(8), 1476–1488 (2009)
65. J. Yuan, H. Zhou, H. Chen, Subjective logic-based anomaly detection framework in wireless sensor networks. Int. J. Distrib. Sens. Netw. **2012**(2012), 13 (2011)
66. S.H. Chi, T.H. Cho, Fuzzy logic anomaly detection scheme for directed diffusion based sensor networks, in *Fuzzy Systems and Knowledge Discovery*, vol. 4223, Lecture Notes in Computer Science, ed. by L. Wang, L. Jiao, G. Shi, X. Li, J. Liu (Springer, Berlin, 2006), pp. 725–734
67. Y. Ponomarchuk, D.W. Seo, Intrusion detection based on traffic analysis and fuzzy inference system in wireless sensor networks. J. Converg. **1**(1), 35–42 (2010)
68. O. Linda, M. Manic, T. Vollmer, J. Wright, Fuzzy logic based anomaly detection for embedded network security cyber sensor, in *IEEE Symposium on Computational Intelligence in Cyber Security (CICS)* (2011), pp. 202–209
69. H. Kumarage, I. Khalil, Z. Tari, A. Zomaya, Distributed anomaly detection for industrial wireless sensor networks based on fuzzy data modelling. J. Parallel Distrib. Comput. **73**(6), 790–806 (2013b)
70. M. Moshtaghi, J. Bezdek, C. Leckie, S. Karunasekera, M. Palaniswami, Evolving fuzzy rules for anomaly detection in data streams. IEEE Trans. Fuzzy Syst. **PP**(99), 1 (2014b)

71. I. Onat, A. Miri, An intrusion detection system for wireless sensor networks, in *Proceedings of IEEE International Conference on Wireless And Mobile Computing, Networking And Communications*, vol. 3 (2005), pp. 253–259
72. V.S. Bhuse, Lightweight intrusion detection: a second line of defense for unguarded wireless sensor networks. Ph.D. thesis, Department of Computer Science, Western Michigan University, 2007
73. D.E. Boubiche, Z. Bilami, Cross layer intrusion detection system for wireless sensor network. Int. J. Netw. Secur. Appl. (IJNSA) **4**(2), 35–52 (2012)
74. M. Becker, M. Drozda, S. Schaust, S. Bohlmann, H. Szczerbicka, On classification approaches for misbehavior detection in wireless sensor networks, J. Comp. 4(5) 357–365 (2009)
75. H. Dai, H. Liu, Z. Jia, T. Chen, A multivariate classification algorithm for malicious node detection in large-scale wsns, in *Proceedings of IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)* (2012), pp. 239–245
76. T. Li, M. Song, M. Alam, Compromised sensor nodes detection: a quantitative approach, in *Proceedings of 28th International Conference on Distributed Computing Systems Workshops* (2008), pp. 352–357
77. I. Krontiris, T. Dimitriou, F.C. Freiling, Towards intrusion detection in wireless sensor networks, in *Proceedings of 13th European Wireless Conference* (2007), p. 7
78. B. Krishnamachari, I. Iyengar, Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks. IEEE Trans. Comput. **3**, 241–250 (2004)
79. A. Lazarevic, A. Ozgur, L. Ertoz, J. Srivastava, V. Kumar, A comparative study of anomaly detection schemes in network intrusion detection, in *Proceedings of the Third SIAM International Conference on Data Mining* (2003), pp. 1–8
80. M. Usman, V. Muthukkumarasamy, X.-W. Wu, S. Khanum, Securing mobile agent based wireless sensor network applications on middleware, in *International Symposium on Communications and Information Technologies (ISCIT)* (2012), pp. 707–712
81. Y.M. Kwon, S. Sundresh, K.L Mechitov, G. Agha, Actornet: an actor platform for wireless sensor networks, in *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems* (2006), pp. 1297–1300
82. C.-L. Fok, G.-C.n Roman, C. Lu, Agilla: a mobile agent middleware for self-adaptive wireless sensor networks. ACM Trans. Auton. Adapt. Syst. **4**(3), 16:1–16:26 (2009)
83. F. Aiello, G. Fortino, R. Gravina, A. Guerrieri, Maps: a mobile agent platform for java sun spots, in *Proceeding of the Third International Workshop on Agent Technology for Sensor Networks* (2009), p. 8
84. C. Muldoon, G.M.P. OHare, R. Collier, M.J. OGrady, Agent factory micro edition: a framework for ambient applications, in *Computational Science*, ed. by V.N. Alexandrov, G.D. van Albada, P.M.A. Sloot, J. Dongarra. Lecture Notes in Computer Science, vol. 3993 (Springer, Berlin, 2006), pp. 727–734
85. G. Vigna, Cryptographic traces for mobile agents, in *Mobile Agents and Security* (1998), pp. 137–153
86. J. Zhang, Y. Wang, V. Varadharajan, A new security scheme for integration of mobile agents and web services, in *Second International Conference on Internet and Web Applications and Services* (2007), p. 43
87. Y. Mu, R.H. Deng, M. Zhang, Q. Zhang, Secure mobile agents with designated hosts, in *Proceedings of IEEE Third International Conference on Network and System Security* (2009), pp. 286–293
88. N.S. Malik, D. Ko, H.H. Cheng, A secure migration process for mobile agents. Softw. Pract. Exp. **41**(1), 87–101 (2011)
89. O. Esparza, J.L. Munoz, J. Tomas-Builart, M. Soriano, An infrastructure for detecting and punishing malicious hosts using mobile agent watermarking. Wireless Commun. Mob. Comput. **11**(11), 1446–1462 (2011)
90. M. Usman, V. Muthukkumarasamy, X.-W. Wu, Formal verification of mobile agent based anomaly detection in wireless sensor networks, in *Proceedings of IEEE 38th Conference on Local Computer Networks Workshops (LCN Workshops)* (2013), pp. 1001–1009

91. S. Nanz, C. Hankin, A framework for security analysis of mobile wireless networks. Theoret. Comput. Sci. **367**(1–2), 203–227 (2006)
92. N. Mezzetti, D. Sangiorgi, Towards a calculus for wireless systems. Electron. Notes Theor. Comput. Sci. **158**(1), 331–353 (2006)
93. M. Merro, F. Ballardin, E. Sibilio, A timed calculus for wireless systems. Theoret. Comput. Sci. **412**(47), 6585–6611 (2011)
94. J. Godskesen, A calculus for mobile ad hoc networks, In *Coordination Models and Languages* (Springer, Berlin Heidelberg 2007), pp. 132–150
95. M. Merro, An observational theory for mobile ad hoc networks. Inf. Comput. **207**(2), 194–208 (2007)
96. A. Singh, C.R. Ramakrishnan, S. Smolka, A process calculus for mobile ad hoc networks, in *Coordination Models and Languages*, vol. 5052, Lecture Notes in Computer Science, ed. by D. Lea, G. Zavattaro (Springer, Berlin Heidelberg, 2008), pp. 296–314
97. Y.W. Law, R. Corin, S. Etalle, P.H. Hartel, A formally verified decentralized key management architecture for wireless sensor networks, in *Personal Wireless Communications*, vol. 2775, Lecture Notes in Computer Science, ed. by M. Conti, S. Giordano, E. Gregori, S. Olariu (Springer, Berlin, 2003), pp. 27–39
98. F. Werner, Applied formal methods in wireless sensor networks. Ph.D. thesis, Faculty of Computer Science, Karlsruhe Institute of Technology, 2009
99. A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. Drielsma, P. Heam, O. Kouchnarenko, J. Mantovani, S. Modersheim, D. Oheimb, M. Rusinowitch, J. Santiago, L. Turuani, M. Vigano, L. Vigneron, The avispa tool for the automated validation of internet security protocols and applications, in *Computer Aided Verification*, ed. by D. Lea, G. Zavattaro (Springer, Berlin, 2005), pp. 281–285
100. T. Murata, Petri nets: properties, analysis and applications. Proc. IEEE **77**(4), 541–580 (1989)
101. D. He, L. Cui, H. Huang, M. Ma, Design and verification of enhanced secure localization scheme in wireless sensor networks. IEEE Trans. Parallel Distrib. Syst. **20**(7), 1050–1058 (2009)
102. H. Rodriguez, R. Carvajal, B. Ontiveros, I. Soto, R. Carrasco, Using petri net for modeling and analysis of an encryption scheme for wireless sensor networks, in *Petri Nets Applications*, ed. by P. Pawlewski (InTech, Croatia, 2010)
103. H.-R. Tseng, R.-H. Jan, W. Yang, A robust user authentication scheme with self-certificates for wireless sensor networks. Secur. Commun. Netw. **4**(8), 815–824 (2011)
104. Z. Sbai, M. Escheikh, Model checking techniques for verication of an encryption scheme for wireless sensor networks, in *Proceeding of the International Conference on Information Processing and Wireless Systems* (2012), p. 6
105. Z. Bankovic, J.C. Vallejo, P. Malagon, A. Araujo, J.M. Moya, Eliminating routing protocol anomalies in wireless sensor networks using ai techniques, in *Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security* (2010), pp. 8–13

Chapter 3
Abnormality Identification and Confirmation System

3.1 Introduction

Abnormalities can severely disrupt the performance of a sensor network application. In a worst-case scenario, abnormalities caused by attacks or faults can completely halt the functioning of a sensor network application. The timely detection of abnormalities and then identification of the source of abnormalities are, therefore, imperative for their effective mitigation. This chapter has introduced an abnormality identification and confirmation system which can not only timely detect the different nature of abnormalities, but also effectively identify the source of abnormalities.

This chapter is structured as follows: A few formal definitions and terminologies are discussed in Sect. 3.2. The network model is drawn in Sect. 3.3. The internal structural details of the presented system are described in Sect. 3.4. The algorithmic specifications of the system and their complexity analysis are presented in Sect. 3.5. The model formulation and its formal analysis are carried out in Sect. 3.6. The extension of the model is made by defining a related Generalized Stochastic Petri net model in Sect. 3.7 to formalize the time-based characteristics of the presented system. The time-based conduct validation of the system through experimentation on a real test bed is performed in Sect. 3.8. The key results of the work are presented in Sect. 3.9. Finally, the conclusion is drawn in Sect. 3.10.

3.2 Terminologies and Formal Definitions

All symbolism is locally defined at the time of their first use, and a list of key notations is provided in Appendix A for reference. The readers are referred to the study [12] for the definition of Petri and Generalized Stochastic Petri nets. The formal definitions of behavioral and structural properties, namely reachability, safeness, boundedness, liveness, and deadlock are given in [1, 2].

M. Usman et al., *Mobile Agent-Based Anomaly Detection and Verification System for Smart Home Sensor Networks*,
https://doi.org/10.1007/978-981-10-7467-7_3

3.3 Network Model

The structure of the network is presumed to be a clustered network with hierarchy. A r_{th} cluster C_r is formed by the collection of s number of motes, msn, at the lowest level, namely leaf level, such that $C_r = \{msn_i|1, , s\}$. Let msn_q represent the node of interest from the cluster C_r. The symbol MSN represents all member nodes in the network. The cluster leader nodes (CLNs) are in-charge nodes of their proximities, and they are equipped with more resources as compared to other ordinary member nodes. The Ξ number of CLNs is linked with a node, namely base station (BS), which is chief of the network and responsible to operate application and handle received data. A generic formation of the network model is illustrated in Fig. 3.1 [3]. The general network functionality has following characteristics:

- The interaction between constituents of the network is non-deterministic due to several aspects such as channel errors and environmental influence [4].
- The CLNs keep track of the statuses of the resources of the member nodes through the synchronized resource management technique which is facilitated by TinyOS [5].
- The agents, namely abnormality agents (AAs), have watermark embedded in their codes. Thus, they are safe from attacks [6]. The key steps of the algorithm used in this study are following: (i) Select **p** and **q** as numbers, namely prime numbers, where $\mathbf{p} \times \mathbf{q} = \mathbf{N}$, (ii) insert **N** in graph G, (iii) generate watermark W to create G, (iv) embed W in the original code O to generate O_o code so that if the given

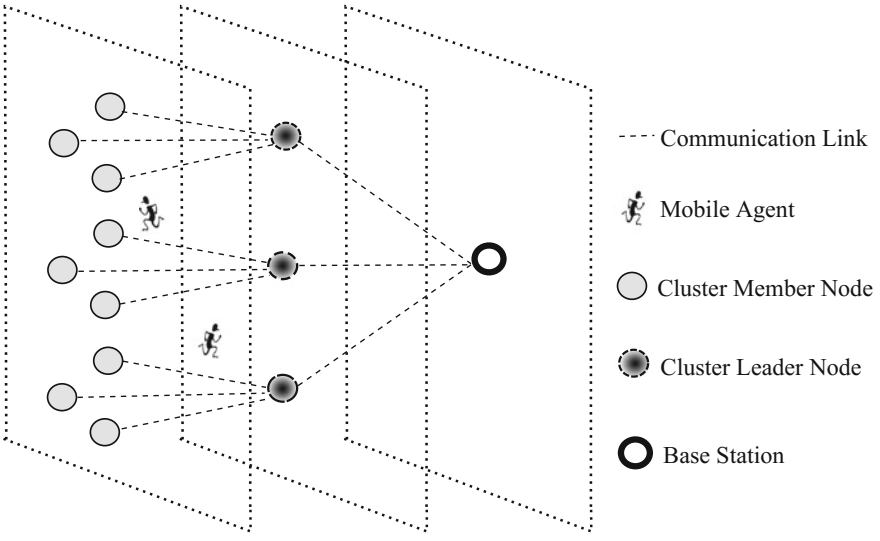

Fig. 3.1 Network model: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. A resource-efficient system for detection and verification of anomalies using mobile agents in wireless sensor networks. Journal of Networks, 9(12):3427–3444, 2014

input I, the recognizer R is able to extract W and N (v) employ tamperproofing to avoid the removal of W (generating O_1), (vi) employ obfuscation to make analysis difficult (generating O_2), at this stage, R and W becomes R 0 and W 0, (vii) extract $R^{'}$ and distribute code O_3, (viii) the attacker can generate O_4 by modifying O_3, and (ix) finally, the cluster leader node can then verify the originality of the code by applying $R^{'}$ to O_4 in the case of an attack.

- The movement of AA is restricted among only cluster leader and member nodes for optimum utilization of energy resources.
- All leaf nodes are assumed to be susceptible to different nature of abnormalities. The other entities in the network are considered as secure.
- The bootstrap values of all member nodes are kept within [0,1], a closed interval. These values vary at run time due to the dynamic changes in the behavior of nodes.

3.4 Architecture of Abnormality Identification and Confirmation Module

The description of the composition of abnormality identification and confirmation module (ADVM) is elucidated below.

3.4.1 Abnormality Identification and Confirmation Module

Every cluster leader node is set up with its own ADVM which accomplishes several imperative tasks such as detection of abnormalities, transmission of agents, and transmission optimization of agents. The ADVM is made up of three sub-modules, namely coordination unit, abnormality agent, and repository Fig. 3.2 [3] depicts the deployment of ADVMs on several cluster leader nodes.

Coordination Unit: This is a main component of ADVM. It facilitates coordination among internal elements of ADVM and detection of abnormalities coordination with the BS node. The element CU fetches readings from inbound data traffic to carry out abnormality identification. The legitimate reading is transmitted to Aggregation Unit (AU) which aggregates sensed data and periodically transmits them to BS. Conversely, that is, in the case of anomalous data, CU is entitled to initiate the following actions: (i) Transmit abnormality agent to perform on-the-spot confirmation of suspicious node, (ii) trigger an alarm to the BS node, (iii) declare the cluster member node as faulty or antagonist, and (iv) reduce interaction with the doubtful node.

Abnormality Agent: It is a tiny piece of code which is made up of four fragments, namely identity, data, code, and itinerary. Every agent possesses an inimitable identity. The agent itinerary is based on the target node address. The code fragment is composed of on-the-spot confirmation procedure code. The data fragment contains

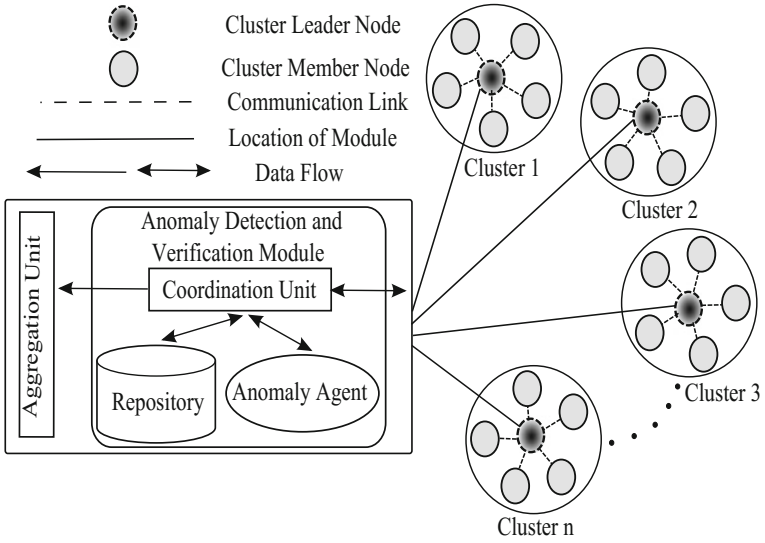

Fig. 3.2 A depiction of the deployment of ADVM on the cluster leader nodes: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. A resource-efficient system for detection and verification of anomalies using mobile agents in wireless sensor networks. Journal of Networks, 9(12):3427–3444, 2014

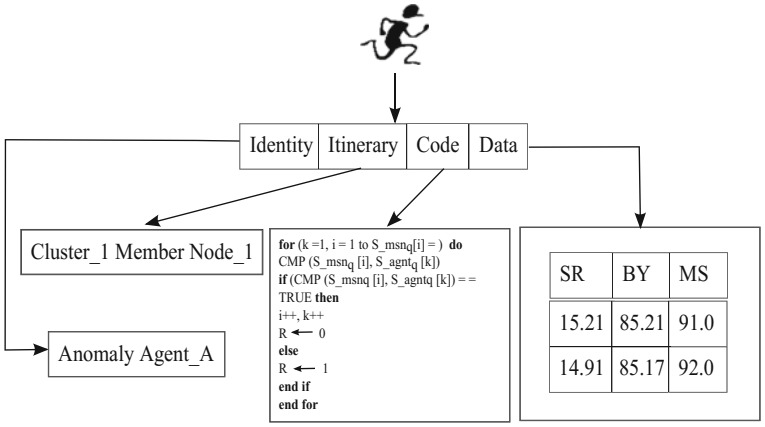

Fig. 3.3 The depiction of the internal composition of an abnormality agent: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. A resource-efficient system for detection and verification of anomalies using mobile agents in wireless sensor networks. Journal of Networks, 9(12):3427–3444, 2014

values of previous data packets that are employed for on-the-spot confirmation of suspicious node. Fig. 3.3 [3] shows the composition of an abnormality agent.

CU can activate an agent for on-the-spot confirmation of the doubtful node after the identification of abnormalities in received data packets. The following cases may occur as a consequence of the agent transmission: (i) the correct execution of an agent on a doubtful node and communication of on-the-spot confirmation result to a cluster leader node, (ii) transfer of agent to doubtful node is endangered to the attack, namely agent-execution manipulation, and (iii) the doubtful node may escape the examination of an agent and as a result it may not transmit back any outcome to the cluster leader node. For the case (i), the agent will transfer the required values of the data to the cluster leader node. For the case of (ii), the mechanisms for the protection of agent-execution integrity may be integrated to secure from the execution-manipulation attack [7]. For the case (iii), if the doubtful node prohibits the examination of the agent, and no outcome is transmitted to the cluster leader node, this will prove abnormal status of node, that is the eventual aim of transmitting that agent.

Repository: It is denoted by RP, and it can store the values of features of interest of the normal profile and other associated data as five tuples. These five tuples then facilitate the procedures of identification of abnormalities and transmission of agents. The composition of repository is conceived as an addition to the preliminary concept stated in [8]. The tuples can be formally defined as shown below.

$$RP = \langle msn_q^{id}, RS, FS, AO, AS \rangle \tag{3.1}$$

In Eq. (3.1), msn_q^{id} is a column vector that keeps the identities of nodes which are members of a particular cluster. The tuple, RS, stores statuses of the resources of nodes. Every member node has several resources, for instance, memory and battery. The values of these resources are kept in the memory in the form of an $\mathbf{m} \times \mathbf{n}$ matrix; here $\mathbf{m}$ represents strength of member nodes in terms of numbers and $\mathbf{n}$ denotes resources. The FS tuple, on the other hand, stores values of features. The single value of FS, for nodes with similar responsibilities in the cluster, is stored on the cluster leader node for the optimum utilization of its memory. If the sufficient memory space is available, then different values of FS can be stored to facilitate high level of security. The fundamental structure of FS is shown below.

$$FS = \langle \lambda, \jmath, \varphi, \upsilon, f \rangle \tag{3.2}$$

In Eq. 3.2, λ outlines the minimum to maximum boundaries of the values of the sensor reading. For instance, λ can keep values in the range of $16-34\,^{\circ}\mathrm{C}$ to set the usual behavior of a member node of the cluster. The $\jmath$ feature represents the time interval, which is used to observe the actions of the member node for a specific duration of time. This activity saves record of the usual behavior of the member node in connection with other factors such as sensor reading and resource status. The φ feature keeps the values for the actions which are entitled such as sleeping, wake-up, sensing, and transmission of sensed data values, which are performed by the member

nodes. The f and υ features represent count of the received packets and status of member node resources, respectively.

The AO tuple is a $\mathbf{w} \times \mathbf{n}$ matrix. This matrix saves $\mathbf{w}$ abnormal observations, which stores the computed values of thresholds that are used to optimize the transmission of abnormality agents. The optimization of agent transmission method is described in the next chapter (see Sects. 4.2.2 and 4.2.3). The structure of the AO is expressed below.

$$AO = FS^{'} \tag{3.3}$$

The $FS^{'}$ feature stores values of $\lambda^{'}$, $\jmath^{'}$, $\varphi^{'}$, $\upsilon^{'}$, and $f^{'}$, the notation $\lambda^{'}$ denotes the abnormal values of sensor readings, the $\jmath^{'}$ feature represents the unusual actions which are performed by the member node, the $\varphi^{'}$ feature denotes actions performed by the member node but they are unauthorized, and the $f^{'}$ feature represents the unusual frequency of the received data packets.

The AS tuple represents the values of the set of actions that aids the functionality of the abnormality identification procedure. The AS tuple is made up of two action classes, viz. κ and τ.

$$AS = \langle \kappa, \tau \rangle \tag{3.4}$$

In Eq. (3.4), κ represents the actions related to the abnormality identification procedure. Subsequently, the ADVM transmits the usual reading to a unit, namely aggregation unit. If the received observation is anomalous, then an abnormality agent would be transmitted by the detection module. Alternatively, it may take diverse action(s): (i) declaring the relevant member node as faulty of anomalous to other member nodes within that cluster, transmitting an alarm to the chief node, i.e., BS, and reducing the interaction with the doubtful node. The τ tuple denotes tuning actions such as changes in the limits of the zones, namely normal, tolerated, and anomalous. It is stated that the κ action is automatically executed by ADVM after the receipt of every observation, on the other hand, the τ action can be started by a user from the BS node. Tables 3.1 [3] and 3.2 [3] present descriptions of the set of actions. Similarly, Figs. 3.4 [3] and 3.5 [3] illustrate flows of the actions. Note that the detection part of the sample set of actions and corresponding flow of those actions are based on the first-order abnormalities detection algorithm which is presented in Sect. 4.2.1.

3.5 Algorithms and Analysis

The trustworthiness of the sensor network applications is mainly reliant on the accurately received data. However, data packets, communicated by the cluster member nodes, are vulnerable to attacks and faults. A resilient abnormality identification system, therefore, must be capable of detecting the origin of abnormalities along with the detection of abnormalities before taking fitting actions against the antagonist cluster member node. The system, described in this book, can discover abnormalities and

Table 3.1 κ actions definitions: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. A resource-efficient system for detection and verification of anomalies using mobile agents in wireless sensor networks. Journal of Networks, 9(12):3427–3444, 2014

State	Description
A	Carry out identification of abnormalities procedure on the basis of the first-order bounds
B	Transmit aggregated data to the related unit, i.e., aggregation unit
C	Cluster leader node verifies the member node behavior to facilitate the communication of the agent
D	Forward an agent to the member node for on-the-spot confirmation of the node
E	Reduce the interaction with the cluster member node
F	Communicate status of faulty node to other leader nodes
G	Transmit an alarm to BS regarding the malicious cluster member node

Table 3.2 τ actions definitions: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. A resource-efficient system for detection and verification of anomalies using mobile agents in wireless sensor networks. Journal of Networks, 9(12):3427–3444, 2014

State	Description
H	Update bounds of the normal zone of FS
I	Update bounds of the tolerance zone of FS
J	Update abnormality identification bounds of the λ and $\jmath$ features
K	Update abnormality identification bounds of the $\jmath$ and υ features
L	Update abnormality identification bounds of the φ and υ features
M	Update abnormality identification bounds of the φ and $\jmath$ features
N	Update abnormality identification bounds of the F and $\jmath$ features

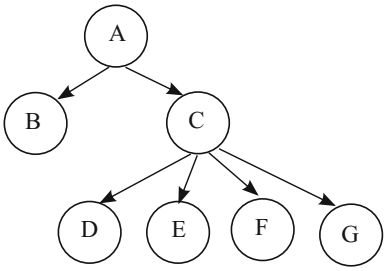

Fig. 3.4 Flow of states of κ actions: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. A resource-efficient system for detection and verification of anomalies using mobile agents in wireless sensor networks. Journal of Networks, 9(12):3427–3444, 2014

then it is capable of using those values for on-the-spot confirmation of the doubtful cluster member nodes by using abnormality identification. This section elucidates algorithmic specifications of the system.

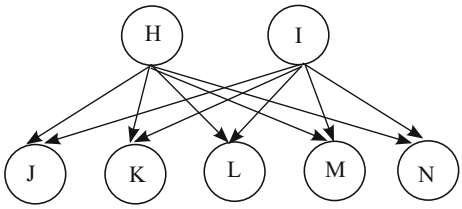

Fig. 3.5 Flow of states of τ actions: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. A resource-efficient system for detection and verification of anomalies using mobile agents in wireless sensor networks. Journal of Networks, 9(12):3427–3444, 2014

Algorithm 3.1 Collection of the values of features by a cluster member node

Input: B_t^l
Output: F_q
1: *At t* time
2: **if** $B_c^l \geq B_t^l$ **then** //confirm the status of battery
3: **while** $i \leq j$ **do**
4: *Clct* ($F_q[i]$) //accumulate the i_{th} feature values of the i_{th} cluster member node
5: $S\_msn_q[i] \leftarrow F_q[i]$ //save i_{th} segment value in the stack
6: $F_q \leftarrow F_q + F_q[i]$ //combine values before dispatch
7: $i \leftarrow i + 1$
8: **end while**
9: *TRNSMT* F_q to cln_q //dispatch data packet, F_q, to cln_q
10: **go to** sleep mode
11: **else**
12: msn_q fails to wake up
13: **end if**

3.5.1 Features Collection by the Cluster Member Node

A node in question, namely msn_q, awakens at t time to verify the status of the battery. If B_c^l, the current level of battery of msn_q, is higher than the already defined threshold, B_t^l, then it accumulates values for v features. The value of v is 3, and $F_q = \{MS, BY, SR\}$, where MS, SR, and BY denote memory status, sensor reading, and battery status, respectively. The msn_q node stores the values of F_q in its stack memory segment after the concatenation of the values to facilitate the on-the-spot confirmation procedure. Then stored values are communicated to, cln_q, the related cluster leader node, for more processing. The msn_q node moves into the sleep state after performing its assigned job. This procedure is performed periodically according the requirements of the application specifications. The pseudocode of this procedure is provided in Algorithm 3.1 [9].

3.5.2 Abnormality Identification by the Cluster Leader Node

A cluster leader node, cln_q, obtains the data traffic, from the msn_q, based on the values of features, that is, $F_q = \{MS, BY, SR\}$, inside the specified timeslot, that is, $T_i^{lb} \leq T_{ar}(F_q) := T_{ar}^{F_q} \leq T_i^{ub}$, where T_i^{lb} represents the initial time, T_i^{ub} denotes the end time, and $T_{ar}(F_q)$ a function that calculates the time of arrival, $T_{ar}^{F_q}$, of F_q. The IEEE standard, namely 802.15.4, categorizes the timeline of communication into two approaches, viz. Contention-Free Period (CFP) and Contention Access Period (CAP) [10]. In the former approach, msn_q attains the timeslot which is guaranteed to interact with the cln_q. The abnormality identification algorithm uses the CFP approach for obtaining the values of F_q. The msn_q node is assumed as abnormal if the received values of F_q are outside of the designated timeslot. A classic example situation is shown in Fig. 3.6 [9], wherein the cln_q node receives the values of F_q outside of the designated timeslot. Therefore, in such cases, the msn_q node must be treated as abnormal. In the situation, if the values of F_q are not received by the cln_q node from the msn_q node within the designated timeslot, then it sends the agent, AA, to the msn_q node to carry out on-the-spot confirmation of its antagonist behavior. The agent transmits old values of F_q to verify the msn_q node is behaving abnormally or if the abnormality has occurred during the communication of the values of F_q among the msn_q and cln_q nodes. In this situation, the cln_q node also decreases the trust counter of the msn_q node by a ς factor (which is a system administrator specified number) and transfers d_i^{al} to BS, which is an alarm. In the situation, when the node, namely cln_q obtains the values of F_q inside the designated timeslot, it carries out the identification of abnormalities procedure by matching the F_q values with that of the related normal profile bounds, $Prf_q = \{(SR^{lb}, SR^{ub}), (MS^{lb}, MS^{ub}), (BY^{lb}, BY^{ub})\}$, to identify

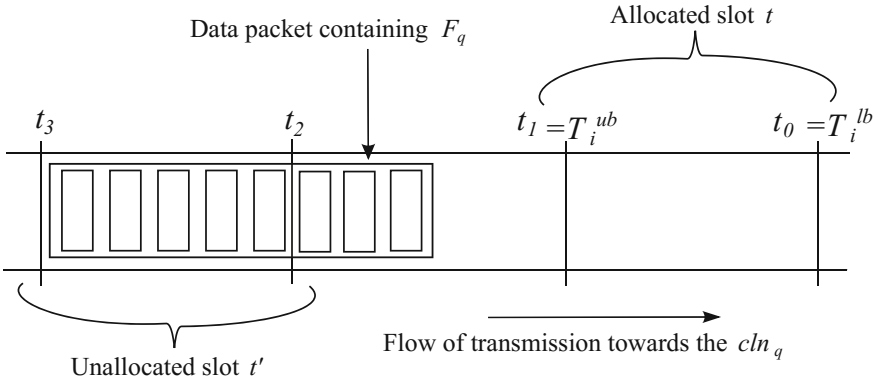

Fig. 3.6 A classic example situation of time-based abnormality: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. Specification and validation of enhanced mobile agent-enabled anomaly detection in resource constrained networks. Journal of Networks, 10(6):353–368, 2015

Algorithm 3.2 Abnormality detection by the cluster leader node

Input: F_q, Prf_q, TR
Output: SR, d_i^{al}, d_i^{ag}, AA, TR

1: **if** $T_i^{lb} \leq T_{ar}(F_q) := T_{ar}^{F_q} \leq T_i^{ub}$ **then** //check F_q is received inside designated timeslot
2: $CHK(F_q, Prf_q)$ //perform abnormality identification
3: **if** $CHK == TRUE$ **then**
4: $A\_unt \leftarrow SR$ //save reading to aggregation unit
5: $TRNSMT\ d_i^{ag}$ to BS //dispatch accumulated data toe BS after T_i^{ub}
6: **break**
7: **else**
8: $TRNSMT\ d_i^{al}$ to BS //trigger abnormality alarm to BS
9: $TRNSMT\ AA$ to msn_q //transmit abnormality agent to msn_q
10: **if** $TR > 0$ **then** //check level of trust
11: Decr TR by ς //decrement the level of trust
12: **end if**
13: **end if**
14: **else**
15: $TRNSMT\ d_i^{al}$ to BS //trigger abnormality alert to BS
16: $TRNSMT\ AA$ to msn_q //transmit abnormality agent to msn_q
17: **if** $TR > 0$ **then** //check level of trust
18: Decr TR by ς //decrease the level of trust
19: **end if**
20: **end if**

the abnormalities. If the matching result is correct, then the values of SR are accumulated by A_unt within a designated interval of time prior to dispatch to BS. In the contrary, if the cln_q node identifies an abnormal reading, then it transmits an agent for on-the-spot confirmation of the antagonist actions of the msn_q node, triggers an alarm, d_i^{al}, to BS, and also reduces the count of trust by the ς value. The pseudocode for the procedure of the abnormality detection is given in Algorithm 3.2 [9].

3.5.3 Anomalous Node Confirmation

A node, namely msn_q, which is a cluster member node, receives AA. It then inserts the abnormalities detection related data values into the stack segment of the memory, denoted by S_msn_q. This is the part of the node memory which is reserved for on-the-spot confirmation of the origin of abnormalities procedure. The AA carries out this procedure by matching the MS, SR, and BY values of the stack segment of the memory of the mote and the values that are brought by the agent. If the results are correct, then the value 0, normal rank of the mote, is saved.

If even not a single value is matched, then the value 1, showing the mote status as under on-the-spot attack or fault, is saved. The either form of the results is forwarded to the related cluster leader mote. It is pertinent to observe that if the presented system is only positioned for those abnormalities which happen due to errors or faults, the

Algorithm 3.3 Anomalous mote confirmation

Input: AA, readings of SR, BY, MS saved in S_msn_q
Output: WR
1: msn_q obtains AA //msn_q receives abnormality agent
2: $PUSHAA$ in S_agnt_q //abnormality agent saves in the dedicated space of memory
3: **for** (k = 1, i = 1 to $S\_msn_q[i] = \epsilon$) **do**
4: CMP ($S\_msn_q[i]$, $S\_agnt_q[k]$) //contrast past saved values of parameters with those values which are carried by the abnormality agent
5: **if** (CMP ($S\_msn_q[i]$, $S\_agnt_q[k]$)) = = $TRUE$ **then**
6: i + +, k + +
7: $R \leftarrow 0$ //save "0" result, showing no abnormalities
8: **else**
9: $R \leftarrow 1$ //save "1" result, representing abnormalities
10: **end if**
11: **end for**
12: $WMA(R) := WR$ //abnormality agent integrate watermark in outcome
13: $TRNSMT\ WR$ to cln_q //dispatch the watermarked outcome to cln_q

value of R will be directly communicated to the related cluster leader mote. On the contrary, if the presented system is also positioned for other types of abnormalities occur due to attacks, the agent is designated to include a watermark in the result. This approach prevents result from the manipulation attack. The insertion of watermark in result, however, increases transmission and communication costs of on-the-spot confirmation procedure. The overhead of the watermarking is examined in Sect. 3.8. The pseudocode of this procedure is provided in Algorithm 3.3 [9].

3.5.4 Status Update on the Cluster Leader Mote

A mote, cln_q, gets WR, the watermark inserted result, inside the specified time limit and disintegrates the result and watermark. If the obtained result shows the antagonist status of the member mote, then the cluster leader mote reduces the value of trust of a doubtful mote by ς factor and transmits an alert to BS to update the status of mote. On the contrary, if the obtained notification is "0", then the member mote is treated as trouble-free mote and the leader mote takes no action against the member mote. If no result, regarding on-the-spot confirmation, is received by the leader mote inside the specified time limit, it reduces the value of trust of the member mote and sends an alert to BS.

It is imperative to note that leader mote would broadcast an alert and reduce the value of trust in situations such as after discovery of abnormalities and after obtaining on-the-spot confirmation results. If on-the-spot confirmation procedure is not successful and BS is not communicated with the abnormality identification result, then a system administrator may have no information about the detection of abnormality. The pseudocode of is listed in Algorithm 3.4 [9].

Algorithm 3.4 Update of status on the cluster leader mote

Input: WR, TR
Output: TR, d_j^{al}

1: **if** $T_j^{lb} \leq T_{ar}(WR) := T_{ar}^{WR} \leq T_j^{ub}$ **then** //check confirmation outcome obtained inside the designated timeslot
2: $RmW(WR) := R$ //separate watermark from outcome
3: **if** $R == 1 \wedge TR > 0$ **then** //check confirmation outcome and values of trust
4: Decr TR by ς //decrease the value of trust
5: $TRANSMT\ d_j^{al}$ to BS //dispatch alert
6: **else**
7: **break**
8: **end if**
9: **else**
10: **if** $TR > 0$ **then**
11: Decr TR by ς //decrease the value or trust
12: $TRANSMT\ d_j^{al}$ to BS //dispatch alert
13: **end if**
14: **end if**

Algorithm 3.5 Status update on the base station mote

Input: $d_i^{ag} \wedge d_i^{al} \vee d_j^{al}$
Output: update A_rep, update A_{data}

1: **if** BS mote receives $d_i^{al} \vee d_j^{al}$ **then** //obtained packet is an alert
2: update A_{rep} //repository of application update
3: **break**
4: **else if** BS mote receives d_i^{ag} **then** //aggregated data inside the obtained packet
5: update A_data //application data update
6: **else**
7: **break**
8: **end if**

3.5.5 Update of Status on Base Station

A mote, namely BS, obtains either the abnormality alarm, denoted by d_j^{al} or d_i^{al}, or the data which is aggregated, represented by d_i^{ag}, from the leader mote. For the former, BS triggers a message for user to inform about antagonist status of the member mote. For the latter, BS stores the accumulated data for further analysis. The pseudocode is listed in Algorithm 3.5 [9].

3.5.6 Complexity Analysis

The space and time complexities of the proposed algorithms are analyzed in this section.

Theorem 3.1 *The space complexity for (i) the procedure for the collection of the values of features by msn_q has upper bound by $l[a]$; (ii) the procedure of abnormality identification on cln_q is $C_a + l[b]$; (iii) the procedure of on-the-spot confirmation of msn_q is $l[c]$; (iv) the procedure of update of status on cln_q is $l[d]$; and (v) the procedure of update of status on BS is $l[e]$.*

Proof (i). Let $FS_1 = \{\lambda, \varphi, \upsilon\}$ represent the features of interest values that are calculated on msn_q and then employed by AA for on-the-spot confirmation procedure. The notation $l[a]$ represents the stack memory length which stores the values of features and $msnq[x]$ denotes the stack memory length, where $msnq[x] > l[a]$. Assuming a as values of maximum number of features that are collected by msn_q, the space complexity has upper bound as $l[a]$.

(ii). Let $F_q = \{SR; MS; BY\}$ be the values of the features which are received by the cln_q. Let $Prf_q = \{(SR^{lb}, SR^{ub})(MS^{lb}, MS^{ub}), (BY^{lb}, BY^{ub})\}$ be the corresponding bounds to perform the abnormality identification on features. Thus, the space of memory consumed by b features, which belong to F^q, becomes $l[b]$. The cln^q takes constant memory spaces C^1, C^2, C^3, and C^4 to store the values of the bounds for the abnormality identification, aggregated sensed data (d_i^{ag}) allocated timeslot values (T_i^{lb} and T_i^{ub}), and the trust value (TR) of the msn^q, respectively. The abnormality confirmation agent consumes C^5 and C^6 spaces of memory to save data and code of the abnormality confirmation agent, correspondingly, where space of memory reserved for data also stores the values of identity and itinerary of the abnormality confirmation agent (AA). Therefore, the overall space of memory consumed by the abnormality identification procedure is $\bigcup_{a=1}^{6} C_a + l[b]$.

(iii). Let $msn_q[y]$ assume as the stack memory length of msn_q which accommodates the procedure of on-the-spot confirmation. The mote $msn_q[y]$ must fulfill the following two conditions: (a) $msn_q[y] > l[b]$ and (b) $msn_q[y] > C_6$, where $C_7 = C_5 \bigcup C_6$. This implies that the msn_q memory must facilitate the accumulated F_q values, and the data and code of AA for on-the-spot confirmation procedure. Let $l[c]$ be an upper bound of total F_q memory for both C_6 and $l[b]$, the space complexity for on-the-spot confirmation of msn_q is $l[c]$.

(iv). The cln_q takes constant memory spaces C_4, C_8, and C_{10} to store the trust value (TR) allocated timeslot values (T_i^{lb}), and T_i^{ub}, and in situ confirmation result value (WR). These memory spaces must hold the relation $C_{10} > C_8 > C_4$, as C_{10} holds the watermarked on-the-spot confirmation result (WR) which takes more space as compared to T_i^{lb} and T_i^{ub} values (stored by C_8 memory space) and the TR value (stored by C_4 memory space). Thus, considering $l[d]$ as the upper bound of the combined C_4, C_8, and C_{10} spaces, the space complexity of the algorithm of the status update on the cln_q is $l[d]$.

(v). The BS mote takes C^{11} memory space to store received abnormality alarm, d_i^{al} or d_i^{al}, from the cln_q. Similarly, the memory space C_{12} is taken by the BS mote to store the received aggregated data (d_j^{al}). Considering $l[e] = C_{11} \bigcup C_{12}$, the space complexity for the status update on the BS is $l[e]4$.

Theorem 3.2 *The time complexity for (i) The procedure for the collection of the values of features by msn_q is $O(l)$; (ii) the procedure of abnormality identification*

on cln_q is U; (iii) the procedure of on-the-spot confirmation of msn_q is $O(m)$; (iv) the procedure of update of status on cln_q is constant time V ; and (v) the procedure of update of status on BS is constant time W.

Proof (i). Time complexity for procedure of collection of features is primarily dependent on $FS_1 = \{\lambda, \varphi, \upsilon\}$; here $FS_1 = \{\iota, f\}$ is calculated on cln_q after obtaining FS_1 values from msn_q. Let msn_q takes l time to accumulate values of FS_1 from its proximity and save them. The mote msn_q consumes U_1, a constant time, to communicate F_q values to cln_q. Taking the case of upper bound, the features collection procedure has $O(l)$ time complexity.

(ii). The cln_q takes constant time U_2 to get the featuresF_q from msn_q, U_3 time to check the condition $T_i^{lb} \leq T_{ar}(Fq) := T_{ar}^{F_q} \leq T_i^{ub}$, that is, F_q received within the allocated timeslot, and U_4 time to perform the abnormality identification, $CHK(Fq, Prf_q)$. The cln_q consumes U_5 time to aggregate the sensor reading, U_6 time to check the trust value consumes, U_7 time to decrement the trust value, U_8 time for the abnormality confirmation agent dispatch to msn_q; here msn_q is antagonist in such situations. The cln_q consumes U_9 time to transmit abnormality alarm, d_i^{al}, to the BS mote. Thus, considering $\bigcup = \sum_{i=2}^{9} \bigcup_i$, the algorithm for the abnormality identification procedure runs in a constant time U.

(iii). The mote msn_q receives AA and transmits on-the-spot confirmation result to cln_q in U_{10} to U_{11} times. The mote msn_q takes U_{12} time to add a watermark in on-the-spot confirmation result. The mote msn_q takes m time to carry out the operation of comparison among the values of S_{msn_q} and S_{agnt_q}. Thus, by taking the upper limit on time consumed by the procedure of comparison, the complexity of on-the-spot confirmation procedure is $O(m)$.

(iv). The cln_q takes constant V_1 time to check confirmation result within the allocated timeslot values. It takes constant time V_2 to remove the watermark from the received result, V_3 to check the confirmation result, V_4 to check trust value, V_5 to decrement the trust value, and V_6 to transmit an abnormality alarm to BS. Thus, considering $V = \sum_{i=1}^{6} V_i$, the algorithm for the status update procedure on the cln_q runs in a constant time V.

(v). The BS mote consumes constant time W_1 to receive an abnormality alarm, d_i^{al} or d_j^{al}, from the cln_q. Similarly, the BS mote consumes time W_2 to store the received aggregated data, d_j^{al}. Thus, considering $W = W_1 + W_2$, the algorithm for the status update procedure on the BS mote runs in a constant time W.

3.6 Formal Model

The above-cited algorithmic specifications are primarily transformed into associated Petri net modules. Subsequently, an integration, viz. bottom-up is performed to validate a unified model that specifies the formalization of the properties of the system. This procedure is initiated by a refined mapping between algorithmic specifications and Petri net modules. In this course of events, both on-the-spot and during the

transmission abnormal states of the member motes are also considered. The states, namely the wakeup by mote, threshold level, and other are formalized using places, which are represented by p, whereas transitions which are denoted by t are used to formalize different actions, such as activation of mote, check of battery level.

The first algorithmic description shows the procedure of collection of F_q by msn_q. The formal procedure description is provided next.

Net module 1: (Collection of features by msn_q). The features collection net module, (PN_1), is a 5-tuple net: $PN_1 = (P_1, T_1, F_1, W_1, (M_0)_1)$, where $P_1 = \{p_1, p_2, p_3, p_4\}$ and $T_1 = \{t_1, t_2, t_3, t_4, t_5\}$ are non-empty, finite, and disjoint sets of places and transitions, correspondingly. $F_1 = \{(p_1, t_1), (t_1, p_2), (p_2, t_2), (p_2, t_3), (p_2, t_4), (t_2, p_3), (t_3, p_3), (t_4, p_3), (p_3, t_5), (t_5, p_4)\}$. $W_1(p_1, t_1) = W_1(t_1, p_2) = W_1(p_2, t_2) = W_1(p_2, t_3) = W_1(p_2, t_4) = W_1(t_2, p_3) = W_1(t_3, p_3) = W_1(t_4, p_3) = 1$, $W_1(p_3, t_5) = W_1(t_5, p_4) = 3$, and $(M_0)_1 = p1$.

The formal description of the identification of the abnormalities is provided below.

Net module 2: (Abnormality identification by the cln_q). The abnormality identification net module, (PN_2), is a 5-tuple net: $PN_2 = (P_2, T_2, F_2, W_2, (M_0)_2)$, where $P_2 = \{p_5, p_6, p_7, p_8, p_9, p_{10}\}$ and $T_2 = \{t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}\}$ are non-empty, finite, and disjoint sets of places and transitions, correspondingly. $F_2 = \{(p_5, t_6), (t_6, p_6), (p_6, t_7), (p_6, t_8), (p_6, t_9), (t_7, p_7), (t_7, p_8), (t_8, p_7), (t_8, p_8), (t_9, p_7), (t_9, p_8), (p_7, t_{10}), (p_8, t_{11}), (t_{10}, p_9), (t_{11}, p_{10}), (p_9, t_{12}), (p_{10}, t_{13})\}$. $W_2(p_5, t_6) = W_2(t_6, p_6) = W_2(p_8, t_{11}) = W_2(t_{11}, p_{10}) = W_2(p_{10}, t_{13}) = 3$, $W_2(p_6, t_7) = W_2(p_6, t_8) = W_2(p_6, t_9) = W_2(t_7, p_7) = W_2(t_7, p_8) = W_2(t_8, p_7) = W_2(t_8, p_8) = W_2(t_9, p_7) = W_2(t_9, p_8) = W_2(p_7, t_{10}) = W_2(t_{10}, p_9) = W_2(p_9, t_{12}) = 1$, and $(M_0)_2 = 3p5$.

The on-the-spot confirmation procedure which is carried out on msn_q is formalized next.

Net module 3: (Anomalous mote, msn_q, confirmation). The abnormal mote confirmation net module 3, (PN_3), is a 5-tuple net: $PN_3 = (P_3, T_3, F_3, W_3, (M_0)_3)$, where $P_3 = \{p_{11}, p_{12}, p_{13}\}$ and $T_3 = \{t_{14}, t_{15}, t_{16}, t_{17}, t_{18}\}$ are non-empty, finite, and disjoint sets of places and transitions, correspondingly. $F_3 = \{(p_{11}, t_{14}), (t_{14}, p_{12}), (p_{12}, t_{15}), (p_{12}, t_{16}), (p_{12}, t_{17}), (t_{15}, p_{13}), (t_{16}, p_{13}), (t_{17}, p_{13}), (p_{13}, t_{18})\}$. $W_3(p_{11}, t_{14}) = W_3(t_{14}, p_{12}) = 3$, $W_3(p_{12}, t_{15}) = W_3(p_{12}, t_{16}) = W_3(p_{12}, t_{17}) = W_3(t_{15}, p_{13}) = W_3(t_{16}, p_{13}) = W_3(t_{17}, p_{13}) = W_3(p_{13}, t_{18}) = 1$, and $(M_0)_3 = 3p_{11}$.

The next procedure elucidates msn_q status update on cln_q after the on-the-spot confirmation procedure. The formal description is provided next.

Net module 4: (Update of status on cln_q). The update of status on the cluster leader mote net module, (PN_4), is a 5-tuple net: $PN_4 = (P_4, T_4, F_4, W_4, (M_0)_4)$, where $P_4 = \{p_{14}, p_{15}, p_{16}, p_{17}\}$ and $T_4 = \{t_{19}, t_{20}\}$ are non-empty, finite, and disjoint sets of places and transitions, correspondingly. $F_4 = \{(p_{14}, t_{19}), (t_{19}, p_{15}), (p_{15}, t_{20}), (t_{20}, p_{16}), (t_{20}, p_{17})\}$. $W_4(p_{14}, t_{19}) = W_4(t_{19}, p_{15}) = W_4(p_{15}, t_{20}) = W_4(t_{20}, p_{16}) = W_4(t_{20}, p_{17}) = 1$ and $(M_0)_4 = p_{14}$.

Last but not least, fifth and final algorithmic description elucidates the procedure of on-the-spot confirmation result and handling of aggregated data by BS.

Net module 5: (Update of status on BS). The update of status on BS net module, (PN_5), is a 5-tuple net: $PN_5 = (P_5, T_5, F_5, W_5, (M_0)_5)$, where $P_5 = \{p_{18}, p_{19}, p_{20}\}$

and $T_5 = \{t_{21}, t_{22}, t_{23}\}$ are non-empty, finite, and disjoint sets of places and transitions, correspondingly. $F_5 = \{(p_{18}, t_{22}), (t_{21}, p_{19}), (t_{22}, p_{19}), (p_{19}, t_{23}), (t_{23}, p_{20})\}$. $W_5(p_{18}, t_{22}) = W_5(t_{21}, p_{19}) = W_5(t_{22}, p_{19}) = W_5(p_{19}, t_{23}) = W_5(t_{23}, p_{20}) = 1$ and $(M_0)_5 = p_{18}$.

Once we formally describe the distinct net modules, the subsequent stage is to systematically build a unified model to portray the complete flow of the work of the system. The unified model ought to fulfill all algorithmic stipulations of the system. This stage is imperative because each net module might formalize a correct and stable conduct, but joining of last transitions or places of preceding net module might not be suitable with initial transitions or places of following modules. Consequently, to retain the consistent model, the prior and posterior conditions of every net are considered in the formation of the overall model, else, the absent transitions or places might bring the system into standstill state, or individual modules might stay disconnected.

To formulate the unified model, net modules are integrated in transition-place or place-transition connection method. Moreover, arcs from places to transitions and vice versa are presented as $F_6 = \{(t_5, p_5), (t_{12}, p_{18}), (t_{13}, p_{11}), (t_{18}, p_{14}), (p_{16}, t_{21})\}$. Then associated weights are specified as $W_6(t_5, p_5) = W_6(t_{13}, p_{11}) = 3$ and $W_6(t_{12}, p_{18}) = W_6(t_{18}, p_{14}) = W_6(p_{16}, t_{21}) = 1$. This demonstrates within unified model, a rudimentary weight of each link is set as 1 or 3 other than those states which model the procedure of the identification of abnormalities and their weights which are computed by executions of the relations g_1, g_2, g_3, g_4, and g_5. It is pertinent to observe that the above-specified five relations are primary fragments of the abnormality detection presented in Sect. 3.5.

The first relation, namely g_1, verifies the current level of battery, B_c^l, of the cluster member mote, msn_q. The msn_q mote may begin its work if B_c^l is equivalent or larger than the already defined level of threshold, B_t^l. If B_c^l of msn_q is lower than B_t^l, then msn_q is treated as dead or malfunctioning. This relation is specified below.

$$g_1 = f(B_c^l, B_t^l) = \begin{cases} 1, & B_c^l \geq B_t^l, \\ 0, & Otherwise. \end{cases} \tag{3.5}$$

The second relation, g_2, confirms a packet, which possesses the F_q values, is collected inside a specified timeslot, $T_{t_1}^l$, or not. This relations is described as

$$g_2 = f(T_i^{lb}, T_{ar}^{F_q}, T_i^{ub}) = \begin{cases} k, & T_i^{lb} \leq T_{ar}^{F_q} \leq T_i^{ub}, \\ 0, & Otherwise. \end{cases} \tag{3.6}$$

The notation $T_{ar}^{F_q}$ denotes an arrival time of packet having features readings. The symbolism T_i^{lb} and symbolism T_i^{ub} represent initial and final times of the allocated timeslot to collect F_q, correspondingly. The notation k denotes features of the usual behavior profile of msn_q, in this case $k = 3$. If a packet is obtained inside the designated timeslot, then k features are used for the identification of abnormalities.

Subsequently, the identification of abnormalities procedure is performed as per nature of features. A related relation among g_3 and g_4 is processed for the features with fixed and bounded value bounds, correspondingly. The features having fixed values, for instance, memory status of msn_q, are discrete in nature because they are based on discrete values, for example, 89% and 86%. On the contrary, the features with bounded values, for instance battery status and sensor reading, are modeled as continuous random variables and they are based on real values. These kind of parameters have lower and upper limit values to outline the usual behavior of msn_q.

$$g_3 = f(P_{rg}^{lb}, m_{rg}^{f_q}, P_{rg}^{ub}) = \begin{cases} 1, & P_{rg}^{lb} \leq m_{rg}^{f_q} \leq P_{rg}^{ub}, \\ 0, & Otherwise. \end{cases} \tag{3.7}$$

$$g_4 = f(m_{fx}^{f_q}, P_{fx}^{f_q}) = \begin{cases} 1, & m_{fx}^{f_q} = P_{fx}^{f_q}, \\ 0, & Otherwise. \end{cases} \tag{3.8}$$

The notation $m_{rg}^{f_q}$ represents a bounded-feature value which is obtained from msn_q. Correspondingly, P_{rg}^{lb} and P_{rg}^{ub} denote minimum and maximum limits, in order to specify the usual conduct of msn_q with regard to $m_{rg}^{f_q}$. The symbol $m_{fx}^{f_q}$ represents received fixed-feature value, and $P_{fx}^{f_q}$ denotes associated value of the normal profile. Lastly, the last relation, g_5, verifies the well-timed receipt of on-the-spot confirmation result.

$$g_5 = f(T_j^{lb}, T_{ar}^{WR}, T_j^{ub}) = \begin{cases} 1, & T_j^{lb} \leq T_{ar}^{WR} \leq T_j^{ub}, \\ 0, & Otherwise. \end{cases} \tag{3.9}$$

The symbolization T_{ar}^{WR} represents the arrival time of watermarked on-the-spot confirmation result, WR. The notations T_j^{lb} and T_j^{ub} show initial and end times of a timeslot which is designated to collect on-the-spot confirmation result, correspondingly.

The descriptions of transitions and places of the unified model are provided in Table 3.3 [9], and the unified model is drawn in Fig. 3.7. States are shown by small circles in Fig. 3.7 [9]. On the contrary, transitions are represented by tiny dark rectangles. A tiny dark circle, known as token, in p_1, represents the start state of the modeled behavior of the system. Transfer (firing) of single token from initial to succeeding place represents the variation in a state within the system. The model is stated below.

Unified model The unified model, PN, is a 5-tuple net: PN = $(\mathcal{P}, \mathcal{T}, \mathcal{F}, \mathcal{W}, \mathcal{M}_0)$, where $\mathcal{P} = \bigcup_{i=1}^{5} P_i$, $\mathcal{T} = \bigcup_{i=1}^{5} T_i$, $\mathcal{F} = \bigcup_{i=1}^{6} F_i$, $\mathcal{W} = 1$ $\forall$ arcs except $\mathcal{W}(p_3, t_5)$ = $\mathcal{W}(t_5, p_4)$ = $\mathcal{W}(t_5, p_5)$ = $\mathcal{W}(t_6, p_6)$ = $\mathcal{W}(p_8, t_{11})$ = $\mathcal{W}(t_{11}, p_{10})$ = $\mathcal{W}(p_{10}, t_{13})$ = $\mathcal{W}(t_{13}, p_{11})$ = $\mathcal{W}(p_{11}, t_{14})$ = $\mathcal{W}(t_{14}, p_{12})$ = 3, $\mathcal{W}(p_1, t_{19}) = g_1$, $\mathcal{W}(p_5, t_6) = g_2$, $\mathcal{W}(p_6, t_7) = g_3$, $\mathcal{W}(p_6, t_8) = g_3$, $\mathcal{W}(p_6, t_9) = g_4$, $\mathcal{W}(p_{14}, t_{19}) = g_5$, and $\mathcal{M}_0 = p1$.

p_1 grasps a token when the working of the system is initiated as a first work cycle. Tokens, in the net module 1, remain equivalent to the quantity of parameters in normal profile. The tokens are three in number. Then the movement of these tokens will

Table 3.3 Descriptions of places and transitions: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. Specification and validation of enhanced mobile agent-enabled anomaly detection in resource constrained networks. Journal of Networks, 10(6):353–368, 2015

Place	Description	Transition	Description
p_1	msn_q wakes up	t_1	msn_q checks the battery level
p_2	msn_q is prepared to accumulate the values of features	t_2	The msn_q accumulates the SR values
p_3	The collected values of features are stored	t_3	msn_q accumulated the MS value
p_4	msn_q goes into the sleep mode	t_4	msn_q collects the value of BY
p_5	cln_q is ready to receive F_q	t_5	msn_q transmits the accumulated F_q values
p_6	cln_q is prepared to carry out the abnormality identification	t_6	cln_q waits for T_i^{ub} to receive F_q
p_7	cln_q is ready to aggregate the data	t_7	cln_q checks the SR value for (SR^{lb}, SR^{ub})
p_8	cln_q is prepared in decrementing a count of trust	t_8	cln_q checks MS value for (MS^{lb}, MS^{ub})
p_9	cln_q is prepared to communicate aggregated data	t_9	cln_q checks BY value for (BY^{lb}, BY^{ub})
p_{10}	cln_q is ready to transmit AA	t_{10}	cln_q aggregates the sensed data
p_{11}	msn_q is ready to receive the AA	t_{11}	cln_q is prepared in decrementing a count of trust
p_{12}	AA is prepared to compare data for on-the-spot confirmation	t_{12}	cln_q transmits the aggregated data
p_{13}	AA is prepared to communicate on-the-spot confirmation result after the insertion of watermark	t_{13}	cln_q transmits AA to msn_q
p_{14}	cln_q is prepared to get on-the-spot confirmation result	t_{14}	msn_q receives AA
p_{15}	cln_q is prepared to check on-the-spot confirmation result	t_{15}	AA compares SR with already saved values
p_{16}	An alarm is transmitted to BS by cln_q	t_{16}	AA compares the values of MS with already saved values
p_{17}	Status result saved by cln_q	t_{17}	AA compares BY with previously stored values
p_{18}	BS is prepared to receive the accumulated data	t_{18}	AA transmits the watermarked on-the-spot confirmation result to cln_q
p_{19}	BS is ready to analyze the data	t_{19}	cln_q waits for T_j^{ub} time to obtain on-the-spot confirmation result
p_{20}	BS is prepared to obtain the application data	t_{20}	cln_q verifies the result
		t_{21}	BS gets an alarm
		t_{22}	BS gets aggregated data
		t_{23}	BS updates repository

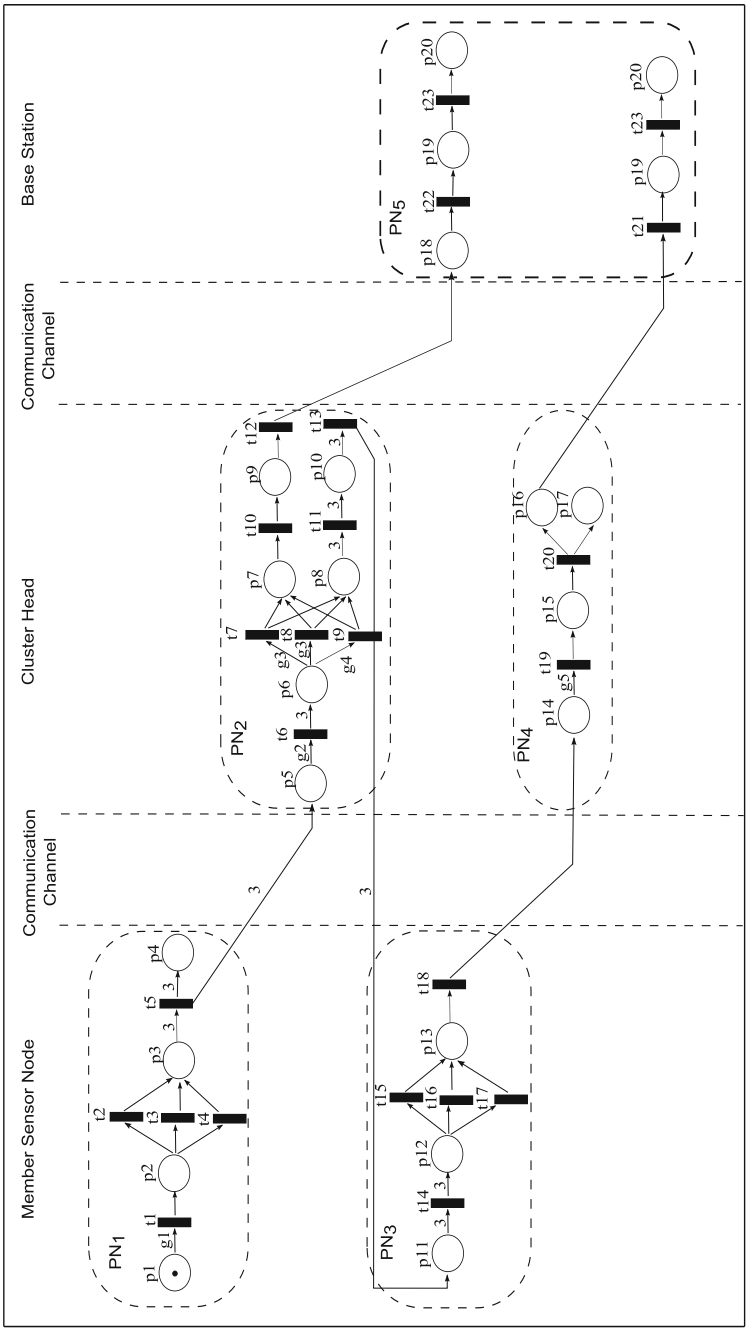

Fig. 3.7 The unified Petri net model: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. Specification and validation of enhanced mobile agent-enabled anomaly detection in resource constrained networks. Journal of Networks, 10(6):353–368, 2015

represent the present states of the behavior of the system. The subsequent functional description articulates the procedure of abnormality identification that is carried out by cln_q. The cln_q mote obtains the F_q values from msn_q and carries out the abnormality identification procedure. Then three tokens in p_5 are accumulated in the net module 2. These tokens are consequential of the firing of t_5, representing the communication of the aggregated values of F_q. These tokens are then amalgamated at p_7, representing accumulation of sensor readings at cln_q. On the contrary, three tokens remain at p_8 and afterward, each token denoting a single feature value. Subsequently, msn_q gets an agent that carries out on-the-spot confirmation procedure.

Firing of t_{12}, denoting the communication of accumulated data, transfers a token into p_{11}, that is the initial place of the net module 5 (i.e., the procedure of the update of status on BS). On the contrary, the firing of t_{13}, representing the communication of agent, produces three tokens into p_{11}. These three tokens, each of them representing single feature values, are amalgamated at p_{13}, denoting the prepared to communicate an agent to cln_q state. The firing of t_{18} produces a token in p_{14}, denoting ready to receive MA from msn_q state.

A token is received by p_{18}, as a consequence of the firing of t_{12}, representing the communication of accumulated data to BS. The aggregated data are then prepared to be saved and examined by the user. Otherwise, the firing of t_{21} produces a token in p_{19} to save and examine an obtained alarm by the user.

The behavioral properties, namely boundedness and liveness, of the proposed system are characterized and verified below. The boundedness expresses the maximum possible number of tokens within the system, i.e., the maximum possible processes can be possessed by the system at a state. Alternatively, the property of liveness represents the system has no deadlock. The flow of work and reachability of the abnormal states in the unified model are also studied below. Figure 3.8 [9], illustrating the reachability tree, formalizes the overall flow of the work of the system.

Theorem 3.3 *The PN, unified model, is 3-bounded.*

Proof For all $p \in \mathcal{P}$ and $\mathcal{M}(p) = 1$ apart from $\mathcal{P}_e = \{p_4, p_5, p_6, p_8, p_{10}, p_{11}, p_{12}\}$, which has three tokens. The corresponding transitions are $\mathcal{T}_e = \{t_5, t_6, t_7, t_8, t_9, t_{11}, t_{13}, t_{14}\}$, $\mathcal{P}_e \subset \mathcal{P}$, and $\mathcal{T}_e \subset \mathcal{T}$. This permits the autonomous handling of each feature for the procedure of abnormality identification. There are three features that move over explicit places of the system; therefore, PN is 3-bounded.

Theorem 3.4 *Every transition in PN is live at level 4 other than T_l transitions that are provisionally live at level 0.*

Proof For all $t \in \mathcal{T}$, the liveness level is 4 apart from $\mathcal{T}_l$, that is live at 0 level iff the output of conditions $g_1, g_3, g_4, g_5 \neq 1$ and $g_2 \neq k$, where $\mathcal{T}_l \subset \mathcal{T}$ and $\mathcal{T}_l = \{t_1, t_6, t_7, t_8, t_9, t_{19}\}$. This property holds given that the condition-based weighted arcs exist which input those transitions that formalize the procedure of abnormality identification. The arc conditions that possess certain weights and associated transitions are (g_1, t_1), (g_2, t_6), (g_3, t_7), (g_2, t_6), (g_3, t_8), (g_4, t_9), and (g_5, t_{19}).

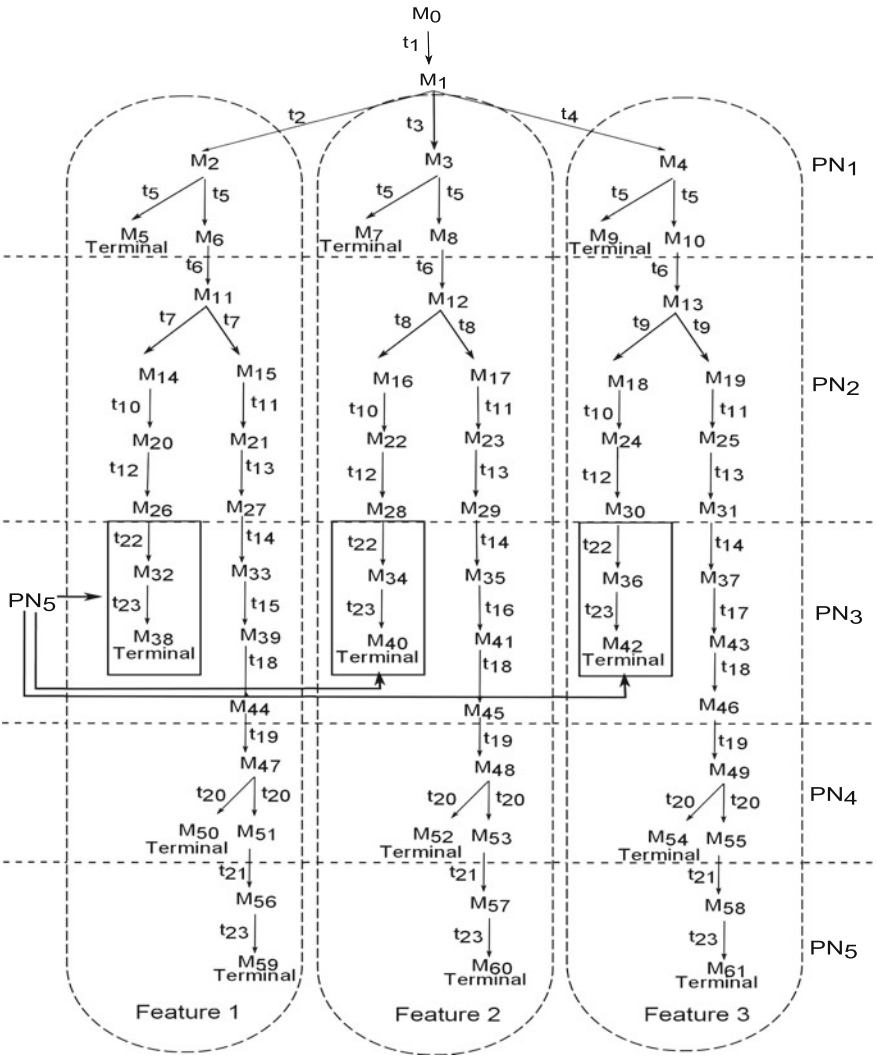

Fig. 3.8 The reachability tree: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. Specification and validation of enhanced mobile agent-enabled anomaly detection in resource constrained networks. Journal of Networks, 10(6):353–368, 2015

Theorem 3.5 *A number of net modules,* $\bigcup_{i=1}^{4} PN_i$*, are executable in a sequence, where* PN_5 *is sequentially executed after* PN_2 *and* PN_4 *to formulate the overall flow of the work of the system.*

Proof PN_1 performs the procedure of the collection of the F_q values by reachable markings $M_0[t_1\rangle\ M_1[t_2\rangle\ M_2[t_5\rangle\ \widehat{M_5}$, $M_0[t_1\rangle\ M_1[t_2\rangle\ M_2[t_5\rangle\ \widetilde{M_6}$, $M_0[t_1\rangle\ M_1[t_3\rangle\ M_3[t_5\rangle\ \widehat{M_7}$, $M_0[t_1\rangle\ M_1[t_3\rangle\ M_3[t_5\rangle\ \widehat{M_8}$, $M_0[t_1\rangle\ M_1[t_4\rangle\ M_4[t_5\rangle\ \widehat{M_9}$, and $M_0[t_1\rangle\ M_1[t_4\rangle\ M_4[t_5\rangle\ \widetilde{M_{10}}$, as shown in Fig. 3.8, where $\widehat{M_{(\cdot)}}$ and $\widetilde{M_{(\cdot)}}$ represent the final and all other states,

correspondingly. The final states denote the sleep status of msn_q, subsequent to the communication of F_q next to their accumulation. Subsequently, only the instance of sensor reading is discussed, which is easy to encompass to other features. The procedure of the abnormality identification is carried out after the procedure of the collection of F_q values by the subsequent reachable marking sequences: $M_6[t_6\rangle$ $M_{11}[t_7\rangle$ $M_{14}[t_{10}\rangle$ $M_{20}[t_{12}\rangle$ $\widetilde{M_{26}}$ and $M_6[t_6\rangle$ $M_{11}[t_7\rangle$ $M_{15}[t_{11}\rangle$ $M_{21}[t_{13}\rangle$ $\widetilde{M_{27}}$, here earlier marking represents the communicated of accumulated data, d_i^{ag}, whereas the final marking denotes the communication of agent to msn_q. Correspondingly, on-the-spot confirmation procedure is performed by $M_{27}[t_{14}\rangle$ $M_{33}[t_{15}\rangle$ $M_{39}[t_{18}\rangle$ $\widetilde{M_{44}}$ marking, that is followed by $M_{44}[t_{19}\rangle$ $M_{47}[t_{20}\rangle$ $\widetilde{M_{51}}$ and $M_{44}[t_{19}\rangle$ $M_{47}[t_{20}\rangle$ $\widetilde{M_{50}}$ markings for saving result of on-the-spot confirmation, R, on cln_q and communication of accumulated data, d_i^{ag}, or watermark inserted confirmation result, WR, to BS, correspondingly. The earlier module is completed, whereas the last module has subsequent $M_{51}[t_{21}\rangle$ $M_{56}[t_{23}\rangle$ $\widehat{M_{59}}$ marking. This implies that the $\bigcup_{i=1}^{4} PN_i$ modules execute in a sequence. A uniquely reachable $M_{26}[t_{22}\rangle$ $M_{32}[t_{23}\rangle$ $\widehat{M_{38}}$ marking is joined after the $M_6[t_6\rangle$ $M_{11}[t_7\rangle$ $M_{14}[t_{10}\rangle$ $M_{20}[t_{12}\rangle$ $\widetilde{M_{26}}$ marking, which is a accessible marking of PN_2. Therefore, PN_5 executes sequentially to PN_4 and PN_2, as shown in Fig. 3.8, denoting a reachability tree.

Theorem 3.6 *The procedure of abnormality identification for each feature can be executed in parallel to each other.*

Proof The proof for this theorem is a direct consequence for the proof of Theorem 1. Let the situation when a model is 3-bounded. In such a situation, a token in p_2, denoting the M_1 state, empowers $\mathcal{T}_q = \{t_1, t_2, ..., t_v\}$ transition set for associated feature set $F_q = \{fs_1, fs_1, ..., fs_v\}$. This confirms an independent and parallel execution of the procedure of abnormality identification for every feature. This is obvious through a pattern that the $M_1[t_2\rangle$ $M_2[t_5\rangle$ $M_6[t_6\rangle$ $M_{11}[t_7\rangle$ $M_{15}[t_{11}\rangle$ $M_{21}[t_{13}\rangle$ $M_{27}[t_{14}\rangle$ $M_{33}[t_{15}\rangle$ $M_{39}[t_{18}\rangle$ $M_{44}[t_{19}\rangle$ $M_{47}[t_{20}\rangle$ $M_{51}[t_{21}\rangle$ $M_{56}[t_{23}\rangle$ $\widehat{M_{59}}$, $M_1[t_3\rangle$ $M_3[t_5\rangle$ $M_8[t_6\rangle$ $M_{12}[t_8\rangle$ $M_{17}[t_{11}\rangle$ $M_{23}[t_{13}\rangle$ $M_{29}[t_{14}\rangle$ $M_{35}[t_{16}\rangle$ $M_{41}[t_{18}\rangle$ $M_{45}[t_{19}\rangle$ $M_{48}[t_{20}\rangle$ $M_{53}[t_{21}\rangle$ $M_{57}[t_{23}\rangle$ $\widehat{M_{60}}$, and $M_1[t_4\rangle$ $M_4[t_5\rangle$ $M_{10}[t_6\rangle$ $M_{13}[t_9\rangle$ $M_{19}[t_{11}\rangle$ $M_{25}[t_{13}\rangle$ $M_{31}[t_{14}\rangle$ $M_{37}[t_{16}\rangle$ $M_{43}[t_{18}\rangle$ $M_{46}[t_{19}\rangle$ $M_{49}[t_{20}\rangle$ $M_{55}[t_{21}\rangle$ $M_{55}[t_{23}\rangle$ $\widehat{M_{61}}$ markings are accessible to represent the sovereign management of features, viz. battery status, memory status, and sensor reading, as illustrated in Fig. 3.8.

The accessibility of the states: $M_j^{'} \in RM(M_0)$, which denotes the procedure of the abnormality identification, is verified subsequently.

Theorem 3.7 *The abnormality identification states, that is, $M_j[t_i\rangle M_j^{'}$, are only reachable when the system is abnormality free, where $M_j = M_0$, $M_j^{'} = (M_1)$, (M_{11}, M_{12}, M_{13}), (M_{14}, M_{15}), (M_{16}, M_{17}), (M_{18}, M_{19}), (M_{47}, M_{48}, M_{49}), and $t_i = t_1, t_6, t_7, t_8, t_9, t_{19}$ for associated accessible markings.*

Proof M_1, a marking, is accessible iff $\mathcal{W}(p_1, t_1) = 1$ in the unified model. If implementation of the relation g_1 produces a number except 1, then msn_q is assumed as

abnormal because of either or faulty status. Therefore, M_1 is not accessible in such situations. Correspondingly, the M_{11}, M_{12}, and M_{13} markings are reachable through the marking sequences $M_0[t1\rangle\ M_1[t2\rangle\ M_2[t5\rangle\ M_6[t6\rangle\ M_{11}$, $M_0[t1\rangle\ M_1[t3\rangle\ M_3[t5\rangle\ M_8[t6\rangle\ M_{12}$, and $M_0[t1\rangle\ M_1[t4\rangle\ M_4[t5\rangle\ M_{10}[t6\rangle\ M_{13}$ markings, correspondingly, iff $\mathcal{W}(p_5, t_6) = k$, that is obtained through the relation g_2. Otherwise, if $\mathcal{W}(p_5, t_6) \neq k$, then these states are not accessible, representing the abnormal conduct of msn_q. Next, t_7, t_8, and t_9 transitions are fired to reach (M_{14}, M_{15}), (M_{16}, M_{17}), and (M_{18}, M_{19}) markings, correspondingly. The relation g_3 specifies the weights of arcs of the t_7 and t_8 markings for the (M_{14}, M_{15}) and (M_{16}, M_{17}) markings, correspondingly. By the same token, the relation g_4 specifies the weight of the arc of the t_9 transition for the (M_{18}, M_{19}) markings. The result, except 1 relation, makes states inaccessible, denoting the abnormal values of usual behavior profile parameters of msn_q. If the outcome of the $g_5 \neq$ relation execution is 1, then the states M_{47}, M_{48}, and M_{49} become inaccessible because of discrepancy among the tokens and weights of arcs in the inbound place of arcs, namely transition-place.

The proof supports the argument the abnormal states are inaccessible in the model and merely those states are accessible that show the usual behavior of msn_q. Therefore, the unified model is proficient in identifying different types of abnormalities that are occurred due to faulty feature values and also due to time-based abnormalities that caused because of the late arrivals of on-the-spot confirmation results and observations.

An imperative inference of the unified model is joining of small but significant descriptions in the specifications of algorithms of the primary design of the system. The enhanced procedure of features collection (i.e., Algorithm 3.1) sets the member mote into sleep mode when it finishes its allocated job. This modification, in the initial functional description, avoids the needless consumption of energy. An additional important modification is made in the procedure of abnormality identification (i.e., Algorithm 3.2), here cln_q waits for the assigned slot of time to get F_q. This aspect permits the system to be able to identify time-based abnormalities which occurred because of the non-arrival or late arrival of F_q at cln_q.

3.7 Unified GSPN Model

The fitting time-based conduct of an identification system is important to identify and confirm the origin of abnormalities in a timely fashion. There exist two important procedures which need the time-based conduct study: (i) the joint procedure of collection of feature values on msn_q and their receipt by cln_q; this is denoted as α procedure in subsequent discussion and (ii) the joint procedure of abnormality identification, agent dispatch, on-the-spot confirmation, and confirmation result receipt on cln_q are denoted as β procedure in subsequent discussion. The analysis of the time-based behavior of the system is carried out in non-deterministic interaction setting of sensor networks that is typically occurred due to channel faults, traffic

features, and environmental factors. The unified model is converted into associated unified GSPN model.

The strategy definitions of model are reliant on rates, transition categories, memory definitions, and server semantics. The transitions are assumed as immediate other than those that are part of the calculation of α and β procedures. The immediate transition weight is set as 1; on the other hand, the timed transition rate is adjustable as per the computation carried out by them. The timed transition rates are defined on the basis of MICAz mote resource capability [11].

The individual firing semantic is adopted for timed transitions, wherein a single token stays in the inbound place till the expiry of timer. The processing of the system is executed in a sequence. Thus, solitary server definitions are selected for every transition, other than (t_2, t_3, t_4), (t_7, t_8, t_9), and (t_{15}, t_{16}, t_{17}) transitions, because these transitions have k-server definitions because of parallel computation in the model. It is evident $k = 3$, i.e., there are three features for abnormality identification. Moreover, the policy of age memory is assumed for all transitions because of the involvement of the continuous operations. On the basis of the above-cited semantics, the formalization of the model is provided next.

Unified GSPN model: GSPN is an 8-tuple net: GSPN = ($\mathcal{P}$, $\mathcal{T}$, $\Pi(\cdot)$, $I^-(\cdot)$, $O^+(\cdot)$, $H(\cdot)$, $\mathcal{W}(\cdot)$, $\mathcal{M}_0$), where $\mathcal{P} = \bigcup_{i=1}^{5} P_i$, $\mathcal{T} = \bigcup_{i=1}^{5} T_i$, $I^-(\cdot) \cup O^+(\cdot) = \mathcal{F} = \bigcup_{i=1}^{6} F_i$. The flow of the work of the beneath Petri net executes sequentially, and there is no inhibition arc. Thus, $\Pi(\cdot) = \emptyset$ and $H(\cdot) = \emptyset$. The time transitions weight $\mathcal{W}(\cdot) = t_1 = t_2 = t_3 = t_4 = 0.25$ms, $t_5 = 6$ms, $t_7 = t_8 = t_9 = t_{11} = t_{15} = t_{16} = t_{17} = 1$ms, $t_{13} = t_{14} = 15$ms, $t_{18} = 1$ms. Lastly, the first marking $M_0 = p1$.

The time-based conduct of the procedures, namely α and β, is examined below through the above-cited model. The time consumed by α is calculated as

$$\alpha = \alpha_1^t + \alpha_2^t + \alpha_3^t \tag{3.10}$$

The symbol α_1^t represents the time consumed by the procedure of the accumulation of F_q values (denoted by the t_1 to t_4 transitions in the model). The α_2^t notation denotes the time consumed by F_q to reach to cln_q from msn_q (formalized by the t_5 transition). Lastly, the notation α_3^t represents the deferment of the receipt of F_q on cln_q because of environmental aspects, channel faults, and traffic features. The deferment happens on the transmission link that is formalized by the arc, viz. (t_5, p_5). The consumption of time by β can be computed from the subsequent formula.

$$\beta = \sum_{i=1}^{6} \beta_i^t \tag{3.11}$$

The symbol β_1^t represents the total time consumed by the procedure of the abnormality identification carried out by cln_q (denoted by the transitions, viz. t_7, t_8, t_9, and t_{11} in the unified GSPN model), the notation β_2^t denotes the communication time consumed by the agent for roaming from cln_q to msn_q (represented by the t_{13} transition), the symbol β_3^t denotes the time consumed by the agent to carry out the

job of on-the-spot confirmation on msn_q (represented by the t_{14} to t_{17} transitions), and the symbol β_4^t represents the time consumed by on-the-spot confirmation result receipt on cln_q (represented by the t_{18} transition). The symbols β_5^t and β_6^t represent the deferment aspects for the receipt of the agent on msn_q and on-the-spot confirmation result on cln_q, correspondingly. The β_5^t and β_6^t factors happen at links that are formalized by the (t_{13}, p_{11}) and (t_{18}, p_{14}) arcs, correspondingly.

The deferment aspects α_3^t, β_5^t, and β_6^t are treated within the [0,1] closed interval for the study; here 0 denotes none delay (i.e., denoting the lower-fault-susceptible transmission link) and 1 represents the 100% deferment (i.e., denoting the higher-fault-susceptible transmission link). Moreover, the deferment aspects possess exponential distributions, that is in line with needs of Petri net-based formal modeling of the probabilistic procedures [12]. The tiny motes, that is, msn_q and cln_q, are treated 10 m apart for theoretical analysis and conforming trials.

Mainly, a sensitivity study on α_3^t, β_5^t, and β_6^t delay aspects was performed. To complete this job, the α_3 procedure was randomized on the [0,1] closed interval. This produced the outcome that consumption of time by a procedure, viz. α in lower-fault-susceptible transmission link remained 6.99 ms. In contrast, a solitary α procedure took 8.00 ms time in the transmission link, which delays transmission of messages between msn_q cln_q up to 100%. The model outcomes of the time-based conduct of the procedure, viz. α are given in the higher part of Fig. 3.9 [9]. The associated statistics which contain standard deviation (σ), mean (μ), minimum (Min), and maximum (Max) values of the outcomes are shown in Table 3.4 [9].

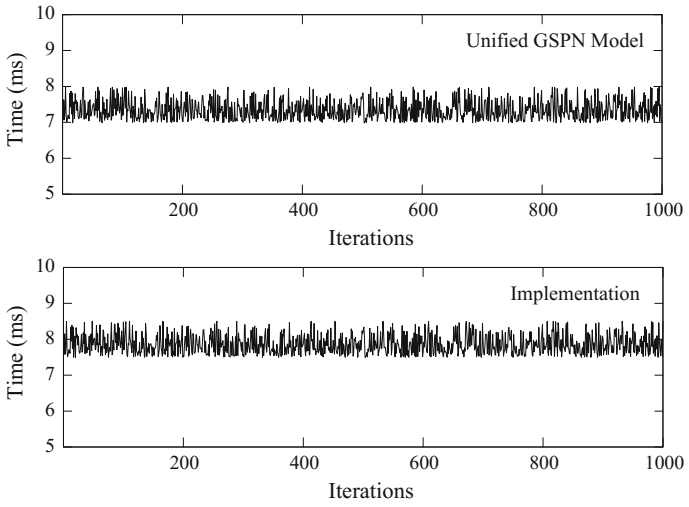

Fig. 3.9 The time-based conduct of the α procedure: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. Specification and validation of enhanced mobile agent-enabled anomaly detection in resource constrained networks. Journal of Networks, 10(6):353–368, 2015

Table 3.4 The α procedure statistics: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. Specification and validation of enhanced mobile agent-enabled anomaly detection in resource constrained networks. Journal of Networks, 10(6):353–368, 2015

Model	μ	σ	Min	Max
Unified GSPN model	7.35	0.273	6.99	8.00
Implementation	7.86	0.274	7.50	8.50

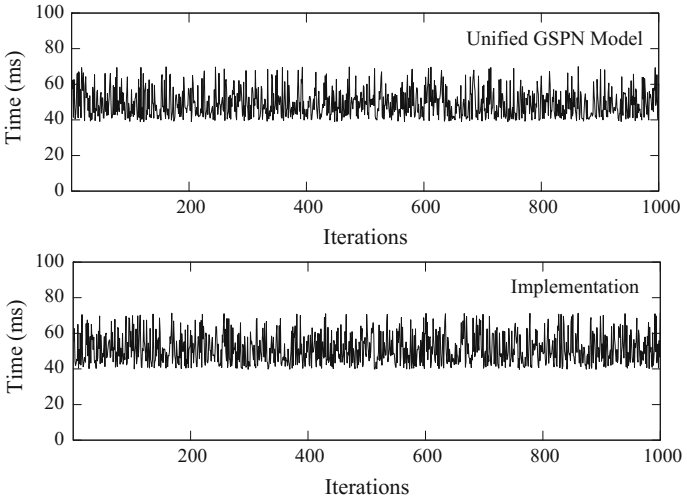

Fig. 3.10 The time-based conduct of the β procedure: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. Specification and validation of enhanced mobile agent-enabled anomaly detection in resource constrained networks. Journal of Networks, 10(6):353–368, 2015

Table 3.5 The β procedure statistics: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. Specification and validation of enhanced mobile agent-enabled anomaly detection in resource constrained networks. Journal of Networks, 10(6):353–368, 2015

Model	μ	σ	Min	Max
Unified GSPN model	49.30	7.88	38.96	69.92
Implementation	51.24	8.54	39.76	71.35

Subsequently, the procedures, viz. β_5^t and $\beta_6^{'}$, were randomized over the [0,1] closed interval to study the time-based conduct of the procedure, viz. β. In such a situation, the β procedure consumed 38.96 ms and 69.92 ms for the lower and higher-fault-susceptible transmission links, correspondingly. The model outcomes are given in the higher part of Fig. 3.10 [9], and the associated statistics are provided in Table 3.5 [9].

3.8 Time-Based Behavior Validation

The time-based conduct is corroborated by the implementation of algorithms on TinyOS, working on resource constrained MICAz motes, that are equipped with the Atmel ATmega128L microcontroller [11]. The moderately higher data transmitter radio of above-cited mote can transfer data at transfer rate of 250 kb/s. MICAz has 4 kb and 128 KB of EEPROM and program flash memory, correspondingly. The mote may be furnished with a supplementary 512-kb serial flash memory that is capable of holding over 100,000 measurements.

The topology of the network is based on two accessible MICAz motes that were positioned to carry out experiments. This topology of network is adequate for the corroboration of the time-based conduct of the procedures, viz. α and β because of the reason that these procedures are carried out by communication and processing between cln_q and msn_q. Thus, one accessible mote was positioned as cln_q; on the other hand, the second was positioned as msn_q. It is imperative to indicate that the outcomes acquired from the topology of network may be generalized by assuming the transmission link between cln_q and msn_q as a unit constituent of the over the network interaction.

The agent was coded, having 762 bytes of size including data and code. The 802.15.4 and Zigbee amenable MICAz motes can only sent 127 bytes (i.e., payload has 102 and header has 25 bytes) in one data packet [8]. Thus, the agent was divided into eight data packets on the mote that transmits the agent and congregated as an agent on the mote that receives the agent to perform its allocated job. The size of header was fixed as 25 bytes for all packets, where the size of the payload for initial seven packets was set as 102 bytes and for the final data packet was set as 48 bytes. Correspondingly, the each data packet size that had the F_q values was fixed as 31 bytes (i.e., payload has 6 and header has 25 bytes). The on-the-spot confirmation result data packet size was fixed as 27 bytes (i.e., payload has 2 and header has 27 bytes). Moreover, the deferment aspect was introduced in the experiments through exponential distribution to imitate the delay that has non-deterministic occurrence in the networks.

Five scenarios were studied to comprehensively examine the time-based conduct of the system. The initial two scenarios were developed for the confirmation of the time-based conduct outcomes of procedures, viz. α and β acquired by formalization performed in Sect. 3.7. The third situation was developed to examine the overhead effect on the time-based conduct of the system that was occurred due to securing the agent by inserting watermark and also by enabling the agent to insert watermark in on-the-spot confirmation outcomes. The final two situations were developed for supplementary analysis of the time-based conduct of the system.

Situation 1: The development of the procedure, viz. α was carried out to confirm the outcomes received by the procedure, viz. α formalized in Sect. 3.7. The outcomes discovered that the time consumed by the procedure, viz. α as compared to the unified GSPN model was somewhat higher, that is, among 7.50 ms and 8.50 ms for lower fault-prone and higher fault-prone transmission links, correspondingly. These out-

comes are given in the lower part of Fig. 3.9 [3] and the associated numbers are provided in Table 3.4 [3]. It is evident from Table 3.4 the initial and final limit variances among formal model and development outcomes for the procedure, viz. α are merely 0.51 and 0.50 ms, correspondingly. This difference is insignificant, and it may have caused because of the ecological effects, namely humidity, temperature, etc.

Situation 2: The development of the procedure, viz. β was performed to confirm the outcomes attained through the procedure, viz. β formalized in Sect. 3.7. In this situation, the on-the-spot confirmation outcomes were communicated to cln_q without inserting watermark. In reality, this situation is valid once the system is configured for the identification and confirmation of the abnormalities occurred due to only on-the-spot on during the transmission faults or errors. The time consumed by the procedure, viz. β was among 39.76 and 71.35 ms in the experiments, as compared to the formal model, it was among 38.96 and 69.92 ms, as depicted in Fig. 3.10. The associated numbers are shown in Table 3.5 [9]. Time taken by a procedure, viz. β is minimal to carry out the jobs of the abnormality identification, on-the-spot confirmation, and on-the-spot confirmation result receipt on cln_q, that is 10 m apart from msn_q.

Situation 3: Then, in the procedure, viz. β, the agent was developed to insert watermarks in on-the-spot confirmation outcomes that were sent to cln_q by the agent of msn_q. In this situation, the Radix-k encoding is employed to insert the watermark in the data and code of the agent [6] to safeguard agent from on-the-spot or in transmission attacks. This situation, in reality, is valid when system is proficient in the identification and confirmation of the abnormalities also occurred due to on-the-spot and in transmission attacks. In this situation, the agent size was increased up to 977 bytes that was initially 762 bytes because of its supplementary ability of inserting watermark in on-the-spot confirmation outcome. The overhead of around 25% was occurred due to insertion of a watermark in the agent. Therefore, the overall size of the agent was 1220 bytes. The agent was divided into 12 data packets for transmission by following the standard, namely Zigbee/802.15.4. The initial 11 data packets had the size of 127 bytes. However, the last packet size remained 123 bytes. In this situation, the on-the-spot confirmation outcome size was 127 bytes.

The trials outcome, in such situations, show the time consumption for situation where the abnormality agent was watermark secured and capable of inserting the watermark in on-the-spot confirmation outcome was 61.71 ms and 113.83 for lower and higher-error-susceptible communication links. In contrast, in the case of the usual abnormality agent, the time taken was 39.76 and 71.35 ms for lower and higher-error-susceptible communication links, correspondingly. This indicates the burden of time consumption was 62.68 and 64.37% in contrast to β procedure with usual abnormality agent (i.e., Case 2). The outcomes are given in Fig. 3.11 [9], and associated numbers are provided in Table 3.6 [9].

Situation 4: In this situation, the distance among msn_q and cln_q was randomized from 3 to 15 m by keeping the other configurations intact. The distance was randomized to examine the consequence of the distance on on-the-spot confirmation procedure that is carried out by the agent. The outcomes illustrate the time taken remained among 61.20 and 61.94 ms if the distance varied among 3 and 15 m,

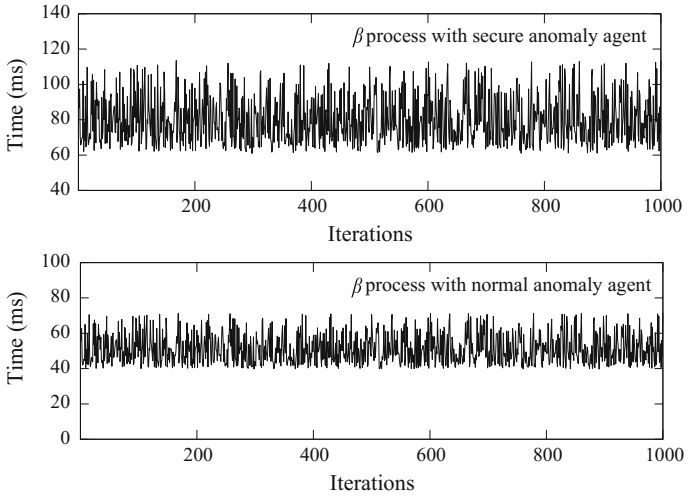

Fig. 3.11 The β procedure with the normal and secure abnormality agent: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. Specification and validation of enhanced mobile agent-enabled anomaly detection in resource constrained networks. Journal of Networks, 10(6):353–368, 2015

Table 3.6 The β procedure statistics: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. Specification and validation of enhanced mobile agent-enabled anomaly detection in resource constrained networks. Journal of Networks, 10(6):353–368, 2015

Model	μ	σ	Min	Max
β procedure with secure abnormality agent	80.81	13.69	61.71	113.83
β procedure with normal abnormality agent	51.24	8.54	39.76	71.35

correspondingly, for the lower-fault-susceptible transmission link. Correspondingly, consumption of time remained among 112.50 and 115.04 ms for the higher-fault-susceptible transmission link if the distance varied among 3 and 15 m, correspondingly. These outcomes show that an surge in the value of distance among msn_q and cln_q has minor effect on the time-based conduct of the procedure, viz. β. These outcomes are shown in Fig. 3.12 and the associated statistics are given in Table 3.7 [9]

Situation 5: The procedure, viz. β was developed, and its time-based conduct was examined for both continuously and periodic data communicating applications. The conduct of a fire-tracking network was studied for the former case, where msn_q transmits sensed data continuously to associated cln_q after the identification of an unusual event [13]. Such an application, the memory, viz. serial flash, was employed to save F_q values that were used by the agent to carry out the job of on-the-spot confirmation on msn_q. On the contrary, in the latter applications, conduct of a built

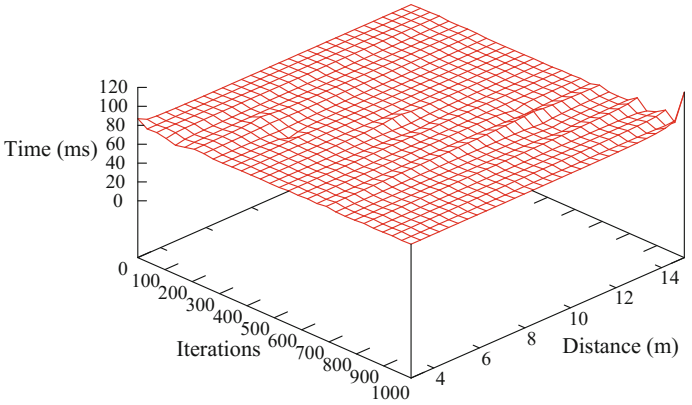

Fig. 3.12 The β procedure with a distance factor: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. Specification and validation of enhanced mobile agent-enabled anomaly detection in resource constrained networks. Journal of Networks, 10(6):353–368, 2015

Table 3.7 The statistics of the β procedure with the distance factor: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. Specification and validation of enhanced mobile agent-enabled anomaly detection in resource constrained networks. Journal of Networks, 10(6):353–368, 2015

Distance (m)	μ	σ	Min	Max
3	80.10	13.34	61.20	112.50
6	80.42	13.47	61.24	112.75
9	80.73	13.61	61.66	113.71
12	81.04	13.81	61.90	113.95
15	81.66	14.30	61.94	115.04

infrastructure observing network, wherein msn_q intermittently communicates data to the associated cln_q [14], was examined.

It was found, in the experiments, that the continuously transmitting data application unavoidably saves additional observations prior to the receipt of the agent to carry out the job of the on-the-spot confirmation on msn_q. Thus, the agent was supposed to devote more time to carry out the job of on-the-spot confirmation for this type of applications. This surges total time of procedure, viz. β. The experiment outcomes show the time consumed by the procedure, viz. β was among 73.29 and 104.25 ms for continuous application in contrast to 39.76 and 71.35 ms for periodic application. Moreover, it was noted msn_q saved 10 additional values of observations for the job of on-the-spot confirmation prior to the receipt of the agent.

This has not only enhanced the consumption of time by the job of on-the-spot confirmation from 3.79 to 37.9 ms, but also took additional memory of 60 bytes to save the values of 10 observations. This outcome infers that on-the-spot confirmation procedure is more suitable to periodically data transmitting applications because of

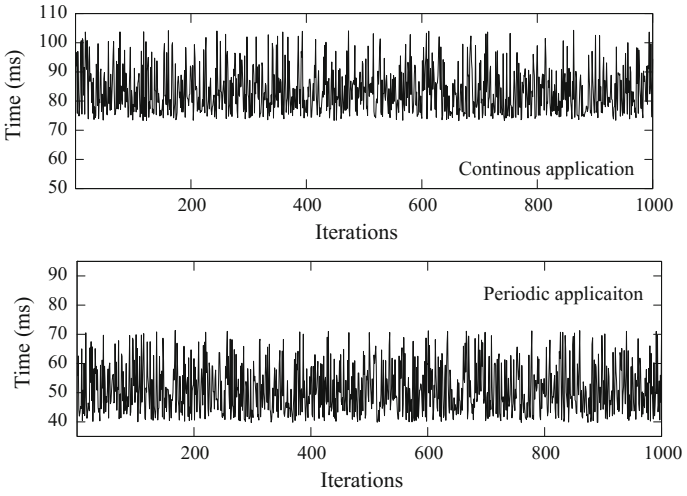

Fig. 3.13 Continuous vs periodic application: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. Specification and validation of enhanced mobile agent-enabled anomaly detection in resource constrained networks. Journal of Networks, 10(6):353–368, 2015

Table 3.8 Statistics of applications: © Academy Publisher, reprinted from M. Usman, V. Muthukkumarsamy, and X.-W. Wu. Specification and validation of enhanced mobile agent-enabled anomaly detection in resource constrained networks. Journal of Networks, 10(6):353–368, 2015

Model	μ	σ	Min	Max
Continuous application	83.63	7.88	73.29	104.25
Periodic application	51.24	8.54	39.76	71.35

less consumption of memory and time. The time-based conduct outcomes are given in Fig. 3.13 [9], and the associated statistics summary is shown in Table 3.8 [9]. The further results of the usage of memory are listed in Sects. 4.4.2.1 and 5.7 in the following chapter.

3.9 Discussion

The main conclusions of the theoretical investigation and experiment outcomes are listed below.

- The structure of the proposed system is comprehensive and correct.
- The proposed system has the ability to identify the time-based abnormalities along with performing on-the-spot confirmation and identification of abnormalities in the data that are received from member motes.
- The proposed system has adequate time-based conduct in a highly non-deterministic transmission environment.

- The proposed system consumes 64.37% more time to carry out the job of on-the-spot confirmation when it is configured for identification and confirmation of abnormalities that are occurred due to attacks, dissimilar to situation when the network is configured only for abnormalities that are occurred due to errors or faults.
- The randomization in near distance has a minor effect on time-based conduct of system. This infers the system is appropriate for smart home, built infrastructure monitoring, and other such applications, where sensor motes are typically positioned in a comparatively adjacent proximity.
- The system is highly suitable for periodic data sending applications in comparison to continuous data sending applications.

3.10 Summary

This chapter has presented a detailed agent-enabled abnormality identification and confirmation system in order to address the research questions 1 and 2 and also to satisfy the corresponding requirements 1 and 2. The architecture of the abnormality identification and confirmation module has been presented in detail. The algorithmic specifications of the proposed system for the network entities such as cluster member, cluster leader, and base station motes have also been elucidated. The space and time complexity analyses were performed to analyze the performance of the proposed algorithms. A unified formal model of the system was formulated to characterize and study its properties. The unified formal model was then extended into a unified GSPN model to characterize the time-based conduct of the system. The time-based conduct of the system was then confirmed through implementation on the real test bed in a number of scenarios. The theoretical analyses and experiment outcomes have advocated the capability of the system to detect behavioral abnormalities occurred due to faulty values of features and time-based conduct abnormalities occurred due to the deferred arrivals of observations and on-the-spot confirmation outcomes at the cluster leader motes. The results have also demonstrated the aptness of the time-based conduct of the system in a communication environment that is non-deterministic. The experimental results also endorsed the fact the system is more adequate for periodic data sending applications like smart home sensor networks in comparison with the continuously data transmitting applications such as the fire-tracking applications.

3.11 Bibliographic Notes

The idea of the resource-efficient abnormality identification and confirmation system was first introduced in [15, 16]. It was then discussed in detail and thoroughly analyzed in [3]. The formal modeling, verification, and analyses have been carried out in [9].

Appendix

Quantified Operations

This appendix quantifies the individual operations involved in α and β processes discussed in Sect. 3.7, Chap. 3. First, Tables 3.9 and 3.10 are presented, which quantifies the operations of α process, reported in Fig. 3.9. Then, Tables 3.11 and 3.12 are presented, which quantifies the operations of β process reported in Fig. 3.10. Tables 3.9, 3.10, 3.11, and 3.12 are given on the following pages.

Table 3.9 Quantification of the operations of the alpha process (Unified GSPN Model) results reported in Fig. 3.9

Iterations	α_1^t	α_2^t	α_3^t
1	1.5	5.5	0.98
2	1.5	5.5	0.95
3	1.5	5.5	1.00
4	1.5	5.5	0.00
5	1.5	5.5	0.52
6	1.5	5.5	0.33
1	1.5	5.5	0.71
8	1.5	5.5	0.15
9	1.5	5.5	1.00
10	1.5	5.5	0.75
11	1.5	5.5	0.93
12	1.5	5.5	0.28
13	1.5	5.5	0.44
14	1.5	5.5	0.65
15	1.5	5.5	0.59
16	1.5	5.5	0.24
17	1.5	5.5	0.66
18	1.5	5.5	0.74
19	1.5	5.5	0.14
20	1.5	5.5	0.00

Table 3.10 Quantification of the operations of the alpha process (Implementation) results reported in Fig. 3.9

Iterations	α_1^t	α_2^t	α_3^t
1	1.75	5.75	0.54
2	1.75	5.75	0.57
3	1.75	5.75	1.95
4	1.75	5.75	0.00
5	1.75	5.75	0.52
6	1.75	5.75	1.00
1	1.75	5.75	0.71
8	1.75	5.75	0.54
9	1.75	5.75	1.00
10	1.75	5.75	0.00
11	1.75	5.75	0.73
12	1.75	5.75	0.88
13	1.75	5.75	0.44
14	1.75	5.75	0.65
15	1.75	5.75	0.57
16	1.75	5.75	0.74
17	1.75	5.75	0.65
18	1.75	5.75	0.65
19	1.75	5.75	0.14
20	1.75	5.75	0.85

Table 3.11 Quantification of the operations of the β process (Unified GSPN Model) results reported in Fig. 3.10

Iterations	β_1	β_2	β_3	β_4	β_5	$\beta_3 6$
1	4.2	30.35	3.5	0.81	18.50	0.32
2	4.2	30.35	3.5	0.81	23.20	0.32
3	4.2	30.35	3.5	0.81	1.77	0.32
4	4.2	30.35	3.5	0.81	23.67	0.32
5	4.2	30.35	3.5	0.81	12.01	0.32
6	4.2	30.35	3.5	0.81	0.00	0.32
7	4.2	30.35	3.5	0.81	4.18	0.32
8	4.2	30.35	3.5	0.81	9.72	0.32
9	4.2	30.35	3.5	0.81	26.71	0.32

(continued)

Table 3.11 (continued)

Iterations	β_1	β_2	β_3	β_4	β_5	β_36
10	4.2	30.35	3.5	0.81	27.28	0.32
11	4.2	30.35	3.5	0.81	2.22	0.32
12	4.2	30.35	3.5	0.81	27.74	0.32
13	4.2	30.35	3.5	0.81	26.68	0.32
14	4.2	30.35	3.5	0.81	8.26	0.32
15	4.2	30.35	3.5	0.81	17.87	0.32
16	4.2	30.35	3.5	0.81	1.99	0.32
1	4.2	30.35	3.5	0.81	30.35	0.32
18	4.2	30.35	3.5	0.81	23.81	0.32
19	4.2	30.35	3.5	0.81	17.53	0.32
20	4.2	30.35	3.5	0.81	18.50	0.32

Table 3.12 Quantification of the operations of the β process (Implementation) results reported in Fig. 3.10

Iterations	β_1	β_2	β_3	β_4	β_5	β_36
1	4.3	30.28	3.79	0.86	15.90	0.17
2	4.3	30.28	3.79	0.86	22.72	0.14
3	4.3	30.28	3.79	0.86	3.62	0.85
4	4.3	30.28	3.79	0.86	1.83	0.41
5	4.3	30.28	3.79	0.86	3.33	0.19
6	4.3	30.28	3.79	0.86	5.55	0.04
7	4.3	30.28	3.79	0.86	4.39	0.43
8	4.3	30.28	3.79	0.86	24.92	0.34
9	4.3	30.28	3.79	0.86	0.03	0.00
10	4.3	30.28	3.79	0.86	11.06	0.47
11	4.3	30.28	3.79	0.86	2.34	0.81
12	4.3	30.28	3.79	0.86	19.86	0.24
13	4.3	30.28	3.79	0.86	2.41.90	0.00
14	4.3	30.28	3.79	0.86	8.77	0.020
15	4.3	30.28	3.79	0.86	30.78	0.85
16	4.3	30.28	3.79	0.86	5.65	0.64
17	4.3	30.28	3.79	0.86	0.64	0.86
18	4.3	30.28	3.79	0.86	3.14	0.28
19	4.3	30.28	3.79	0.86	6.49	0.24
20	4.3	30.28	3.79	0.86	3.08	0.09

References

1. F. Ahmad, S.A. Khan, Specification and verification of safety properties along a crossing region in a railway network control. Appl. Math. Model. **37**(7), 5162–5170 (2013)
2. J.L. Peterson, *Petri Net Theory and the Modeling of Systems*, 1st edn. (Prentice Hall, Englewood Clis, N. J., USA, 1981)
3. M. Usman, V. Muthukkumarsamy, X.-W. Wu, A resource-efficient system for detection and verification of anomalies using mobile agents in wireless sensor networks. J. Netw. **9**(12), 3427–3444 (2014)
4. J.C. Dagher, M.W. Marcellin, M.A. Neifeld, A theory for maximizing the lifetime of sensor networks. IEEE Trans. Commun. **55**(2), 323–332 (2007)
5. J. Waterman, G.W. Challen, M. Welsh, Peloton: Coordinated resource management for sensor networks, (Switzerland, 2009), pp. 1–5
6. O. Esparza, J.L. Munoz, J. Tomas-Builart, M. Soriano, An infrastructure for detecting and punishing malicious hosts using mobile agent watermarking. Wirel. Commun. Mob. Comput. **11**(11), 1446–1462 (2011)
7. C. Muldoon, G.M.P. OHare, R. Collier, M.J. OGrady, Agent factory micro edition: a framework for ambient applications, in *Computational Science*, ed. by V.N. Alexandrov, G.D. van Albada, P.M.A. Sloot, J. Dongarra. Lecture Notes in Computer Science, vol. 3993 (Springer Berlin Heidelberg, 2006), pp. 727–734
8. D.E. Denning, An intrusion-detection model. IEEE Trans. Softw. Eng. **SE-13**(2), 222–232 (1987)
9. M. Usman, V. Muthukkumarsamy, X.-W. Wu, Specification and validation of enhanced mobile agent-enabled anomaly detection in resource constrained networks. J. Netw. **10**(6), 353–368 (2015)
10. Part 15.4: Wireless Medium Access (MAC) and, Physical Layer (PHY) specications for low-rate wireless Personal Area Network (LR-WPANs). IEEE Std. **802**(15), 4 (2006)
11. J. Polastre, R.Szewczyk, C. Sharp, D. Culler, The mote revolution: low power wireless sensor network devices, in *Proceeding of the Hot Chips* (2004)
12. A.M. Marsan, G. Conte, G. Balbo, A class of generalized stochastic petri nets for the performance evaluation of multiprocessor systems. ACM Trans. Comput. Syst. **2**(2), 93–122 (1984)
13. C. Lino, C. T. Calafate, A. Diaz-Ramirez, P. Manzoni, J.-C. Cano, Studying the feasibility of ieee 802.15.4 based wsns for gas and re tracking applications through simulation, in *Proceedings of the 36th IEEE Conference on Local Computer Networks (LCN)*, (2011), pp. 875–881
14. E.U. Gaura, J. Brusey, R. Wilins, J. Barnham, in *Wireless sensing for the built environment: enabling innovation towards greener, healthier homes* (United Kingdom, 2011), pp. 1–6
15. M. Usman, V. Muthukkumarsamy, X.-W. Wu, S. Khanum, Wireless smart home sensor networks: mobile agent based anomaly detection, in *Proceedings of the 9th IEEE International Conference on Ubiquitous Intelligence Computing and 9th International Conference on Autonomic Trusted Computing (UIC/ATC)*, Sept 2012, pp. 322–329
16. M. Usman, Agent-enabled anomaly detection in resource constrained wireless sensor networks, in *Proceedings of the 15th IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2014, pp. 1–2

Chapter 4
First-Order Abnormalities: Agent Transmission Optimization

4.1 Introduction

The system, presented in the preceding chapter, is not only able to identify different types of abnormalities, but also empowers agents to use the synchronized resource management scheme information to carry out on-the-spot diagnosis of the member motes to discover the origin of abnormalities. This chapter introduces a method which exploits the statistical association among different features of the synchronized resource management scheme-based observations to discover a number of first-order abnormalities that are occurred due to denial-of-sleep attacks, battery exhaustion attacks, and faulty motes. The constrained energy resources of networks demand careful communication of agents. Therefore, two novel methods, namely 2-sigma and weighted sum, for abnormality confirmation agent transmission optimization are presented in this chapter.

This structure of this chapter is described below. Section 4.2 presents algorithms of the proposed methods along with their complexity analysis. The formal modeling and analysis of the proposed methods is carried out in Sect. 4.3. The details of the simulation setup, corresponding results and analysis, implementation results, and comparative study are discussed in Sect. 4.4. Lastly, Sect. 4.5. recapitulates the contributions of this chapter.

4.2 Algorithms and Analysis

This section elucidates the algorithmic specifications of the first-order abnormality identification, 2-sigma, and weighted-sum optimization methods.

M. Usman et al., *Mobile Agent-Based Anomaly Detection and Verification System for Smart Home Sensor Networks*,
https://doi.org/10.1007/978-981-10-7467-7_4

4.2.1 First-Order Abnormality Identification by the Cluster Leader Mote

One of the key goals of this work is to maximize the use of received synchronized resource management scheme-based observation, O_j, values and computed information of $FS_2 = \{\jmath, f\}$ for the abnormality identification process, where $O_j = FS_1 = \{\lambda, \varphi, \nu\}$. To this end, the statistical associations between $FS = \{\lambda, \varphi, \nu, \jmath, f\}$ features have been exploited to identify certain nature of group abnormalities that are occurred by on-the-spot attacks or faults, resource exhaustion attacks, denial-of-sleep attacks, and faults on motes. The first-order join is specified as two-dimensional linkage between two features by setting bounds of every feature to calculate normal profile region. A first-order join among λ and $\jmath$ features has been established to detect abnormalities caused by on-the-spot faults or attacks. The sign of such abnormalities on cln_q is receiving of faulty sensor reading values outside the allocated timeslots. The combined usual region for $\jmath$ and λ features can be derived by (4.1)

$$N(\lambda, \jmath) = \int_{\lambda_{il}}^{\lambda_{fl}} \int_{\jmath_{il}}^{\jmath_{fl}} f(\lambda, \jmath)\, d\jmath\, d\lambda \tag{4.1}$$

where notation N represents the usual zone in relation to λ and $\jmath$ features, and subscripts il shows start and fl denotes finish boundary of the particular feature. The subsequent join is made up by combining features, namely $\jmath$ and ν, to discover abnormalities occurred due to the attack, viz. resource exhaustion. The sign of these abnormalities is an unforeseen increase in the consumption of resources in regard to the timeslot. The usual region of an attack, viz. resource exhaustion, is calculated by (4.2).

$$N(\jmath, \nu) = \int_{\jmath_{il}}^{\jmath_{fl}} \int_{\nu_{il}}^{\nu_{fl}} f(\jmath, \nu)\, d\nu\, d\jmath \tag{4.2}$$

The succeeding two joins are $(\varphi, \jmath)$ and (φ, ν) which discover abnormalities that caused because of faults or attacks on the resources of a mote. In such situations, the signs are unfamiliar consumption of resources during the execution of usual jobs and unapproved activities executed by the mote in regard to time. Usual boundaries for joins $(\varphi, \jmath)$ and (φ, ν) are computed by (4.3) and (4.4).

$$N(\varphi, \jmath) = \sum_{\varphi_{il}}^{\varphi_{fl}} \int_{\jmath_{il}}^{\jmath_{fl}} f(\varphi, \jmath)\, d\jmath \tag{4.3}$$

$$N(\varphi, \nu) = \sum_{\varphi_{il}}^{\varphi_{fl}} \int_{\nu_{il}}^{\nu_{fl}} f(\varphi, \nu)\, d\nu \tag{4.4}$$

Lastly, a join is formed between features, viz. f and $\jmath$, to discover the abnormalities occurred due to an attack, viz. denial-of-sleep and defective mote. The sign for

Table 4.1 Joins, associated abnormalities, and their details: © Academy Publisher, reprinted from Usman et al. [1]

Join	Abnormalities	Description
$N(\lambda, \jmath)$	On-the-spot attack or fault	Abnormal sensed data in regard to time
$N(\jmath, \nu)$	Resource exhaustion attack	The unexpected rise in the consumption of battery in regard to time
$N(\varphi, \jmath)$	Faulty mote	The unapproved actions performed by the tiny mote in regard to time
$N(\varphi, \nu)$	Fault on a mote, attack on the resources of a mote	The excessive battery consumption while carrying out usual jobs
$N(f, \jmath)$	Denial-of-sleep attack, faulty mote	The monotonous communication of packets in regard to time

these abnormalities at the leader mote of the cluster is the incorrect count of packet. The usual boundary for the join is derived by (4.5).

$$N(f, \jmath) = \sum_{f_{il}}^{f_{fl}} \int_{\jmath_{il}}^{\jmath_{fl}} f(f, \jmath)\, d\jmath \tag{4.5}$$

The above-mentioned joins, associated abnormalities, and their details are given in Table 4.1 [1].

To discover the abnormalities, cln_q obtains O_j from msn_q. The unit, namely coordination, then excerpts the $FS_1 = \{\lambda, \varphi, \nu\}$ values from obtained O_j to carry out the abnormality identification procedure by employing joins. If msn_q is observed as normal after receiving O_j, then sensed data is aggregated by (A_unt). On the contrary, if msn_q is observed as abnormal, then the abnormality identification algorithm initiates the optimization of the (abnormality) agent transmission process (i.e., Phase 2 of either Algorithms 4.2 or 4.3 as specified by the system administrator). Phase 2 of Algorithm 4.3 yields msn_q behavior (Beh) as abnormal (BA), tolerated category 1 (BT_η), tolerated category 2 (BT_γ), or tolerated category 3 (BT_ζ). For abnormal conduct, the abnormality identification and confirmation module sends the agent to msn_q to carry out on-the-spot confirmation. On the contrary, for the categories, viz. 1, 2, and 3, the abnormality identification and confirmation module broadcasts msn_q as abnormal to other member motes and leader motes, reduces the interaction with msn_q, and issues an alarm to BS, correspondingly. Phase 2 of Algorithm 4.2, on the contrary to Phase 2 of Algorithm 4.3, performs the abnormality agent transmission optimization process and takes adequate action by itself; that is, it does not return any value to Algorithm 4.1 and rest of the processing is performed by Phase 2 of Algorithm 4.2 by itself. The abnormality identification procedure pseudocode is provided in Algorithm 4.1 [1].

Algorithm 4.1 First-order abnormality identification by the cluster leader mote

Input: O_j

Output: AA, store d^{ag}, transmit d_i^{al}, announce msn_q as anomalous, minimize communication with msn_q

1: cln_q receives O_j from msn_q
2: CU extract $FS_1 = \{\lambda, \varphi, \nu\}$ from O_j
3: Compute $FS_2 = \{ \jmath, f \}$
4: CHK $(An_f^o) = \int_{\lambda_{il}}^{\lambda_{fl}} \int_{\jmath_{il}}^{\jmath_{fl}} f(\lambda, \jmath)\, d\jmath\, d\lambda$
$\wedge\, N(\jmath, \nu) = \int_{\jmath_{il}}^{\jmath_{fl}} \int_{\nu_{il}}^{\nu_{fl}} f(\jmath, \nu)\, d\nu\, d\jmath$
$\wedge\, N(\varphi, \nu) = \sum_{\varphi_{il}}^{\varphi_{fl}} \int_{\nu_{il}}^{\nu_{fl}} f(\varphi, \nu)\, d\nu$
$\wedge\, N(\varphi, \jmath) = \sum_{\varphi_{il}}^{\varphi_{fl}} \int_{\jmath_{il}}^{\jmath_{fl}} f(\varphi, \jmath)\, d\jmath$
$\wedge\, N(f, \jmath) = \sum_{f_{il}}^{f_{fl}} \int_{\jmath_{il}}^{\jmath_{fl}} f(f, \jmath)\, d\jmath$ //perform first-order abnormality identification
5: **if** CHK $(An_f^o) == TRUE$ **then**
6: $A\_unt \leftarrow SR$ //aggregate sensed data in aggregation unit
7: **else**
8: CALL $Beh = AAO(FS)$ //invoke Phase 2 of Algorithm 4.3
9: **if** CHK $(Beh = BA) == TRUE$ **then** //msn_q behavior found as anomalous
10: TRNSMT AA to msn_q //transmit abnormality agent to the msn_q
11: **else if** CHK $(Beh = BT_\eta) == TRUE$ **then** //msn_q behavior found as BT_η tolerated
12: $A\_unt \leftarrow SR \wedge CU$ announce the msn_q as anomalous to msns and other clns
13: **else if** CHK $(Beh = BT_\gamma) == TRUE$ **then** //msn_q behavior found as BT_γ tolerated
14: $A\_unt \leftarrow SR \wedge CU$ reduces the interaction with msn_q
15: **else if** CHK $(Beh = BT_\zeta) == TRUE$ **then** //msn_q behavior found as BT_ζ tolerated
16: $A\_unt \leftarrow SR \wedge CU$ transmits d_i^{al} to BS
17: **end if**
18: **end if**

4.2.2 2-Sigma Optimization by the Cluster Leader Mote

Abnormality agent should not be spontaneously communicated on the network because of the costly transmission operation. The curtailment of the agent transmission, however, should be carefully designed so that it should not disturb the performance of the identification and confirmation system. The 2-sigma technique uses two standard deviations in order to outline a curtailment region on probability distributions of parameters in feature set, FS. An observation which stays between first and second deviations (i.e., $1\sigma < FS \leq 2\sigma$ or $-1\sigma > FS \geq -2\sigma$) is treated as tolerated, and the one which stays outside the two deviations (i.e., $FS > 2\sigma$ or $FS < -2\sigma$) is considered as appropriate to send the abnormality agent to carry out the job of on-the-spot confirmation of msn_q. In the situation of the usual observation, the value of the SR is accumulated by the aggregation unit. On the contrary, the value of trust of msn_q is decremented in the case if the observation lies in the tolerance zone. The abnormality agent is triggered to carry out the job of on-the-spot confirmation of msn_q once the trust level of the msn_q reaches the lower bound, that is, 0. Setting up a tolerance zone results in the fewer communications of agents that decreases the consumption of energy by both the msn_q and cln_q motes; hence, the overall lifetime

Algorithm 4.2 2-sigma optimization

Input: FS
Output: $-1\sigma, -2\sigma, 1\sigma, 2\sigma$ for all $\mathbf{f}_i \in FS$
Phase 1: Compute zones (FS)
1: At t_k time
2: **for** each $\mathbf{f}_i \in FS$ **do**
3: Compute $-1\sigma, 1\sigma, -2\sigma, 2\sigma$ //compute values of threshold
4: Compute $(N^z) = -1\sigma \leq \mathbf{f}_i \leq 1\sigma$ //compute normal zone (N^z)
5: Compute $(T^z) = 1\sigma < \mathbf{f}i \leq 2\sigma \wedge -1\sigma > \mathbf{f}_i \geq -2\sigma$ //compute tolerance zone (T^z)
6: Compute $(A^z) = \mathbf{f}_i > 2\sigma \wedge \mathbf{f}_i < -2\sigma$ //compute anomalous zone (A^z)
7: **end for**

Input: FS
Output: SR, d_i^{al}, AA
Phase 2: 2-Sigma transmission optimization (FS)
1: **for** FS **do**
2: **if** $-1\sigma \leq FS \leq 1\sigma$ **then** //check for normal behavior
3: $Beh \leftarrow BN$
4: Goto 13
5: **else if** $1\sigma < FS \leq 2\sigma \wedge -1\sigma > FS \geq -2\sigma$ **then** //check for tolerated behavior
6: $Beh \leftarrow BT$
7: Goto 15
8: **else** //$FS > 2\sigma \wedge FS < -2\sigma$ implies that the behavior is anomalous
9: $Beh \leftarrow BA$
10: Goto 22
11: **end if**
12: **end for**
13: $A\_unt \leftarrow SR$ //aggregate sensed data in the aggregation unit
14: Goto 24
15: **if** $TR > 0$ **then** //check the trust value
16: Decr TR by ς //decrement the trust value
17: **else**
18: $TRNSMTd_i^{al}$ to BS //send abnormality alert to BS
19: $TRNSMTAA$ to msn_q //send abnormality agent to msn_q
20: Goto 24
21: **end if**
22: $TRNSMTd_i^{al}$ to BS //send abnormality alert to BS
23: $TRNSMTAA$ to msn_q //send abnormality agent to msn_q
24: **break**

of the sensor network increases. A conceptual view of the 2-sigma method is depicted in Fig. 4.1 [2]. The 2-sigma agent transmission optimization procedure pseudocode is given in Algorithm 4.2 [2].

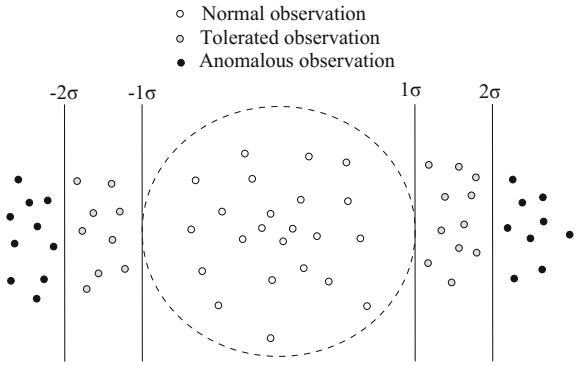

Fig. 4.1 Visualization of the 2-sigma method: © 2012 IEEE. Reprinted, with permission, from Usman et al. [2]

4.2.3 Weighted-Sum Optimization

The abnormality identification and confirmation module may bear the anomalous conduct of msn_q to optimize the agent communication for on-the-spot confirmation procedure. An essential approach for the optimization of agent communication can be to carry out the analysis of every feature, $\mathbf{f}_i$, to check its existence in the usual (i.e., $-1\sigma \leq \mathbf{f}_i \leq 1\sigma$), tolerated (i.e., $1\sigma < \mathbf{f}_i \leq 2\sigma \; OR \; -1\sigma > \mathbf{f}_i \geq -2\sigma$), or abnormal (i.e., $\mathbf{f}_i > 2\sigma \; OR \; \mathbf{f}_i < -2\sigma$) zones (as detailed in Sect. 5.2.2). The transmission of agent can then be restricted for the zone, viz. tolerance and transmitted only for the abnormal region. This tactic, however, may not treat the previous conduct of the mote and send the agent simply on the basis of the existence of existing abnormal observation (i.e., due to the temporary anomalous behavior), and it might result in unnecessary broadcast of the agent. Thus, the agent communication procedure should consider the weighted sum of the historical instances and existing observations for more robust abnormality agent transmission decision making. The historical observation score, S_{msn_q}, of the msn_q can be computed from the Eq. (4.6).

$$S_{msn_q} = \alpha_1(\frac{\Omega_{i1}}{h}) + \alpha_2(\frac{\Omega_{i2}}{h}) \tag{4.6}$$

In the above equation, α_1 and α_2 are the weighting influences for tolerated and abnormal occurrences of $\mathbf{f}_i$, correspondingly. The weighting influences $\alpha_2 > \alpha_1$ and $\alpha_1 + \alpha_2 = 1$. The weighting factor α_2 is allocated with a larger value because of the reason that it is related with abnormal occurrences of $\mathbf{f}_i$. In (4.6), S_{msn_q} possesses value in [0, 1] unit interval, and it is derived as a function of two key factors:

- Ω_{i1}: $\mathbf{f}_i$ number of occurrences from h past measurements that observe the condition $1\sigma < \mathbf{f}_i \leq 2\sigma$ or $1\,\sigma > \mathbf{f}_i \geq -2\sigma$.
- Ω_{i2}: $\mathbf{f}_i$ number of occurrences from h past observations that observe the condition $\mathbf{f}_i > 2\sigma$ or $\mathbf{f}_i < -2\sigma$.

It is evident from above-cited descriptions that the factors, viz. $\Omega_{i1} + \Omega_{i2} = h$. To optimize weighting factors' values, viz. α_1 and α_2, the term $\alpha_1 + \alpha_2 = 1$ can be adjusted as $\alpha_2 = 1 - \alpha_1$, h can be set as c constant, and $\Omega_{i1} + \Omega_{i2} = h$ can be adjusted as $\Omega_{i2} = c - \Omega_{i1}$. Consequently, (4.6) can be amended as shown below.

$$S_{msn_q} = \alpha_1(\frac{\Omega_{i1}}{c}) + \alpha_2(\frac{c - \Omega_{i1}}{c}) \tag{4.7}$$

Since $\alpha_1 + \alpha_2 = 1$ and $\alpha_2 > \alpha_1$, thus $\alpha_1 \in [0, 0.5)$. In the succeeding discussion, the window extent of the past observations is set as c, here $c = 10$. This process is simple to generalize. This shows that $\Omega_{i1} \in [1, 10]$, here $\Omega_{i1} = 1$ denotes there is only solitary measurement in h which occurs in a zone, viz. tolerance, whereas $\Omega_{i1} = 10$ represents there are all previous measurements in a zone, viz. tolerance, and none measurement in a zone, viz. anomalous.

The statistical *mean* values of objective functions are selected and associated *mean* value of initial parameters (i.e., α_1 and β_1) that are linked with the tolerated occurrences of $\mathbf{f}_i$ is identified to optimize parameters. The high value of initial parameters is not selected because of the reason that the succeeding parameters (i.e., α_2 and β_2) possess high primacy because of their linkage with the abnormal occurrences of $\mathbf{f}_i$.

The subsequent objective function, which is obtained from (4.7), is defined to optimize the parameters α_1 and α_2 to the extend that S_{msn_q} obtains the *mean* value, where x and w denote Ω_{i1} and α_1, correspondingly.

$$f(w, x) = (1 - w) + (2w - 1)\frac{x}{10} \tag{4.8}$$

Assume $x \in [1, 10]$ and $w \in [0, 0.5)$ due to the reasons that $\alpha_1 \in [0, 0.5)$ and $\Omega_{i1} \in [1, 10]$. This provides the $f(w, x) = 0.47$ *mean* value of the objective function. The associated w value, which produces objective function $f(w, x)$ *mean* value, stays in the limit [0.35, 0.47], having deviation 0.005 on either sides of value. The *mean* of bound produces $0.415 \approx 0.42$ that is taken as an optimal w value for above-cited situation. Figure 4.2 [1] shows a three-dimensional vision of the $f(w, x)$ objective function.

The optimum $\alpha_1 = 0.42$ value is obtained from above-cited process. By the implication of the relation $\alpha_2 = 1 - \alpha_1$, the value for α_2 may be inferred as 0.58. The concluding score for the transmission of an agent can be computed from (4.9).

$$S^t_{AA} = \frac{S_{msn_q}}{2} + \beta_1\mathbf{u} + \beta_2\mathbf{v} \tag{4.9}$$

where the β_1 and β_2 parameters are weighting influences for the recent observation that stay in the zones, viz. $1\sigma < \mathbf{f}_i \leq 2\sigma \; OR \; -1\sigma > \mathbf{f}_i \geq -2\sigma$, and $\mathbf{f}_i > 2\sigma \; OR \; \mathbf{f}_i < -2\sigma$, correspondingly. It is imperative to observe that the weighting influencing sets $\{\alpha_1, \alpha_2\}$ and $\{\beta_1, \beta_2\}$ are autonomous and possess no relation. The weighting influences $\beta_2 > \beta_1$ and $\beta_1 + \beta_2 = 1$. The weighting influence β_2 possesses the large value as it is linked with the present abnormal occurrence of $\mathbf{f}_i$.

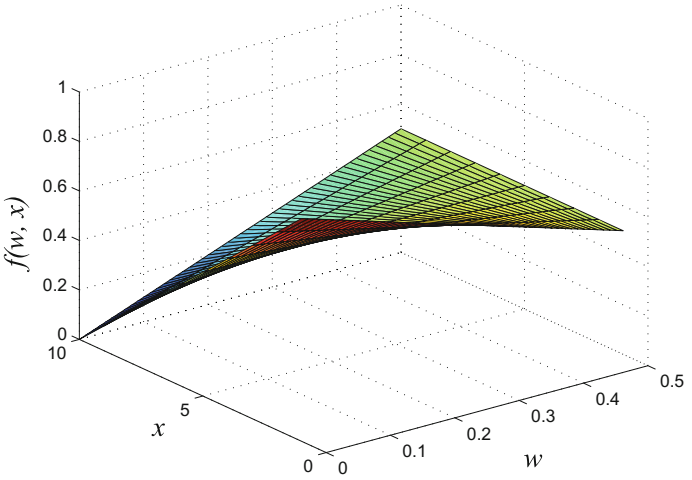

Fig. 4.2 Visualization of the objective function $f(w, x)$: © Academy Publisher, reprinted from Usman et al. [1]

In (4.9), the notation S^t_{AA} possesses number in [0, 1] unit interval and it is computed as a three-parameter function, viz. S_{msn_q}, **u**, and **v**. The S_{msn_q} is derived from (4.10), where the definitions of **u** and **v** are elucidated next.

- $\mathbf{u} = 1$, iff present $\mathbf{f}_i$'s occurrence fulfills the $1\sigma < \mathbf{f}_i \leq 2\sigma$ or $-1\sigma > \mathbf{f}_i \geq -2\,\sigma$ condition, else $\mathbf{u} = 0$.
- $\mathbf{v} = 1$, iff present $\mathbf{f}_i$'s occurrence fulfills the $\mathbf{f}_i > 2\sigma\ OR\ \mathbf{f}_i < -2\sigma$ condition, else $\mathbf{v} = 0$.

The association among **u** and **v** parameters is described as $\mathbf{v} = 1 - \mathbf{u}$. The weighting influences equality $\beta_1 + \beta_2 = 1$ can be reordered as $\beta_2 = 1 - \beta_1$. Consequently, (4.9) may be rearranged as given below.

$$S^t_{AA} = \frac{S_{msn_q}}{2} + 1 + 2\beta_1\mathbf{u} - \beta_1 - \mathbf{u} \tag{4.10}$$

Since $\beta_1 + \beta_2 = 1$ and $\beta_2 > \beta_1$, thus $\beta_1 \in [0, 0.5)$ and $S_{msn_q} \in [0, 1]$. The subsequent objective function, which is computed from (4.10), is defined to enhance the β_1 and β_2 parameters to the extent that S^t_{AA} attains the *mean* value, where y and z denote β_1 and S^t_{AA}, correspondingly.

$$f(y, z) = \begin{cases} \frac{z}{2} + 1 - y, & \mathbf{u} = 0, \\ \frac{z}{2} + y, & \mathbf{u} = 1. \end{cases} \tag{4.11}$$

To calculate the objective function $f(y, z)$ mean value, assume $z \in [0, 1)$ and $y \in [0, 0.5)$ and equivalent to $S_{msn_q} \in [0, 1]$ and $\beta_1 \in [0, 0.5)$, correspondingly.

The $f(y, z)$ objective function may has two situations, namely situation 1: $\mathbf{u} = 0$ and situation 2: $\mathbf{u} = 1$. The situation 1 has the minimum (*min*) (1), maximum (*max*) (1.01), and *mean* (1.0049) values. On the contrary, the situation 2 has *min* (0), *max* (0.99), and *mean* (0.5048) values. The associated y value of both situations stays in the bound [0.2401, 0.2499]. The interval *mean* produces $0.245 \approx 0.25$ that is considered as an optimal parameter y value.

On the basis of the calculation of the $f(y, z)$ objective function, the optimized $\beta_1 = 0.25$ value is derived. Again by the implication of the relation $\beta_2 = 1 - \beta_1$, the β_2 value is acquired as 0.75.

The computed value of agent transmission, S^t_{AA}, must be larger than already set threshold ψ to send an agent, where $\psi \in (0, 1)$. It is imperative to observe that the terminating upper limit level of total score of agent dispatch, S^t_{AA}, is fixed as 1. Nonetheless, this would not change decision of agent dispatch, because any value higher than ψ is considered as reasonable for the communication of agent. Moreover, if the score of agent dispatch is below than ψ, the abnormality identification and confirmation module may take other usual actions, for instance, reducing the interaction with msn_q, announcing msn_q as abnormal to other leader motes and member motes, and transmitting an alarm to BS for tolerated categories, viz. 1, 2, and 3, correspondingly (as discussed in Sect. 4.2.1). This tactic initiates less frequent communications of agents which decreases consumption of energy and enhances total lifetime of the network. The pseudocode of the procedure of the optimization of agent transmission is provided in Algorithm 4.3 [1]. Observe that Algorithm 4.3 (Phase 1) is executed only when the system is deployed and on every occasion when tuning action (denoted by τ and discussed in Sect. 3.4.1) is carried out by the user; on the contrary, Phase 2 is computed by the abnormality identification and confirmation module for every received abnormal observation.

4.2.4 Complexity Analysis

The space and time complexity of the algorithms is discussed below.

Theorem 4.1 *The space complexity for (i) the procedure for the abnormality identification on cln_q is $C_n + l[n]$ and (ii) the procedure for the optimization of the abnormality agent transmission on cln_q is constant C_v.*

Proof (i) Assume $FS_2 = \{\jmath, f\}$ be the numbers which are computed on cln_q after obtaining the O_j observation, where O_j possesses FS_1 values. Therefore, $FS = FS_1 \cup FS_2 = \{\lambda, \varphi, \upsilon, \jmath, f\}$. FS_2 consumes $l[j]$ memory space. The entire space consumed by n FS features thus becomes $l[n] = l[j] \cup l[m]$; here, $l[j] < l[m]$. The cln_q mote consumes constant spaces, viz. C_1, C_2, and C_3, to save values of thresholds for joins, combined value, and outcome of the optimization of the transmission of agent procedure, correspondingly. The agent takes C_4 and C_5 spaces to save data and code, correspondingly. Therefore, the entire memory space consumed by the abnormality identification procedure is $\cup_{n=1}^{5} C_n + l[n]$.

Algorithm 4.3 Weighted sum optimization

Input: FS
Output: $-1\sigma, -2\sigma, 1\sigma, 2\sigma$ for all $\mathbf{f}_i \in FS$
Phase 1: Compute zones (FS)
1: At t_k time
2: **for** each $\mathbf{f}_i \in FS$ **do**
3: Compute $-1\sigma, 1\sigma, -2\sigma, 2\sigma$ //compute values of threshold
4: Compute $(N^z) =$ -1 $\sigma \leq \mathbf{f}_i \leq 1\ \sigma$ //compute normal zone (N^z)
5: Compute $(T^z) = 1\ \sigma < \mathbf{f}i \leq 2\ \sigma \wedge$ -1 $\sigma > \mathbf{f}_i \geq$ -2 σ //compute tolerance zone (T^z)
6: Update $(A^z) = \mathbf{f}_i > 2\sigma \wedge \mathbf{f}_i < -2\sigma$ //compute anomalous zone (A^z)
7: **end for**

Input: FS
Output: Beh
Phase 2: Weighted sum transmission optimization (FS)
1: **for** each $\mathbf{f}_i \in FS$ **do**
2: Compute $S_{msn_q} = \alpha_1(\frac{\Omega_{i1}}{w}) + \alpha_2(\frac{\Omega_{i2}}{w})$ //compute the historical observation score S_{msn_q}
3: Compute $S^t_{AA} = \frac{S_{msn_q}}{2} + \beta_1 y + \beta_2 z$ //compute the abnormality agent transmission score S^t_{AA}
4: **if** $S^t_{AA} \geq \psi$ **then** //check for the anomalous behavior
5: $Beh \leftarrow BA$
6: break
7: **else if** $S^t_{AA} < \psi \wedge \geq \zeta$ **then** //check for the BT_ζ tolerated behavior
8: $Beh \leftarrow BT_\zeta$
9: break
10: **else if** $S^t_{AA} < \zeta \wedge \geq \gamma$ **then** //check for the BT_γ tolerated behavior
11: $Beh \leftarrow BT_\gamma$
12: break
13: **else if** $S^t_{AA} < \gamma \wedge \geq \eta$ **then** //check for the BT_η tolerated behavior
14: $Beh \leftarrow BT_\eta$
15: break
16: **else** //$S^t_{AA} < \eta$ implies that the behavior is normal
17: $Beh \leftarrow BN$
18: **end if**
19: **end for**
20: return Beh to Algorithm 4.1.

(ii) The cln_q takes constant memory spaces C_7, C_8, C_9, and C_{10} to save weighting factors α_1, α_2, β_1, and β_2 values, correspondingly. The memory spaces C_{11} and C_{12} are consumed by cln_q to save f_i instances values from w past tolerated and abnormal observations. Correspondingly, cln_q consumes C_{13} space to save the tolerated or abnormal occurrence value of the present-obtained observation. The C_{14}, C_{15}, C_{16}, and C_{17} memory spaces are consumed by cln_q to save the score of the past observations of msn_q, the score of agent transmission, the value of the threshold for the agent transmission, and the conduct status of msn_q, correspondingly. Assuming $C_v = \cup_{v=7}^{17} C_v$, the complexity of procedure of dispatch of agent is constant C_v.

Theorem 4.2 *(i) the abnormality identification procedure on cln_q executes in constant D time for usual measurements and it has the $O(n)$ time complexity for*

abnormal observations and (ii) for the optimization of transmission of agent procedure on cln_q is $O(y)$.

Proof (i) The cln_q consumes the constant time D_2 to receive the observation from msn_q, D_3 time to retrieve FS_1 values, and D_4 time to calculate the FS_2 values. To accumulate sensor reading, msn_q takes the D_5 time to send the outcome to BS for further examination. cln_q consumes D_6 time to carry out the abnormality identification procedure by employing the joins and allocating the appropriate behavior to msn_q the value of that is obtained after the calculation carried out by the procedure of optimization of transmission of agent, where $D_6 > D_5$. cln_q takes D_7 time for the transmission of agent to msn_q; msn_q is abnormal in such a situation. Therefore, assuming $D = \sum_{i=2}^{7} D_i$, the procedure of the identification of abnormality executes in D constant time. Furthermore, Algorithm 4.2 calls the phase, viz. agent transmission optimization for abnormal observations that has the time complexity of $O(n)$.

(ii) cln_q consumes y time to examine multiple zones for n features to categorize as usual, tolerated, or abnormal to carry out the optimization of agent dispatch procedure. By taking the upper limit case, time complexity for optimization of dispatch of agent procedure becomes $O(y)$.

4.3 Formal Modeling and Analysis

This section first addresses the individual formal specifications of the algorithmic specifications presented in this chapter. This is followed by the unified model formulation. Finally, the behavioral and structural properties are formulated and analyzed.

4.3.1 Model Formulation

The first algorithmic specification elucidates the method for the first-order abnormality identification by the cln_q. The formal specification of the method is shown below.

Net module FO: The first-order abnormality identification module, (PN_{FO}), is a 5-tuple net: $PN_{FO} = (P_{FO}, T_{FO}, F_{FO}, W_{FO}, (M_0)_{FO})$, where $P_{FO} = \{p_{21}, p_{22}, p_{23}, p_{41}\}$ and $T_{FO} = \{t_{24}, t_{25},..., t_{31}, t_{47},..., t_{51}\}$ are non-empty, finite, and disjoint sets of places and transitions, correspondingly. $F_{FO} = \{(p_{21}, t_{24}), (p_{21}, t_{25}), (p_{21}, t_{26}), (p_{21}, t_{27}), (p_{21}, t_{28}), (t_{24}, p_{22}), (t_{25}, p_{22}), (t_{26}, p_{22}), (t_{27}, p_{22}), (t_{28}, p_{22}), (t_{24}, p_{23}), (t_{25}, p_{23}), (t_{26}, p_{23}), (t_{27}, p_{23}), (t_{28}, p_{23}), (p_{22}, t_{29}), (p_{23}, t_{30}), (p_{23}, t_{30}), (p_{41}, t_{47}), (p_{41}, t_{48}), (p_{41}, t_{49}), (p_{41}, t_{50}), (p_{41}, t_{51})\}$ is set of arcs, $W_{FO} = 1$ is weight for all arcs, and $(M_0)_{FO} = p_{21}$ denotes solitary token in the initial place.

The execution of the first-order abnormality identification algorithm is followed by the invocation of the 2-sigma abnormality agent transmission optimization method, which is formally characterized next.

Net module 2S: The 2-sigma optimization module, (PN_{2S}), is a 5-tuple net: $PN_{2S} = (P_{2S}, T_{2S}, F_{2S}, W_{2S}, (M_0)_{2S})$, where $P_{2S} = \{p_{24}, p_{25},..., p_{29}\}$ and $T_{FO} = \{t_{32}, t_{33},..., t_{38}\}$ are non-empty, finite, and disjoint sets of places and transitions, correspondingly. $F_{FO} = \{(p_{24}, t_{32}), (t_{32}, p_{25}), (t_{32}, p_{26}), (p_{25}, t_{33}), (p_{26}, t_{34}), (t_{34}, p_{27}), (t_{34}, p_{28}), (p_{27}, t_{35}), (p_{27}, t_{36}), (p_{28}, t_{37}), (t_{37}, p_{29}), (p_{29}, t_{38})\}$ is set of arcs, $W_{2S} = 1$ is weight for all arcs, and $(M_0)_{2S} = p_{24}$ denotes solitary token in the initial place of PN_{2S}.

Alternatively, depending upon the requirements of the user, the execution of the first-order abnormality identification algorithm can be followed by the invocation of the weighted-sum abnormality agent transmission optimization method, which can be formally defined as given below.

Net module WS: The weighted-sum optimization module, (PN_{WS}), is a 5-tuple net: $PN_{WS} = (P_{WS}, T_{WS}, F_{WS}, W_{WS}, (M_0)_{WS})$, where $P_{WS} = \{p_{30}, p_{31},..., p_{40}\}$ and $T_{WS} = \{t_{39}, t_{40},..., t_{46}\}$ are non-empty, finite, and disjoint sets of places and transitions, correspondingly. $F_{WS} = \{(p_{30}, t_{39}), (p_{30}, t_{40}), (t_{39}, p_{31}), (t_{40}, p_{31}), (p_{31}, t_{41}), (t_{41}, p_{32}), (t_{41}, p_{33}), (p_{32}, t_{46}), (p_{33}, t_{42}), (t_{42}, p_{34}), (t_{42}, p_{35}), (p_{34}, t_{46}), (p_{35}, t_{43}), (t_{43}, p_{36}), (t_{43}, p_{37}), (p_{36}, t_{46}), (p_{37}, t_{44}), (t_{44}, p_{38}), (t_{44}, p_{39}), (p_{38}, t_{46}), (p_{39}, t_{45}), (t_{45}, p_{40}), (p_{40}, t_{46})\}$ is set of arcs, $W_{WS} = 1$ is weight for all arcs, and $(M_0)_{WS} = p_{21}$ denotes solitary token in the initial place of PN_{WS}.

In order to formulate the unified model, the modules are joined in the transition-place manner. In this process, the following additional arcs have been introduced: $F_a = \{(t_{30}, p_{30}), (t_{31}, p_{24}) (t_{46}, p_{41})\}$. The corresponding weights are defined as $W_a(t_{30}, p_{30}) = W_a(t_{31}, p_{24}) = W_a(t_{46}, p_{41}) = 1$. The resultant unified model, (PN_u), is depicted in Fig. 4.3 and formally described below.

Unified model The (PN_u), a unified PN model, is a 5-tuple net: $PN_u = (P_u, T_u, F_u, W_u, (M_u)_0)$, where $P_u = \{P_{FO}, P_{2S}, P_{WS}\}$ and $T_u = \{T_{FO}, T_{2S}, T_{WS}\}$ are non-empty, finite, and disjoint sets of places and transitions, correspondingly. Similarly, $F_u = \{F_{FO}, F_{2S}, F_{WS}, F_a\}$ and $W_u = 1$ is weight for all arcs including additional arcs. Finally, $(M_0)_u = p_{21}$.

4.3.2 Formal Characterization and Analysis

On the basis of the formal specifications stated above, the behavioral properties, namely safeness and liveness of the proposed algorithmic specifications, can be formally verified as shown below.

Theorem 4.3 *The model, PN_u, is safe.*

Proof For all $p \in P_u$, $M(p) = 1$. Although firing of the transitions t_{24}, t_{25}, t_{26}, t_{27}, and t_{28} yields 5 tokens in the place p_{22} or p_{23}, these tokens, however, merge together and flow as a single token in the rest of the net. This yields that the unified Petri net model, PN_u, is safe.

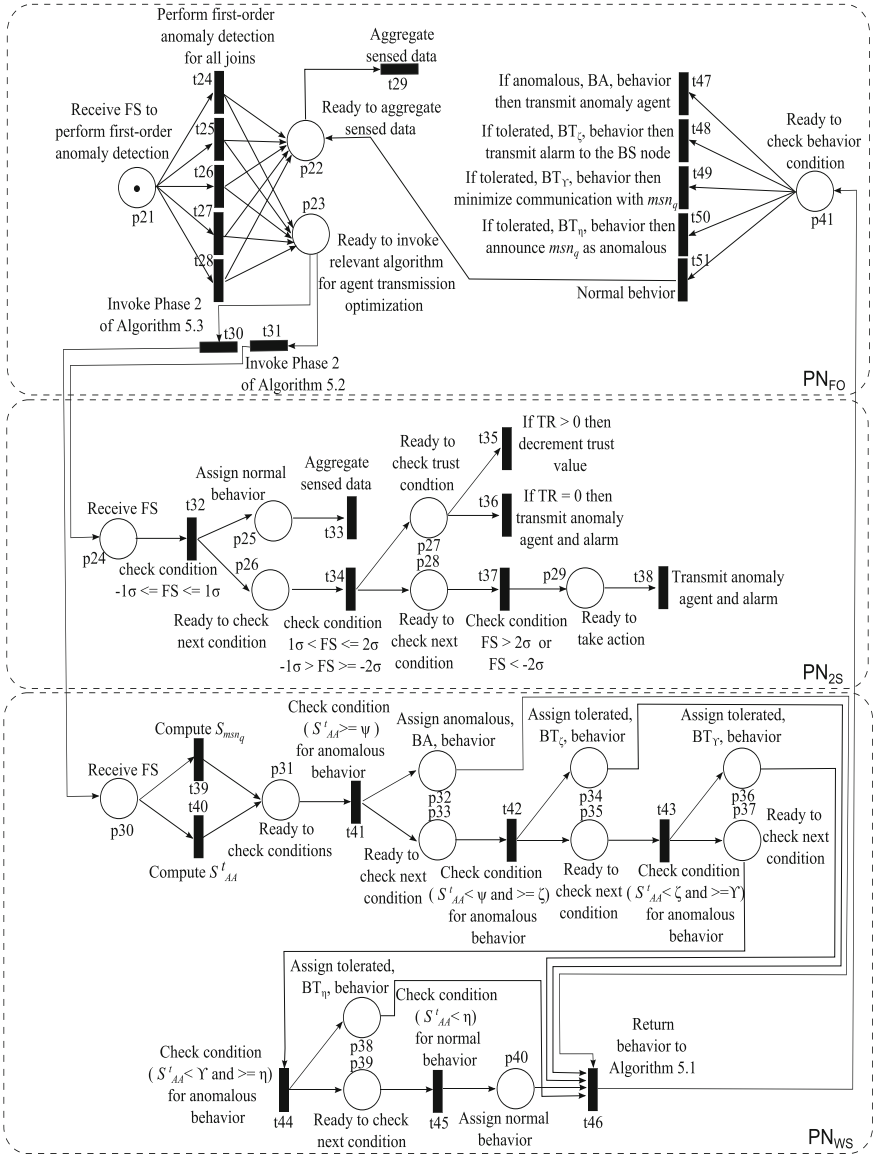

Fig. 4.3 The unified formal model

Theorem 4.4 *The unified Petri net model, PN_u, is level 4 live.*

Proof The terminal transitions t_{29}, t_{33}. t_{35}, t_{36}, t_{38}, and $\{t_{47}, t_{48}, t_{49}, t_{50}\}$ are level 4 live as there are firing sequences for all these transitions. The firing sequence for the transition t_{29} is $\{t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\} \rightarrow t_{29}$. Then, the firing sequence for the transition t_{33} is $\{t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\} \rightarrow t_{31} \rightarrow t_{32} \rightarrow t_{33}$. Similarly, the firing sequence for the transitions t_{35} and t_{36} is $\{t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\} \rightarrow \rightarrow t_{31} \rightarrow t_{32} \rightarrow t_{34}$ which then leads to either t_{35} or t_{36} depending on the state of the system. Next, the firing sequence for the transition t_{38} is $\{t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\} \rightarrow \rightarrow t_{31} \rightarrow t_{32} \rightarrow t_{34} \rightarrow t_{37} \rightarrow t_{38}$. The firing sequence for the set of transitions $\{t_{47}, t_{48}, t_{49}, t_{50}\}$ is $t_1 \rightarrow \{t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\} \rightarrow t_{30} \rightarrow \{t_{39}, t_{40}\} \rightarrow \{\{t_{41}, t_{46}\}, \{t_{41}, t_{42}, t_{46}\}, \{t_{41}, t_{42}, t_{43}, t_{46}\}, \{t_{41}, t_{42}, t_{43}, t_{44}, t_{46}\}, \{t_{41}, t_{42}, t_{43}, t_{44}, t_{45}, t_{46}\}\} \rightarrow \{t_{47}, t_{48}, t_{49}, t_{50}\}$. Finally, the firing sequence for the transition t_{29} can also achieved as $t_1 \rightarrow \{t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\} \rightarrow t_{30} \rightarrow \{t_{39}, t_{40}\} \rightarrow \{\{t_{41}, t_{46}\}, \{t_{41}, t_{42}, t_{46}\}, \{t_{41}, t_{42}, t_{43}, t_{46}\}, \{t_{41}, t_{42}, t_{43}, t_{44}, t_{46}\}, \{t_{41}, t_{42}, t_{43}, t_{44}, t_{45}, t_{46}\}\} \rightarrow t_{51} \rightarrow t_{29}$. Thus, PN_u is level 4 live, that is, no deadlock exists in the model, PN_u.

4.4 Performance Evaluation

The temporal behavior of the abnormality identification and confirmation system has been thoroughly investigated through implementation on a small-scale real test bed in the previous chapter. A simulation study has been carried out and elucidated to analyze the conduct of the system in a large-scale network. The proposed algorithms have also been implemented on a real mote to measure their consumption of memory, consumption of energy, and processing time. Finally, the comparison of the system has been made with a few recent competing schemes.

4.4.1 Simulation Study

In the simulation study, the first-order abnormality identification algorithm was employed for identifying abnormalities. Similarly, the 2-sigma and weighted-sum algorithms were employed for abnormality agent transmission optimization. The simulation scenarios have been developed to analyze the system performance, in discrete event and object-oriented environment that emulates events in a sequential order [3]. The environment of simulation was based on the following setup.

4.4.1.1 Simulation Setup

- *Network model*: The horizon of simulation had the $Wd \times Lg$ square meter area. k motes were arbitrarily positions in multiple simulation scenarios. The BS was positioned at (a, b) location.
- *Mote model*: The standard capability of resources of MICAz was employed [4]. The flash data logger, SRAM, and program memories were set as M_{FLASH}, M_{SRAM}, and M_P, correspondingly. The overall mote energy was set as N_E at the start of network lifetime. Energy consumption, by a mote during the sleep state, was supposed as N_{Eslp}.
- *Mote categorization*: The scenario of smart home scenario was assumed to study the consumption of energy. The motes were characterized as non-security and security motes. In practice, the non-security sensors may include temperature, humidity, and pressure sensors, whereas the light and motion detectors may be considered as security sensors. A limit, σ, was fixed for security motes, whereas a comparatively lenient limit, 2σ, was fixed for the non-security sensors.
- *Model for dissipation of communication energy*: A common model was employed for energy consumption [5]. The following relations were employed to evaluate the dissipation of energy by radio hardware to send l-bits on d distance.

$$E_{Tx}(l, d) = E_{Tx-elec}(l) + E_{Tx-amp}(l, d) \tag{4.12}$$

$$P_U(u) = \begin{cases} lE_{elec} + l\epsilon_{fs}d^2 & d < d_0, \\ lE_{elec} + l\epsilon_{mp}d^4 & d \geq d_0 \end{cases} \tag{4.13}$$

Correspondingly, to get l-bits, the consumption of energy was evaluated from the relation.

$$E_{Rx}(l) = E_{Rx-elec}(l) + lE_{elec} \tag{4.14}$$

where E_{elec} denotes the electronic energy dissipation which has multiple aspects such as filtering, digital coding, modulation, and spread of signals. Correspondingly, $\epsilon_{fs}d^2$ and $\epsilon_{mp}d^4$ are the energies of amplifier that relies on multiple aspects, viz. distance among receivers and senders and acceptable bit-fault rate. It is imperative to note that E_{elec} is equal to $(E_{Tx} + E_{DA})$ for cluster leader motes and E_{Tx} for cluster member motes for communication; here, E_{Tx} represents the transmission energy and E_{DA} denotes the energy of data accumulation. On the contrary, the E_{elec} values were equal to E_{Rx} for leader and member motes while getting packets.
- *Network lifetime*: An approach, viz. "first mote die" approach, was employed for analysis [6].
- *Propagation model*: The d^2 (free space, f_s) and d^4 (multipath fading, mp) power loss models were employed as propagation models [5]. If distances among receivers and senders were less than d_0 threshold, then f_s, otherwise mp model, was employed.

- *Traffic model*: The cluster member motes were set to periodically transmit sensed data. The cluster member motes sent sensed data after each t seconds. The sent sensed data had Pkt size, having both payload and header. The payload of usual packet, shown in Fig. 4.4 [1], had FS_1 values that were sent by cluster member motes. The features, viz. υ, T, and λ, were assumed as continuous variables in analysis; on the contrary, F and φ were treated as discrete variables.
- *Cluster formation*: The groups (clusters) were made by employing LEACH, a famous cluster creation algorithm [5]. The scenarios were based on heterogeneous motes (in terms of resource capability), and LEACH facilitates such motes to create the cluster-oriented topology.
- *Abnormality agent*: The developed agent size was 762 bytes (having itinerary, data, code, and identity). An agent is not dispatched as solitary packet due to its larger size. Thus, the agent was divided into several frames according to the specification of 802.15.4 standard [7]. The maximum allowed frame limit is 127 bytes (i.e., with the payload and header sizes of 102 and 25 bytes, correspondingly) on the basis of 802.15.4 specifications. Thus, the agent was divided into eight data packets. The size of header was fixed as 25 × 8 = 200 bytes. On the contrary, the size of the payload of initial seven data packets was 102 × 7 = 714 bytes, and for final packet, 48 × 1 = 48 bytes. The total size of initial seven packets, by employing the ((*header size* × *number of packet*) + (*payload* × *size number of packets*)) formula, was (25 × 7) + (102 × 7) = 175 + 714 = 889 bytes. Correspondingly, the final packet size was (1 × 25) + (1 × 48) = 25 + 48 = 73 bytes. Therefore, the agent size was 962 bytes after its division. The packet size, which takes the outcome of on-the-spot confirmation procedure, was 27 bytes (i.e., 2 and 25 payload and header sizes), as that packet just transmits the outcome of the confirmation procedure as either 0" or 1 denoting 'usual" or abnormal status, correspondingly. The structure of data packets of agent and confirmation procedure is shown in Figs. 4.5, 4.6, and 4.7 [1].
- *Anomalous traffic*: The 25% anomalous traffic was randomly generated by abnormal motes in the simulation horizon. The usual and abnormal traffics of data were accumulated and afterward employed for computing first-order bounds.
- *Thresholds for agent transmission optimization*: The six-sigma rule-based values, viz. -2σ, -1σ, 1σ, and 2σ, were employed to optimize transmission of agent [8]. The weighting factors values were fixed as α_1, α_2, β_1, and β_2. The agent transmission threshold was set as ψ; on the contrary, the zones for tolerance were set as Tol_η, Tol_γ, and Tol_ζ.
- Iterations: The reported outcomes are based on 30 iterations of the simulated experiments.

The subsequent seeds were employed to form simulation cases: $k = 30 - 150$, $Wd = 100$, $Lg = 100$, $a, b = 50$, $E_{Tx}, E_{RX} = 50 \times 10^{-9}$ Joules, $\epsilon_{mp} = 1.3 \times 10^{-3} \times 10^{-12}$ $Joules/bit/m^4$, $\epsilon_{fs} = 10 \times 10^{-12}$ $Joules/bit/m^2$, $E_{DA} = 5 \times 10^{-9}$ $Joules/bit/signal$, $M_p = 128\ kb$, $M_{SRAM} = 4\ kb$, $M_{flash} = 512\ kb$, $N_E = 1e^4\ nJ$, $t = 0.1$, $Pkt = 31$ bytes, $N_{Eslp} = 1\ nJ/t$, -1σ to $1\sigma = 0.68$, -2σ to $2\sigma = 0.95$, λ = 13 °C to 39 °C, T = 0–1, 2–3, 4–5 s, $\varphi = 1$ for allowed job carried out and 0 for not-allowed job carried out, υ = 100% (i.e., full battery) → 0% (i.e., empty battery),

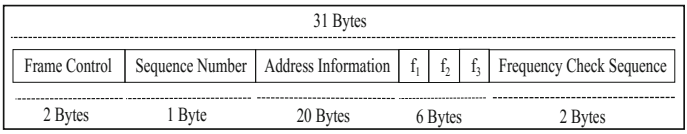

Fig. 4.4 Usual data packet: © Academy Publisher, reprinted from Usman et al. [1]

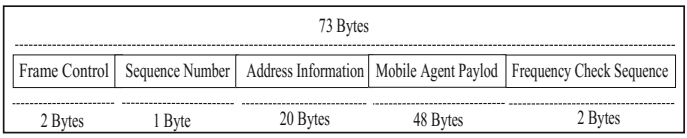

Fig. 4.5 Agent data packets (excluding last packet): © Academy Publisher, reprinted from Usman et al. [1]

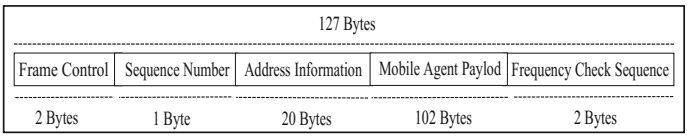

Fig. 4.6 Last data packet of agent: © Academy Publisher, reprinted from Usman et al. [1]

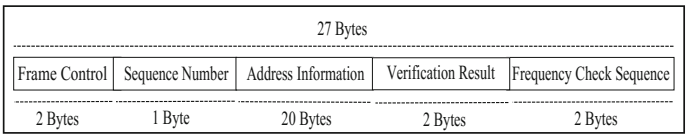

Fig. 4.7 On-the-spot confirmation outcome data packet: © Academy Publisher, reprinted from Usman et al. [1]

$F = 1$ for every timely obtained and 0 for each deferred obtained packet, $h = 10$, $\alpha_1 = 0.42$, $\alpha_2 = 0.58$, $\beta_1 = 0.25$, $\beta_2 = 0.75$, $\eta = 0.50$, $\psi = 0.55$, $\zeta = 0.40$, and $\gamma = 0.45$.

4.4.1.2 Results and Analysis

The comprehensive performance of the abnormality identification and confirmation system was measured in regard to the subsequent performance measures:

- *Rate of abnormality detection*: This measure provides the percentage of the abnormalities identified from total abnormalities.
- *Estimation of Energy consumption*: The consumption of energy is assessed for both usual and abnormal network traffics. This facilitates to measure the effect of employing agents for on-the-spot confirmation on the energy level of a mote with limited resources.

- *Number of transmitted agents*: This measure assists in computing the number of agents that are sent with employing and without employing optimization of agent communication procedure.

The initial group of experiments was carried out to measure the rate of detection of first-order abnormalities that occurred due to on-the-spot faults (see $N(\lambda, T)$ join, as described in Sect. 4.2.1). In this group of experiments, member motes of cluster uninterruptedly sent FS_1 values in place of periodic communications to the corresponding leader motes of clusters. The produced network traffic was comprised of 5000 packets that were sent by thirty member motes to their respective leader motes. In such a situation, the abnormality identification and confirmation module discovered 99% of the abnormalities. For network traffic having 7000, 9000, 11000, and 13000 measurements sent by 60, 90, 120, and 150 member motes, the abnormality identification rates were 98.80%, 98.40%, 98.20%, and 98%, correspondingly. The trials' outcomes illustrate the identification rate of abnormalities occurred due to on the spot faults remained in the limit between 98 and 99%.

The second and next abnormalities are associated with consumption of resources by member motes. A resource exhaustion attack was performed on the member motes, in which member motes sent low-value status of battery rather than values that are expected residual battery values (see second join $N(T, \upsilon)$, as elucidated in Sect. 4.2.1). The total rate of detection of abnormalities occurs due to exhaustion of resource attack randomized among 98.60 and 99%. The rates of detection for 5000, 7000, 9000, 11000, and 13000 observations remained 99%, 98.80%, 98.80%, 98.60%, and 98.60%, correspondingly. Then, the abnormality identification and confirmation modules on the leader motes were configured to identify the abnormalities occurred due to faulty motes (see third $N(\varphi, \upsilon)$, as described in Sect. 4.2.1). In these trials, the energy stock of defective member motes consumed rapidly because of the unapproved activities carried out by the motes. As a result, the motes sent low-level status of battery in place of expected values to the related leader motes. In this group of experiments, the rates of detection were 98.2%, 98.80%, 99%, 97.80%, and 98.80% for 5000, 7000, 9000, 11000, and 13000 observations, correspondingly.

Then, the case of faulty motes was investigated (see fourth join $N(\varphi, T)$, as described in Sect. 4.2.1), wherein defective member motes sent abnormal values of the allowed activities in regard to time duration. In such situations, the abnormality identification and confirmation modules, positioned on every leader mote of the cluster, spotted 98.2%, 98.80%, 99.2%, 98%, and 98.80% abnormalities for 5000, 7000, 9000, 11000, and 13000 observations, correspondingly. Lastly, the denial-of-sleep attack situations were introduced (see fifth join $N(F, T)$, as elucidated in Sect. 4.2.1). In these situations, the member motes of clusters unremittingly sent observations other than intermittent communications. In this group of trails, the rates of detection for 5000, 7000, 9000, 11000, and 13000 observations were 98%, 98.60%, 99%, 97.80%, and 98.80%, correspondingly. In all above-cited situations, agents were able to recognize the origin of abnormalities which have caused on the spot or in transmission. The plot revealed in Fig. 4.8 [1] gives the abnormality

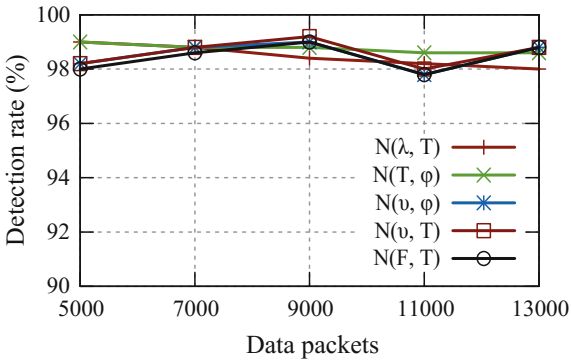

Fig. 4.8 Rate of detecting first-order abnormalities: © Academy Publisher, reprinted from Usman et al. [1]

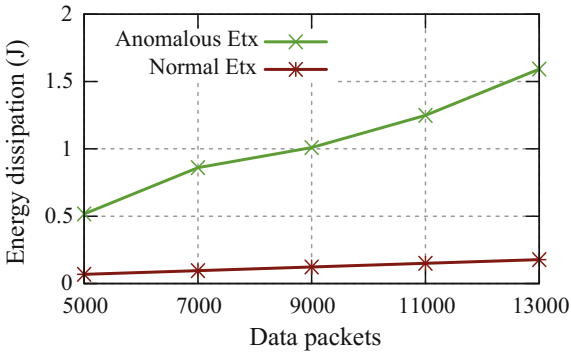

Fig. 4.9 Energy consumption by transmission of packets: © Academy Publisher, reprinted from Usman et al. [1]

identification outcomes. It is evident that the accuracy of detection is higher for every join as it remained among 97.80 and 99.20%.

Next, the consumption of energy in situations, i.e., usual and abnormal packets are sent in variable mote density situations, is examined. The usual data traffic had packets that were sent from member motes to their respective leader motes of the clusters. On the contrary, the abnormal traffic had packets of agents that were sent from leader motes to the doubtful member motes for on-the-spot confirmation and outcomes of on-the-spot confirmation procedures that were sent back to leader motes of the clusters. It is obvious from Fig. 4.9 [1] that the line that represents the consumption of energy by transmission of 5000, 7000, 9000, 11000, and 13000 usual packets that were sent by 30, 60, 90, 120, and 150 member motes to corresponding leader motes has a stable evolution in positive path beside x-axis. This illustrates the steady surge in consumption of energy as statistic of packets rises in network traffic, whereas the consumption of energy by abnormal traffic of the network which has agents and on-the-spot confirmation packets incline to change and has a moderately non-stable evolution. This is because of the randomization in the detection accuracy of related leader motes of the clusters and also variations involved in underlying abnormal traffic.

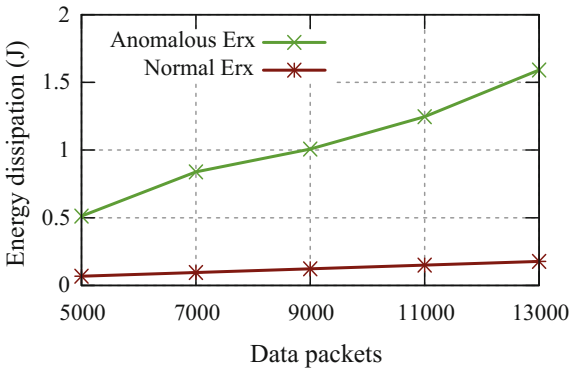

Fig. 4.10 Energy consumption by receiving packets: © Academy Publisher, reprinted from Usman et al. [1]

The energy consumption cost of receiving packets is studied next. Figure 4.10 [1] shows the consumption of energy outcomes for receiving both usual and abnormal packets. The consumption of energy due to the reception of packets trails the similar tendency as displayed by the packets sent with a somewhat minor cost. The traffic, having 5000–7000 packets, instigated the consumption of 0.0686–0.177 J energy for sending and 0.0682–0.176 J energy for getting the usual traffic. Likewise, for sending and getting abnormal traffic, the consumed energy by network was 0.517–1.592 J and 0.51215–1.591 J, correspondingly. The moderately low consumption of energy during getting data traffic is because of the reason the communication procedure includes distance influence along with the static quantity of energy spent by transceiver for dispatch of data. An imperative fact that is clear from Figs. 4.9 and 5.10 is variance in scale of consumption of energy by usual traffic with abnormal traffic. This variance of consumption of energy scale is due to the additional traffic sent on the network as agents and on-the-spot confirmation outcome packets for the situation of abnormal traffic. The resource-wealthy group head motes can effortlessly manage such an overhead. Though, the resource-constrained member motes may rapidly consume their energy that promotes the usage of optimization of agent transmission procedure to save the energy resources of resource-constrained motes.

The abnormality identification and confirmation modules, positioned on cluster leader motes, sent 1240, 1770, 2140, 2570, and 3249 agents subsequent to the identification of same number of abnormalities by not using optimization of agent transmission procedure in the traffic having 5000, 7000, 9000, 11000, and 13000 data packets, correspondingly. As cited above, this rapidly consumed energy budget of resource-constrained cluster member motes. The proposed abnormality agent transmission optimization methods (i.e., Algorithms 4.2 and 4.3) were, therefore, employed. As a result, the transmissions of agents were decreased to 818, 1204, 1477, 1825, and 2177 in the case of 2-sigma optimization method (i.e., Algorithm 4.2). On the other hand, the transmissions of agents were decreased to 600, 993, 1080, 1513, and 1560 in the case of the weighted-sum optimization method (i.e., Algorithm 4.3). These results are shown in Fig. 4.11 [1]. This decreased the consumption of energy by both reception and transmission abnormal traffics on the network to around 29–34%

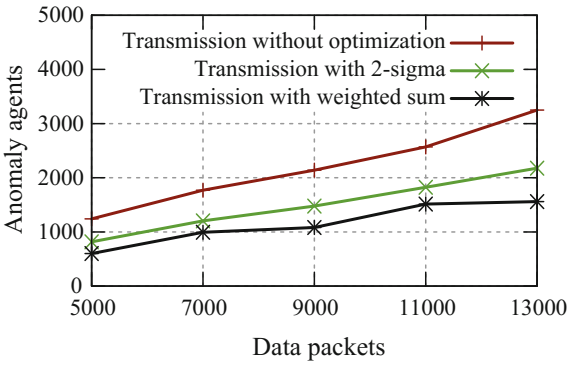

Fig. 4.11 Optimization of agent transmission: © Academy Publisher, reprinted from Usman et al. [1]

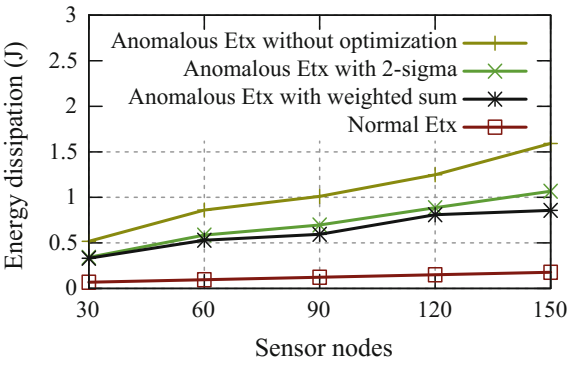

Fig. 4.12 Consumption of energy by transmission of agent: © Academy Publisher, reprinted from Usman et al. [1]

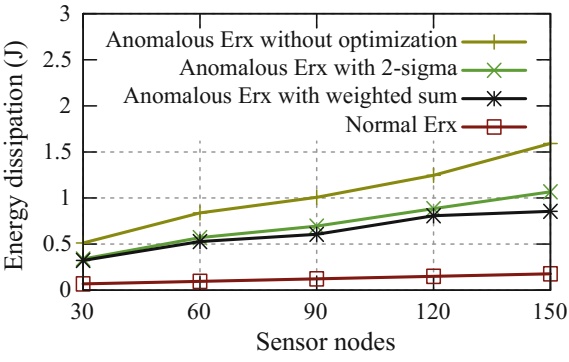

Fig. 4.13 Consumption of energy by receiving agent: © Academy Publisher, reprinted from Usman et al. [1]

for the situation of the usage of 2-sigma optimization procedure. The reduction of the energy dissipation was even higher in the case of the weighted-sum optimization method, that is, down to around 42–52%, as depicted in Figs. 4.12 [1] and 4.13 [1].

The above-mentioned analysis reveals that the abnormality identification and confirmation system is not merely able to identify first-order abnormalities with higher rate of detection but also able to effectively carry out on-the-spot confirmation job on

suspicious motes. Moreover, the optimization of agent transmission procedure that considers both previous and present observations to enhance the procedure of agent transmission can extend the lifetime of the network l by approximately 42–52%. Correspondingly, the optimization of agent transmission procedure that outlines the restriction zones on the probability distribution of features extends the lifetime of the network approximately 29–34%.

As indicated by the simulation results, most of the network energy is consumed by transmission and reception of agents in the functioning of the abnormality identification and confirmation system. This study has proposed two methods, namely 2-sigma and weighted-sum optimization, to optimize the agent transmission. However, because of the inherent characteristics of the agent-enabled tiny sensor mote networks, if our proposed system is deployed on homogeneous networks with battery-operated mote, then transmissions and receptions of agents will result in quick depletion of battery resources of certain cluster leaders and some of their member motes. This may result in re-clustering of the network, which is an energy-consuming process. The re-clustering of a network is also not suitable due to the sensed data stored for on-the-spot confirmation purposes. The transmission of this data from one cluster leader mote to another cluster leader mote will also consume energy, and the old data may not remain suitable for on-the-spot confirmation procedure. Thus, the system is better suited for heterogeneous sensor networks.

4.4.2 Implementation

The algorithms have been implemented on MICAz [4], having TinyOS 2.1.1 [9], to study their effect on low resources of real sensor motes. MICAz has a microprocessor (ATmega128L). The configuration EEPROM, data logger, and flash memories of MICAz have 4, 512, and 128 kb storage memories, correspondingly. The algorithms are developed using nesC [10] that is employed for developing applications in TinyOS. The aim of development is estimation of consumption of energy and memory resources with processing time consumed by algorithms. It is pertinent to note that the worst situation, that is, all relations within the algorithms were finished in order to measure the effect of algorithms on resources of MICAz. Moreover, only those algorithms, of the system, are implemented which execute either on cluster member motes or on cluster leader motes.

4.4.2.1 Memory Consumption

The nesC produces a report for consumption of memory by programs in the phase of compilation. The report shows the statuses of ROM and RAM of MICAz in bytes. The overall consumption of memory (i.e., ROM and RAM together) by Algorithms 4.1, 4.2, 4.3, and 4.4 were 1567, 2132, 3130, and 1124 bytes, correspondingly. This outcome shows that the consumption of memory by Algorithms 4.2 and 4.3 is

more than Algorithms 4.1 and 4.4 due because of the former algorithms include the processing, transmission, and reception of agent. Algorithms 4.2 and 4.4 run on resource-wealthy leader motes of the clusters. Therefore, the high consumption of memory may be achieved by such motes. The running of Algorithms 4.1 and 4.3, on member motes of the clusters, takes 1567 and 3130 bytes memory, correspondingly, that may be accommodated by MICAz memory subsystem. Moreover, the running of Algorithm 4.3 is infrequent; that is, it runs only when a doubtful cluster member mote is identified within the cluster and the leader mote of the cluster transmits an agent for on-the-spot confirmation of the doubtful cluster member mote after the consideration of the optimization of agent transmission procedure.

It is pertinent to note that the abnormality agent inhabits the memory of the cluster member mote for a limited amount of time. Per se, an agent is removed that clears engaged space of the memory as early as it sends on-the-spot confirmation outcome to the leader mote of the cluster. Thus, the effect of agent on the memory of the member mote of the cluster is for relatively short term.

The memory consumption by Algorithms 4.1, 4.2, and 4.3 was 2559, 1066, and 1206 bytes, correspondingly. This shows that the abnormality identification with the first-order joins (i.e., Algorithm 4.1) consumes more memory as compared to the abnormality identification on individual features (i.e., Algorithm 4.2). Similarly, 2-sigma method for optimization of agent transmission (i.e., Algorithm 4.2) consumes less memory as compared to the weighted-sum method (i.e., Algorithm 4.3). This indicates that although the weighted-sum method is more efficient as it causes less frequent abnormality agent transmission, it consumes more resources during execution of the algorithm. The memory consumption outcomes are given in Table 4.2 [1].

4.4.2.2 Processing Time

The processing times for Algorithms 4.1, 4.2, 4.3, and 4.4 were 6.93, 32.51, 7.65, and 4.01 ms, correspondingly. Algorithms 4.2 and 4.3 carry out the task of the identification of abnormality, on-the-spot confirmation, and their total spent time was 40.16 ms. Algorithms 4.2 and 4.3 consumed large amount of processing time in

Table 4.2 Memory consumption: © Academy Publisher, reprinted from Usman et al. [15]

Algorithm	RAM	Rom
4.1	29	1538
4.2	41	2091
4.3	59	3071
4.4	23	1101
5.1	50	2509
5.2	22	1044
5.3	24	1182

Table 4.3 Processing time and energy consumption: © Academy Publisher, reprinted from Usman et al. [15]

Algorithm	Processing time (ms)	Energy consumption (μJ)
4.1	6.93	55.60
4.2	32.51	3647.12
4.3	7.65	1570.96
4.4	4.01	39.73
5.1	37.25	4039.91
5.2	3.97	38.67
5.3	4.13	40.02

comparison to Algorithms 4.1 and 4.4 because of the immersion abnormality agent processing. This processing time result is in line with theoretical outcomes illustrated in Sect. 3.5.6.

The processing times for Algorithms 4.1, 4.2, and 4.3 were 37.25. 3.97, and 4.13 ms, correspondingly. This shows that the combined elapsed time for the first-order abnormality identification (i.e., Algorithm 4.1) and on-the-spot confirmation procedure (i.e., Algorithm 4.3) was 44.9 ms. If a cluster leader mote also employs the procedure of optimization of agent transmission, then the overall process consumes 48.87 ms time in the case of the 2-sigma method (i.e., Algorithm 4.2) and 49.03 ms in the case of the weighted-sum method (i.e., Algorithm 4.3), which are quite efficient. These results are shown in Table 4.3 [1].

4.4.2.3 Energy Consumption

The procedure of collection of features by the cluster member mote (i.e., Algorithm 4.1) consumed 55.60 μJ of energy in each iteration. Algorithms 4.2 and 4.4 consumed 3647.12 μJ and 39.73 μJ of energy, correspondingly. These algorithms, however, run on resource-wealthy cluster leader motes that can manage such a consumption of energy. The dissipation of energy by Algorithm 4.3 was 1570.96 μJ. Nevertheless, this energy is merely spent by doubtful cluster member mote for on the spot confirmation procedure. Similarly, Algorithms 4.1, 5.2, and 5.3 consumed 4039.91, 38.67, and 40.02 μJ energy. Algorithm 4.1 consumed higher amount of energy as it involves the transmission of the abnormality agent. This energy consumption is, however, manageable by resource-rich cluster leader motes. The energy consumption results are provided in Table 4.3 [1].

4.4.3 Comparative Study and Discussion

The abnormality identification and confirmation system is contrasted with three relatively latest associated techniques presented by Ketel [11], Eludiora and colleagues [12], and Khanum and colleagues [13]. As discussed in Sect. 2.2.2.5, these schemes have been chosen due to the similarity of the use of agent technology for abnormality/intrusion/attack identification in networks. It is pertinent to mention that the job of the abnormality confirmation agent in abnormality identification and confirmation system is to verify that the abnormalities are caused on the spot or in transition unlike the work presented by Wagner [14] which assumes a secure link among a sensor mote and BS. Therefore, no comparison of our work can be made with this secure aggregation technique. However, our abnormality identification and confirmation system can be used in conjunction with the Wagner's approach to achieve more reliable functionality of sensor networks. It is pertinent to note that the primary contribution of this study is not to improve the detection accuracy, but to identify abnormalities caused due to the different nature of faults and attacks, in addition to on-the-spot confirmation of doubtful motes and optimization of agent dispatch. However, a fuzzy logic-established cross-layer abnormality identification technique has been presented in Chap. 5 which increases the detection accuracy and that work has been compared with the related abnormality identification schemes (see Sect. 5.7).

The comparison has been made in the following six aspects: (i) role of the agent, (ii) nature of identified abnormalities, (iii) agents per mote, (iv) identification time complexity, (v) homogeneous or heterogeneous nature of sensor network, and (vi) agent dispatch optimization. In the system, the agents are mobile and solitary agent per mote is sent for on-the-spot confirmation of the doubtful conduct of a cluster member mote by using the resources of the mote. On the other hand, the scheme presented by Ketel [11] employs three agents, namely static, mobile, and nodal agents, for the abnormality identification procedure. The technique put forwarded by Eludiora et al. [12] employs an agent for the inter-BS regulatory interaction. Correspondingly, the work by Khanum and colleagues [13] employs two static agents, namely management and coordination, and an agent that is mobile and employed to perform the abnormality identification procedure. Both Khanum et al. [13] and Ketel [11] used three agents for abnormality identification. The usage of numerous agents not only surges the cost of computation, but it also needs added calculation for inter-agent interaction. The usage of numerous agents also surges the interaction and computation workload of a network.

An imperative variance among the system and other techniques is communication of an agent by a specific kind of mote. In the system, an agent is only sent by resource wide motes (i.e., by cluster leader motes) and obtained by cluster member motes in comparison to other techniques, where agents are sent by all motes. This method of other techniques can rapidly consume the resources of energy of member motes. In contrast, the member motes in the system obtain the agent only for on-the-spot confirmation procedure (i.e., not very frequently). This tactic puts the minimum

load on the resource-constrained member motes. Moreover, the system does not facilitate the movement of agent among cluster member motes (i.e., among non-cluster leader motes inside or outside). This policy successfully uses total resources of network without refuting the job of the agent in the abnormality identification and confirmation system.

The abnormalities identified in the studies carried out by Ketel [11] and Eludiora and colleagues [12] are DoS attack-based and mote abnormalities, correspondingly. The technique studied by Khanum et al [13] only identifies reading abnormalities. In contrast, the system can discover numerous natures of first-order abnormalities that are produced by denial-of-sleep threat, exhaustion of battery, and other attacks. The complexity of detection of the technique put forwarded by Ketel [11] cannot be computed, because it is a high-level technique and no implementation (or algorithmic) descriptions of the abnormality identification procedure are given. The abnormality identification complexity of Eludiora et al. [12] technique is $O(n^2)$ [12]. The technique by Khanum et al. [13] and system discovers abnormalities within $O(n)$ time. Moreover, the system is appropriate for networks with heterogeneous nature of motes.

Moreover, not any among existing techniques has solved the problem of optimization of agent transmission [11–13]. The system improves the procedure of agent transmission by taking into account the past and present occurrences of the abnormal observations. A comparative summary of the techniques is given in Table 4.4 [1].

Table 4.4 Comparison summary: © Academy Publisher, reprinted from Usman et al. [15]

Technique/system	Agent role	Abnormalities	Agents/mote	Complexity of detection
Ketel [11]	Abnormality information collection	Mote abnormalities through neighbor monitoring	3	Not applicable
Eludiora et al. [12]	Inter-BS control communication	Abnormalities caused by DoS attack	1	$O(n^2)$
Khanum et al. [13]	Local abnormality	Sensor reading abnormalities	3	$O(n)$
The system	On-the-spot confirmation	Denial-of-sleep and resource exhaustion attacks along with mote faults	1	$O(n)$

4.5 Summary

The algorithm, presented in this chapter, maximizes the use of synchronized resource management scheme-based observations and defines an association among features of interest to detect abnormalities occur due to on-the-spot attacks or faults, resource exhaustion attacks, attacks on the resources of motes, and denial-of-sleep attacks. This chapter has also addressed the third research question and satisfied the corresponding third requirement by proposing two methods, namely 2-sigma and weighted sum, for abnormality confirmation agent transmission optimization. The former method defines the curtailment regions on underlying probability distributions to curtail transmission of abnormality confirmation agents, whereas in the latter method, the cluster leader mote considers both the current and anomalous behaviors of the cluster member sensor motes in order to compute the abnormality confirmation agent transmission score and consequently transmitting abnormality confirmation agents to carry out the job of on-the-spot confirmation.

The performance of the algorithms has been thoroughly investigated in terms of a complexity analysis, formal modeling and analysis, a simulation study, implementation on a real mote, and a comparison study. The outcomes have indicated that the first-order abnormality identification algorithm can discover the occurrence of numerous first order abnormalities because of the denial-of-sleep attack, resource exhaustion attack, and errors on motes with the accuracy between 97.80 and 99.20%. Similarly, the 2-sigma and weighted-sum abnormality confirmation agent transmission optimization algorithms reduced the abnormality confirmation agent transmission overhead by as much as 29–34% and 42–52%, correspondingly, unlike related schemes which have not considered the optimization of agent transmission [11–13]. The implementation on a real mote advocated the suitability of the algorithms for low-resource sensor motes. Finally, the comparative study has demonstrated that the system can detect denial-of-sleep and resource exhaustion attacks and errors on motes with detection complexity $O(n)$, dissimilar to related techniques presented by Eludiora et al. [12] and Khanum et al. [13] which can detect DoS attack and sensor reading abnormalities with the detection complexity of $O(n_2)$ and $O(n)$, correspondingly.

A scheme for cross-layer abnormality identification and abnormality confirmation optimization has been introduced in the next chapter.

4.6 Bibliographic Notes

The notion of the first-order anomalies was elucidated in [1]. The concept of 2-sigma optimization was presented in [2]. The idea of weighted-sum optimization

method was introduced in [1]. The simulation study results and analyses were initially reported in [1].

References

1. M. Usman, V. Muthukkumarsamy, X.-W. Wu, A resource-efficient system for detection and verification of anomalies using mobile agents in wireless sensor networks. J. Netw. **9**(12), 3427–3444 (2014b)
2. M. Usman, V. Muthukkumarsamy, X.-W. Wu, S. Khanum, Wireless smart home sensor networks: mobile agent based anomaly detection, in *Proceedings of the 9th IEEE International Conference on Ubiquitous Intelligence Computing and 9th International Conference on Autonomic Trusted Computing (UIC/ATC)*, September 2012, pp. 322–329
3. A. Varga, The omnet++ discrete event simulation system, in *Proceeding of the European Simulation Multiconference* (2001), p. 7
4. J. Polastre, R. Szewczyk, C. Sharp, D. Culler, The mote revolution: low power wireless sensor network devices, in *Proceeding of the Hot Chips* (2004)
5. W.B. Hinzelman, A.P. Chandrakasan, H. Balakrishnan, An application-specic protocol architecture for wireless microsensor networks. IEEE Trans. Wirel. Commun. **1**(4), 660–670 (2002)
6. J.C. Dagher, M.W. Marcellin, M.A. Neifeld, A theory for maximizing the lifetime of sensor networks. IEEE Trans. Commun. **55**(2), 323–332 (2007)
7. Part 15.4: Wireless Medium Access (MAC) and 2006 Physical Layer (PHY) specications for low-rate wireless Personal Area Network (LR-WPANs), IEEE Std. 802.15.4
8. D.H. Stamatis, *Six Sigma and Beyond: Statistics and Probability*, vol. III (CRC Press, USA, 2002)
9. P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, D. Culler, Tinyos: an operating system for sensor networks, in *Ambient Intelligence*, ed. by W. Weber, J. Rabaey, E. Aarts (Springer, Berlin Heidelberg, 2005), pp. 115–148
10. D. Gay, P. Levis, R. Von Behren, M. Welsh, E. Brewer, D. Culler, The nesc language: a holistic approach to networked embedded systems, in *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation* (2003), pp. 1–11
11. M. Ketel, Applying the mobile agent paradigm to distributed intrusion detection in wireless sensor networks, in *40th Southeastern Symposium on System Theory*, March 2008, pp. 74–78
12. S.I. Eludiora, O.O. Abiona, A.O. Oluwatope, S.A. Bello, M.L. Sanni, D.O. Ayanda, C.E. Onime, E.R. Adagunodo, L.O. Kehinde. A distributed intrusion detection scheme for wireless sensor networks, in *IEEE International Conference on Electro/Information Technology (EIT)*, May 2011, pp. 1–5
13. S. Khanum, M. Usman, A. Alwabel, Mobile agent based hierarchical intrusion detection system in wireless sensor networks. Int. J. Comput. Sci. Issues (IJCSI) **9**(3), 101–108 (2012)
14. D. Wagner, Resilient aggregation in sensor networks, in *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks* (2004), pp. 78–87
15. M. Usman, V. Muthukkumarsamy, X.-W. Wu, Specification and validation of enhanced mobile agent-enabled anomaly detection in resource constrained networks. J. Netw. **10**(6), 353–368 (2015a)

Chapter 5
Cross-Layer Identification and Transmission of Agent Using Fuzzy Logic

5.1 Introduction

As highlighted in the preceding chapters, the motes and their transmitted data are susceptible to on-the-spot and in transmission abnormalities. An agent-enabled abnormality identification technique in such scenarios not only identifies abnormalities in a smart home, but also provides a mechanized service to confirm the origin of abnormalities prior to informing an administrator regarding the abnormalities. The community of researchers has also studied other roles of agents in the abnormality identification applications for sensor networks [1–3]. These roles include arbitrary specimen collection of network traffic and distribution of network management statistics. Previous studies, however, do not take into account the state of the connection among interaction motes prior to the dispatch of agents. A low-quality state of communication link might initiate faults in code or data of agent during communication that might eventually disturb its allocated functionality.

The past agent-enabled abnormality identification techniques have identified crisp data boundaries to illustrate the motes behavior [1–3]. This may trigger needless alarms in cases when received data values stay closer to the boundaries of the normal profile. For instance, take an example of the smart home situation, where a mote is nominated to detect and transmit the apartment temperature. The usual conduct of the mote is bounded with the [14 °C, 21 °C] closed interval. In this situation, any value near to 14 °C or 21 °C, such as 13.9 °C or 21.2 °C, will trigger a needless alarm for a user. Fuzzy logic may be helpful in such cases to set soft limits for related choice making [4]. An abnormality identification technique, that embodies the conduct of motes merely using fuzzy logic, is incapable to deliberate randomness of underlying data to define usual conduct of motes. Moreover, the existing agent-enabled abnormality identification techniques have also not taken into account the state of the communication link for both the abnormality identification and agent transmission [1–3].

M. Usman et al., *Mobile Agent-Based Anomaly Detection and Verification System for Smart Home Sensor Networks*,
https://doi.org/10.1007/978-981-10-7467-7_5

To tackle the above-cited shortcomings, this chapter has presented an agent-enabled cross-layer abnormality identification technique. The presented technique relies on statistical processes that take into account the stochastic inconsistency in the data to identify three zones, viz. usual, tolerance, and abnormal, on a feature space that is formed on cross-layer features. The usual zone describes usual conduct of a mote. The tolerance zone is identified to handle measurements that lie near to the usual zone. The technique reduces trust count of a mote if a measurement from mote stays inside the tolerance zone, and the administrator will merely be informed when the level of trust decreases under an already defined bound. The agent is sent for on-the-spot confirmation of the mote to confirm the origin of abnormalities, only if a measurement stays in the abnormal zone or trust value approaches the minor limit. The lenient bounds between the abnormal and tolerance zones and fuzzy logic-enabled rule-base are formulated to identify cross-layer abnormalities and to efficiently send agents. The presented technique is applied on mote-based test bed, and experiment outcomes show its capability to identify cross-layer abnormalities having higher accuracy and increase the life of network.

5.2 Network Model

The network is supposed to be a digraph that is described as $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, here $\mathbf{V}$ denotes vertices (i.e., motes) and $\mathbf{E}$ represents edges (i.e., interaction channels) in smart home. Motes $\mathbf{V} = \bigcup_{i=1}^{3} V_i$ form smart home, here V_1 is a top-level mote that could be a desktop or a laptop; it acts as a network chief, and interconnected with m resource-extensive cluster leader motes, i.e., $V_2 = v_1, v_2, ..., v_m$. The motes $V_3 = \bigcup_{j=1}^{m} V_{3j}$ form m clusters, here $V_{3j} = v_j, s_{j1}, s_{j2}, ..., s_{jk}$. The notation V_{3j} represents the j_{th} cluster inside a network, v_j denotes the cluster leader mote in that particular cluster, and k represents member motes within that cluster. The mote sets cardinality should hold the relation $|V_1| \leq |V_2| \leq |V_3|$ to create a hierarchical smart home sensor network, where V_1, V_2, and V_3 are the upper, middle, and leaf (lower) level motes, correspondingly.

The motes that belong to the mote types, namely V_2 and V_3 are IEEE 802.15.4-compliant MICAz sensors. V_2 kind of motes are resource-extensive, as they have supplementary memory and possess continued supply of power. In contrast, the motes, viz. V_3 have minimal battery and memory assets. These motes are positioned to sense their vicinity, save the observations and status of battery in memory for on the spot confirmation procedure, and then send those observations to the associated V_2 type mote. The V_2 kind of motes identify cross-layer abnormalities on the obtained packets and send agents for on-the-spot confirmation of the mote after taking into account the state of the communication link.

5.3 Cross-Layer Abnormality Identification Module Architecture

Every V_2 type mote is armed with a cross-layer abnormality identification module that identifies cross-layer abnormalities and also does the job of transmission of agent after taking into account the state of the communication channel. Cross-layer abnormality identification module has three elements, viz. expert (cross-layer) system , controlling unit, and mobile agent, as shown in Fig. 5.1 [5].

Controlling unit works as a controller within intra-module and among inter-module elements to support the abnormality identification and agent transmission procedures. Controlling unit gets packets from V_3 motes and forwards those packets to an element, namely cross-layer expert system, that carries out the jobs of abnormality identification and agent transmission, and transmits back the outcome of controlling unit. The typical sensed data, that is, without abnormality, is sent to aggregation unit, that saves it for a predetermined amount of time before sending it to V_1 mote for the next necessary action. If the measurement is found abnormal, controlling unit transmits an agent to perform on-the-spot confirmation of V_3 mote to discover the origin of the abnormalities.

The agent employs received values of past packets to perform the job of on-the-spot confirmation on V_3 mote. The agent then carries out a comparison among saved data of the sensed readings and battery status with that of data carried by agent to execute the job of on-the-spot confirmation. If data is matched, then V_3 mote is assumed to be abnormality free. Else, the abnormal position of the mote is commu-

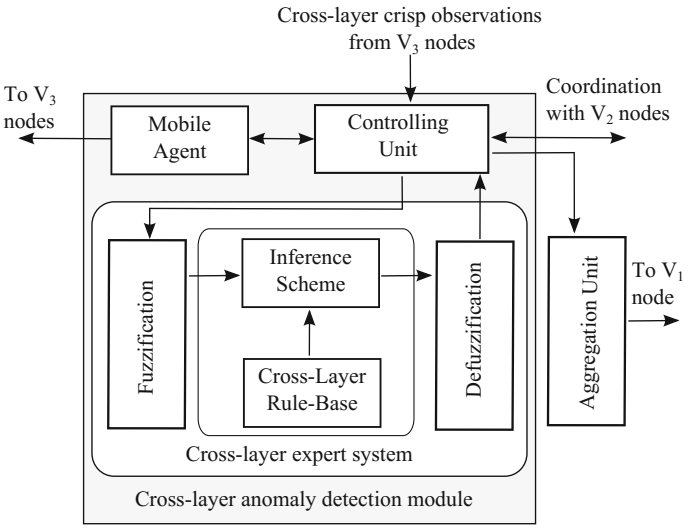

Fig. 5.1 Cross-layer abnormality identification module architecture: © 2015 IEEE. Reprinted, with permission, from Usman et al. [5]

nicated to the associated V_2 mote. For more description of on-the-spot confirmation procedure, readers are referred to Sect. 3.5.3.

A fuzzy scheme fuzzifies data of crisp nature into corresponding fuzzy values and computes them by employing a rules set to get fuzzy nature of data [6]. Then fuzzy data is converted back to get a single (crisp) outcome that initiates the already defined action execution. The expert system then gets cross-layer (crisp) features data from coordinator and fuzzifies data by employing functions, namely membership functions discussed in Sect. 5.4.2. The fuzzified data is then employed for the decision making regarding agent transmission and cross-layer abnormality identification, as illustrated in Sect. 5.4.3. Lastly, the unit, viz. defuzzification defuzzifies the output by using a method, namely maximum, that is, by choosing the data value that has the highest membership value.

5.4 The Proposed Scheme

This section first discusses cross-layer feature set which is employed for abnormality identification and abnormality confirmation agent transmission decision making. This is followed by an elucidation of the regions computation method. Finally, the initial arrangement of the cross-layer rule-base is elucidated in detail.

5.4.1 Cross-Layer Feature Set

The conduct of V_3, that is, IEEE 802.15.4-acquiescent MICAz, mote is formalized by features of mote and communication link. The values of the mote features are sent by V_3 motes to their associated V_2 motes. Other features comprises of Battery Status (BS) and Sensor Reading (SR). The SR might comprise of observations of motion, temperature, and pressure detection sensors.

The state of communication channel is formalized on the basis of three communication link features, viz. Received Signal Strength Indicator (RSSI), Packet Error Rate (PER), and Link Quality Indicator (LQI) for making decisions on agent transmission and abnormality identification. The communication link features' values are obtained by V_2 mote from obtained data of V_3 mote. The (MICAz) mote works on 2.4 GHZ radio frequency [7]. Its data rate is 250 Kbps, and range of adaptive transmission power is $-25 \sim 0$ dBm. The transceiver chip (CC2420 RF) calculates the values of average correlation (CORR) and RSSI of every obtained data packet to calculate the value of LQI feature. The CORR value provides the raw information of communication link inside the [8, 9] closed interval, where 50 denotes the worst and 110 represents the best case values. This research has treated LQI = CORR to calculate the value of LQI as suggested by Tang et al. [7]. This indicates that the motes necessitate not to carry out any added calculation in order to calculate the RSSI and LQI values to make agent transmission and abnormality identification decisions.

The PER values are imperative for the accurate execution of on-the-spot confirmation procedure, because the SR and BS values acquired from received packets are employed by the confirmation procedure. The ignorance of errors in obtained packet may result in incorrect outcome of on the spot confirmation procedure. Hence, only those data packets that clear the 16-bit CRC are assumed to be collected [7]. The value of PER is calculated as count of collected packets over entire sent packets.

5.4.2 Regions Computation

The restricted energy resource of motes requires cautious transmission of agents to perform the function of on-the-spot confirmation. Therefore, the cross-layer abnormality identification module divides the feature space of each feature of V_3 motes into three zones, viz. usual, tolerance, and abnormal. The usual zone sets the usual conduct of motes. If values of features of collected packets are not inside the usual zone, however, in its vicinity, then it might not be adequate to instantly send an agent to perform on-the-spot confirmation procedure because of energy expensive transmission operation [10]. The abnormality identification module treats this zone as a tolerance zone and decreases the value of trust of V_3 mote after getting the packets having measurements in tolerance zone. An agent is sent when V_2 mote loses trust on V_3 mote to some extent. Packets that have measurements outer than tolerance zone are considered as abnormal by the abnormality identification module, and an agent is instantly sent to perform its allocated job. The procedure for zones computation is illustrated below.

Formally, consider X be a Universe of Discourse (UoD), denoting feature space of one V_3 mote feature, here $X = \{N, T, A\}$. The (fuzzy) numbers, namely N, T, and A represent usual, tolerance, and abnormal zones, correspondingly in X UoD. The fuzzy number domains are outlined below.

$N = [c^*, d^*]$

$T = [a^*, c^*] \bigcup [d^*, f^*]$

$A = [-\infty, b^*] \bigcup [e^*, +\infty]$

In the above definitions, $a^* = a \pm s^l$, $b^* = b \pm A_r^l$, $c^* = c \pm (s/\sqrt{n})^l$, $d^* = d \pm (s/\sqrt{n})^r$, $e^* = e \pm A_r^r$, and $f^* = f \pm s^r$, and the parameters a^* to f^* must follow the association $f^* \geq e^* \geq d^* \geq c^* \geq b^* \geq a^*$ to outline the fuzzy numbers domains. The notation s denotes the standard deviation of n trial measurements that are employed to calculate the zones. The symbol A_r represents the abnormal zone limit. The superscripts r and l do not show the power, in fact these are right-side and left-side parameters values on horizontal axis.The left parameter number is computed by the subtraction operation, whereas the right parameter number is calculated by the addition operation of the statistic (computed by a statistical process) from or to the value of mean. The a to f variables are custom tuning variables that are employed to tune parameter values to update the calculated zones. The adjustment variable values are not dependent on the parameter values that are computed using statistical methods.

The limit values of domains are computed by performing the statistical calculations on n sampled measurements. The usual zone that is outlined by N is calculated using standard deviation of the mean value of n measurements, i.e., $s/\sqrt{n}$. Then the left and right sides of the mean ($\overline{x}$) along the x-axis are bounded by the values $c^* = c \pm (s/\sqrt{n})^l$ and $d^* = d \pm (s/\sqrt{n})^r$, respectively, to outline the usual conduct of V_3 mote.

Correspondingly, the limits of the tolerance zone, outlined by T, are calculated by computation of s on n measurements. On the basis of this calculation, the limits $[d^* = d \pm (s/\sqrt{n})^r,\ f^* = f \pm s^r]$ and $[a^* = a \pm s^l,\ c^* = c \pm (s/\sqrt{n})^l]$ outline the right and left tolerance zones, correspondingly. Lastly, the abnormal region is computed using the following relation.

$$A_r^l = (s/\sqrt{n})^l + s^l/2 \tag{5.1}$$

$$A_r^r = (s/\sqrt{n})^r + s^r/2 \tag{5.2}$$

Above relations outline the upper limit abnormal zone and lower limit of right abnormal zones, correspondingly. The limits $[e^* = e \pm A_r^r, +\infty)$ and $(-\infty, b^* = b \pm A_r^l]$ outline the right and left abnormal zones domains, correspondingly, on the basis of above-cited calculations, as illustrated in Fig. 5.2 [5].

The membership function for N can be computed as

$$M_N(x) = \begin{cases} 1, & c^* < x < d^* \\ 0, & x \le c^*, x \le d^* \end{cases} \tag{5.3}$$

Similarly, the membership function for T is defined as

$$M_T(x) = \begin{cases} 1, & x = c^*, x = d^* \\ (x - a^*)/(c^* - a^*), & a^* \le x \le c^* \\ (f^* - x)/(f^* - d^*), & d^* \le x \le f^* \\ 0, & x < a^*, x > f^*, c^* < x < d^* \end{cases} \tag{5.4}$$

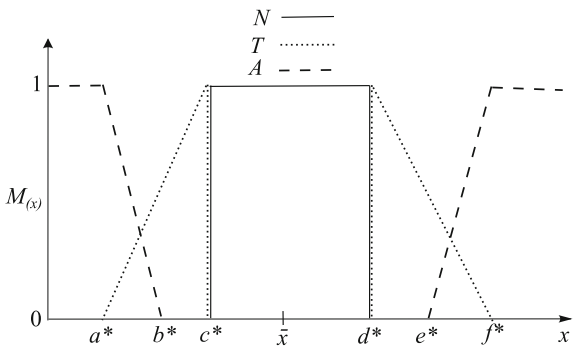

Fig. 5.2 Membership functions: © 2015 IEEE. Reprinted, with permission, from Usman et al. [5]

Lastly, the membership function for A is calculated by

$$M_A(x) = \begin{cases} 1, & x < a^*, x > f^* \\ (b^* - x)/(b^* - a^*), & a^* \leq x \leq b^* \\ (x - e^*)/(f^* - e^*), & e^* \leq x \leq f^* \\ 0, & b^* < x < e^* \end{cases} \tag{5.5}$$

The N, T, and A fuzzy number membership functions are illustrated in Fig. 5.2. The N is based on membership function with crisp values. Note that this is a distinct situation of membership functions. The design characteristics are selected to enable the system to decrease trust count of V_3 mote immediately after falling of cross-layer measurements in zones computed by T, though values are nearer to frontier of N. An example is provided below to show the calculation of parameter values and subsequently outline the zones.

Example 1 Let, for instance, a situation where a V_3 mote senses its vicinity and intelligence the readings of temperature to corresponding V_2 mote. Consider X be a UoD for temperature readings, $n = 51$, $\overline{x} = 21.11$, $s = 3.47$, and a to $f = 0$. This infers $s/\sqrt{n} = 0.47$. Therefore, the usual zone can outlined as $[c_* = c \pm (s/\sqrt{n})^l = 19.62, d^* = d \pm (s/\sqrt{n})^r = 20.58]$. Next, the tolerance zones are demarked as $[a^* = a \pm s^l = 16.71, c^* = c \pm (s/\sqrt{n})^l] = 19.62]$ and $[d^* = d \pm (s/\sqrt{n})^r = 20.58, f^* = f \pm s^r = 23.49]$. Lastly, the abnormal zones are calculated as $(-\infty, b^* = b \pm A_r^l = 18.17]$ and $[e^* = e \pm A_r^r = 22.03, \infty)$. Tuning variable values (i.e., a to f) are fixed as 0. However, an administrator can set numbers to recompute the calculated zones in practice. Moreover, membership values could be allocated by employing relations (5.3) to (5.5).

5.4.3 Cross-Layer Rule-Base

The system rule-base has cross-layer features' rules that execute the obtained network traffic for decision making about agent transmission and abnormality identification. The rule-base is comprises of *IF antecedent(s), THEN consequent(s)* rules, here antecedents are based on five input linguistic (i.e., cross-layer features) variables, viz. Battery Status (BS), Sensor Reading (SR), Packet Error Rate (PER), Received Signal Strength Indicator ($RSSI$), and Link Quality Indicator (LQI). These variables are associated with a logical operator: AND. Every input linguistic variable comprises of three values, namely A, T, and N, in its term-set. It is important to observe that the term-set scale may be adjusted according to the choice of user to adjust the system performance.

The resultant (i.e., the output linguistic variable represented by D) has three values, viz. D_1, D_2, and D_3, where D_1 represents the outcome of the accumulated data for situation where obtained data is usual, D_2 decrements the value of trust, and finally, D_3 transmits the agent to carry out on-the-spot confirmation. The D_1, D_2, and D_3 decisions are based on triangular-shaped functions that are defined by three values

Table 5.1 Cross-layer rule-base: © 2015 IEEE. Reprinted, with permission, from Usman et al. [5]

Rule no.	SR	BS	LQI	RSSI	PER	D
1	N_{SR}	N_{BS}	N_{LQI}	N_{RSSI}	N_{PER}	D_1
2	T_{SR}	N_{BS}	N_{LQI}	N_{RSSI}	N_{PER}	D_2
3	A_{SR}	N_{BS}	N_{LQI}	N_{RSSI}	N_{PER}	D_3
4	N_{SR}	T_{BS}	N_{LQI}	N_{RSSI}	N_{PER}	D_2
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
242	T_{SR}	A_{BS}	A_{LQI}	A_{RSSI}	A_{PER}	D_3
243	A_{SR}	A_{BS}	A_{LQI}	A_{RSSI}	A_{PER}	D_3

(t_r, t_m, t_l), where t_r, t_m, and t_l are the right, middle, and left values on horizontal axis. The decision variable parameters have following values: $D_1 = (0, 0.2, 0.4)$, $D_2 = (0.3, 0.5, 0.7)$, and $D_3 = (0.6, 0.8, 1)$. A key design aspect of rule-base is rules perform the D_3 action in the case when merely solitary feature value is abnormal. The initial rule syntax, as an example, is shown below.

$(SR = N_{SR}) \wedge (BS = N_{BS}) \wedge (LQI = N_{LQI}) \wedge (RSSI = N_{RSSI}) \wedge (PER = N_{PER}) \rightarrow D = D_1$

Conceptually, in the ancestor segment of the above-cited rule, the N_{SR}, N_{BS}, N_{LQI}, N_{RSSI}, and N_{PER} are the values possessed by input linguistic variables, namely SR, BS, LQI, $RSSI$, and PER, correspondingly. Consequent resultant segment facilitates accumulation of data. Input linguistic variables are 5 in presented techniques and every variable could possess 3 values. Therefore, entire rules, with every likely combination, are 243. The generic rule-base structure is provided in Table 5.1 [5].

5.5 Algorithm and Analysis

The algorithm executes on V_2 motes, that are resource rich, after receiving the network traffic from V_3 motes. The algorithm comprises of two procedures, viz. Initialization and Main. The former is accountable for the calculation of the zones. It is initially run at the system deployment time and thereafter runs only if user aims to recalculate the zones and rule-base update. It takes the sampled measurement values and value of n variable as input to calculate $\overline{x}$, $(s/\sqrt{n})^l$, $(s/\sqrt{n})^r$, s^l, s^r, A_r^l, and A_r^r parameters. The zone is then determined using these numbers, and membership functions are computed by using relations (6.3) to (6.5). A user-set empirical rule-base is produced next to running initial phase.

The later procedure carries out the tasks of agent transmission and abnormality identification by using the rule-base. The phase runs after obtaining each packet value from V_3 motes. In this stage, the abnormality identification module fetches the

crisp values of features, viz. BS, SR, $RSSI$, and LQI, from the obtained packets and also calculates the PER value. Then, these numbers are fuzzified by employing membership functions that are outlined in (6.3) to (6.5), and further executed by the rule-base. Next, the process of defuzzification of (decision) variable is carried out by employing the maximum method, to perform actions, viz. accumulation of reading, decrease in the value of trust of V_3 mote, or agent transmission. The procedure pseudocode is described Algorithm 5.1 [5].

Algorithm 5.1 Cross-Layer Abnormality Identification and Agent Transmission

Initialization Procedure
Require: n sampled measurements, n value
Ensure: Membership functions
1: **for** SR, BS, LQI, $RSSI$, PER **do**
2: Calculate: Eset = { $E(1)$, $E(2)$, $E(3)$, $E(4)$, $E(5)$, $E(6)$, $E(7)$} {//$E(1) = \overline{x}$, $E(2) = s^l$, $E(3) = s^r$, $E(4) = (s/\sqrt{n})^l$, $E(5) = (s/\sqrt{n})^r$ $E(6) = A_r^l$, $E(7) = A_r^r$}
3: $EstReg(Eset)$ {//Compute zones for every feature}
4: $ConstructMeb\ M_N(x)$, $M_T(x)$, $M_A(x)$ {//Build membership functions of every feature}
5: **end for**

Main procedure
Require: $DatPkt$, $RlBs$ {//data packet and cross-layer rule-base}
Ensure: Accumulate SR and save $SR\ BS\ LQI$, $RSSI$, PER, $NewTrust$, or transmit MA
1: **for** each $DatPkt$ **do**
2: $GetVal(SR, BS\ LQI, RSSI, PER)$
3: Fuzzify: $Fuzzset = \{fuzz(SR), fuzz(BS), fuzz(LQI), fuzz(RSSI), fuzz(PER)\}$ {//using (6.3) to (6.5) for every feature}
4: **for** $Fuzzset = \{fuzz(SR), fuzz(BS), fuzz(LQI), fuzz(RSSI), fuzz(PER)\}$ **do**
5: $EvalRlBs(Fuzzset)$ {//Evaluate rule-base}
6: **end for**
7: $DefuzzDes(D_1, D_2, D_3)$ {// Defuzzify decision}
8: **if** $D == D_1$ **then** {//checking decision}
9: $Agg(SR) \wedge Str(SR, BS, LQI, RSSI, PER)$ {//accumulate SR and save values for every feature}
10: **else if** $D == D_2$ **then** {//checking decision}
11: $DecrTrst(T_r)$ {Decrease trust value}
12: **else**
13: $TrnsmtMA$ {Send agent}
14: **end if**
15: **end for**

5.5.1 Complexity Analysis

This part of the chapter provides the space and time complexities of the algorithm.

Theorem 5.1 *(i) The space complexity for Procedure 1 of the algorithm is* $nl[i]$ *and (ii) Procedure is* $nl[j]$.

Proof (i) V_2 mote takes constant memory spaces C_{33} to C_{39} for storing $E(1)$ to $E(7)$ values, respectively, for a cross-layer feature. Similarly, the V_2 type mote takes memory space C_{40} to store the values of an estimated region and C_{41} space to store the values of a constructed membership function of the feature. Considering $l[i]$ as the total memory space taken by a single cross-layer feature to execute Phase 1 (i.e., Initialization Procedure) of Algorithm 5.1, the memory space taken by n features is $nl[i]$.

(ii) The V_2 type mote takes constant space C_{42} to store the value of a cross-layer feature to perform an abnormality identification and abnormality confirmation agent transmission decision. The V_2 type mote takes memory spaces C_{43}, C_{44}, C_{45} to store a fuzzified value, relevant rules in the rule-base, and a dufuzzified decision of a cross-layer feature. Let $l[j_a] = C_{42}, C_{31}, C_{44}, C_{45}$ be the memory space taken by a single cross-layer feature, then the total memory space taken by n features is $nl[j_a]$. The V_2 type mote further takes C_2, C_4, C_5, and C_6 memory spaces to store aggregate sensed data, trust values, and code and data of abnormality confirmation agent, respectively. Let $l[j_b] = C_2, C_4, C_5$, and C_6. Thus, considering $nl[j] = l[j_a] + l[j_b]$, the memory space taken by Phase 2 (i.e., Main procedure) of Algorithm 5.1 is $nl[j]$.

Theorem 5.2 *The cost of computation of (i) Procedure 1 is $O(n)$ and (ii) Procedure 2 is $O(n2)$.*

Proof (i) The procedure of calculating the statistical parameter values, zone computation, and calculation of membership functions consumes constant time for every job for n features. Therefore, taking $n(1 + 1 + 1)$ as overall complexity, the cost is $O(n)$.

(ii) Procedure 2 consumes constant time in order to carry out every jobs, viz. receiving cross-layer feature values from obtained packets, defuzzification, fuzzification, and decision making on n number of features jobs. Finally, time n consumed by Procedure 2 to execute n rules. Therefore, the computation complexity is $O(n^2)$.

Observe that $O(n^2)$ represents a larger complexity than $O(n)$ in the proof of Theorem 5.1 due to the reason that the Procedure 2 involves the execution of rule-base, that is a processing intensive task in comparison with other jobs within the algorithm.

5.6 Formal Modeling and Analysis

The algorithmic specifications of the proposed cross-layer abnormality identification and abnormality confirmation agent transmission algorithm are first formally defined in this section. The functionality of the algorithm is then formally characterized and analyzed. **Cross-layer model**: The unified Petri net model, (PN_c), is a 5-tuple net: $PN_c = (P_c, T_c, F_c, W_c, (M_c)_0)$, where $P_c = p_{42}, p_{42}, ..., p_{51}$ and $T_c = t_{52}, t_{53}, ..., t_{67}$ are non-empty, finite, and disjoint sets of places and transitions, respectively. $F_c = (p_{42}, t_{52}), (p_{42}, t_{53}), (p_{42}, t_{54}), (p_{42}, t_{55}), (p_{42}, t_{56}), (p_{42}, t_{57}), (p_{52}, t_{43}), (p_{53},$

t_{44}), (p_{54}, t_{45}), (p_{55}, t_{46}), (p_{56}, t_4), (p_{57}, t_{48}), (p_{43}, t_{58}), (p_{44}, t_{59}), (p_{45}, t_{60}), (p_{46}, t_{61}), (p_{47}, t_{62}), (p_{48}, t_{63}), (t_{58}, p_{49}), (t_{59}, p_{49}), (t_{60}, p_{49}), (t_{61}, p_{49}), (t_{62}, p_{49}), (t_{63}, p_{49}), (p_{49}, t_{64}), (t_{64}, p_{50}), (p_{50}, t_{65}), (t_{65}, p_{51}), (p_{51}, t_{66}), (p_{51}, t_{67}). The weight for all arcs is 1 except the weight of the arc (p_{50}, t_{65}) which is determined through the execution of the function g_6, that is, $W_c((p_{50}, t_{65}) = g_6$. Finally, $(M_c)0 = p_{42}$.

The above definition formally defines the cross-layer abnormality identification and abnormality confirmation agent transmission method, and the corresponding formal model is depicted in Fig. 5.3. The reachability tree for Petri net model PNc is elucidated in Appendix C. The place p_{42} possess solitary token while receiving a data packet, $DatPkt$, from the V_3 type mote, which enables transitions t_{52} to t_{57} to fire the token. The transitions t_{52} to t_{57} model the process of obtaining the cross-layer feature values from obtained $DatPkt$. The firing of the token from the place p_{42} yields one token in each of the places p_{43} to p_{48} and enables transitions t_{58} to t_{63} which model the fuzzification process of cross-layer features. The firing of tokens from the places p_{43} to p_{48} then yields six token at the place p_{49}. The six tokens merge into a single token, which flows through the rest of the model to formulate the workflow of the algorithm. The merged token in the place p_{49} enables the transition t_{64} which denotes the execution of the rule-base. The fine-grained mapping of the

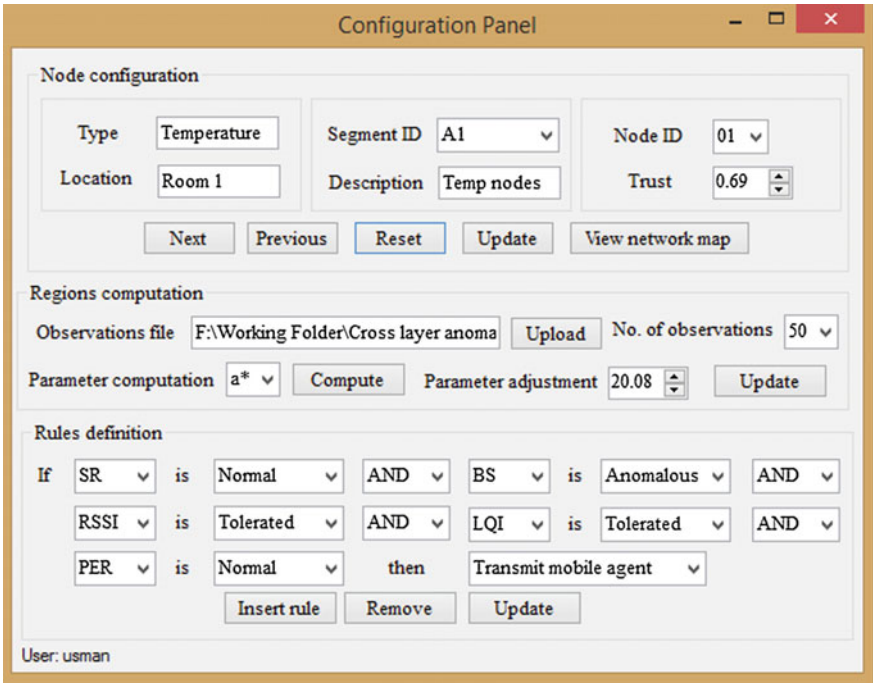

Fig. 5.3 Configuration panel GUI: © 2015 IEEE. Reprinted, with permission, from Usman et al. [5]

cross-layer rule-base is carried out in the Petri net model depicted in the dotted line inset in Fig. 5.3, where transitions t_{64}^{1} t_{64}^{243} represent rules in the rule-base.

The firing of the token from the place p_{49} yields a token in the place p_{50} which enables the transition t_{65}, denoting the process of the defuzzification of the decision. The firing of the token from the place p_{50} is, however, determined by the execution of the condition g_6, which models the decision defuzzification process along with the transition t65. The decision D_1 or D_2 yields single token into the final place, that is p_{51}. In contrast, in the case of abnormal values of features, the outcome "0" of the condition g_6 will assign "0" weight to arc (p_{50}, t_{65}). This means the abnormal state is not reachable in the cross-layer formal model. The condition g_6 is shown below.

$$g_6(x) = \begin{cases} 1, & D = D_1 \text{ or } D_2 \\ 0, & D = D_3 \end{cases} \tag{5.6}$$

On the basis of the above definition and discussion, the following results can be derived.

Theorem 5.3 *The PN_c is 6-bounded.*

Proof For all $p \in PN_c$, $M(p) = 1$. The firing of the token from the place p_{42}, however, yields one token each in places p_{43} to p_{48}. This means PN_c has six tokens at a time. Though those tokens merge at the place p_{49} to constitute the rest of the workflow of the PN_c, the full number of obtained tokens in PN_c is 6. Thus, PN_c is 6-bounded.

Theorem 5.4 *Transitions in PN_c are live at level 4 other than the t_{65} transition that is provisionally live at 0 level.*

Proof For all $t \in T$, the liveness level is 4 other than t_{65}, which is live at 0 level iff the condition g_6 output is 6 = 1. This implies that PN_c goes into the deadlock state only when abnormalities are found in the cross-layer features. This result is valid till the weighted arc that are based on condition exists, that is (g_6, t_{65}), which inputs the transition which models the defuzzification process after performing the cross-layer abnormality identification process.

5.7 Performance Evaluation

The performance is assessed with regards consumption of memory and energy, detection accuracy, and estimation of processing time.

The algorithms are programmed on two MICAz and solitary laptop mote-based test bed, denoting a minimum functional smart home as POC, that is, proof of concept. A mote was positions as V_3 mote that was answerable to sense its vicinity and sending the readings of temperature to V_2 mote. The V_2 mote was accountable for abnormality identification, accumulation and then communication of accumulated

data to V_1 mote, and transmission of agent to V_3 mote for on-the-spot confirmation procedure. An application was programmed to control the abnormality identification system.

The application has five working layers. The lower most layer (i.e., layer 5) carries out main tasks such as vicinity temperature sensing and on-the-spot confirmation on V_3 mote, the agent transmission and abnormality identification on V_2 mote, and zones calculation and zone update on V_1 mote. The next layer, that is, layer 4 takes care of storage job through the network. On V_3 mote, it saved status of battery and sensed readings, that are employed by agent for on-the-spot confirmation procedure. Layer 4 also preserves the accumulated data, rule-base, and trust count on V_2 mote. In the end, on V_1 mote, the abnormality identification outcomes that are sent by V_2 mote and an update regarding mote identities are also saved by layer 4.

Layer 3 handles communication interfaces of motes and carries out the following important jobs: (i) the communication of sensor reading from V_3 to V_2 motes, (ii) the abnormality identification message and accumulated data communication from V_2 to V_1 motes, (iii) the communication of agents for on-the-spot confirmation from V_2 to V_3 motes, and (iv) finally, the communication of on-the-spot confirmation result from V_3 to V_2 motes. The upper layer, that is, layer 2 fetches the data from collected packets and forwards them to the next upper layer, that is, layer 5, to carry out its designated tasks. Layer 2 also generates data packets and hands over them to next upper layer, that is, layer 3 for packet transmission. In the end, the topmost layer, that is, layer 1 offers GUIs to handle and supervise the abnormality identification system.

Software application has two modules, viz. configuration and report panels. Observe that because of the modular method, the accessible choices of parameter selection on module graphical user interfaces (GUIs) could be changed or new modules may be integrated according to user choice. Configuration panel GUI is based on three constituents, viz. (i) mote configuration, (ii) regions (zones) computation, and (iii) rules definition. Initial constituent is accountable for identifying class of motes, namely pressure, temperature, and motion, and also setting motes' identities and their locations in smart home. The constituent provides a facility to increment or decrement the trust level for a certain mote. It empowers users to look a house plan to identify the position of motes in the network. The second constituent facilitates user to calculate or tune the parameter values to define zones. The third and final constituent provides a facility to set fuzzy rules. GUI module, namely configuration panel, is depicted in Fig. 5.3 [5].

Report Panel module offers a service to retrieve an abnormalities report. The report could be produced in regard to the network component identity, identity of room, and type of mote for particular duration of time and date. Three abnormality observations, in the inverse sequential order, are shown inside the window of report panel. Report may be watched through Detailed View option button. Report panel GUI is depicted in Fig. 5.4 [5].

The network traffic of 1000 rounds transmitted from V_3 to V_2 mote was sampled to calculate zones and configuring expert systems for experiments. The BS, LQI, SR, and RSSI feature values of every packet were stored, whereas the PER values

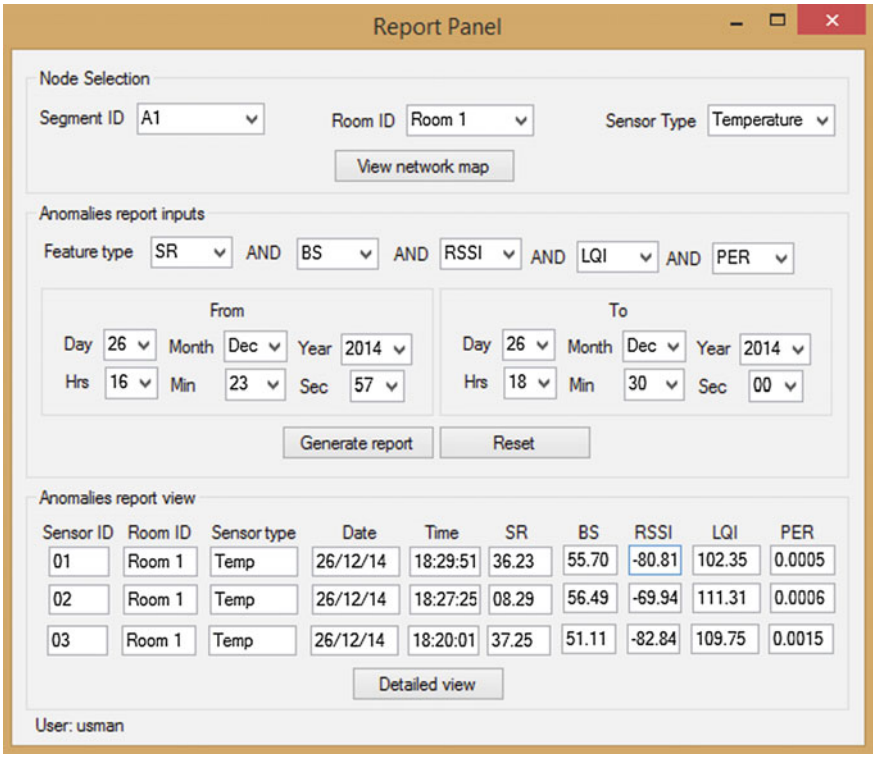

Fig. 5.4 Report panel GUI: © 2015 IEEE. Reprinted, with permission, from Usman et al. [5]

were calculated for each five packets. The measurements of link and mote features are shown in Figs. 5.5 and 5.6 [5], correspondingly, and statistics of measurements are plotted in Table 5.2 [5].

The n parameter value was fixed as 50. As a result, the values shown in Table 5.3 [5] were acquired to set the zones for trials. The 10% arbitrarily produced abnormal traffic was integrated in data. Rule-base was formed by employing configuration panel. The decrease in the trust count was fixed as 0.33 for measurements in the tolerance zone. The agent was communicated in only those cases where the value of trust was 0. The value of trust was reset as 1 every time it touched 0 for the sake of trials. In practice, though, the algorithm must transmit an alarm as early as the value of trust value will touch the lower limit.

The trials were also carried out with a well-known algorithm, namely decision tree to offer the baseline results. The decision tree detection accuracies were 98.8% for SR, 98.5% for BS, 98.7% for LQI, 98.4% for $RSSI$, and 98.7% for PER. In contrast, the accuracy was stable at around 100% for the presented algorithms. These results are depicted in Fig. 5.7 [5]. The presented algorithm, however, needs the domain knowledge to adequately define rule-base to identify abnormalities with higher accuracy.

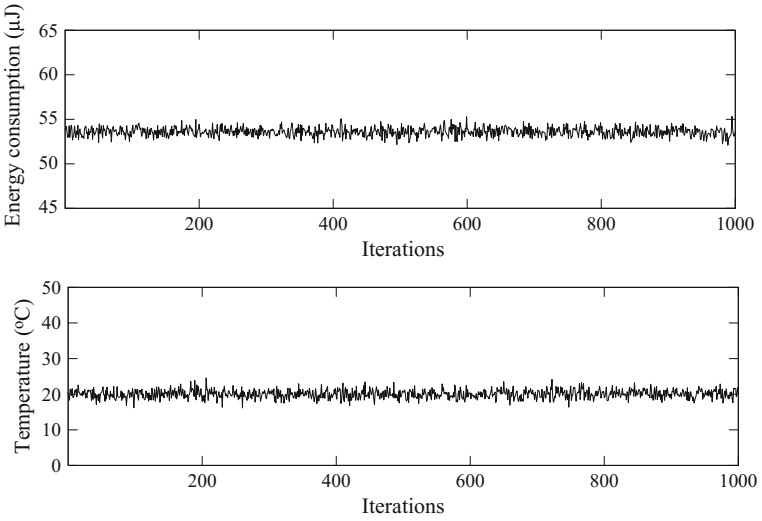

Fig. 5.5 Mote features: © 2015 IEEE. Reprinted, with permission, from Usman et al. [5]

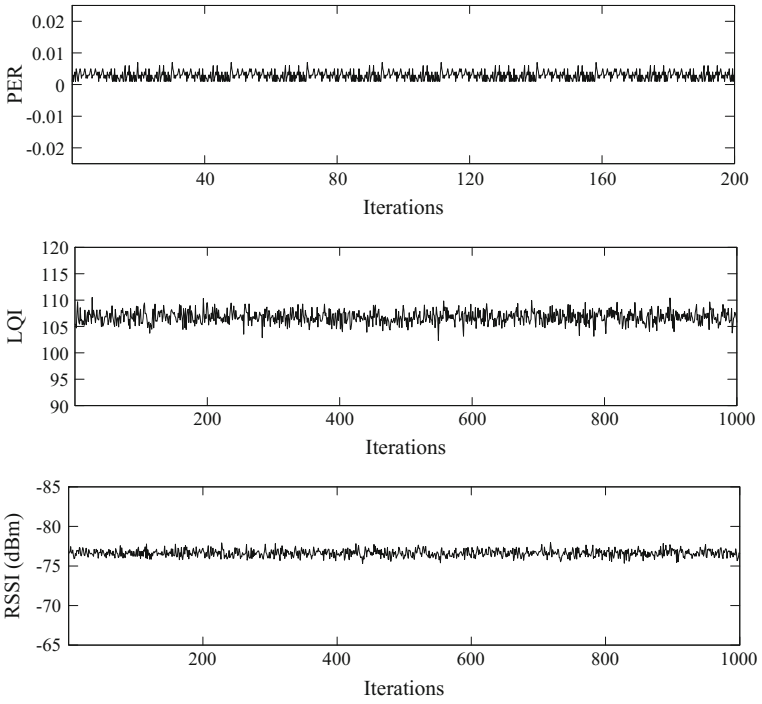

Fig. 5.6 Features of link: © 2015 IEEE. Reprinted, with permission, from Usman et al. [5]

Table 5.2 Sampled cross-layer data statistics: © 2015 IEEE. Reprinted, with permission, from Usman et al. [5]

Feature category	Feature	Mean	Standard deviation
mote	SR	20.08	1.43
mote	BS	53.70	0.50
Link	$RSSI$	−76.65	0.33
Link	LQI	106.75	1.28
Link	PER	0.0033	0.001

Table 5.3 Cross-layer expert system parameters: © 2015 IEEE. Reprinted, with permission, from Usman et al. [5]

Features	a^*	b^*	c^*	d^*	e^*	f^*
SR	18.65	19.26	19.88	20.28	20.90	21.51
BS	53.20	53.41	53.63	53.77	53.99	54.20
$RSSI$	−76.98	−76.83	−76.70	−76.60	−76.46	−76.32
LQI	105.47	106.02	106.57	106.93	107.48	108.03
PER	0.0020	0.0024	0.0029	0.0031	0.0036	0.0040

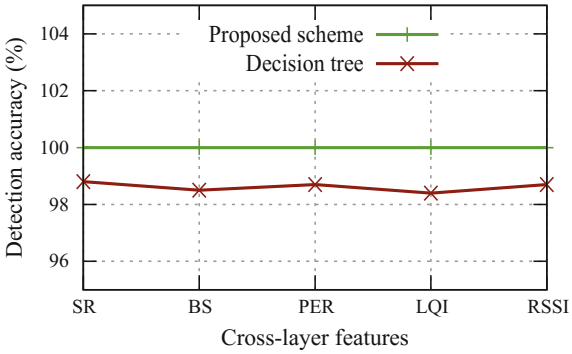

Fig. 5.7 Detection accuracy: © 2015 IEEE. Reprinted, with permission, from Usman et al. [5]

The trials were conducted by using cross-layer technique (i.e., the presented technique) and by without using cross-layer technique (i.e., the prevailing techniques) for the consumption of energy estimation by agent transmission procedure [1–3]. For the existing techniques (i.e., without taking into account the state of the communication for the decision of agent transmission), the expenditure of energy for the transmission of agent was among 3764.32 J for 200 to 18821.60 J for 1000 packets. In contrast, for the presented technique case, the dissipation of energy was among 1613.28 J for 200 to 10217.44 J for 1000 packets, as given in Fig. 5.8 [5]. The outcomes of trials show the presented technique could save 42.85 to 54.29% J energy in comparison to prevalent techniques that do not take into account state of communication channel prior to agents dispatch.

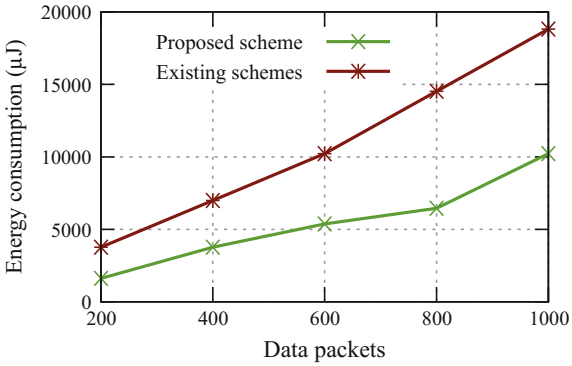

Fig. 5.8 Consumption of energy: © 2015 IEEE. Reprinted, with permission, from Usman et al. [5]

Table 5.4 Memory, processing, time, and consumption of energy outcomes: © 2015 IEEE. Reprinted, with permission, from Usman et al. [5]

Procedure	RAM (bytes)	ROM (bytes)	Processing time (ms)	Consumption of energy(µJ)
Phase 1	81	4013	12.73	113.12
Phase 2	1439	73303	282.86	2554.76

The outcomes for implementation of algorithm for usage of memory, consumption of energy, and processing time are summarized in Table 5.4. These outcomes yield two findings: (i) the procedure is adequate for constrained resource motes and (ii) the outcomes of implementation are in line with the theoretical outcomes described in Sect. 5.5.1.

To further establish the effectiveness of the proposed abnormality identification and confirmation system, the results have been compared with the two approaches [11]. The first approach employs linear programming-based hyperellipsoidal formulation. The second approach, on the contrary, employs hypersphere to capture the usual behavior in higher dimensional space. The results of CESVM and QSSVM show that these approaches can achieve the best detection performance of 99.32 and 80%, respectively. In contrast, the detection accuracy of our proposed scheme was steady at 100%.

5.8 Discussion

The main points from the experiments are as follows. Firstly, the domain expertise is needed to adequately setup the abnormality identification system. A simple training or a user manual may be required to enable users to operate the developed application software in real world settings. The user can then manage the performance of the system using configuration and report panels.

The technique can discover cross-layer abnormalities caused in features, namely SR, BY, LQI, $RSSI$, and PER with high accuracy. The technique employs statistical procedures to define abnormal, tolerated, and usual regions using fuzzy logic to discover cross-layer abnormalities and optimize abnormality confirmation agent transmission. This approach has resulted in high-detection accuracy (i.e., up to 100%), and it can save 54.29–42.85% energy resources in comparison to the other techniques [1–3] that do not take into account the state of the communication link for the dispatch of abnormality confirmation agent.

It is pertinent to observe that the non-consideration of the poor communication link-state may result in non-reliable transmission of the abnormality identification agent. In such situations, abnormality identification system must be able to use other measures for instance updating a user regarding abnormalities. The related schemes have not considered this key factor [1–3]. On the contrary, the scheme presented in this chapter empowers the cross-layer abnormality identification module to facilitate other jobs like updating the user regarding abnormalities in the situations when abnormality confirmation agent cannot be reliably transmitted due to poor communication link-state. This advocates that the proposed scheme offers more reliable and energy efficient service to transmit abnormality confirmation agents to carry out their designated task.

5.9 Summary

A robust abnormality identification system is indispensable for smart homes to update users in a timely way regarding abnormalities occur due to errors in transmission, attacks, or mote faults. This chapter has discussed the design and implementation rationales of an innovative cross-layer abnormality identification technique for smart homes in order to address the first and third research questions and also to satisfy the corresponding first and third requirements. The technique uses simple but effective statistical processes with fuzzy logic to identify cross-layer abnormalities. The technique also provides the service to dispatch abnormality confirmation agents after taking into account the state of the communication link.

The technique has been implemented on real mote-based test bed. The detection accuracy results of the proposed scheme have been compared with crisp-logic classification algorithm. The outcomes show the detection accuracy of the proposed scheme which was steady at 100%. In contrast, the detection accuracies of the crisp-logic case were 98.8% for SR, 98.5% for BY, 98.7% for LQI, 98.4% for $RSSI$, and 98.7% for PER. The results also show capability of the proposed scheme to save the energy consumed by the dispatch of the abnormality confirmation agent in poor state of the communication link situations as high as 42.85 to 54.29% in contrast to the other techniques. A software application has been programmed to control the abnormality identification system in smart home that allows users to improve the functioning of the technique.

5.10 Bibliographic Notes

The method for cross-layer abnormality identification and transmission of agent transmission optimization using fuzzy logic was presented in [5]. The implementation and theoretical analysis results were reported in [5].

References

1. M. Ketel, Applying the mobile agent paradigm to distributed intrusion detection in wireless sensor networks, in *40th Southeastern Symposium on System Theory*, pp. 74–78, March 2008
2. M. Pugliese, A. Giani, F. Santucci, Weak process models for attack detection in a clustered sensor network using mobile agents, in *Sensor Systems and Software*, vol. 24, ed. by S. Hailes, S. Sicari, G. Roussos. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, (Springer, Berlin, 2010), pp. 33–50
3. S. Khanum, M. Usman, A. Alwabel, Mobile agent based hierarchical intrusion detection system in wireless sensor networks. Int. J. Comput. Sci. Issues (IJCSI) **9**(3), 101–108 (2012)
4. K. Kapitanova, S.H. Son, K.D. Kang, Using fuzzy logic for robust event detection in wireless sensor networks. Ad Hoc Netw. **10**(4), 709–722 (2012)
5. M. Usman, V. Muthukkumarsamy, X.-W. Wu, Mobile agent-based cross-layer anomaly detection in smart home sensor networks using fuzzy logic. IEEE Trans. Consum. Electron. **61**(2), 197–205, 2015b
6. R. Alsaqour, M. Abdelhaq, R. Saeed, M. Uddin, O. Alsukour, M. Al-Hubaishi, M. Alahdal, Dynamic packet beaconing for gpsr mobile ad hoc position-based routing protocol using fuzzy logic. J. Netw. Comput. Appl. **47**, 32 (2015)
7. L. Tang, K.C. Wang, F. Huang, Y. Gu, Channel characterization and link quality assessment of ieee 802.15.4-compliant radio for factory environments. IEEE Trans. Ind. Inform. **3**(2), 99–110 (2007)
8. D.-I. Curiac, O. Banias, F. Dragan, C. Volosencu, O. Dranga, Malicious node detection in wireless sensor networks using an autoregression technique, in *Third International Conference on Networking and Services*, pp. 83–83, June 2007
9. S. Nanz, C. Hankin, A framework for security analysis of mobile wireless networks. Theor. Comput. Sci. **367**(1–2), 203–227 (2006)
10. M. Moshtaghi, C. Leckie, S. Karunasekera, S. Rajasegarar, An adaptive elliptical anomaly detection model for wireless sensor networks. Comput. Netw. 64, 195–207, 2014a
11. S. Rajasegarar, C. Leckie, J.C. Bezdek, M. Palaniswami, Centered hyperspherical and hyperellipsoidal one-class support vector machines for anomaly detection in sensor networks. IEEE Trans. Inf. Forensics Secur. **5**(3), 518–533 (2010)

Chapter 6
Conclusions

The number of state-of-the-art sensor network applications has been increasing with the passage of time due to advancements in the broader field of the information technology. This has increased the demand of the design and development of innovative services to cater for the unique requirements of the sensor network applications. This book has elucidated a novel service for on-the-spot confirmation of suspicious malicious motes to discover the origin of abnormalities in sensor network applications, for example, smart home and built infrastructure monitoring. The identification of the source of abnormalities is imperative in order to timely and effectively mitigate them. The internal structure and details of the abnormality identification and confirmation system, along with a number of related methods for the abnormality identification and agent transmission optimization, have been introduced throughout this text.

In order to conclude the discussion on the work presented in this text, this chapter first outlines the work presented in this book. The limitations of the discussed methods and corresponding precautionary measures are discussed next. Finally, a discussion on the possible avenues for future research is provided.

6.1 Book Outlook

To offer a big picture of the abnormality identification schemes in the literature, a taxonomy has been formulated in Chap. 2, refer to Sects. 2.1 and 2.2 [1]. This process has not only provided the big picture of the literature, but also helped in the identification of the problem domain. More specifically, related literature was analyzed and broadly classified into statistical, artificial intelligence and agent-enabled, machine learning, and other schemes. The related literature of the agents security and formal modeling and analysis has also been reviewed to set the research context.

M. Usman et al., *Mobile Agent-Based Anomaly Detection and Verification System for Smart Home Sensor Networks*,
https://doi.org/10.1007/978-981-10-7467-7_6

An agent-enabled abnormality identification and confirmation system, with its internal architecture and algorithmic specifications, has been introduced in Chap. 3, refer to Sects. 3.4 and 3.5 [5–7]. The algorithmic specifications include features collection by cluster member mote, abnormality identification by cluster leader mote, anomalous mote confirmation, and update of status on the leader mote and BS. The system has been designed to not only detect abnormalities which occur due to the erroneous values of the synchronized resource management technique-based features and temporal abnormalities that are occurred due to the arrival delay of measurements, but also to perform on-the-spot confirmation of the member motes using the information accumulated by the synchronized resource management technique.

The on-the-spot confirmation procedure is, however, time-sensitive. To examine the temporal conduct of the system, the formalization of individual functionalities has been, therefore, performed using formalism of Petri nets in Chap. 3, refer to Sects. 3.6, 3.7, and 3.8 [2]. The bottom-up amalgamation of the individual net components (i.e., each algorithmic specification) was then performed to make a unified model that embodies the conduct characteristics like liveness and boundedness, and also the global workflow of the system. The analysis has shown that the design of the system is deadlock free and correct in order to complete its designated tasks, as there was no deadlock state in the model, which can halt the system, and the tokens were able to successfully reach the terminal states, denoting the correctness of the model. The standard model was prolonged into an equivalent high category Generalized Stochastic model to characterize and examine the time-based conduct of the system in an extremely non-deterministic interaction environment of the sensor networks. The Generalized Stochastic Petri Net-based conduct of the system was then endorsed via experiments performed on a test bed composed of resource-limited MICAz motes.

A number of case scenarios have been experimented to systematically study the time-based conduct of the system. The outcomes have demonstrated that the system consumes 64.36% more time to carry out the job of on-the-spot confirmation when system is also configured for the identification and confirmation of the abnormalities occur due to attacks, dissimilar to when it is configured only for the abnormalities occur due to errors or faults. The additional time is taken due to the overhead involved to secure the abnormality confirmation agent by employing the water marking technique. The impact of the change in the distance on the temporal behavior of the system has also been studied. The results have indicated that the change in the close proximity has minimum effect on the time-based conduct of the system. This shows that the system is largely adequate for sensor applications, viz. smart home and built infrastructure monitoring where motes are typically configured in close vicinity from each other.

The results have also indicated that the system is more adequate to periodically data sending applications as compared to continuous data sending applications, as continuous data sending applications store more number of observations for on the spot confirmation procedure. As a consequence, the abnormality confirmation agent requires more resources and time to carry out the job of on-the-spot confirmation in continuously data sending applications as compared to periodically data sending applications.

A method for detecting first-order abnormalities in the synchronized resource management technique-based observations has been elucidated in Chap. 4, refer to Sect. 4.2.1 [2]. The method has exploited statistical association among features of normal profile to identify numerous kinds of abnormalities such as occurred due to on-the-spot attacks or faults, exhaustion of resource threats, faulty motes, attacks on the resources of motes, and denial-of-sleep attacks. The results have shown that the method can identify first-order abnormalities with a detection accuracy of between 97.80 and 99.20%. It has also been observed that the method can detect more types of abnormalities which occur due to denial-of-sleep threat, exhaustion of resource threat, and mote faults with $O(n)$ detection complexity, unlike related techniques presented by Eludiora and colleagues [3] and Khanum and colleagues [4] which can detect DoS attack and sensor reading abnormalities with the detection complexity of $O(n_2)$ and $O(n)$, respectively.

Two methods for the optimization of dispatch of agents were also elucidated in Chap. 4, refer to Sects. 4.2.2 and 4.2.3, to extend network lifetime. The first method, namely two-sigma, employed two standard deviations in order to outline curtailment zones on probability distributions of the normal profile features to optimize transmission of agents [5, 6]. On the other hand, the second method, namely weighted sum optimization, incorporated the previous and present conducts of the mote to optimize dispatch of agent procedure to enhance the network lifetime [7]. The experiment outcomes have shown that the 2-sigma and weighted sum abnormality confirmation agent transmission optimization methods reduced the abnormality confirmation agent transmission overhead by as much as $29\% - 34\%$ and $42\% - 52\%$, respectively, unlike related schemes [3, 4, 8] which have overlooked the optimization of abnormality confirmation agent dispatch.

A fuzzy logic-oriented cross-layer abnormality identification and agent transmission optimization scheme has been elucidated in Chap. 5, refer to Sects. 5.3, 5.4, and 5.5 [9]. The main characteristics of the method are as follows: (i) a zone calculation technique, on the basis of statistical processes, was presented to define multiple zones for making decisions regarding abnormality identification and agent dispatch, (ii) a fuzzy logic-oriented rule-base was formulated and an associated method was studied to identify the cross-layer abnormalities and dispatch an agent after taking into account the state of communication channel, and (iii) finally, the techniques were deployed on mote-based test bed with a developed application to assess its performance in smart home.

The outcomes have indicated the detection accuracy of the proposed scheme was steady at 100%. On the contrary, the identification accuracies of crisp logic, a baseline situation, were 98.8% for SR, 98.5% for BY, 98.7% for LQI, 98.4% for $RSSI$, and 98.7% for PER. The experiments have also shown the ability of the technique to save energy consumed by the abnormality confirmation agent dispatch in poor state situations of communication link as high as 42.85 to 54.29% in contrast to the related schemes [4, 8, 10], that do not take into account the state of the communication link for the dispatch of abnormality confirmation agent.

In summary, this book has elucidated an agent-enabled abnormality identification and confirmation system for sensor networks. The system, in addition to detecting

different types of abnormalities, offers a novel service to discover the origin of abnormalities after their identification. The temporal behavior of the system has been thoroughly investigated to establish its aptness for applications, namely smart home sensor network and built infrastructure monitoring. A number of methods, namely two-sigma and weighted sum optimization, have been designed and analyzed to effectively transmit agents in order to increase the network longevity. An agent-enabled cross-layer abnormality identification and confirmation scheme has been elucidated, which is capable to identify cross-layer abnormalities and effectively dispatch agent after taking into account the state of communication link.

6.2 Limitations

The agent-enabled abnormality identification and confirmation system and related methods, which are presented in this text, can detect different kinds of abnormalities with high accuracy, optimize agent transmission using a number of methods, and perform on-the-spot diagnosis of motes to identify the origin of abnormalities. The system and associated methods, however, have a few limitations which may affect their performance in certain situations. In most of the situations, some precautionary measures can be taken to either entirely negate or minimize that effect. The limitations with corresponding precautionary measures are discussed below.

The system is designed for cluster-based sensor networks. The abnormality identification module is designed to be configured on resource-redundant cluster leader motes. At the time of abnormality identification application deployment, the system administrator must consider the size of clusters in the network. Cluster leader motes should have enough memory to accommodate different normal profiles of cluster member motes. In some applications, such as smart home sensor networks, different sensors with similar roles in the network can be grouped together to have a single normal profile to optimize consumption of memory.

The transmission of agents in a network can cause the communication bottleneck at cluster leader motes. This can change the temporal behavior of on-the-spot confirmation procedure. It is, therefore, recommended that the system administrator must perform a worst case temporal behavior analysis at the time of abnormality identification application deployment. That is, transmit agents to all cluster member sensor motes and then compute the time taken from the agent transmissions to on-the-spot confirmation results arrival at cluster leader motes. The computed time then must be integrated into on-the-spot confirmation decision making process to accommodate the presence of a particular number of motes at certain positions within a cluster.

Another potential limitation of the elucidated abnormality identification and confirmation system, like other cluster-based applications, is the possibility of the failure of a cluster leader mote. It is, therefore, imperative that the cluster leader mote must share the abnormality identification and confirmation related information with other

trusted motes to avoid any such failure. The frequency of the information sharing, however, must be carefully set by considering the available memory and energy resources of the sensor network.

6.3 Further Research

The work carried out in this study can be extended in a number of directions.

On-the-spot confirmation using multihop agent itinerary: This study has focused on a single mote agent itinerary model, that is, a agent can only traverse from a cluster leader mote to a cluster member mote and vice versa. Although single mote agent itinerary is suitable for cluster-based applications, viz. smart home and built infrastructure monitoring sensor networks, it may not be suitable for large sensor networks. The new algorithms and protocols, therefore, could be designed to effectively perform on-the-spot confirmation of motes which are away from the base station. The energy consumption by the process of the agent transmission over a multihop link must be carefully analyzed while designing new algorithms.

Higher-order abnormality identification: This study has considered abnormalities in the synchronized resource management technique-based individual features, and their first-order joins to detect abnormalities caused by erroneous values of features, mote faults and attacks, resource exhaustion attacks, attacks on the resources of motes, and denial-of-sleep attacks. Another possible extension could be the exploitation of high-order joins. This may result in detection of more complex and different natures of abnormalities. The system designer, however, must consider the suitability of the computational complexity of the detection mechanism for low resource sensor networks while designing a high-order joins algorithm.

Contextual and spatiotemporal abnormality identification: Another interesting future work direction could be the construction of different normal profiles for multiple kinds of motes within a sensor network. For instance, in a smart home, different normal devices in regards to their role and hardware capability should have different normal profiles. Not only the contextual information of sensors, but also the spatiotemporal correlation (if any) can be integrated in the normal profile of motes for the robust abnormality identification. The resultant abnormality identification mechanism, however, should satisfy the hardware constraints of multiple sensors within the network.

Mobile sensor networks and cyber-physical systems: The feasibility of utilizing elucidated abnormality identification and on-the-spot confirmation system for mobile sensor networks and cyber-physical systems can also be investigated as a future work. An interesting research direction could be prediction of itinerary of an agent

in a mobile wireless sensor network by considering the past movement of motes within the network. Similarly, the stochastic model of the system may be extended to cyber-physical systems to verify the aptness of the system.

References

1. M. Usman, V. Muthukkumarsamy, X.-W. Wu, S. Khanum, *Anomaly Detection in Wireless Sensor Network: Challenges and Future Trends* (Auerbach publications Taylor and Francis Group, USA, in security for multihop wireless networks edition, 2014)
2. M. Usman, V. Muthukkumarsamy, X.-W. Wu, Specification and validation of enhanced mobile agent-enabled anomaly detection in resource constrained networks. J. Netw. **10**(6), 353–368 (2015)
3. S.I. Eludiora, O.O. Abiona, A.O. Oluwatope, S.A. Bello, M.L. Sanni, D.O. Ayanda, C.E. Onime, E.R. Adagunodo, L.O. Kehinde, A distributed intrusion detection scheme for wireless sensor networks, in *IEEE International Conference on Electro/Information Technology (EIT)* (2011), pp. 1–5
4. S. Khanum, M. Usman, A. Alwabel, Mobile agent based hierarchical intrusion detection system in wireless sensor networks. Int. J. Comput. Sci. Issues (IJCSI) **9**(3), 101–108 (2012)
5. M. Usman, V. Muthukkumarsamy, X.W. Wu, S. Khanum, Wireless smart home sensor networks: mobile agent based anomaly detection, in *Proceedings of the 9th IEEE International Conference on Ubiquitous Intelligence Computing and 9th International Conference on Autonomic Trusted Computing (UIC/ATC)* (2012), pp. 322–329
6. M. Usman, Agent-enabled anomaly detection in resource constrained wireless sensor networks, in *Proceedings of the 15th IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)* (2014), pp. 1–2
7. M. Usman, V. Muthukkumarsamy, X.-W. Wu, A resource-efficient system for detection and verification of anomalies using mobile agents in wireless sensor networks. J. Netw. **9**(12), 3427–3444 (2014)
8. M. Ketel, Applying the mobile agent paradigm to distributed intrusion detection in wireless sensor networks, in *40th Southeastern Symposium on System Theory* (2008), pp. 74–78
9. M. Usman, V. Muthukkumarsamy, X.-W. Wu, Mobile agent-based cross-layer anomaly detection in smart home sensor networks using fuzzy logic. IEEE Trans. Consum. Electron. **61**(2), 197–205 (2015)
10. M. Pugliese, A. Giani, F. Santucci, Weak process models for attack detection in a clustered sensor network using mobile agents, in *Sensor Systems and Software*, vol. 24, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, ed. by S. Hailes, S. Sicari, G. Roussos (Springer, Heidelberg, 2010), pp. 33–50

Appendix A
Reachability Trees

The reachability trees for the Petri net models, namely unified formal model (refer to Chap. 4, Sect. 4.3) and cross-layer model (refer to Chap. 5, Sect. 5.6), are elucidated in this appendix. Figure A.1 depicts the reachability tree for the unified Petri net model, PN_u. The Petri net model PN_u executes the first-order anomaly detection by the cluster leader nodes through the reachable markings $M_{62}[t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\rangle \widetilde{M_{63}}$ and $M_{62}[t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\rangle \widehat{M_{64}}$, where $\widetilde{M_{(\cdot)}}$ and $\widehat{M_{(\cdot)}}$ denote the non-terminal and terminal states, respectively. The non-terminal state denotes the state where the cluster leader node invokes a relevant algorithm for the anomaly verification agent transmission optimization. The terminal state, on the other hand, denotes the state where the cluster leader node aggregates the sensed data.

The Petri net model PN_u then executes the 2-sigma anomaly verification agent transmission optimization algorithm through the markings $M_{62}[t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\rangle M_{63}\ [t_{31}\rangle\ M_{66}\ [t_{32}\rangle \widehat{M_{68}}$, $M_{62}[t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\rangle\ M_{63}\ [t_{31}\rangle\ M_{66}\ [t_{32}\rangle\ M_{69}\ [t_{34}\rangle \widehat{M_{72}}$, and $M_{62}[t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\rangle\ M_{63}\ [t_{31}\rangle\ M_{66}\ [t_{32}\rangle\ M_{69}\ [t_{34}\rangle\ M_{73}\ [t_{37}\rangle \widehat{M_{76}}$. The state $\widehat{M_{68}}$ denotes the normal behavior of msn_q, $\widehat{M_{72}}$ represents the state of checking the trust value of the cluster member node, and $\widehat{M_{76}}$ shows the state where the cluster leader node transmits the anomaly verification agent and anomaly alarm to the cluster member node and the base station, respectively.

Alternatively, the Petri net model PN_u executes the weighted-sum optimization algorithm through the markings $M_{62}\ [t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\rangle\ M_{63}\ [t_{30}\rangle\ M_{65}[t_{39}, t_{40}\rangle\ M_{67}\ [t_{41}\rangle\ M_{70}\ [t_{46}\rangle \widehat{M_{82}}$,$M_{62}\ [t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\rangle\ M_{63}\ [t_{30}\rangle\ M_{65}[t_{39}, t_{40}\rangle\ M_{67}\ [t_{41}\rangle\ M_{70}\ [t_{46}\rangle\ M_{83}\ [t_{51}\rangle \widehat{M_{64}}$, $M_{62}\ [t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\rangle\ M_{63}\ [t_{30}\rangle\ M_{65}[t_{39}, t_{40}\rangle\ M_{67}\ [t_{41}\rangle\ M_{71}\ [t_{42}\rangle\ M_{74}\ [t_{46}\rangle \widehat{M_{82}}$, $M_{62}\ [t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\rangle\ M_{63}\ [t_{30}\rangle\ M_{65}[t_{39}, t_{40}\rangle\ M_{67}\ [t_{41}\rangle\ M_{71}\ [t_{42}\rangle\ M_{74}\ [t_{46}\rangle\ M_{83}\ [t_{51}\rangle \widehat{M_{64}}$, $M_{62}\ [t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\rangle\ M_{63}\ [t_{30}\rangle\ M_{65}[t_{39}, t_{40}\rangle\ M_{67}\ [t_{41}\rangle\ M_{71}\ [t_{42}\rangle\ M_{75}\ [t_{43}\rangle\ M_{77}\ [t_{46}\rangle \widehat{M_{82}}$, $M_{62}\ [t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\rangle\ M_{63}\ [t_{30}\rangle\ M_{65}[t_{39}, t_{40}\rangle\ M_{67}\ [t_{41}\rangle\ M_{71}\ [t_{42}\rangle\ M_{75}\ [t_{43}\rangle\ M_{77}\ [t_{46}\rangle\ M_{83}\ [t_{51}\rangle \widehat{M_{64}}$, $M_{62}\ [t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\rangle\ M_{63}\ [t_{30}\rangle\ M_{65}[t_{39}, t_{40}\rangle\ M_{67}\ [t_{41}\rangle\ M_{71}\ [t_{42}\rangle\ M_{75}\ [t_{43}\rangle\ M_{78}\ [t_{44}\rangle\ M_{79}\ [t_{46}\rangle \widehat{M_{82}}$, $M_{62}\ [t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\rangle\ M_{63}\ [t_{30}\rangle\ M_{65}[t_{39}, t_{40}\rangle\ M_{67}\ [t_{41}\rangle\ M_{71}\ [t_{42}\rangle\ M_{75}\ [t_{43}\rangle\ M_{78}\ [t_{44}\rangle\ M_{79}\ [t_{46}\rangle\ M_{83}\ [t_{51}\rangle \widehat{M_{64}}$, $M_{62}\ [t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\rangle\ M_{63}\ [t_{30}\rangle\ M_{65}[t_{39}, t_{40}\rangle\ M_{67}\ [t_{41}\rangle\ M_{71}\ [t_{42}\rangle\ M_{75}\ [t_{43}\rangle\ M_{78}$

M. Usman et al., *Mobile Agent-Based Anomaly Detection and Verification System for Smart Home Sensor Networks*,
https://doi.org/10.1007/978-981-10-7467-7

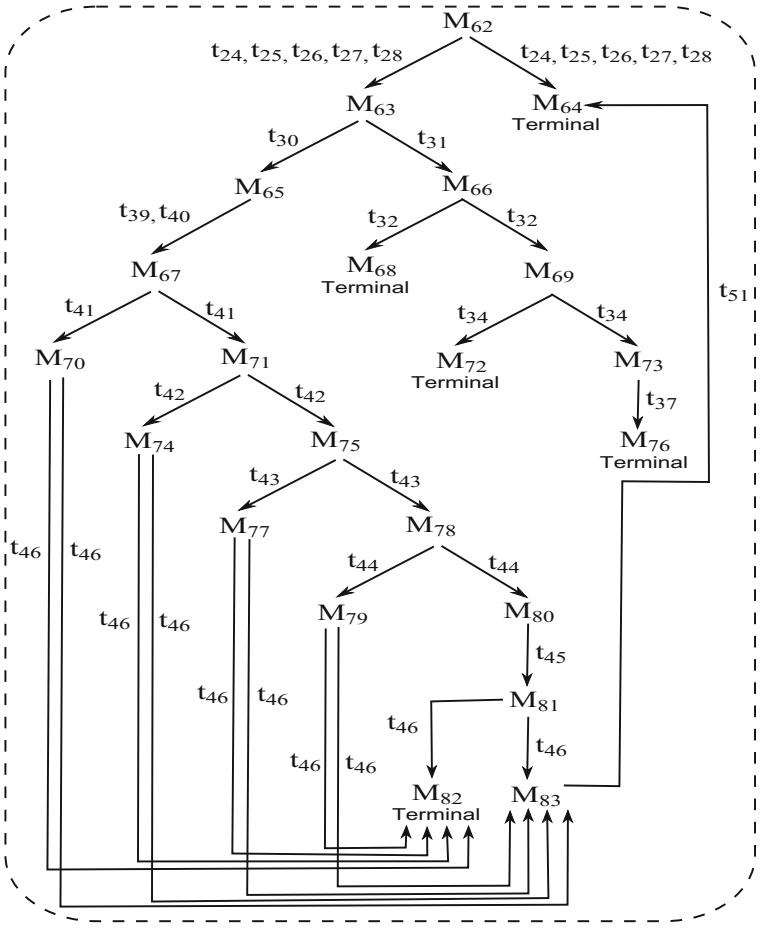

Fig. A.1 Reachability tree for Petri net model PN_u

$[t_{44}\rangle\ M_{80}\ [t_{45}\rangle\ M_{81}\ [t_{46}\rangle\ \widehat{M_{82}}$, and $M_{62}\ [t_{24}, t_{25}, t_{26}, t_{27}, t_{28}\rangle\ M_{63}\ [t_{30}\rangle\ M_{65}[t_{39}, t_{40}\rangle\ M_{67}\ [t_{41}\rangle\ M_{71}\ [t_{42}\rangle\ M_{75}\ [t_{43}\rangle\ M_{78}\ [t_{44}\rangle\ M_{80}\ [t_{45}\rangle\ M_{81}\ [t_{46}\rangle\ M_{83}\ [t_{51}\rangle\ \widehat{M_{64}}$ to optimize anomaly verification agent transmission.

The cross-layer Petri net model PN_c, shown in Fig. A.2, executes the cross-layer anomaly detection and anomaly verification agent transmission optimization algorithm through the markings $M_{84}\ [t_{52}, t_{53}, t_{54}, t_{55}, t_{56}, t_{57}\rangle\ M_{85}, M_{86}, M_{87}, M_{88}, M_{89}, M_{90}\ [t_{58}, t_{59}, t_{60}, t_{61}, t_{62}, t_{63}\rangle\ M_{91}\ [t_{64}^1, t_{64}^2, ..., t_{64}^{243}\rangle\ M_{93}\ [t_{65}\rangle\ \widehat{M_{92}}$ and $M_{84}\ [t_{52}, t_{53}, t_{54}, t_{55}, t_{56}, t_{57}\rangle\ M_{85}, M_{86}, M_{87}, M_{88}, M_{89}, M_{90}\ [t_{58}, t_{59}, t_{60}, t_{61}, t_{62}, t_{63}\rangle\ M_{91}\ [t_{64}^1, t_{64}^2, ..., t_{64}^{243}\rangle\ M_{93}\ [t_{65}\rangle\ \widehat{M_{94}}$, where $\widehat{M_{92}}$ and $\widehat{M_{94}}$ are terminal states which show states of the cluster leader node to aggregate sensed data and decrement the trust value of the cluster member node, respectively.

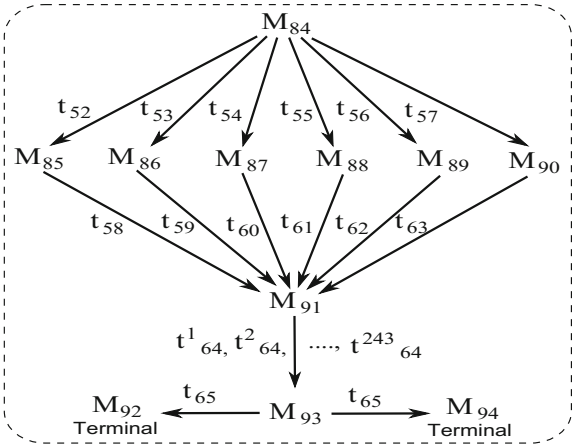

Fig. A.2 Reachability tree for Petri net model PN_c

Bibliography

B. Sundararaman, U. Buy, A.D. Kshemkalyani, Clock synchronization for wireless sensor networks: a survey. Ad Hoc Netw. **3**(1), 281–323 (2005)

Y. Gu, T. He, Dynamic switching-based data forwarding for low-duty-cycle wireless sensor networks. IEEE Trans. Mobile Comput. **10**(12), 1741–1754 (2011)

M. Usman et al., *Mobile Agent-Based Anomaly Detection and Verification System for Smart Home Sensor Networks*,
https://doi.org/10.1007/978-981-10-7467-7