

# **Resource Inquisition and Optimal Calibration on Internet-of-Things**

*A Thesis*

*submitted in partial fulfillment of the requirements for the award of degree of*

**Doctor of Philosophy**

*by*

**Monika Bharti**

Registration No. 901411011

under the guidance of

**Dr. Rajesh Kumar**  
Professor

**Dr. Sharad Saxena**  
Associate Professor



**Computer Science and Engineering Department  
Thapar Institute of Engineering and Technology  
Patiala-147004, Punjab, India  
July 2018**

## *Certificate*

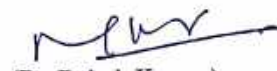
I hereby certify that the work which is being presented in this thesis entitled "**Resource Inquisition and Optimal Calibration on Internet-of-Things**", in partial fulfillment of the requirement for the award of degree of "Doctor of Philosophy" submitted in Computer Science and Engineering Department of Thapar Institute of Engineering and Technology, Patiala, Punjab, India is an authentic record of my own work carried out under the supervision of Dr. Rajesh Kumar and Dr. Sharad Saxena. The work has referred research work of others which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for the award of any other degree or to any other university.

  
(Monika Bharti)

Registration No. 901411011

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

  
(Dr. Rajesh Kumar)  
Professor  
Computer Science and  
Engineering Department

  
(Dr. Sharad Saxena)  
Associate Professor  
Computer Science and  
Engineering Department

## *Acknowledgement*

I would like to express my sincere appreciation to my supervisors, **Dr. Rajesh Kumar**, Professor and **Dr. Sharad Saxena**, Associate Professor, for being pillars of support and encouragement throughout my research work. Their experiences, strengths, tenderness and willfulness, have taught me valuable lessons of life, which are going to be of immense help in taking decisions for moving forward.

My sincere thanks are due to **Prof. O. P. Pandey**, Dean of Research/Scientific Projects, **Prof. R. K. Sharma**, Dean of Faculty Affairs, **Prof. Maneek Kumar**, Dean of Student Affairs, **Prof. S. S. Bhatia**, Dean of Academic Affairs and **Prof. Maninder Singh**, Head, CSED, TIET, Patiala; for providing me the necessary administrative assistance in completion of the work. I am also thankful to **Dr. Seema Bawa**, **Dr. Sanmeet Bhatia** and **Dr. Kulbir Singh**, TIET, Patiala for their constructive comments and regularly ensuring the progress of my research work. Lastly, I am thankful to all the **faculty** and **staff** members of the CSED for their kind support, especially, **Dr. Shalini Batra**.

I offer my deepest gratitude to my loving parents, **Sh. Pitander Kumar**, **Smt. Tulsia Rani**, nephew **Adavya Singh**, brother **Er. Rohit Bharti**, husband **Dr. Himanshu Jindal**, sister **Dr. Renu Bharti** and brother-in-law **Dr. Aman Gopal** whose love, affectionate, encouragement, motivation, and confidence have been a constant source of inspiration in making my vision a reality. I also acknowledge the cooperation and encouragement extended to me by my friends, **Dr. Amritpal Singh**, **Mr. Abhishek Khanna**, and **Ms Garima Kadian**. I would like to acknowledge the financial support given by the CSED, TIET under **Ministry of Electronics and Information Technology (MeitY)** to complete my doctoral studies. I am also thankful to authorities of **National Highways Authority of India** for their kind support.

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

Patiala  
July, 2018

*Monika Bharti*  
(Monika Bharti)

## ***Abstract***

Internet-of-Things (*IoT*) is an emerging and widely applied research area that envision integration of physical world into digital. For the same, it requires automation of the interconnected resources such that they can sense, process and interpret *via* Internet-connected infrastructures. A resource is either software or hardware with fundamental characteristics such as physical embodiment, unique identifier, offered service, location, information, operating system and mode of communication.

To facilitate communication and computation among the resources, it is required to discover them across distributed systems having distinct communication capabilities and intelligence. It implies that discovery is possible with heterogeneous views like human operators, application softwares and autonomous smart objects. Such views face challenges because resources are generating huge volume of data coupled with mobility and dynamicity. Hence, it would slow down the process of discovery because of issues like heterogeneity, interoperability, lack of standardization, and periodic evaluations. Moreover, these issues limit the resource discovery on *IoT* paradigm leading to challenges such as to provide answers to knowledge-based queries, to develop models to exchange applications' context, to bridge the gap between device level and across applications, indexing and splitting of multiple parameters, and sophisticated techniques for managing metadata. Thus, it concludes resource discovery as a fundamental challenge for the realization of *IoT* vision and has gained the researcher's interest all over the world.

To address the challenge, resources are required to undergo both retrieval and ranking processes. In short, resource discovery needs organization and analysis of the gathered complex descriptions as data. For the purpose, it requires techniques or algorithms that cover basic fundamental search principles for indexing, clustering, knowledge representation, and content being searched. This will ease in searching and would cover a multitude of functionality and dimensionality. Moreover, due to factors such as noise, limited memory, opportunistic presence of the resources, availability and data integrity, *IoT* needs to consider various parameters with respect to discovery. The parameters are to automate and access, several dimensions related to query and to understand the search evaluation metrics.

In this thesis, a novel clustering technique, *namely*, Iterative K-means Clustering Algorithm

(*IKm-CA*) and three frameworks, *i.e.*, Context-Aware Search Optimization Framework on Internet-of-Things (*CASOF-IoT*), Intelligent Resource Inquisition Framework on Internet-of-Things (*IRIF-IoT*) and Middleware Approach for Reliable Resource Selection on Internet-of-Things (*MARRS-IoT*) for resource discovery and selection are proposed. The *IKm-CA* targets concrete cluster formation using similarity coefficients of vector space model and performs efficient search against matching criteria. It mitigates the problems of selecting erroneous or empty data points to clusters, reduces consumption time, noise factor and the number of clusters that are manually input to the system. Though, the technique eliminates the heterogeneity in data challenge to resource discovery and selection on *IoT*, yet it has not accounted the issues such as context of the gathered data, scalability and search metrics on *IoT*. These issues if not resolved, would lead resource discovery and selection to emerge as a non-linear constrained specific problems. Therefore, it is required to optimize them to ease the inter-communication and interaction. For the purpose, *CASOF-IoT* is introduced that targets knowledge presentation through schema, discovery *via* multi-modal search algorithm and its optimization through an Iterative Gradient Descent algorithm. The multi-modal search algorithm through keywords, value or spatial-temporal indices perform resource discovery by finding the suited matches as search set from search-space. Though, it performs efficient resource discovery yet it could not address interoperability challenge at various levels like semantic, radio access and context with respect to resource discovery for the automation and authentication of the infrastructure. Hence, another framework, *IRIF-IoT* is suggested that link resources through usage of semantic description and ontology, their discovery with Semantic Matchmaking Engine using Bipartite Graph (*SMEBG*) and to access information *via* web terminal for users. The framework has enhanced the system performance significantly but has not considered trust issue. This issue arises on a global network infrastructure like the *IoT* due to continuously changing mobility patterns, heterogeneity, interoperability and scalability on the network that imposes restriction to process an optimal decision. Keeping the trust perspective, *MARRS-IoT* is presented that helps to achieve maximum reliable resource for communication that maintain data integrity *via* trust evaluations and improve system performance. It dynamically processes information *locally* as well as *globally* with less time consumption and minimizes complex searches.

# Table of Contents

Certificate . . . . .	iii
Certificate . . . . .	iii
Acknowledgment . . . . .	iii
Abstract . . . . .	iii
List of Figures . . . . .	ix
List of Tables . . . . .	xii
List of Algorithms . . . . .	xiii
List of Symbols . . . . .	xiii
List of Abbreviations . . . . .	xxi
<b>1 Introduction</b>	<b>1</b>
1.1 Internet-of-Things . . . . .	1
1.1.1 Building Blocks . . . . .	2
1.1.2 Common Standards . . . . .	5
1.1.3 Current Landscape . . . . .	8
1.2 Resource Discovery and Selection on Internet-of Things . . . . .	8
1.2.1 Resources and their Graphical Forecast . . . . .	9
1.2.2 Resource Discovery . . . . .	9
1.2.3 Motivation behind Resource Discovery . . . . .	10
1.2.4 Traditional Web Search versus Internet-of-Things Resource Discovery	11
1.2.5 Evaluation Metrics for Resource Discovery . . . . .	12
1.2.6 Challenges in Resource Discovery . . . . .	13
1.3 Thesis Organization . . . . .	14

<b>2</b>	<b>Literature Review</b>	<b>19</b>
2.1	Internet-of-Things . . . . .	19
2.2	Foraging Loop: Data Gathering and Modeling . . . . .	25
2.2.1	Data Gathering . . . . .	27
2.2.2	Data Modeling . . . . .	30
2.3	Sense Making Loop: Resource Discovery and Selection . . . . .	38
2.3.1	Resource Discovery: Categorization and Approaches . . . . .	39
2.3.2	Resource Selection . . . . .	51
2.4	Issues in Resource Discovery and Selection . . . . .	73
2.5	Problem Formulation . . . . .	74
2.6	Thesis Objectives . . . . .	75
<b>3</b>	<b>Clustering based Resource Discovery and Selection</b>	<b>77</b>
3.1	Introduction . . . . .	77
3.2	Resource Clustering . . . . .	78
3.2.1	Vector Space Model . . . . .	79
3.2.2	k-means Algorithm . . . . .	81
3.3	Proposed Iterative K-means Clustering Algorithm . . . . .	86
3.3.1	Cluster Formation . . . . .	86
3.3.2	Iterative K-means Clustering Algorithm . . . . .	86
3.3.3	Matching Conditions for Discovery . . . . .	89
3.4	Experimental Evaluations . . . . .	91
3.4.1	Similarity Computation . . . . .	91
3.4.2	Clustering Analysis . . . . .	93
3.4.3	System Performance . . . . .	97
3.4.4	Complexity Analysis . . . . .	100
3.5	Conclusion . . . . .	105
<b>4</b>	<b>Context-Aware Resource Discovery and Selection using Convex Optimization</b>	<b>107</b>
4.1	Introduction . . . . .	107
4.2	Context Awareness and Internet-of-Things . . . . .	108
4.3	Non-Linear Programming . . . . .	108

4.3.1	Gradient Descent Method . . . . .	109
4.3.2	Newton Method . . . . .	110
4.3.3	Quasi-Newton Method . . . . .	113
4.4	Intelligent Transportation System . . . . .	116
4.4.1	Electronic Toll Collection . . . . .	116
4.4.2	Study Area: Shambhu Toll Plaza . . . . .	117
4.5	Proposed Context-Aware Search Optimization Framework on Internet-of-Things . . . . .	118
4.5.1	Context Acquisition Layer . . . . .	120
4.5.2	Context Annotation-Search Layer . . . . .	120
4.5.3	Search Optimization Layer . . . . .	123
4.5.4	Selection-Distribution Layer . . . . .	127
4.6	Experimental Results and Discussion . . . . .	128
4.6.1	Framework Implementation . . . . .	128
4.6.2	System Performance . . . . .	129
4.6.3	Performance Metrics . . . . .	133
4.6.4	Complexity Analysis . . . . .	136
4.7	Conclusion . . . . .	138
<b>5</b>	<b>Semantic based Resource Discovery and Selection in the Internet-of-Things</b>	<b>139</b>
5.1	Introduction . . . . .	139
5.2	Semantics and Internet-of-Things . . . . .	140
5.3	Ontology . . . . .	141
5.4	Intelligent Resource Inquisition Framework for the Internet-of-Things . . .	142
5.4.1	Perception Layer . . . . .	143
5.4.2	Discovery Layer . . . . .	143
5.4.2.1	Mapping of Local Databases . . . . .	143
5.4.2.2	Searching Process . . . . .	148
5.4.3	Application Layer . . . . .	153
5.5	Experimental Results and Discussion . . . . .	155
5.5.1	Framework Analysis . . . . .	155

5.5.2	System Performance . . . . .	158
5.5.3	Performance Metrics . . . . .	159
5.5.4	Complexity Analysis . . . . .	163
5.6	Conclusion . . . . .	164
<b>6</b>	<b>Trust based Decision Making for Optimal Resource Selection in the Internet-of-Things</b>	<b>167</b>
6.1	Introduction . . . . .	167
6.2	Neighbor Discovery in the Internet-of-Things . . . . .	169
6.2.1	Trust Evaluation for Decision Making . . . . .	169
6.2.2	Routing . . . . .	171
6.3	Middleware Approach for Reliable Resource Selection on Internet-of-Things	172
6.3.1	Sensor-Actuator Layer . . . . .	173
6.3.2	Middleware . . . . .	173
Knowledge Acquisition . . . . .	174	
Decision Making . . . . .	179	
6.3.3	Human-Machine Interaction Layer . . . . .	187
6.4	Results and Discussion . . . . .	190
6.4.1	Framework Analysis . . . . .	190
6.4.2	Performance Evaluation of Framework . . . . .	194
6.4.3	Communication . . . . .	197
6.4.4	Complexity Analysis . . . . .	199
6.4.5	Resolving Security Issues . . . . .	201
6.5	Conclusion . . . . .	202
<b>7</b>	<b>Conclusion and Future Scope</b>	<b>203</b>
7.1	Conclusion . . . . .	203
7.2	Future Scope . . . . .	207
	<b>References</b>	<b>208</b>
	<b>List of Publications</b>	<b>225</b>

# List of Figures

1.1	Internet-of-Things [1] . . . . .	2
1.2	Building Blocks for <i>IoT</i> [3] . . . . .	3
1.3	A Graphical Forecast of <i>IoT</i> Explosion [30] . . . . .	10
2.1	Evolution: Internet-of-Things [179] . . . . .	20
2.2	Discovery and Selection on Internet-of-Things . . . . .	26
2.3	Architecture of <i>GPS</i> -based <i>ETC</i> System [57] . . . . .	27
2.4	Overview of <i>CASSARAM</i> [61] . . . . .	28
2.5	Sensor Web Enablement Architecture [65] . . . . .	29
2.6	3D-ray Launching Schematic View [66] . . . . .	30
2.7	Energy Management System for Smart Homes [74] . . . . .	32
2.8	Architecture of Ontology Mapping [76] . . . . .	33
2.9	Hypercat Ontology [77] . . . . .	33
2.10	<i>ForwarDS-IoT</i> Architecture [99] . . . . .	37
2.11	<i>DSHMP-IOT</i> Movement Prediction Stages based on <i>DSHMP</i> -Tree Network Scheme [102] . . . . .	38
2.12	Interaction Pattern: ‘Searching Around Me’ [119] . . . . .	39
2.13	Interaction Pattern: ‘Searching on My Network’ [119] . . . . .	40
2.14	Interaction Pattern: ‘Searching in Directories’ [119] . . . . .	41
2.15	Interaction Pattern: Accessing Thing Metadata [119] . . . . .	42
2.16	Architecture of Discovery Framework [36] . . . . .	43
2.17	Vertical Hand-off Decision Controller Implementation based on <i>IEEE</i> 802.21 [137] . . . . .	52

2.18	Model-Driven Approach for <i>CEP</i> in <i>SOA</i> 2.0 [140]	53
2.19	Proof-of-Concept System Functional Architecture [141]	54
2.20	Context-Aware Sensor Configuration Model (CASCoM) [41]	55
2.21	System Model [143]	55
2.22	Airport Dynamic Social Application Environment and Equipment: (a) Example of Smart Services at the Airport, (b) Airport Dynamic Social Environment, (c) Airport Dynamic Social Equipment [142]	56
2.23	Schematic Genetic Algorithm [146]	58
2.24	Representation of <i>V2V</i> and <i>V2I</i> Technologies (a) <i>V2V</i> Communication, (b) <i>V2I</i> Communication [156]	60
3.1	Similarity Measures Coefficients	80
3.2	Clustering Methodology	87
3.3	Matching Conditions	90
3.4	Similarity Computation using (a) Euclidean Distance, (b) Dice, (c) Jaccard, (d) Cosine	92
3.5	Fuzzy k-means Clustering using (a) Euclidean Distance, (b) Dice, (c) Jaccard, (d) Cosine	93
3.6	FCM Clustering using (a) Euclidean Distance, (b) Dice, (c) Jaccard, (d) Cosine	95
3.7	<i>IKm-CA</i> Clustering using (a) Euclidean Distance, (b) Dice, (c) Jaccard, (d) Cosine	96
3.8	Similarity Searches	99
3.9	Searching Time	99
4.1	Components for <i>ETC</i> [8]	117
4.2	Scenario (a) Application Area (b) Shambhu Toll Plaza, NH-1, India	117
4.3	Dataset collected from Shambhu Toll Plaza, NH-1, India	118
4.4	Context-Aware Search Optimization Framework on Internet-of-Things	119
4.5	Indices to Red-Stores (a) Text-Keyword, (b) Spatial-Temporal, (c) Value-Keyword	121
4.6	Data Modeling	128

4.7	Evaluation Results for Distinct Objective Functions . . . . .	131
4.8	Comparison between <i>IGD</i> , <i>Newton</i> and <i>Quasi-Newton</i> . . . . .	132
4.9	Performance Metrics . . . . .	134
5.1	Intelligent Resource Inquisition Framework on Internet-of-Things ( <i>IRIF-IoT</i> )	143
5.2	Sequence Steps for Partial Mapping . . . . .	145
5.3	Searching Process . . . . .	149
5.4	Fuzzy Operation . . . . .	156
5.5	Comparison of Various Approaches . . . . .	159
5.6	Simulation Environment of Toll Plaza (a) 3D Side View (b) Front View . .	160
5.7	Performance Metrics . . . . .	161
6.1	Middleware Approach for Reliable Resource Selection on Internet-of-Things	173
6.2	Static fraction of participants (a) Broadcasting Message (b) Acknowledgment Message . . . . .	178
6.3	Global Trust Evaluation . . . . .	185
6.4	Resource Discovery on the Network for its Connecting Time during (a) Global Trust (b) Local Trust . . . . .	193
6.5	Resource Discovery on the Network for its Processing Time during (a) Global Trust (b) Local Trust . . . . .	193
6.6	Comparison among algorithms for decision making with respect to (a) CPU Cycles (b) Execution time . . . . .	195
6.7	System Performance (a) Alive resources (b) Dead resources (c) Energy consumption (d) Throughput (e) Success Rate . . . . .	198

# List of Tables

1.1	Building Blocks for <i>IoT</i> . . . . .	3
1.2	Common Standards for <i>IoT</i> . . . . .	6
1.3	Comparison between Traditional Web and Discovery on <i>IoT</i> . . . . .	12
2.1	Identified Key Research Gaps . . . . .	22
2.2	Approaches for Resource Discovery on <i>IoT</i> . . . . .	44
2.3	Literature Survey on Clustering . . . . .	48
2.4	Summary of Literature Survey . . . . .	62
3.1	Experiment for Queries at Threshold = 5 . . . . .	98
4.1	Objective Functions( <i>OF</i> ) . . . . .	129
4.2	Objective Functions( <i>OF</i> ), Elapsed Time ( <i>ET</i> ) . . . . .	132
5.1	Fuzzy Operations . . . . .	142
5.2	Degree of Match . . . . .	148
5.3	Weight Calculation . . . . .	151
5.4	Fuzzy Control Rules . . . . .	156
6.1	Simulation Parameters . . . . .	191
6.2	Comparison for Computation of Resources by Various Approaches . . . . .	192
6.3	Comparison among approaches . . . . .	196
7.1	Comparison among Proposed Techniques . . . . .	205

# List of Algorithms

3.1	Fuzzy k-means Clustering Algorithm ( <i>FKM</i> ) . . . . .	83
3.2	Fuzzy c-means Clustering Algorithm ( <i>FCM</i> ) . . . . .	85
3.3	Iterative K-means Clustering Algorithm ( <i>IKm-CA</i> ) . . . . .	87
4.1	Newton Algorithm . . . . .	112
4.2	Quasi- Newton Algorithm . . . . .	115
4.3	Multi-modal Search Algorithm . . . . .	123
4.4	Iterative Gradient Descent Algorithm ( <i>IGD</i> ) . . . . .	127
5.1	Crawling Algorithm . . . . .	150
5.2	Semantic Matchmaking Using Bipartite Graph . . . . .	152
5.3	Optimized <i>SMEBG</i> Algorithm . . . . .	153
5.4	Reasoning Algorithm . . . . .	154
5.5	Poisson Searching Algorithm . . . . .	158
6.1	<b>Neighbor Discovery Algorithm (<i>NDA</i>)</b> . . . . .	180
6.2	<b>Trust Evaluation Algorithm</b> . . . . .	186
6.3	<b>Fuzzy-Decision Algorithm (<i>FDA</i>)</b> . . . . .	188
6.4	<b>Hybrid M-Gear Algorithm</b> . . . . .	189

# List of Symbols

<b>Symbols</b>	<b>Meaning</b>
$c$	Clusters
$Y$	Partition Matrix
$Z$	Set of Objects
$sed$	Squared Euclidean Distance between two Objects
$O$	Domain
$Prob$	Problem
$wt_{m,n}S_{BoS}$	Weight
$Q_a$	Query
$s_m, t_n$	Pair
$simCos(G, H)$	Cosine based Similarity Value
$G, H$	Vectors
$C_1, C_2$	Concepts
$simDice(G, H)$	Dice based Similarity Value
$simEuc(G, H)$	Euclidean based Similarity Value
$simJac(G, H)$	Jaccard based Similarity Value
$V_a[n]$	Array of Vectors
$n$	Maximum Number of Clusters
$cent_i[m]$	Centroid of Clusters
$c_n[m]$	Clusters
$cent_i[m]$	Centroid Array
$best[m]$	Best Matched Solution
$A_c, A_{c_1}$	Associated Cluster for Array of Vectors
$sim(A, B)$	Similarity Function
$best_n[m]$	Updated Vector of Clusters
$L_{final}$	Reassigned New Clusters
$total_{new}$	Final Vector of Cluster Formed
$total_{new}$	Final Clusters
$F_m(I, J)$	Matching Function

$Th$	Threshold
$(A_c[m])$	Associated Clusters
$m_{d_{ij}}$	Membership Degree
$L_j$	Degree of Sample
$\acute{U}$	Fuzzy Partitioned Matrix
$E_{d_{ji}}$	Euclidean Distance
$cc_i$	Centroid Cluster
$s$	Exponent
$f_{l,k}$	Degree of Membership
$g_l$	Prototype of Centroid of Cluster
$F$	Membership Matrix
$M \times N$	Total Number of Resources
$dist$	Distance among the Centroid of Membership Matrix
$cent_{cal}$	Total Iterations of Centroids
$O(N_Z + K)$	Storage
$N_Z$	Number of Resources
$C$	Number of Clusters
$N$	Data Points
$D$	Dimension
$I$	Iterations
$O(N)$	Complexity
$N_{nd}$	Nodes
$N_{att}$	Attributes
$dist_{cal}$	Distance Calculated among Clusters
$comp$	Total Number of Comparison
$cent_{cal}$	Calculated Centroids of Clusters
$J_e$	Iteration
$N_w$	Total Comparisons among Resource's Distances
$Loc$	Location of Sampled Value
$Ptrn$	Format of Sampling
$T$	Time of Sampling
$Val$	Actual Value
$T_j$	Time
$objId$	Identification of Object
$objDesc$	Description of Object

$PhyAddr$	Physical Address
$k$	Keyword
$Idxnode$	Address of the Child Node
$Ptr$	Pointer
$PT$	Partition Table
$IST$	Instance Sequence Time
$Idx_i$	Node Server
$G_A$	Geographical Area
$Idxnode$	Index Node Server
$G_sJ$	Grid-sketched by the Dynamic Object
$Q_a$	Query
$Q_{bool}$	Storing Atomic Searches
$(z)$	Output
$s(z)$	Objective Function
$g_0$	Initial Critical Point
$G_d$	Gradient Descent
$t_r$	Iteration
$\delta_{t_r}$	Step Size
$t_r$	Trajectory
$L_s$	Level Set
$\theta$	Point-to-set Mapping of Critical Points
$I_A$	Armijo's Rule
$M_{in}$	Inner Mapping
$M_{out}$	Outer Mapping
$z$	Search Space
$maxitr$	Maximum Iteration
$niter$	Number of iterations
$\delta$	Step Size
$d_{xmin}, d_x$	Change in Step Size
$tolr$	Tolerance
$O_n$	Ontology
$C_s$	Conceptual Sets
$H_{C_s}$	Corresponding Hierarchy with respect to subclasses
$\mathcal{I}$	Instance
$\mathcal{R}$	Relation

$\tilde{A}$	Axioms to infer knowledge
$H_{C_s\check{R}}$	Hierarchy for relations
$\check{t}, k$	Terms
$v$	Vocabulary
$ParMap$	Partial Mapping Function
$S_c$	Similarity Computation
$\acute{I}$	Iteration
$S_{cEntity}$	Entity Equatability
$g, y, E$	Entities
$S_{cDef}$	Definite Equatability
$S_{cStrEqu}$	String Equatability
$j, e$	Strings
$S_{cDice}$	Dice Coefficient
$T, K$	Two sets of Entities
$S_{cSet}$	Similarity Set
$S_{cAgg}$	Identity Agglutination
$D_m$	Degree of Match
$Q_{in}$	List of Input Concepts
$M_s$	Matching Sets
$List_i$ and $List_j$	Vertical Lists
$M_{s_{deref}}$	Unseen/blacklisted Sets
$W_a, W_b, W_c$	Weight
$max(wt_i)$	Maximum Weighted Edge
$G_B$	List of Matching Sets
$V_x$	Vertices of Entities
$H_M$	Hungarian Match
$G_B$	Bipartite Graph
$R_c$	Rules
$A_{xi}$	Axiomatic Triples
$T^*$	Searching Time
$P_{T^*}(n)$	Poisson Distribution
$n$	Number of Vehicles
$\lambda$	Total Time
$\beta$	Approximate Searching Time
$F^*(\lambda)$	Probability Density Function

$V_h$	Number of Vehicles
$S_T$	Search Time Set
$P_{cbit}$	Power Consumption
$Y_{i,j}$	Number of Hops among End Points
$Energy_{resource}$	Energy consumed in network
$Energy_{nodes}$	Power consumption at the end points
$S_1$	Sum of Data Packet
$S_2$	Sum of Data Packets Generated by the Source
$P_{loss}$	Packet Loss
$P_{gen}$	Generated Packets
$P_{rec}$	Received Packets
$P_{size}$	Total Size of Packets
$\tilde{N}$	Number of Search Iterations
$p_{k,l}$	Position
$l$	Step
$Y_{k,l}$	Point Observation
$F_{p_{k,l}}$	Spatial Function
$\epsilon_{k,l}$	Gaussian noise
$Y_{k,l}$	Gaussian
$F_{k,l}$	Latent Variable
$CV_{NN}$	Covariance Matrix
$p(F_{N,l})$	Complete Prior Distribution
$R_{N,l}$	Regression
$p_{*,l}$	Test Point
$p_{k,l}, Y_{N,l}$	Training Data Set
$\mu_{*,l}$	Predictive Mean Function
$\sigma_{*,l}$	Predictive Covariance Function
$CR_k$	Communication Range
$r_{k,l}, s_{k,l}$	Local Hyper-Parameters
$ER_k$	Effective Range
$B_k$	Predictive Neighbor Set
$N_k$	Cardinality
$P_{ack}$	Probability
$s$	Slot
$\lambda$	Rate

$j_l$	Number of Participants Joining System
$\alpha$	Load Factor
$rv_0, rv_1$ and $rv_2$	Number of Slots with Acknowledgment Message
$EXP$	Expected Value
$\sigma_0$	Variance
$F_s$	Frame Size
$pd$	Prior Distribution
$\mu$	Predictive Mean Function
$MyIndex$	Parent Index Service
$NS$	Neighbor Set
$T_s$	Trust Score
$NS_{sub}$	Shortlisted Resources
$TT_{i,j}$	Transaction Table
$ST_{i,j}$	Successful Transaction
$PT_{i,j}$	Positive Transaction
$TTL$	Time to Live
$R$	Reputation
$DT_{i,j}$	Direct Trust
$\gamma, \chi$	Behavior of Resources
$DT_{i,j}^{e-1}$	Direct Trust at Latest Time Duration
$DT_{i,j}^{\Delta e}$	Direct Trust at Previous Time Duration
$C_{th}$	Communication Time
$DT_{i,j}^e$	Direct Trust Evaluation
$RS_{i,j}$	Number of Request Solicits
$PS_{i,j}$	Number of Positive Solicits
$C_{th}$	Threshold
$IT_{i,j}$	Indirect Trust
$T_{i,j}$	Total Trust Evaluation
$W_d$	Value for Weight of Direct Trust
$W_{ind}$	Value for Weight of Indirect Trust
$NS_{i,l,m,\dots,j}$	Neighbor Sets
$a[T_s]_1$	Array with Trust Score equal to 1
$ns_p$	Participating Resources
$i$	Counter
$ns_{lp}[i], ns_{gp}[i]$	Resultant Array

$nS_{newlp}[i]$ ,	Sorted Array
$nS_{newgp}[i]$	
$p_{snew}[i], q_{snew}[i]$	Sum Arrays
$p_{avail}[i], q_{avail}(i)$	Shortlisted Available Array
$ns_p$	Participants
$BS$	Base Station
$CH$	Cluster-Head
$area$	Area
$egy$	Energy
$gatewayresource$	Gateway Resources
$region$	Region
$table$	Data Table
$T_{storage}$	Storage Complexity
$R_d(DT)$ ,	Storage Repositories
$R_d(IDT)$	
$O(N^2)$	Number of Iterations
$R_d(IDT)$	Storage repositories for the Gathered Data by Indirect Trust
$R_d(sortIDT)$	Storage Repositories for the Gathered Data by Evaluating Gathered Data after Sorting

# List of Abbreviations

<b>Abbreviations</b>	<b>Description</b>
<i>AODV</i>	Ad-hoc On-demand Distance Vector
<i>AMQP</i>	Advanced Message Queuing Protocol
<i>APTS</i>	Advanced Public Transportation System
<i>ARTS</i>	Advanced Rural Transportation System
<i>ATMS</i>	Advanced Traffic Management Systems
<i>ATIS</i>	Advanced Traveler Information System
<i>AVCS</i>	Advanced Vehicle Control System
<i>ACO</i>	Ants Colony Optimization
<i>API</i>	Application Programming Interface
<i>ABC</i>	Artificial Bee Colony
<i>AI</i>	Artificial Intelligence
<i>APFs</i>	Artificial Potential Fields
<i>ANPR</i>	Automatic Number Plate Recording
<i>BS</i>	Base Station
<i>BGA</i>	Binary Genetic Algorithm
<i>BG</i>	Bipartite Graph
<i>BLE</i>	Bluetooth Low Energy
<i>CLARANS</i>	Clustering Large Applications based on RANdomized Search
<i>CVO</i>	Commercial Vehicle Operations
<i>CORBA</i>	Common Object Request Broker Architecture
<i>CEP</i>	Complex Event Processing
<i>CoAP</i>	Constrained Application Protocol
<i>CAL</i>	Context Acquisition Layer
<i>CASL</i>	Context Annotation-Search Layer

<i>CARD</i>	Context Aware Resource Discovery
<i>CASOF-IoT</i>	Context-Aware Search Optimization Framework on the Internet of Things
<i>CASSARAM</i>	Context-Aware, Sensor Search, Selection and Ranking Model
<i>CPU</i>	Control Processing Unit
<i>CASAGRAS</i>	Coordination and Support Action for Global RFID-related Activities and Standardization
<i>CA</i>	Cultural Algorithm
<i>DDS</i>	Data Distribution Service
<i>DM</i>	Decision-Making
<i>DSRC</i>	Dedicated Short Range Communications
<i>DeitY</i>	Department of Electronics and Information Technology
<i>DHT</i>	Distributed Hash Table
<i>DNS</i>	Domain Name System
<i>ETC</i>	Electronic Toll Collection
<i>ETP</i>	Electronic Toll Plaza
<i>EXI</i>	Efficient XML Interchange
<i>EAR</i>	Energy Aware Routing
<i>EPC</i>	Electronic Product Code
<i>ESB</i>	Enterprise Service Bus
<i>ETSI</i>	European Telecommunications Standards Institute
<i>ForwarDS-IoT</i>	Federated Discovery Service
<i>FIB</i>	Forwarding Information Base
<i>FCM</i>	Fuzzy c-Means
<i>FCL</i>	Fuzzy Control Logic
<i>FKM</i>	Fuzzy k-Means
<i>FDA</i>	Fuzzy-Decision Algorithm
<i>GA</i>	Genetic Algorithm
<i>GERI</i>	Global Enterprise Resource Integration
<i>GPS</i>	Global Positioning System
<i>HSMM</i>	Hidden Semi-Markov Model
<i>HPC</i>	High Performance Cluster
<i>HMIL</i>	Human-Machine Interaction Layer
<i>HTTP</i>	Hyper Text Transfer Protocol

<i>IMPPF</i>	Improved Firefly Algorithm
<i>IRIF-IoT</i>	Intelligent Resource Inquisition Framework on Internet-of-Things
<i>IEEE</i>	Institute of Electrical and Electronics Engineers
<i>ITS</i>	Intelligent Transportation System
<i>IETF</i>	Internet Engineering Task Force
<i>IDC</i>	International Data Corporation
<i>IoT</i>	Internet-of-Things
<i>IPSec</i>	Internet Protocol Security Protocol Suite
<i>IPv4</i>	Internet Protocol Version 4
<i>IoT-DS</i>	Internet-of-Things Directory System
<i>IPs</i>	Internet Protocols
<i>IGD</i>	Iterative Gradient Descent
<i>IPv6</i>	Internet Protocol Version 6
<i>IKm-CA</i>	Iterative K-means Clustering Algorithm
<i>JMS API</i>	Java Message Service API
<i>KA</i>	Knowledge-Acquisition
<i>LEACH</i>	Low-Energy Adaptive Clustering Hierarchy Protocol
<i>LTE-A</i>	Long Term Evolution-Advanced
<i>LoWPAN</i>	Low-power Wireless Personal Area Networks
<i>MQTT</i>	Message Queue Telemetry Transport
<i>MeitY</i>	Ministry of Electronics and Information Technology
<i>mDNS</i>	Multicast DNS
<i>MARRS-IoT</i>	Middleware Approach for Reliable Resource Selection on Internet-of-Things
<i>M2M</i>	Machine-to-Machine
<i>MOD-LEACH</i>	MODified Low-Energy Adaptive Clustering Hierarchy
<i>NDN</i>	Named Data Networking
<i>NH-1</i>	National Highway-1
<i>NHAI</i>	National Highways Authority of India
<i>NICT</i>	National Institute of Information and Communications Technology
<i>NFC</i>	Near Field Communication
<i>NDP</i>	Nebula Data Plane
<i>NVENT</i>	Nebula Virtual and Extensible Networking Techniques
<i>NDP</i>	Neighbor Discovery Protocol

<i>NAT</i>	Network Address Translation
<i>NCore</i>	Network Core
<i>NWGN</i>	New Generation Network
<i>NPN</i>	New Paradigm Network
<i>NXGN</i>	Next Generation Network
<i>ONS</i>	Object Name Service
<i>OF</i>	Objective Functions
<i>OS</i>	Operating System
<i>OGC SOS</i>	Open Geospatial Consortium's Sensor Observation Service
<i>OWL</i>	Web Ontology Language
<i>PDR</i>	Packet Delivery Rate
<i>PSO</i>	Particle Swarm Optimization
<i>PIT</i>	Pending Interest Table
<i>PC</i>	Personal Computer
<i>P2P</i>	Point-to-Point
<i>PEGASIS</i>	Power-Efficient Gathering in Sensor Information Systems
<i>QoS</i>	Quality-of-Service
<i>RFID</i>	Radio Frequency Identification
<i>RUNES</i>	Reconfigurable Ubiquitous Networked Embedded Systems
<i>REST</i>	Representational State Transfer
<i>Red-Stores</i>	Resource Data Stores
<i>RTOS</i>	Real Time Operating Systems
<i>RDFs</i>	Resource Description Frameworks
<i>ROA</i>	Resource Oriented Architecture
<i>RPL</i>	Routing over Low Power and Lossy Networks
<i>SE</i>	Search Engine
<i>SOL</i>	Search Optimization Layer
<i>SDL</i>	Selection-Distribution Layer
<i>SMEBG</i>	Semantic Matchmaking Engine using Bipartite Graph
<i>SAL</i>	Sensor-Actuator Layer
<i>SOA</i>	Service Oriented Architecture
<i>SPDP</i>	Service Publish and Discovery Protocol
<i>SSDP</i>	Simple Service Discovery Protocol

<i>SIoT</i>	Social Internet-of-Things
<i>SQL</i>	Structured Query Language
<i>SDN</i>	Software Defined Network
<i>SOC</i>	System-On-a-Chip
<i>SOS</i>	System-Oriented Services
<i>TEDI</i>	Traffic Network Editor
<i>TCP/IP</i>	Transmission Control Protocol/Internet Protocol
<i>transID</i>	Transaction Identification
<i>uCode</i>	Ubiquitous Codes
<i>UDDI</i>	Universal Description, Discovery and Integration
<i>UWB</i>	Ultra Wide Bandwidth
<i>URL</i>	Uniform Resource Locator
<i>URI</i>	Uniform Resource Identifier
<i>UDP</i>	User Datagram Protocol
<i>VNS</i>	Variable Neighbor Search
<i>VSM</i>	Vector Space Model
<i>WS</i>	Web Service
<i>WSDL</i>	Web Service Description Language
<i>WiFi</i>	Wireless Fidelity
<i>WWW</i>	World Wide Web
<i>W3C</i>	World Wide Web Consortium
<i>XMPP</i>	Extensible Messaging and Presence Protocol

---

# CHAPTER 1

---

---

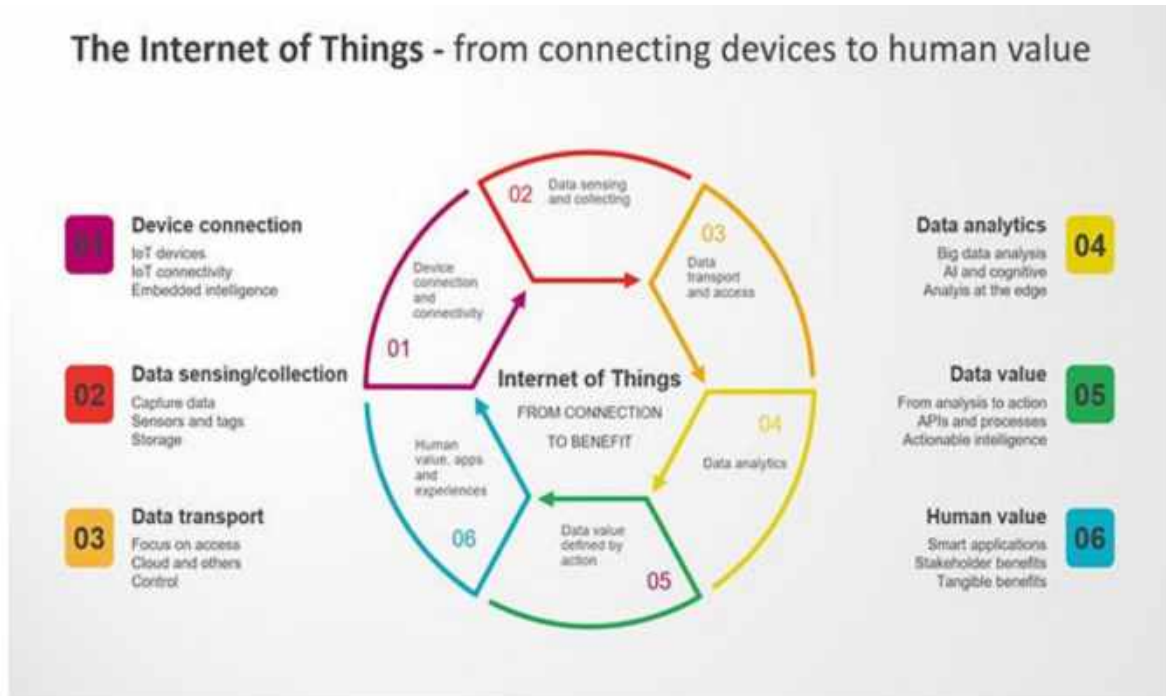
## Introduction

---

This chapter provides a general overview on Internet-of-Things (*IoT*), its basic building blocks and common standards along with its current landscape. *IoT* due to distributed systems, distributed intelligence, communication and computational capabilities pose various challenges to resource discovery and selection. This is due to resources' diverse nature, their capabilities, properties and communication technologies, which would together add to the data complexity, and hence, affects the discovery. This chapter also summarizes the challenges to the resource discovery and provides the organization of the thesis.

### 1.1 Internet-of-Things

*IoT* is a paradigm where resources are embedded with capabilities such as identification, sensing, networking, communication and computation that will allow them to communicate among each other to provide services over the Internet (*see* Figure 1.1). Ultimately, *IoT* resources will be ubiquitous, context-aware and will enable ambient intelligence. Such transformation from real world objects into intelligent virtual resources that can be realized with identification technology, path technology, communication technology, *etc.* [1].

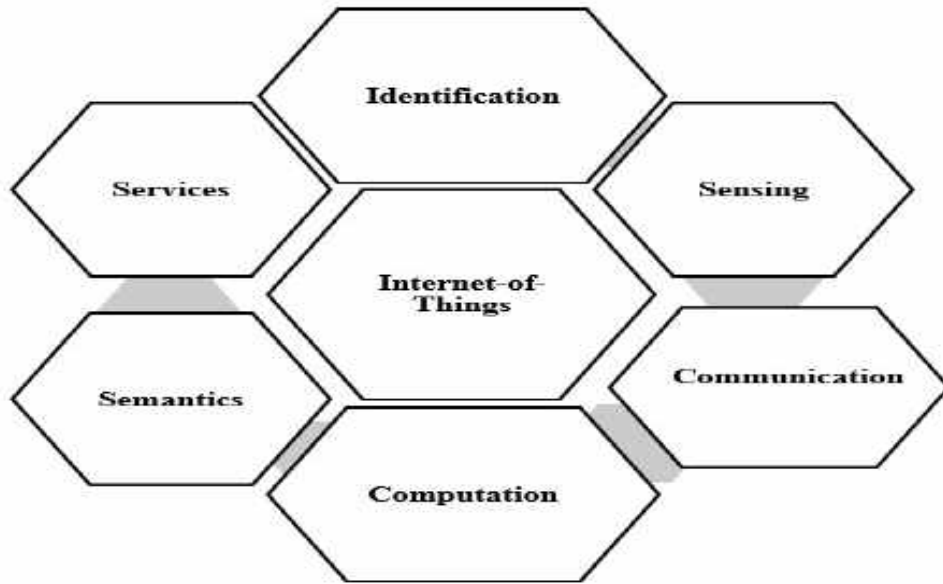


**Figure 1.1:** Internet-of-Things [1]

In short, *IoT* envision convergence between Internet and Sensor Networks that would allow direct Machine-to-Machine (*M2M*) communication over the Internet. It would bring more resources online and allowing them to participate in the web as a vast network of autonomous, self-organizing devices [2].

### 1.1.1 Building Blocks

In order to gain a better insight into the real meaning of the *IoT*, it is required to understand the building blocks that are needed to deliver the functionality of the *IoT* as illustrated in Figure 1.2. These blocks will confirm *IoT* to be more efficient, scalable, reliable, secure and trustworthy. Table 1.1 presents the building blocks with examples.



**Figure 1.2:** Building Blocks for *IoT* [3]

**Table 1.1:** Building Blocks for *IoT*

<i>IoT</i> Building Blocks		Samples
Identification	Naming	EPC, uCode
	Addressing	Internet Protocol Version 4 (IPv4), IPv6
Sensing		Smart Sensors, Wearables sensing devices, Embedded sensors, Actuators, RFID Tag
Communication		RFID, NFC, UWB, Bluetooth, BLE, IEEE 802.15.4, Z-Wave, WiFi, WiFiDirect, LTE-A
Computation	Hardware	Arduino, Intel Galileo, Raspberry Pi, Smart Phones
	Software	OS(Contiki, TinyOS, LiteOS, RIoTOS, Android); Cloud(Nimbits, Hadoop)
Semantic		RDF, OWL, EXI
Service		Shipping, Smart Grid, Smart Home, Ubiquitous Smart City

**Identification:** The identification of the services on *IoT* platform is imperative in order to name and match them. Moreover, identification methods to differentiate name and

address of the object are not globally unique, hence, addressing of the objects helps in their identification uniquely. Also, these objects use Internet Protocols (*IPs*) that are public in nature. For the same, different identification methods exist such as *EPC* and ubiquitous codes (*uCode*) [4] that provide a clear identification for objects. The Low-power Wireless Personal Area Networks (*6LoWPAN*) [5,6], provides a compression mechanism over Internet Protocol version 6 (*IPv6*) headers that makes this addressing appropriate for low power wireless networks.

**Sensing:** The sensing intends to acquire data from the resources connected to the network and forward it to the database or cloud, where it is analyzed to perform particular actions [6]. The resources as sensors, actuator or smart devices provides the required data to the end users and are connected to the centralized server or portal.

**Communication:** The communication technologies on *IoT* connect non-homogeneous resources together to deliver specific smart services. Typically, the resources should operate using low power in the presence of lossy and noisy communication links. Examples of communication protocols used for the *IoT* are Wireless Fidelity (*WiFi*), Bluetooth, Institute of Electrical and Electronics Engineers (*IEEE*) 802.15.4, Z-wave, and Long Term Evolution-Advanced (*LTE-A*). Some specific communication technologies like Radio Frequency Identification (*RFID*), Near Field Communication (*NFC*) and Ultra-Wide Bandwidth (*UWB*) are also used [7–9].

**Computation:** The processing units and software applications represent the computational ability of the *IoT*. For the purpose, number of hardware platforms such as Raspberry PI, Arduino, Intel Galileo, *etc.* are developed. Among these platforms, Operating Systems (*OS*) are vital since they run for the whole activation time of a device. There are several Real-Time Operating Systems (*RTOS*) that are good candidates for the development of *RTOS*-based *IoT* applications, *e.g.*, the Contiki *RTOS* [10], TinyOS [11], LiteOS [12] and Riot *OS* [13]. The cloud platforms facilitate smart objects to send their data to the cloud, for big data to be processed in real-time, and eventually for end-users to benefit from the knowledge extracted from the collected big data [14].

**Semantics:** Semantics on *IoT* includes ontologies, contexts, and structured metadata that can make intelligent sensors to be seamlessly, securely, and trustworthy interconnected for

enabling automated high-level smart applications. For this, it undergoes various processes such as data modeling, its representation, interpretation and reasoning for decision making [15].

**Services:** *IoT* services are classified into four classes, *i.e.*, Identity-related, Information Aggregation, Collaborative-Aware and Ubiquitous Services [16]. Among them, the foremost service, *i.e.*, Identity-related consists of two major components, *namely*, things and read device. The former is equipped with identification identifier like *RFID* whereas latter reads the identity of things and make a request to name resolution server to access more detailed information about particular device. Therefore, Identity-related services can read, transmit or actively send information to other connected devices. The service which is responsible for the collection of raw sensory data, processing of acquired data, transmitting and reporting that data to the *IoT* based application is Information Aggregation. From the application, Collaborative-Aware Services obtain this data for further analysis to take decisions accordingly. Finally, the last service, *i.e.*, Ubiquitous targets provision of the Collaborative-Aware Services to the end user irrespective of time [17–19]. For example, Internet-of-Vehicles [20] represents a typical application of *IoT* in Intelligent Transportation System (*ITS*) that set future trends as converging technology like Electric Vehicle, Electric Smart Grid and Autonomous Vehicle [21].

### **1.1.2 Common Standards**

The number of groups such as World Wide Web Consortium (*W3C*), Internet Engineering Task Force (*IETF*), Institute of Electrical and Electronics Engineers (*IEEE*) and European Telecommunications Standards Institute (*ETSI*) have developed and defined *IoT*' protocols to facilitate seamless interoperability between things, gateways, applications and users that will enable scalability, cross things added value applications, services and cost optimization. The *IoT*' protocols are broadly categorized in to four types, *namely*, application, service discovery, infrastructure and other influential protocols meant for semantic, multi-layer frameworks, security and vertical specific interface standards. However, not all of these protocols have to be bundled together to deliver a given *IoT* application. Moreover, based on the nature of the *IoT* application, some standards may not be required to be supported in an

application. Some of the most prominent protocols are summarized in Table 1.2.

**Table 1.2: Common Standards for IoT**

Application Protocols		DDS	CoAP	AMQP	MQTT	MQTT-SXMP	HTTP REST
Service Discovery		mDNS			DNS-SD		
Infrastructure Protocol	Routing Protocol	RPL					
	Network Layer	6LoWPAN				IPv4/IPv6	
	Link Layer	IEEE 802.15.4					
	Physical/Device Layer	LTE-A	EPC global		IEEE 802.15.4	Z-Wave	
Influential Protocols		IEEE 1883.3, IPSec				IEEE 1905.1	

**Data Distribution Service:** Data Distribution Service (*DDS*) is data-centric that allows fine and extensive control of *QoS* parameters, including reliability, bandwidth, delivery deadlines, and resource limits. Also, it uses interoperable information exchanges and follows a server-less approach with direct end to end communication, making it independent from the platform. The *DDS* helps in transmitting information using transportation schemas like multicast, Transmission Control Protocol/Internet Protocol (*TCP/IP*), etc. It also supports multiple physical connections and has automatic route finding [22].

**Constrained Applications Protocol:** Constrained Applications Protocol (*CoAP*) is a web based client/server model which helps unassisted devices to send information *via* alter routes. It uses User Datagram Protocol (*UDP*), broadcasting and multicast addressing mechanisms for sending information in one to one protocol. *TCP* is not supported by *CoAP*. It also helps in finding routes for exchanging data by connectionless communication and thus supporting route discovery and content negotiation [23].

**Advanced Message Queuing Protocol:** In Advanced Message Queuing Protocol (*AMQP*),

the replacement of proprietary and non-interoperable messaging systems in the financial sector results into a message-centric brokering protocol, with key features like message orientation, queuing, routing, reliability and security. It works at the application layer with an underlying reliable transport layer providing message oriented communication, flow controlled delivery of messages, encryption and authentication [24].

**Message Queue Telemetry Transport:** Message Queue Telemetry Transport (*MQTT*) is an easily available protocol that has simple and basic model that works through a central broker for passing messages among end users *via IP* connections. It is well suited for constrained environments and supports the hierarchical file system structure. It uses asynchronous communication for sending offline messages to devices. It also helps in storing and forwarding the most recent messages. The various versions are available to counteract particular problems [25].

**Extensible Messaging and Presence Protocol:** Extensible Messaging and Presence Protocol (*XMPP*) is decentralized customer server model which uses text messages or applications for communicating massively to cluster of nodes and is broker less. The binary data are encoded with *base64* before transmission and thus providing security to devices during large, potentially unwanted complicated traffic [26].

**Representational State Transfer:** The lightweight protocol, Representational State Transfer (*REST*) is stateless, end to end and cached storage for communication among devices to the cloud *via TCP/IP*. It is independent of language and operating system having asynchronous, publish or subscribe of information exchanges and it does not support cookies [27].

**Java Message Service API:** Java Message Service API (*JMS API*) helps in creating, reading, sending and receiving messages among two or more clients. It intends to separate application and transport layer functions and allows the communication between different components of distributed applications to be loosely coupled, reliable and asynchronous over *TCP/IP*. *JMS* supports both point-to-point and publish/subscribe models using message queuing and durable subscription [28].

### 1.1.3 Current Landscape

The idea of transformation of the Internet as a combination of classic networks and networked objects has resulted in *M2M* interaction without human intervention by means of integrating various processes such as sensing, communication, data analytics, *etc.* Such transformation would offer great research opportunities as new means of communications, living, entertainment, interactions, sensing and identification [29], but here, the question arises that why it took so long for this transformation though, *IoT* is at its peak of inflated expectations [30, 31]. To identify the problems for the realization of *IoT* through the processes, the various research domains are identified, *namely*, identification technology, architecture technology, communication technology, network topology and discovery, algorithmic approach, hardware solutions, data and signal processing, searching techniques, energy optimization, standardization, security and privacy [32–34]. These research domains, due to technological diversity, raise conflicts such as identity and repository management, delegating policy decisions, context switching architectures, autonomic computing and networking; that are envisaged to be resolved by end of 2025.

Among these domains, the thesis evaluates the current state of *IoT* and argues for a paradigm shift addressing why and how discovery can significantly impact the future of *IoT* and moreover, becomes a necessary component for the *IoT* success story [35]. It is due to the estimated explosion of resources on *IoT* in the coming years which raise the need to rethink how it can deliver value to the end-user. The resources required to be discovered in order to allow users to build *IoT* apps, services and applications using intelligent resources without the need for a priori knowledge of these resources.

## 1.2 Resource Discovery and Selection on Internet-of Things

*IoT* is an area which enables heterogeneous resources to sense, process and interpret *via* Internet-connected infrastructures. However, the exponential growth of resources would add to the *IoT* data which is not limited to sensors and machines but data from social

networks, the web, and other user submitted physical observations and measurements. Such huge amount of data from real or virtual world will be available globally and in vast amounts which is to be shared among applications and devices for event detection, context and situational awareness based decision making, enhanced service creation, and driving event-based actuation without human intervention. Such a system has to be reactive, efficient, and effective as it will have to continuously respond to changes in the user's situation. However, such a system can also be proactive where its behavior is based on predicted situations evaluated with some level of confidence and probability. Therefore, there is a need to understand the mechanisms to facilitate automatic discovery and selection of the resources on *IoT* platforms [36]. These mechanisms are discussed in this sub-chapters.

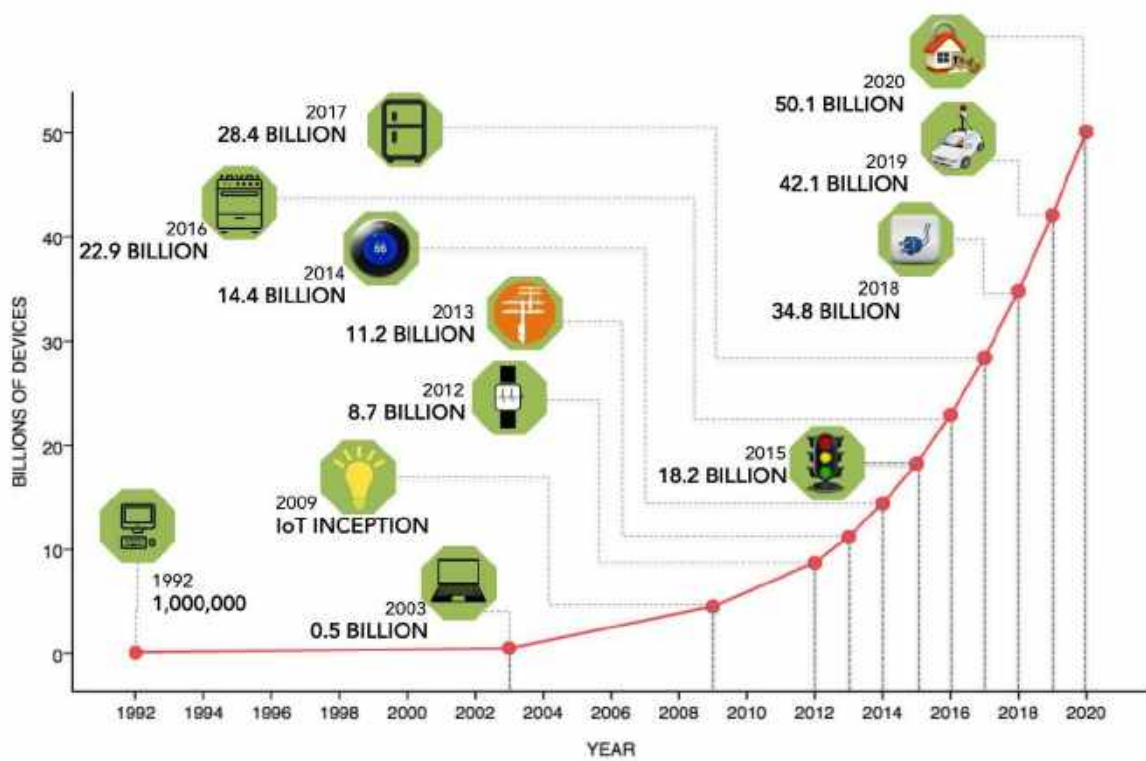
### **1.2.1 Resources and their Graphical Forecast**

A resource (refers to physical objects; the terms object, device, resource, entity, and things are used interchangeably) is either software or hardware with fundamental characteristics such as physical embodiment, unique identifiers, offered service, location, processed information, operating system, languages and modes of communication. Moreover, they have either local or global domain having both physical and digital representation. In order to facilitate interaction, communication, and computation, resources are required to interact with the environment and among themselves [37].

The connected *IoT* devices or resources as predicted by International Data Corporation (*IDC*) projects, would grow to 212 billion that would drive *IoT* and the associated ecosystem to be a \$1.7 trillion market by 2020 [30]. Figure 1.3, presents a graphical forecast of *IoT* explosion over the coming years as estimated by *Cisco*. Such an increase in the number of resources would produce huge amount of data through processes defined for the realization of *IoT* and hence, limits the discovery [38].

### **1.2.2 Resource Discovery**

The resource discovery mechanism enables users from different applications to access *IoT* data wherein users does not require to know the originating source, location, time and



**Figure 1.3:** A Graphical Forecast of *IoT* Explosion [30]

description of the data. It leads discovery mechanism to emerge as a challenging task since activities such as data acquisition, modeling, integration, assessment and reasoning on *IoT* varies with respect to data providers and end publishers or brokers. These activities helps to facilitate data linking, knowledge representation and context-driven search and therefore, on their basis, resource discovery mechanism is broadly categorized in to two successive loops, *namely*, foraging and sense-making [39, 40]. In former loop, originating sources are identified and assessed for knowledge extraction which is further formatted into consumable form. In latter, *i.e.*, sense-making loop, the extracted knowledge is analyzed, interpreted and exploited for the provision of service in accordance to a particular query.

### 1.2.3 Motivation behind Resource Discovery

An implicit property of *IoT* is heterogeneity imposed by plethora of resources. These resources have diversity with respect to communication, computational capabilities, representation, storage, search types and data formats. Such diversity leads to unmanaged big

data driven by its velocity, variety, value, and volume which poses significant challenges to realize the vision of *IoT* [41]. Moreover, to ensure availability of the resources, data would require to be efficiently stored in widely distributed, heterogeneous information systems. It would lead to another challenge related to data retrieval from the information systems which is a non-trivial task without a common machine-readable data representation. Hence, interoperability will need to be addressed to design efficient mechanisms for discovering available resources and capabilities.

Finally, there is a scope for improvement of innovative low-footprint hardware solutions which are embedded into vertical software middleware silos of *IoT* solution stacks. In these silos, data by itself is of limited value unless it is computed with other related data and context. Since, the tremendous amount of data is generated through advancement in technologies, it is necessary to find the needle across zettabytes of multiple haystacks. Here, the challenge is to describe the precision despite increasing heterogeneity and complexity in the data. This is where discovery combined with analytics in the *IoT* can deliver new insights from the data produced by things [42].

#### **1.2.4 Traditional Web Search versus Internet-of-Things Resource Discovery**

In Traditional Web, the basic unit of addressing in World Wide Web (*WWW*) is the Uniform Resource Locator (*URL*) which is accessed over the application protocol, Hyper Text Transfer Protocol (*HTTP*). Traditional resource discovery in *WWW* is performed by search engines which dispatch web crawlers to pull the requested data. On the other hand, resource discovery on the *IoT* are more susceptible to interference, loss of connectivity being power constrained and connected through lossy wireless networks. Moreover, these resources are generally deployed in semi-closed infrastructure or network [43]. The comparison between web search for pages and resource search in *IoT* with respect to *CoAP* is summarized in Table 1.3.

**Table 1.3:** Comparison between Traditional Web and Discovery on *IoT*

<b>Traditional Web Search</b>	<b>IoT based Resource Discovery</b>
Target websites are fetched using Web Crawlers and provided to interested nodes.	Target devices push their services on the resource discovery.
Nodes can fetch the complete Websites, links, Uniform Resource Identifier ( <i>URI</i> ) meta-data and application content.	No application content is transferred to node except <i>URI</i> , meta-data and hyper-links.
The transfer protocol is <i>HTTP</i> .	The transfer protocol is <i>CoAP</i> .
Resources can be discovered in the global scope throughout Internet.	Resources can be discovered locally within the scope of resource directory.
Resource discovery results are ranked.	Efficient ranking is not supported.
Resource discovery results are not machine readable.	Resource discovery results are machine readable in Link Format.

### 1.2.5 Evaluation Metrics for Resource Discovery

The *IoT* is a dynamic global network infrastructure that is composed of billions of interconnected resources, consequently, the scope of the research for discovery techniques is broad. To gain a quick in-depth overview of the research, it would be better to define some prominent metrics or dimensions which are explained as follows [44].

**Data format:** It defines the format in which data is to be represented.

**Access approach:** It alludes to how customers of the search functionality can get to the output of the query.

**Search type:** It tells about the type of the fundamental search techniques based on which the discovery frameworks are designed, such as keyword-based search. The core strategies utilized are critical in deciding the adequacy of the search.

**Scale of experiment:** It presents the scale of the experiment considered in a specific research work, *e.g.*, the number of sensors and entities, amount of data, *etc.*

**Dynamicity:** The dynamicity of the resources refers to their changing factors such as status,

location, generated value, faults in hardware, *etc.* These factors have direct or indirect impact on the performance of the resources, therefore, discovery mechanism should provide support for handling their consequences.

**Architecture:** It defines the type of platform such as centralized or distributed for which discovery approaches are designed.

**Implementation:** It refers to the distinct programming languages and models for the implementation of the different discovery approaches.

### 1.2.6 Challenges in Resource Discovery

The various challenges in resource discovery and selection are described as below [45,46].

**Heterogeneity in data:** With the development of *IoT*, more and more sensors, actuators and mobile devices have been deployed into our daily lives. As such, tremendous data are produced and it is urgent to dig out hidden information behind these voluminous data. However, data generated by multi-modal sensors or devices show great differences in formats, domains and types, which pose challenges for machines to process and understand. Moreover, due to the lack of interoperability, information generated by different sensors or devices cannot share with each other, which has become a severe challenge. Although nearly 45% data created on the Internet can be processed, it is challenging to mine and dig out the hidden information. Also, cross-domain knowledge becomes increasingly difficult to share with others because of the heterogeneity of data.

**Semantic interoperability and data management:** *IoT* deals with big data that needs to be transformed into useful information as knowledge. This transformation is possible through the usage of standardized data formats and semantic models with the detailed description of metadata.

**Self-organization capabilities:** For autonomic capabilities, the smart objects on *IoT* are needed to be embedded with intelligence such that they can take decision in accordance to the situation and react accordingly. For the same, smart objects are supposed to organize themselves into transient ad-hoc networks, providing the basic means for sharing data and for performing coordinated tasks.

**Scalability:** With increase in the number of connected resources on *IoT*, scalability

issue arise at various levels that includes naming and addressing, data communication and networking, information and knowledge management and service provisioning and management.

***Security and Privacy:*** The entire infrastructure of *IoT* is distributed and virtual; communication among resources is highly dimmed by malfunctioning and attacks with dynamic addition of the resources. This means that security should be considered a key system-level property, and is taken into account for the design of architectures and methods for discovery solutions.

### **1.3 Thesis Organization**

The organization of the thesis is as follows.

#### ***Chapter 1: Introduction***

This chapter introduces the context of the research to be presented in this thesis. It starts with an introduction to *IoT*, highlighting its building blocks, common standards and current landscape. Then, it discusses the basic bottleneck to *IoT*, *i.e.*, resource discovery and selection. Further, it presents an overview on resources and their graphical forecast. Next, it describes the resource discovery, motivation, evaluation metrics and challenges. The chapter then presents the primary contributions of this research and ends with a discussion on the organization of the rest of the thesis.

#### ***Chapter 2: Literature Review***

This chapter provides a comprehensive survey of work done in the area of *IoT* and highlights the major research gaps. Among the research gaps, it highlights the contributions in the field of resource discovery and selection on *IoT* with respect to foraging and sense-making loop. The former loop, *i.e.*, foraging includes approaches for data acquisition, modeling, integration, assessment and reasoning on *IoT*. The latter covers techniques for both resource discovery and selection. Next, the chapter summarizes the issues in resource discovery and selection on *IoT*. Then, it formulates the problem formulation and objectives for the thesis work.

### ***Chapter 3: Clustering based Resource Discovery and Selection***

The chapter addresses the various challenges that arise due to heterogeneity, lack of standardization, periodic evaluations, *etc.* with respect to resource discovery and selection on the *IoT*. These challenges are to develop techniques for data coupling, indexing and ranking of multiple parameters to cover fundamental search principles. To resolve the resource discovery and selection on *IoT*, a novel clustering technique, *namely*, Iterative K-means Clustering Algorithm (*IKm-CA*) is proposed. It targets concrete cluster formation and their storage using similarity coefficients of vector space model and performs efficient search against matching criteria with respect to complexity. It is simulated on *MATLAB*, and the obtained results are compared with fuzzy k-means and fuzzy c-means clustering algorithm with similarity coefficients of vector space model against exponential increase in the number of resources. The obtained results justify *IKm-CA* to be an optimal solution for the resource discovery and selection on *IoT*.

### ***Chapter 4: Context-Aware Resource Discovery and Selection using Convex Optimization***

This chapter accounts issues such as context of the gathered data, scalability and search metrics on the *IoT*. These issues, if not resolved, would lead resource discovery and selection to emerge as a non-linear constrained specific problems. Such problems experience higher search complexity in terms of time. Therefore, it is required to optimize them to ease the inter-communication and interaction. To address these issues, the chapter proposes a Context-Aware Search Optimization Framework on Internet-of-Things (*CASOF-IoT*) which targets knowledge presentation through schema, discovery *via* multi-modal search algorithm and its optimization through an Iterative Gradient Descent (*IGD*) algorithm. The multi-modal search algorithm through keywords, value or spatial-temporal indices performs resource discovery by finding the best matches against a query from Red-Stores and stores them as search set. The search set is further evaluated *via IGD* for optimization that reduces the search complexity using Armijo's Rule, which eradicates the unpredictable step size. It is tested using various objective functions and is compared with Newton and Quasi-Newton methods. The obtained results depict the efficiency of the algorithm graphically with respect to the searching time.

### ***Chapter 5: Semantic based Resource Discovery and Selection in the Internet-of-Things***

The chapter incorporates the interoperability challenge at various levels like semantic, radio access and context with respect to resource discovery. Moreover, to discover resources continuously and tracing its localization position in a network, there is a need of semantic tagging for both metadata and information. To meet the challenge, an Intelligent Resource Inquisition Framework on Internet-of-Things (*IRIF-IoT*) is proposed. It eradicates the challenge through its three layers, *namely*, perception, discovery, and application. Its main features are to link resources through usage of semantic description and ontology, to discover them with Semantic Matchmaking Engine using Bipartite Graph (*SMEBG*), to optimize the search using Hungarian approach and to access information through web terminal for users. The framework enhances system performance significantly and is proved to have better search efficiency with minimal operational complexity.

### ***Chapter 6: Trust based Decision Making for Optimal Resource Selection in the Internet-of-Things***

This chapter focuses on trust, security and privacy issues on a highly dynamic network infrastructure, *IoT*. To address the issues, a framework, *namely*, A Middleware Approach for Reliable Resource Selection on Internet-of-Things (*MARRS-IoT*) is suggested. It works through three layers, *i.e.*, Sensor-Actuator Layer (*SAL*), Middleware that works through its two building components, *i.e.*, Knowledge-Acquisition (*KA*), Decision-Making (*DM*) and Human-Machine Interaction Layer (*HMIL*). It is self-configurable, explorable, adaptive, efficient that allows resource discovery and automated decision making with lower latency, higher accuracy and reliability. *MARRS-IoT* performs decision making using fuzzy based trust model system which dynamically processes information *locally* as well as *globally* with less time consumption and minimizes complex searches. Moreover, it adapts trust computation scheme. The trustworthiness is calculated *via* direct, indirect trust evaluations and recommendations with respect to its neighboring resources. Also, a *Hybrid M-Gear* protocol is proposed to provide efficient communication *via* routing and helps in reducing traffic snarls. It routes the data among the trust worthy resources with less packet losses and minimum energy consumption during the communication. The selection of trust worthy

resources are processed on server in order to reduce traffic congestion problem. The obtained results support the efficacy of the framework with respect to throughput and execution time.

### ***Chapter 7: Conclusion and Future Scope***

This chapter concludes thesis outcome and provides an insight into the future scope of the work. *IKm-CA* in collaboration with Cosine coefficient is simulated using *MATLAB* against exponentially growing resources to analyze the search efficiency. The results supports algorithm's efficiency in terms of time. In *CASOF-IoT*, continuous update of search sets is done through *IGD* in collaboration with Armijo Rule that helps to reduce the unpredictable step size and to obtain the best neighbor with less computational complexity, providing efficient search for resources. *IRIF-IoT* has eradicated the syntax limitation through the usage of *Ontology*, performed searching *via SMEBG*. The framework has better search efficiency with minimal operational complexity. *MARRS-IoT* performs a search through Neighbor Discovery Algorithm and evaluates trust score of the discovered resources, both *locally* and *globally* using Fuzzy-Decision Algorithm and performs efficient communication among resources *via* Hybrid M-gear protocol. It supports automated decision making by selecting the rightful resource.

In future, the proposed approaches and frameworks could extendedfor performing data control, resolving traffic congestion, multiple neighbor selection, security and privacy challenges on *IoT*. Moreover, they can be implemented for selection of leader resource and to deal with fault tolerance.



---

## CHAPTER 2

---

---

### Literature Review

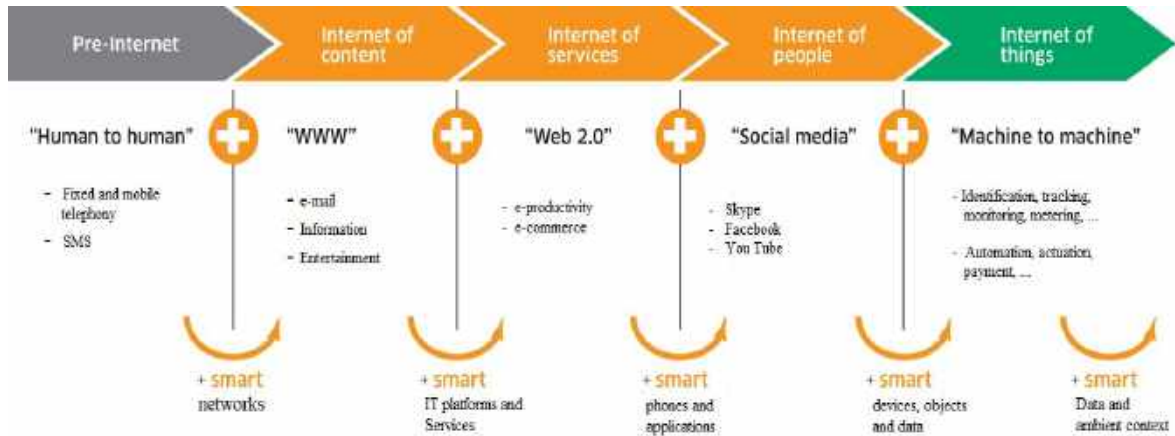
---

The chapter provides an overview on *IoT* and summarizes the research gaps in various domains to realize its vision. The chapter also highlights the most comprehensive contributions in the field of *IoT* with respect to foraging and sense-making loop. The former loop, *i.e.*, foraging includes activities such as data acquisition, modeling, integration, assessment and reasoning on *IoT*. The latter covers techniques for both resource discovery and selection. The discovery techniques are broadly categorized in terms of domain such as around me, on my network, in directories and accessing metadata. The techniques vary with respect to data providers and end publishers. On the other hand, resource selection involves different techniques for decision making such as pattern based, nature-inspired, supervised learning, optimization techniques and trust based. A general, comprehensive and structured overview of existing techniques are described.

#### 2.1 Internet-of-Things

*IoT* concept initially was not named until 1999 officially [48]. The first example of *IoT* came in early 1980's as a Coca Cola machine in Carnegie Melon University. The programmers used to check the availability of drinks, with help of Internet connected to this machine. In 1991, Mark Weiser stated that “The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it” in a seminal paper, *namely*, “The Computer of the 21st Century” [47]. In 1994, *IoT* is

described in *IEEE Spectrum* as transmitting large data into small packets among nodes in order to integrate and automate things that include everything from equipments to whole systems in factories (*see* Figure 2.1).



**Figure 2.1:** Evolution: Internet-of-Things [179]

However, only in 1999, the field start gathering momentum as Neil Gershenfeld from the Massachusetts Institute of Technology (*MIT*) Media Lab in his book, “When Things Start to Think” introduced fundamentals for Internet-of-Things. Also, Sir Kevin Ashton at Auto-ID Centre, *MIT* quoted *IoT* as a standardized way to make communication possible between physical and real world and envisioned computation to be pervasive and operate seamlessly [48]. The Auto-ID Labs had helped to develop the Electronic Product Code (*EPC*), a global Radio Frequency Identification (*RFID*) system that intends to replace the *UPC* bar code. Due to the evolution of *EPC*, the research begins to unite physical and real world things, and is followed by International Telecommunications Union (*ITU*), European Union (*EU*), Cluster of European Research Projects (*CERP*), *etc.*, having distinct focus but common objectives. These organizations have named *IoT* differently like Web-of-Things, Embedded Intelligence and Connected Devices with key features like sensing, identification, communication, localization and tracking capabilities, addressability, actuation, security, and privacy [49]. According to *CERP*, *IoT* is the business, information and social processes that have smart devices or active participants that are interconnected for communication with one another and also with surroundings using information sharing sensors [50]. Forrester commented that, *IoT* is a technology that combines data and communication technologies to make services more responsive, communicable and competent, provided by public safety,

health-care, education, real estate and also in defining components of decisive infrastructure [51]. The *IoT* European Research Cluster (*IERC*) has concluded that *IoT* has a vibrant worldwide system support having its own customs and inter-operable communication set of syntax and semantics for assimilation among real and virtual things having their own Identification (*ID*), corporal properties and information network *via* intellectual interfaces [52]. *IoT-GSU* has given a new definition of *IoT* as interconnection of real and virtual objects on the basis of obtainable and embryonic inter-operable information and announcement technologies for providing highly developed services for the information society [53]. The Internet Architecture Board (*IAB*) has begun *RFC-7452* “Architectural Considerations in Smart Object Networking”, which stated *IoT* as a trend in which large set of embedded devices provide communication related services using various Internet standards. These devices are usually called as “smart objects” and are not operable by humans. The Internet Engineering Task Force (*IETF*) have used the term smart object networking in *IoT* and quoted that “smart objects” are gadgets having important constraints related to its energy consumption, storage and processing capabilities. The *IETF* has provided specific requirements to attain network compatibility among various smart objects. *IEEE Communications Magazine* has stated *IoT* as a framework having specific representation and presence of smart things on the Internet. It is suggested that *IoT* offers new services to connect physical and virtual worlds together to enable interactions among things on cloud. Despite the buzz of *IoT* globally, there is not a single accepted definition for the term. However, in compendium, *IoT* is buzzed as a new period which expects to make objects intelligent such that they can communicate among themselves and participate with different things to make new applications or services and achieve basic objectives utilizing wired or wireless connections and unique addressing scheme [54]. In order to envision the *IoT*, various research domains that are identified for the integration are identification technology, architecture technology, communication technology, network topology and discovery, algorithmic approach, hardware solutions, data and signal processing, searching and browsing techniques, energy optimization, standardization, security and privacy. These research domains, due to technological diversity, raise few research gaps which need to be focused separately. These research gaps are summarized in tabular form below (*see* Table

2.1) [55].

**Table 2.1:** Identified Key Research Gaps

<b>Promising Key Domains</b>	<b>Research Gaps</b>
Identification Technology	<ul style="list-style-type: none"> <li>● Global unique Ids or schemas.</li> <li>● Identity and repository management.</li> <li>● Securing of schemas <i>i.e.</i>, authentication, authorization.</li> <li>● Prevention methods for pseudonymity.</li> </ul>
Architecture Technology	<ul style="list-style-type: none"> <li>● Design of middleware to support distributed open architectures.</li> <li>● Semantic approach to support interoperability.</li> <li>● Standards for implementing higher level of protocol stack.</li> <li>● Integration of semantic sensor web and linked open data.</li> <li>● Delegating policy decisions.</li> <li>● Concept of proxy functionalities to ensure backward compatibility in gateways.</li> <li>● Data models, rules and regulation policy.</li> </ul>
Communication Technology	<ul style="list-style-type: none"> <li>● Adding autonomic capabilities to network devices.</li> <li>● Energy efficient communication.</li> <li>● Software programming contributing high performance and scalable algorithms.</li> <li>● Extension of <i>IPv4</i> to enable connectionless communication.</li> </ul>

to be cont'd on next page

**Table 2.1: Identified Key Research Gaps (contd.)**

<b>Promising Key Domains</b>	<b>Research Gaps</b>
Network Technology and Discovery	<ul style="list-style-type: none"> <li>● Efficient indexing methods to support scalability.</li> <li>● Autonomic computing and networking.</li> <li>● Password and identity distribution mechanisms at layer level.</li> <li>● Chip communication architectures for dynamic configuration design.</li> <li>● Design of power-aware networks to minimize energy consumptions.</li> <li>● Techniques to support dynamic routing scheme.</li> </ul>
Algorithmic Approach and Software	<ul style="list-style-type: none"> <li>● Algorithm for automatically configuring active sensors.</li> <li>● Counting software for active nodes on network.</li> <li>● Accumulating more functionality to gateway in order to deal with fault tolerance in a way to replace the failed node.</li> <li>● Languages to support communication.</li> <li>● Bio-Inspired algorithms for self-organization capabilities.</li> <li>● Algorithmic solutions to support decision making and for optimal assignment of resources in dynamic environment.</li> <li>● Data sharing, propagation and collaboration.</li> </ul>

to be cont'd on next page

**Table 2.1: Identified Key Research Gaps (contd.)**

<b>Promising Key Domains</b>	<b>Research Gaps</b>
Searching Techniques	<ul style="list-style-type: none"> <li>● Metadata and semantic tagging of information, hence context searching algorithms.</li> <li>● To develop algorithms for distributed service consensus and efficient data propagation.</li> <li>● Device communication, disseminated stores.</li> <li>● Positioning and confinement.</li> <li>● Mapping of genuine, advanced and virtual substances.</li> <li>● Universal authentication mechanisms.</li> <li>● Design of techniques to make context aware and optimized usage of offered resources.</li> </ul>
Hardware Solutions	<ul style="list-style-type: none"> <li>● Context switching architectures.</li> <li>● Solutions crossing over nano and small scale frameworks.</li> <li>● Communication radio wires, vitality effective <i>RF</i> front closures.</li> <li>● Nano gadgets and advances with own capabilities.</li> </ul>
Data and Signal Processing	<ul style="list-style-type: none"> <li>● Semantic interoperability and semantic sensor web.</li> <li>● Information sharing.</li> <li>● Sovereign operators.</li> <li>● Real and virtual interaction.</li> </ul>
Security and Privacy	<ul style="list-style-type: none"> <li>● All the security goals must be achievable like freshness, confidentiality, non-repudiation, authorization, integrity, availability and authentication.</li> <li>● Managing privacy policies, access and use rights.</li> </ul>

to be cont'd on next page

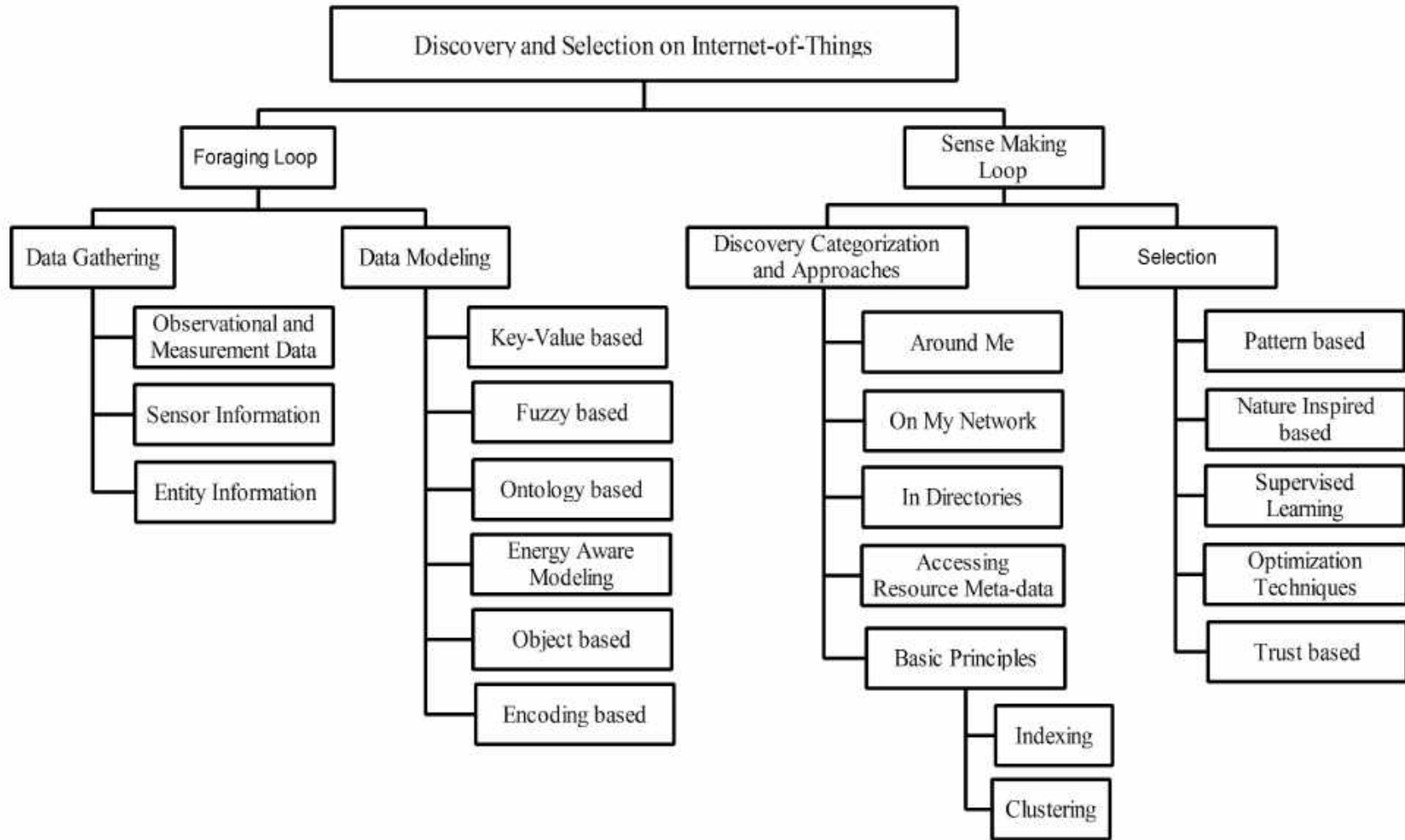
**Table 2.1:** Identified Key Research Gaps (contd.)

<b>Promising Key Domains</b>	<b>Research Gaps</b>
Energy Optimization	<ul style="list-style-type: none"><li>• Issue is to conserve or minimize the energy consumption or harvest energy from natural resources.</li><li>• Energy storage and micro super capacitor technologies.</li></ul>

In order to address these gaps, first it is required to understand the basic definition of *IoT* that target resources to be identifiable, to communicate and to interact. For the same, the prerequisite is to discover them on the network from both ends, *i.e.*, service provider and service requester. Therefore, this literature survey is carried out to understand the mechanism for resource discovery and selection on *IoT*. Also, the mechanism of discovery and selection of resources on *IoT* require captured data from the intelligent nodes to be modeled, represented as knowledge, interpreted with proper reasoning for taking optimal decision. Hence, the mechanism is broadly categorized in to two loops, *namely*, foraging and sense-making (*see* Figure 2.2). The former is responsible for data gathering and its modeling whereas latter targets resource discovery and selection. These loops are studied thoroughly to understand the existing comprehensive contributions as well as research gaps.

## **2.2 Foraging Loop: Data Gathering and Modeling**

The loop is responsible for identifying the originating source of the resource and to extract the meaningful information from the collected data for knowledge formation that can be made available for users in consumable form. For the same, the loop works into two processes, *i.e.*, data gathering and its modeling. Each process is discussed below.



**Figure 2.2:** Discovery and Selection on Internet-of-Things

## 2.2.1 Data Gathering

It targets to collect raw data from the active devices such as sensors, actuators, *etc.* through the usage of various communication technologies that includes *RFID, NFC, BLE, 6LoWPAN* and many more. A literature survey is carried out to understand the techniques, models and architectures for collecting the data. For example, few researchers have surveyed the developments and improvements in *ITS* that describe the generalized transportation infrastructure of the world [56]. The authors have provided all the perspectives with respect to *ITS* in reference to areas of industry, research, development, academic and public policy. S. Lu *et al.* in [57] have described the composition and working flow of Electronic Toll Collection (*ETC*) systems as a part of *ITS*. It is suggested that *GPS-based ETC* system will be widely used in internal expressway network charging as shown in Figure 2.3.

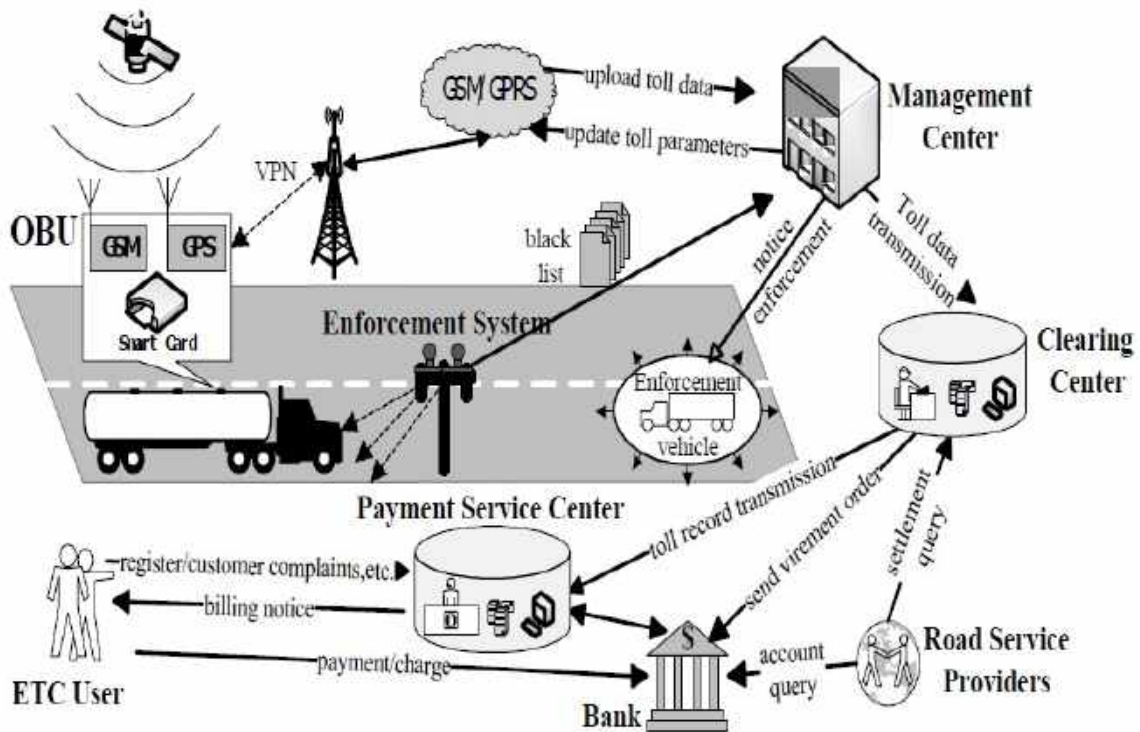
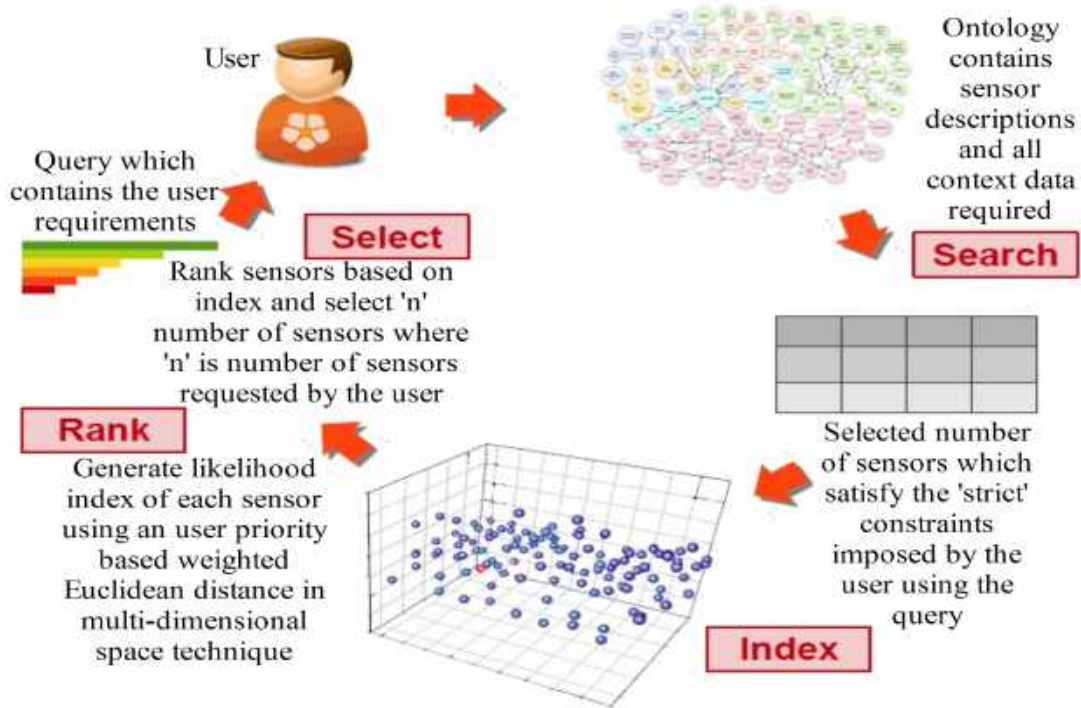


Figure 2.3: Architecture of GPS-based ETC System [57]

Few researchers have developed Gaussian regression process *namely*, Distributed Gaussian Process Regression (*DGPR*) to generate spatial functions for mobile nodes [58, 59]. The process face few disadvantages that it is not tested practically in real-time environment

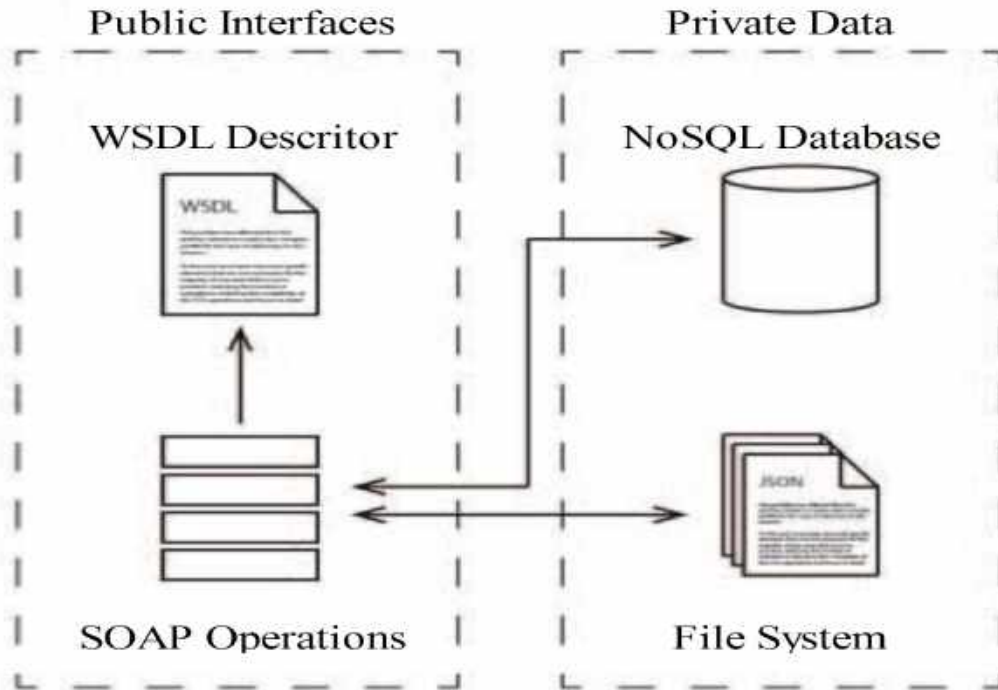
[60]. To this improvement, a Context-Aware Sensor search, Selection, And Ranking Model (*CASSARAM*) for *IoT* has been presented to deal with challenges of sensor selection at the time of overlapping of large set of sensors and redundant information is generated [61] (see Figure 2.4).



**Figure 2.4:** Overview of *CASSARAM* [61]

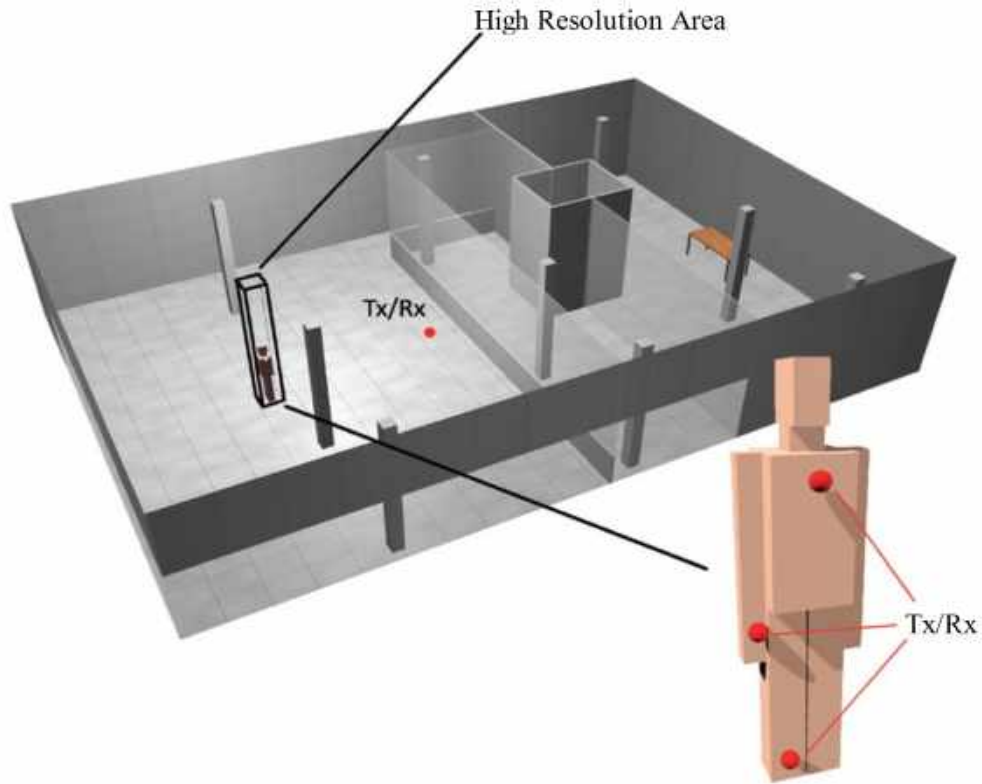
The model has limitations for achieving efficient results and to provide more functionality due to closely related semantic and quantitative reasoning. Few researchers have outlined and exemplified synergy between three complementary paradigms as Semantic Computing (*SC*), Cognitive Computing (*CC*) and Perceptual Computing (*PC*) [63]. The *SC* provides the ability to deal with the challenges of big data. The *CC* helps to improve the understanding of data for decision-making and provides the ability to utilize relevant knowledge. The *PC* has the potential to bring tremendous value through its ability to provide personalized and contextual abstractions over massive amounts of multi-modal data, originating from the physical, cyber, and social domains [64]. These approaches have their disadvantages for having to maintain a database for the storage of tremendous amount of data. In order to remove these gaps, J. Pradilla *et al.* have postulated Sensor Observation Service (*SOS*) to

fit small sensors network environments which do not need robust system to compute and provide standard platform [65] (*see* Figure 2.5).



**Figure 2.5:** Sensor Web Enablement Architecture [65]

The *SOS* increases interoperability by providing interdependency of heterogeneous data from manufacturers and sensor networks. To increase the performance of networks, few researchers have developed wireless body area network based systems for transmitting medical information within complex indoor e-health scenarios, on hand-held android device [66]. They have employed a 3-Dimensional ray launching algorithm that provides an in-house radio frequencies, power delays, and delay spread for indoor scenarios (*see* Figure 2.6).



**Figure 2.6:** 3D-ray Launching Schematic View [66]

Few researchers have highlighted Thing Description (*TD*) from CoRE Link Format to semantic based descriptions that are represented in terms of actions, attributes, *etc.* to enable sensors semantically [37]. The framework has its limitations to provide its extensibility for connected vehicles.

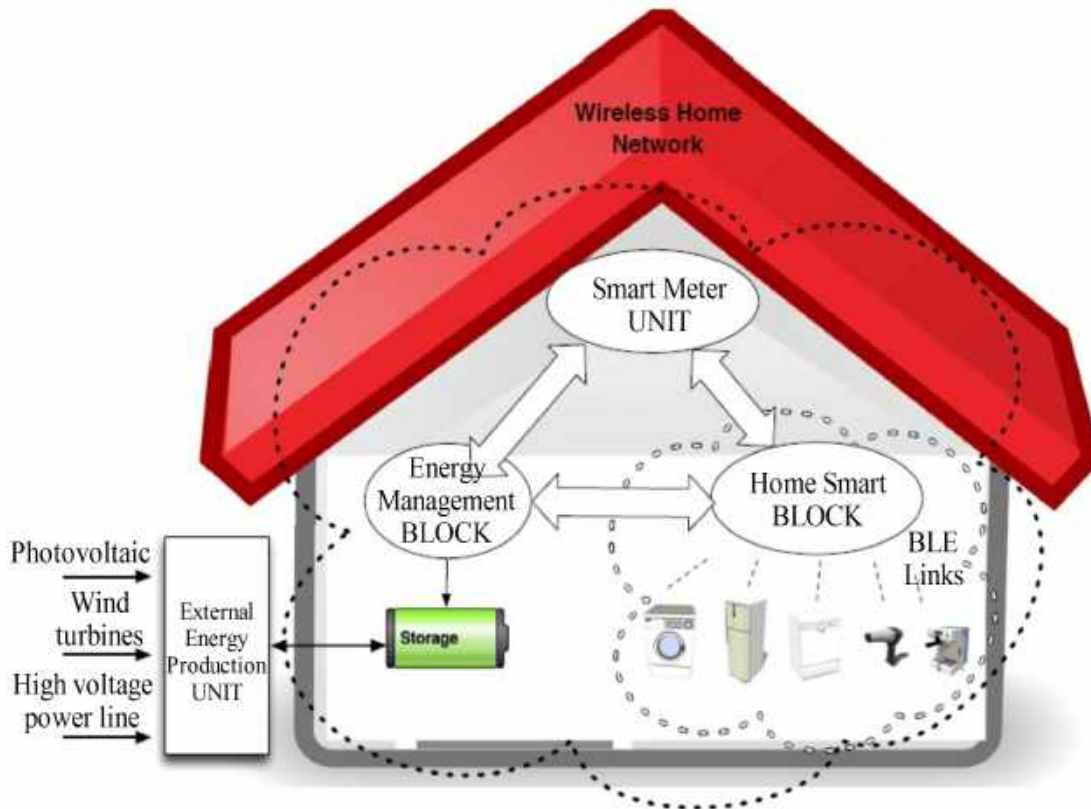
## 2.2.2 Data Modeling

The data modeling is responsible for knowledge formation that helps to build resource repository with defined rules and conditions. For the same, it uses various modeling techniques based on Key-value, Fuzzy, Ontology, Energy-Aware, Object and Encoding. Each of the technique is discussed one by one.

***Modeling based on Key-Value:*** The modeling technique is simple, flexible, easy to manage and application oriented with temporary storage. It does not support storage of complex data structures. For example, few authors have presented an approach for discovering web

services. The approach is based on modified *CLIQUE* algorithm [67]. This approach indexes web services using Key Performance Index (*KPI*) and is tested on virtual logistic system. The approach face few limitations of embedding web services dynamically on to the virtual logistic system [68]. To provide dynamic searches, a technique *namely*, Hybrid Search Engine technique has been postulated. The technique is based on Spatial-Temporal, Value-based, and Keyword-based conditions [69]. The produced results prove the engine as a good solution for real time searches. B. Djamaa and A. Yachir have postulated a Proactive Resource-Directory (*PRD*) discovery mechanism [70]. The *PRD* increases network performance by advertising *RD* in an efficient manner. It has its disadvantage of providing optimizability towards multiple directories.

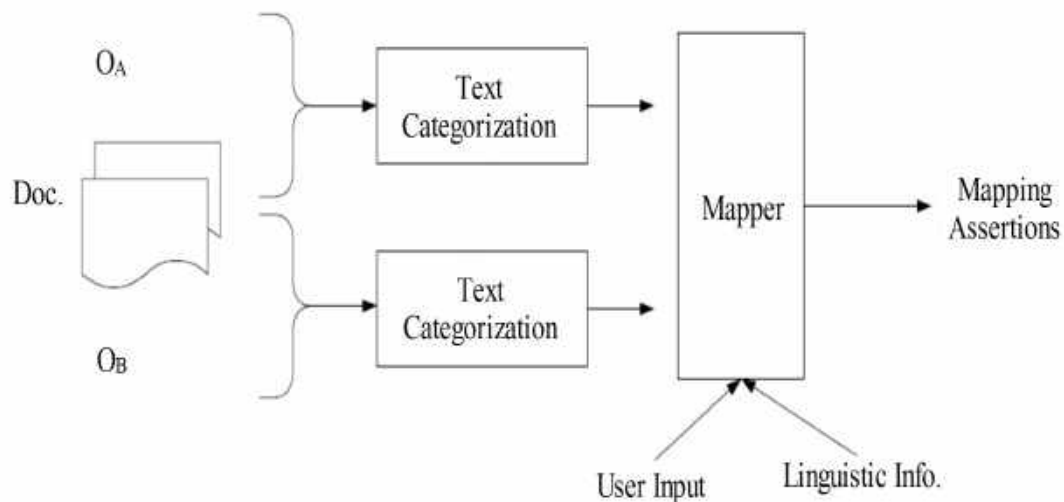
***Modeling based on Fuzzy Logic:*** Fuzzy-based techniques permit approximate reasoning instead of fixed and crisp reasoning. It is simple to define, extendable, handle uncertainty and allows more natural representation using truth table. For example, M. A. Quaddus *et al.* uses fuzzy logic to develop a Map Matching (*MM*) algorithm [71]. The algorithm takes *GPS* augmented data of sensors as input to provide continuous navigation. It is tested for various road networks, however, its performance is not evaluated for urban condition due to unavailability of high accuracy. In order to evaluate positioning performance, a novel interval type-2 fuzzy controller architecture has been postulated [72]. The architecture resolves non-linear problems related to vehicle suspension. The architecture has its limitations for its implementations on electric vehicle suspension system in real-time. To provide dynamic inputs and clustering, smart energy management solution based on fuzzy has been formulated [75] (*see* Figure 2.7).



**Figure 2.7:** Energy Management System for Smart Homes [74]

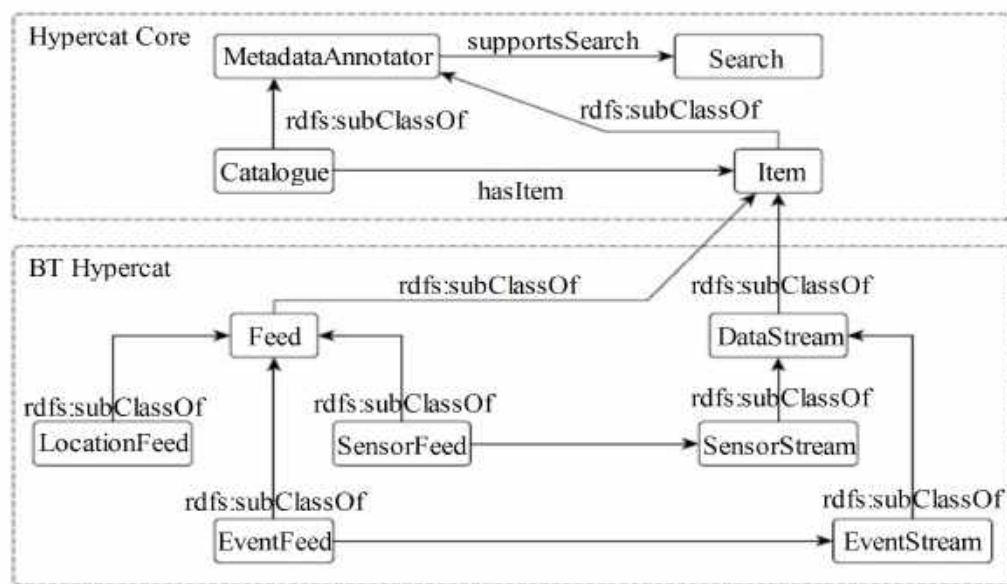
The solution manages consumer's feedback using fuzzy simulations. The results justify the efficiency of the suggested solution in terms of reducing peak load demand and electricity consumption charges.

**Modeling based on Ontology:** The modeling is based on description logic, which is a family of logic based knowledge representations of formalisms. It is mainly supported by *RDFs* and *OWL*. It allows re-usability of knowledge, is simple, flexible, application independent and expressive. As an example, a preliminary solution has been postulated using external information to mitigate the problem of ontology mapping which is pervasive in context as shown in Figure 2.8.



**Figure 2.8:** Architecture of Ontology Mapping [76]

The text categorization is used to automatically assign documents to the concepts in the ontology and similarities are computed between concepts in two ontologies [76]. To provide extensibility, the semantic enrichment of the Hypercat Data Hub has been formulated [77]. This enrichment provides a center point to share a wide range of resources of available datasets. To translate existing data into RDF schema, the Hypercat Ontology has been introduced (*see* Figure 2.9).



**Figure 2.9:** Hypercat Ontology [77]

The ontology has its limitations that it does not provide direct access to spatial information that is currently available in the Hypercat Data Hub.

***Modeling based on Energy-Aware Modeling:*** It covers modeling techniques that integrate internet connectivity that allow devices to be controlled remotely by users, or central management *via* a cloud-based interface, and enable functions like scheduling [78, 79]. For example, C. Intanagonwiwat *et al.* have explored the directed diffusion paradigm for performing sensing of environmental phenomena [80]. It is found that the paradigm has potential for significant energy efficiency and it outperforms an idealized traditional data dissemination scheme like omniscient multi-cast with relatively non-optimized path selection [81]. The paradigm has few limitations that it is not explored for entire space of alternative designs and features like data aggregation and caching. Few of the authors have designed Adhoc On-Demand Distance Vector (*AODV*) routing protocol for use in ad hoc mobile networks [82]. It is tested on simulation based environment for variety of scenarios. The protocol has its disadvantages that it consumes more energy in finding the shortest route between source and destination [83]. In order to reduce energy consumption, a new scheme called energy aware routing is postulated that uses sub-optimal paths occasionally to provide substantial gains [84]. It uses probabilistic forwarding mechanism to send traffic through different routes and helps in using the node resources more equitably. It considers network survivability, for deciding the efficacy of network protocols [85]. S. Lindsey and C. S. Raghavendra have formulated Power-Efficient GATHERing in Sensor Information Systems (*PEGASIS*), a near optimal chain-based protocol that is an improvement over *LEACH* [87]. In *PEGASIS*, each node communicates only with a close neighbor and takes turns transmitting to the base station, thus reducing the amount of energy spent per round. But it faces certain limitations due to its dependency on the platform and it may lead to slow down the performance of the system.

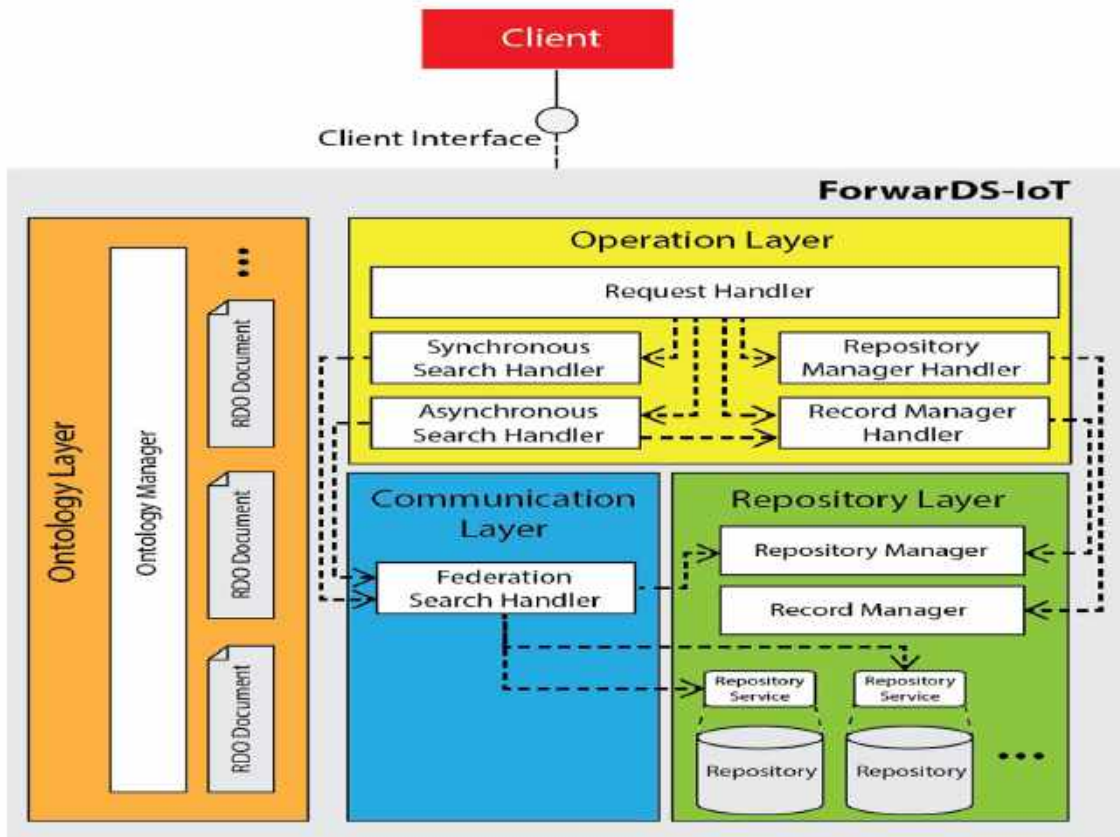
To increase system's performance, an improved algorithm based on *LEACH* has been postulated to balance regional energy of the network [88]. Simulation results show that the suggested algorithm is effective in balancing the network energy consumption and prolonging the network lifetime. The algorithm face few limitations that it consumes extra

energy while receiving control message from sink node. A hierarchical *LEACH* based routing strategy has been formulated to refresh clusters periodically on the basis of residual energy and distance [89]. Re-clustering helps to enhance network's lifetime by distributing workload among different nodes. The protocol has its disadvantages that it does not find an optimal value for the dynamic threshold to conserve energy at cluster head. In order to find optimal clustering, a Multi-Level Clustering Multiple Sink (*MLCMS*) with IPv6 protocol over Low Wireless Personal Area Networks (*6LoWPAN*) has been promoted [90]. It enhances the lifetime of a network and by combining *6LoWPAN* with an *IP* network, the system's lifespan is increased by 12% and a substantial extension of the network is realized.

***Modeling based on Object:*** It models using classes promoting encapsulation, abstraction and is suitable to be used as internal, non-shared, code-based applications. It does not provide inbuilt reasoning capabilities, hence, validation is difficult. For example, the advantage of using a similarity measure instead of a sub-sumption criterion in the object-based representations has been demonstrated [91]. The author has formulated a general approach of the similarity combining some results stemming from Data Analysis, Cognitive Sciences and Machine Learning. The approach allows to evaluate the degree of similarity between complex objects and this measure can be used both in the frame of classification and categorization. Due its quadratic complexity in terms of number of objects involved, its computational cost is high. To decrease its computational cost, a novel vector location-based algorithm based on Distributed Hash Tables (*DHT*) algorithm has been demonstrated [92]. This vector location is applied to resource discovery in Mobile Peer-to-Peer (*MP2P*), dynamically using location information to determine the optimal route. This algorithm improves the efficiency of query, deduces the cost of network by bi-directional querying and optimizes the performance of *MP2P*. Also, A. J. Jara *et al.* have analyzed the requirements and design issues to apply various discovery techniques in smart objects [93]. The authors have provided an evaluation of (*mDNS*) and *DNS* Service Directory (*DNS-SD*) for smart objects. J. Han and X. Y. Li have designed a neighbor discovery protocol *namely*, Pick-up game, for finding set of neighbors [94]. This protocol collects *IDs* of neighboring nodes and minimizes the latency of neighbor-discovery process. The protocol has its limitations that it

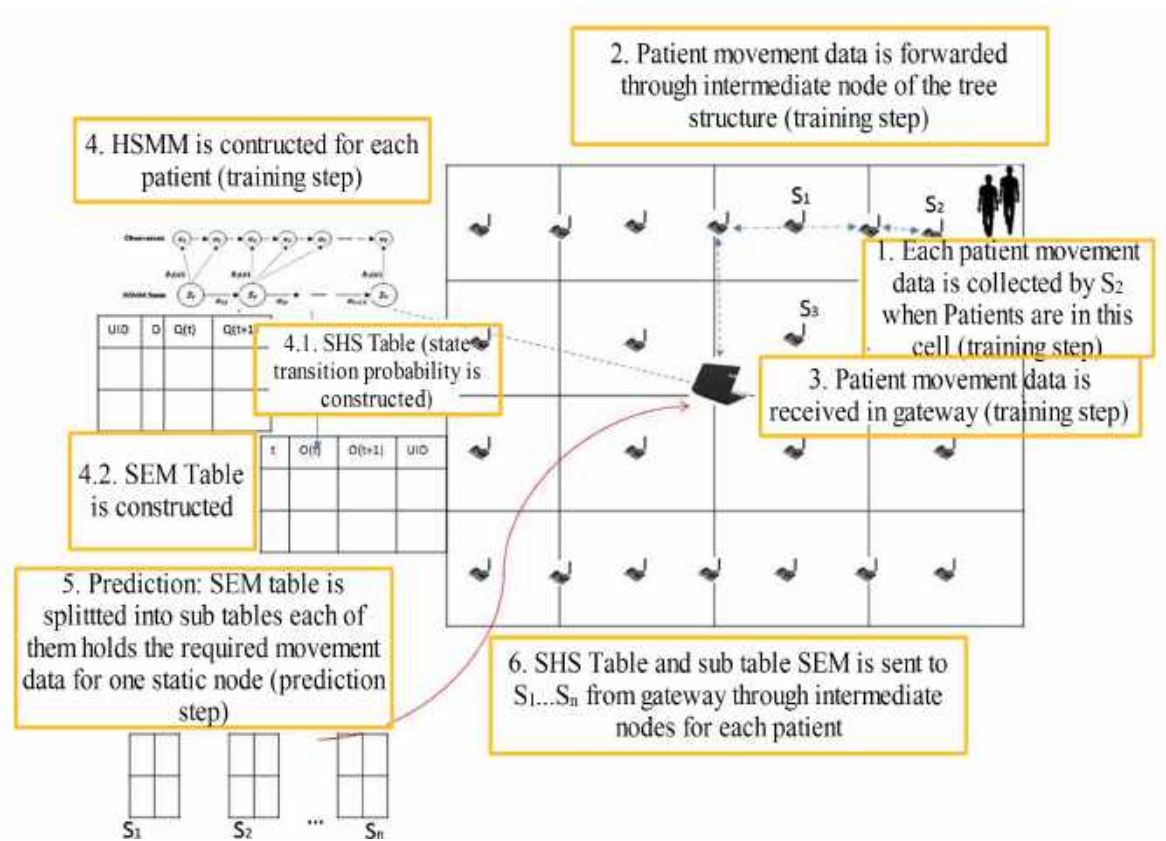
does not account for multi-hop neighbor discovery and has to be at active state during the process that leads to consumption of more energy.

***Modeling based on Encoding:*** It models data using tags and is found to be more flexible, structured, efficient. It supports efficient data retrieval and validation using schema definition such as *XML*. As an example, M. Kodialam and T. Nandagopal have presented estimation algorithms to enable the identification of *RFID* tags in less time [95]. The authors have designed collision-based estimator, the probabilistic estimators and two unified estimation algorithms that have complementary properties. The technique helps in eradicating neighborhood estimation problem, the multiple *RFID* reader problem, and privacy related issues in *RFID* networks. An approach to model contextual information using the concept of context aware in-car infotainment system has been presented [96]. The approach is beneficial for modeling calculation of reliabilities of contextual information. R. Pozza *et. al* have postulated a Context Aware Resource Discovery (*CARD*) framework that leverages Q-Learning techniques [97]. The framework extends the working of asynchronous neighbor discovery protocols and reduces wastage of energy, delays in resource search [98]. The framework has faster computation for discovering resources and has limited processing capabilities. Also, a Federated Discovery Service for *IoT* (*ForwarDS-IoT*) model has been proposed that provides resource search on the basis of attributes, range and queries [99]. The model uses information based on ontology to describe resources and services semantically as shown in Figure 2.10.



**Figure 2.10:** *ForwarDS-IoT* Architecture [99]

It face certain challenges that are investigated by [100]. The author has addressed *IoT* middleware and classified various approaches based on existing techniques of resource search with respect to security, privacy, trust and bootstrapping. To provide security, the potential power of a *DDoS* attack has been determined when using *IoT* devices as a reflector [101]. To determine the power of attacks, key factors like bandwidth, and device availability are combined. The suggested method helps in determining only limited attack vectors to abuse *IoT* devices. To its improvement, a Distributed Self-Healing Movement Prediction scheme has been introduced for *IoT* applications (*DSHMP-IOT*), to predict movement direction of mobile *IP*-based sensors in a multi-user environment, such as a health-care system [102] (see Figure 2.11).



**Figure 2.11:** DSHMP-IOT Movement Prediction Stages based on DSHMP-Tree Network Scheme [102]

The scheme uses an Artificial Intelligence (AI) solution to predict the direction of the mobile nodes. The scheme has the feature like (i) it helps in eliminating the need for special hardware, (ii) it is not sensitive to noise and speed.

## 2.3 Sense Making Loop: Resource Discovery and Selection

The sense-making loop intends to analyze the extracted knowledge such that it can use this to provide services to the service requester. For the same, it requires to explore techniques with respect to discovery mechanism. Also, it needs to emphasize on the selection criteria for taking appropriate decision in order to provide service. In short, the loop focuses on discovery mechanism for resources and their selection, *i.e.*, Resource Discovery and Resource Selection. Both of them are detailed with their major contributions.

### 2.3.1 Resource Discovery: Categorization and Approaches

The mechanism for resource discovery on *IoT* is broadly categorized as (i) Searching around me, (ii) Searching on my network, (iii) Searching on directories, (iv) Accessing Thing meta-data, (v) Searching through peer-to-peer networks, and (vi) Searching through semantic web methodologies on the basis of interaction patterns. It is beneficial for application developers in decision making that would decrease standards dependency and ease the discovery mechanism [15, 103].

**Searching Around Me:** This category enables discovery of smart objects as things that have close spatial proximity. For discovery, it uses all comprised technologies having spatial “nearness” of the client. The interaction pattern for “Searching Around Me” is shown in Figure 2.12. Few examples of discovery technologies are Near Field Communication

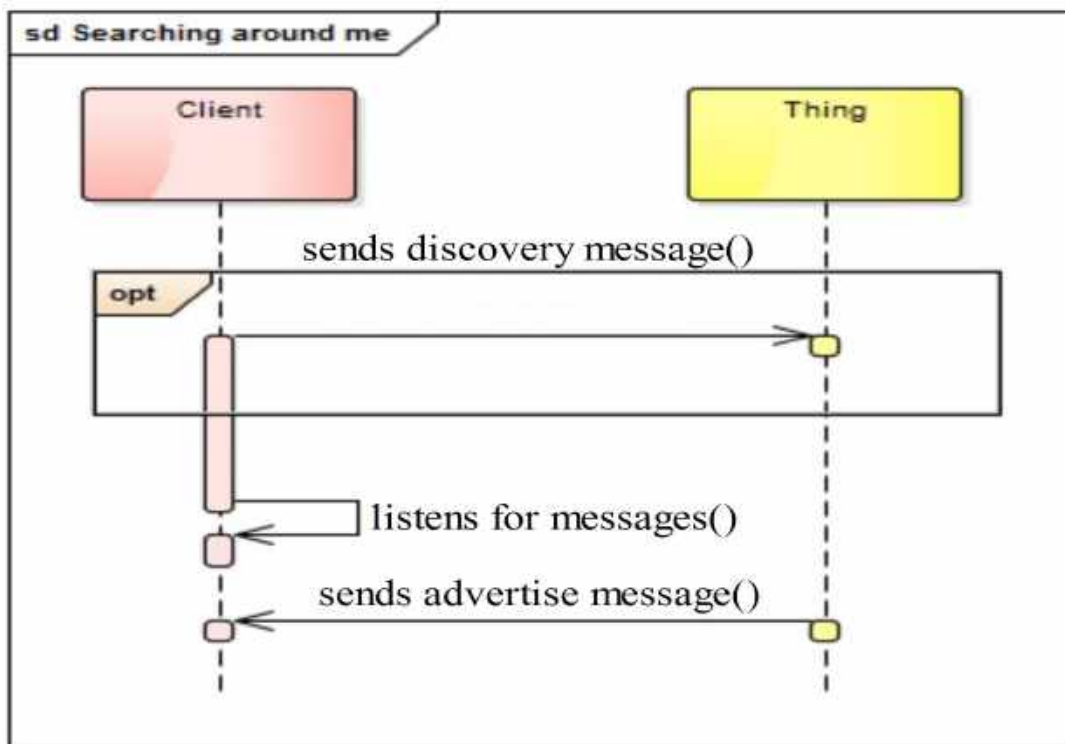
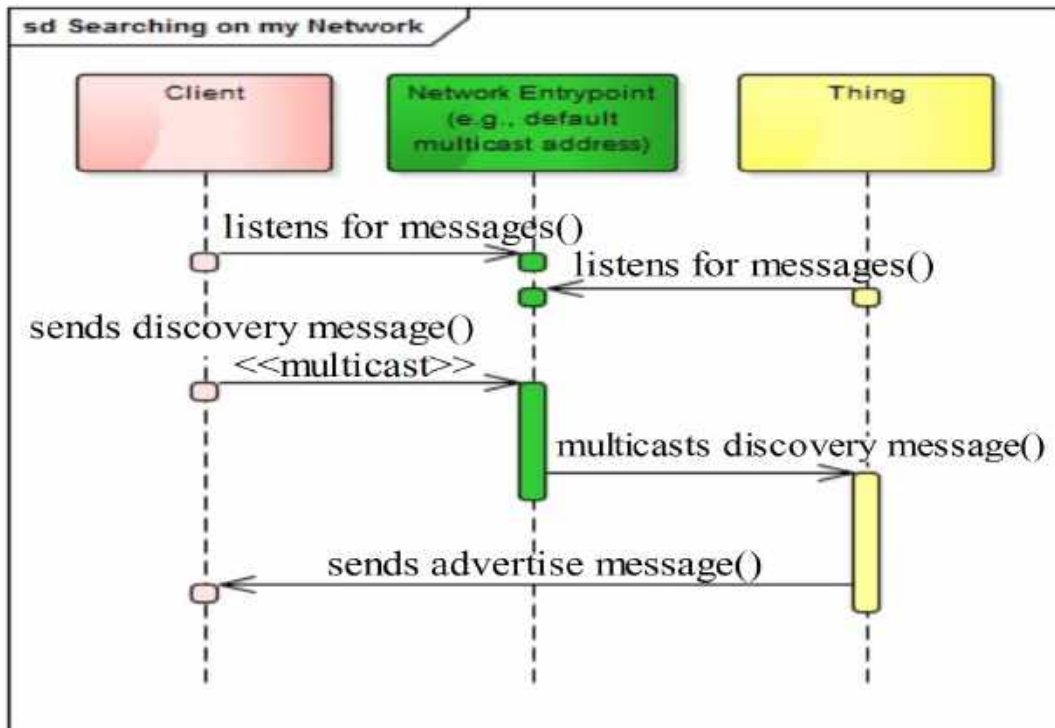


Figure 2.12: Interaction Pattern: ‘Searching Around Me’ [119]

(*NFC*) and Bluetooth Low Energy (*BLE*). *NFC* works on the principle of *RFID* and allows two-way communication among nodes [104]. *BLE* is used to transmit advertising packets

periodically [105]. These packets are comprised of different payloads.

**Searching on My Network:** The technologies under this category allow discovering nodes on network only. The interaction pattern is shown in Figure 2.13.

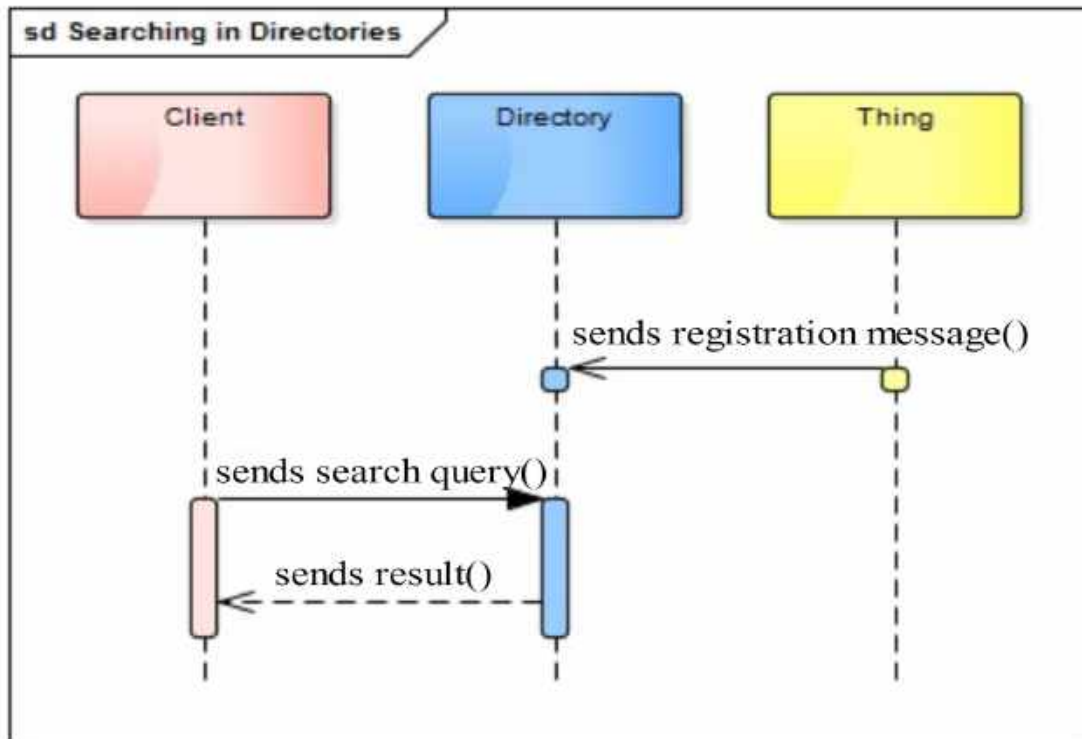


**Figure 2.13:** Interaction Pattern: ‘Searching on My Network’ [119]

Few examples under this category are Web Service (*WS*) Discovery, Multicast *CoAP*, Simple Service Discovery Protocol (*SSDP*) and Multicast *DNS mDNS*. All these examples work on the principle of Internet Protocol (*IP*) and User Datagram Protocol (*UDP*). OASIS’ Device Profile for Web Services (*DPWS*) uses *WS* discovery based communication via *SOAP* messages. It generally avoids centralized registries for smaller networks [106]. The *SOAP*-based message exchange confirms with the identified pattern in ad-hoc manner. Multicast *CoAP* are used by clients to discover things, as *CoAP* servers [107]. The approach is further combined with *CoRE* Link Format to get request to specific multicast address to receive a list of all available resources [108]. Universal Plug and Play (*UPnP*) uses *SSDP* for discovery [109]. The *SSDP* client multicasts a discovery request to the *SSDP* multicast channel and port [110]. *mDNS* resolves host names to *IP* addresses [111] and *mDNS* client

discovers things' endpoint by sending an *IP* multicast query message over the network.

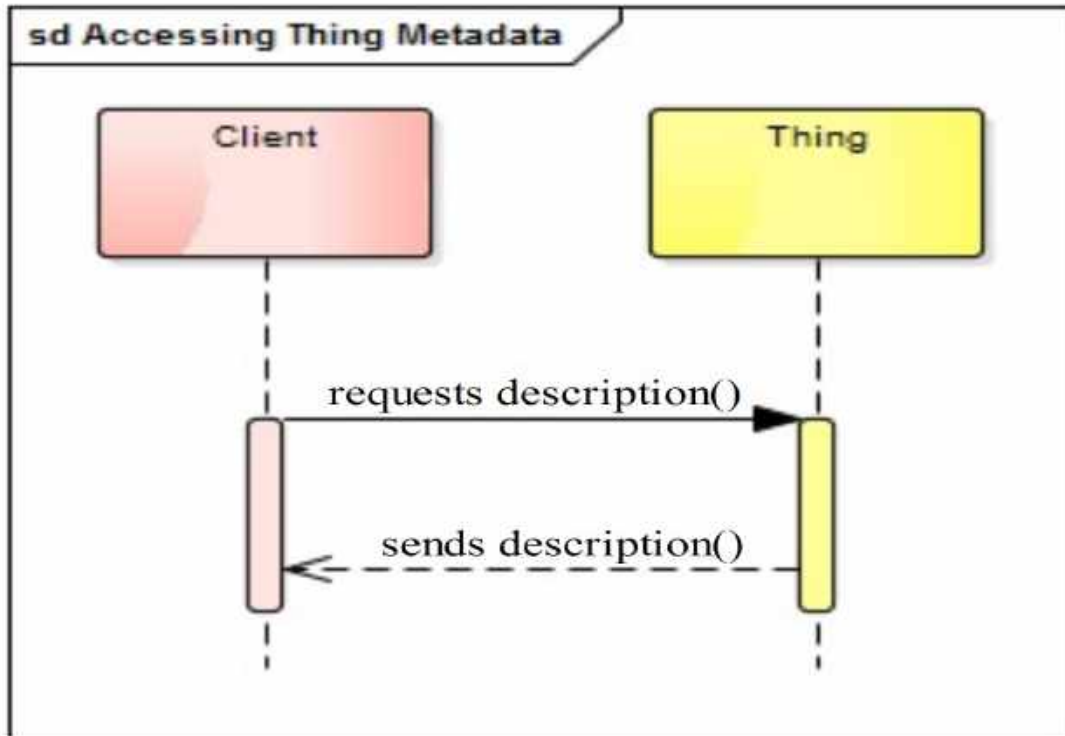
**Searching in Directories:** This category uses central directory for discovering things and its resources. User queries are directly submitted to the directory for searching. Its interaction pattern is shown in Figure 2.14.



**Figure 2.14:** Interaction Pattern: ‘Searching in Directories’ [119]

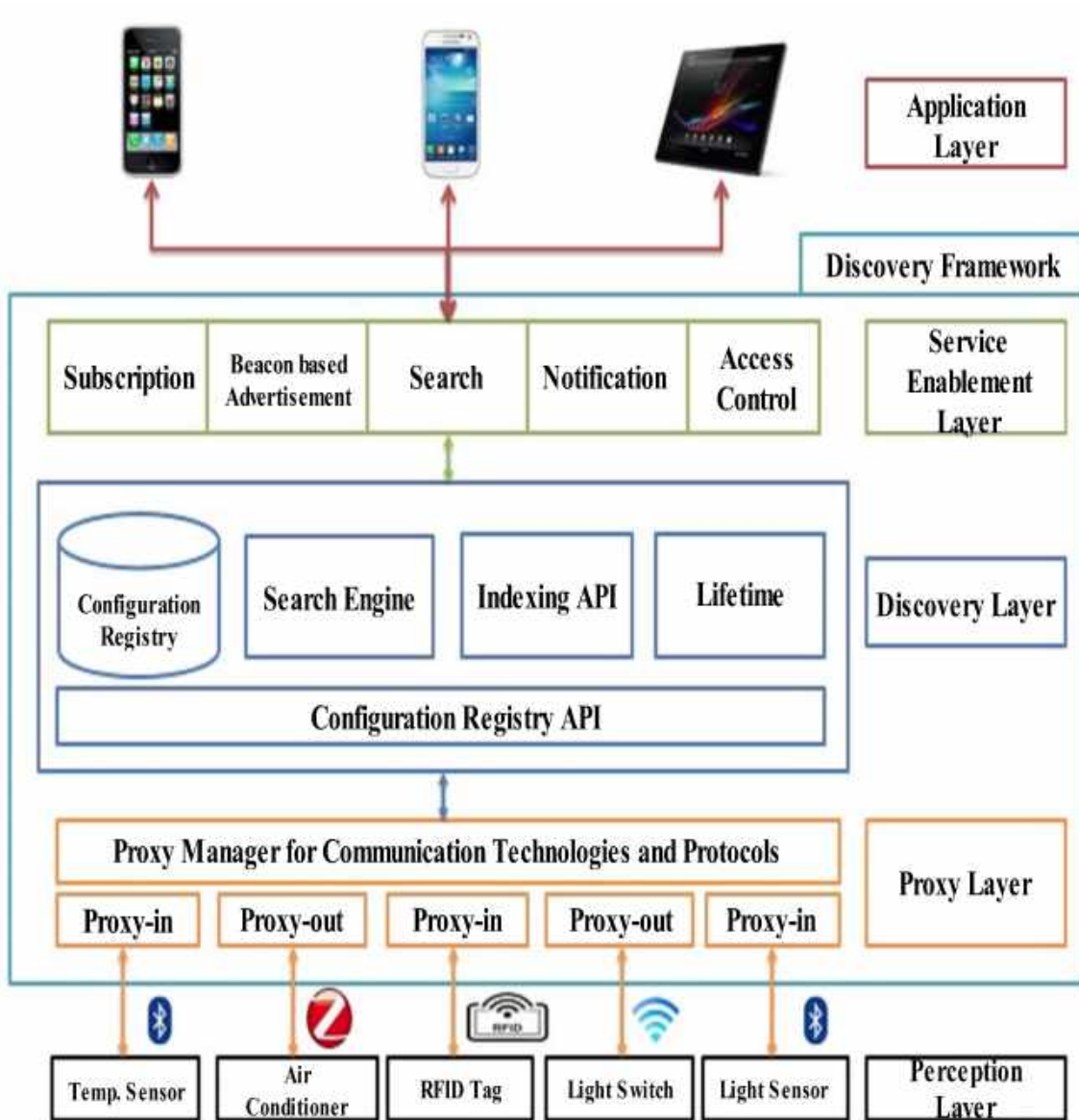
Few examples under this category are *CoRE* Resource Directory [113], HyperCat [114] and *XMPP* [115]. *CoRE* Resource Directory hosts descriptions of resources on other *CoAP* servers and allows lookups for those resources. Resource is discovered using lookup interface which supports simple search queries. Resource encoding in *JSON* are described using lightweight catalog of HyperCat specification. Another example, *i.e.*, *XMPP* based *IoT* Discovery describes the interface of a Thing Registry based on *XMPP* messages. By sending *XMPP* messages, things are registered in Thing Registry and is searched for resources or their meta-data. Discovery is performed using two or more comparison operators within search request to the registry.

**Accessing Thing Metadata:** The resources as Things are discovered using above said approaches but it is necessary to access their meta-data (description or measurements). Therefore, this category provides few technologies that are used by clients to implement a ranking or filtering system for the discovered Things [117].



**Figure 2.15:** Interaction Pattern: Accessing Thing Metadata [119]

Few examples under this category are *CoRE* Link Format [108] and Open Geospatial Consortium’s Sensor Observation Service (*OGC SOS*) [118]. *CoRE* Link Format works on the principle of *IETF* standard which is used to access and communicate Thing descriptions. *OGC SOS* provides web service interface to query meta-data information of things and their measured data. *OGC SOS* supports temporal filtering to access correct meta-data descriptions. S. K. Dutta *et al.* have presented a discovery framework which contains three layers namely proxy, discovery and service enablement (*see* Figure 2.16) [36].



**Figure 2.16:** Architecture of Discovery Framework [36]

The suggested framework provides standards to preserve interoperability of data. The framework has its limitations of not providing optimizability against search and ranking algorithms. In order to provide optimized search, A. Boring *et al.* have analyzed discovery mechanisms based on crucial parameters *i.e.*, discovery range, queries, bootstrapping and ranking of results [119]. This evaluation has shown great differences in the analyzed technologies. Based on the categories for resource discovery, few approaches are described in Table 2.2.

**Table 2.2:** Approaches for Resource Discovery on *IoT*

Category	Approaches	Location	Indexing	Ranking	Data Storage
Searching Around Me	<ul style="list-style-type: none"> <li>• Near field Communication (<i>NFC</i>) [20]</li> <li>• Blue-tooth Low Energy (<i>BLE</i>) [40]</li> </ul>	10-100 cm	×	×	×
Searching on My Network	<ul style="list-style-type: none"> <li>• Web Service Discovery (<i>WS</i>) [120]</li> <li>• Multicast CoAP [21]</li> <li>• Simple Service Discovery Protocol (<i>SSDP</i>) [121]</li> <li>• Multicast DNS (<i>mDNS</i>) [122]</li> </ul>	Remote within network	×	×	✓
Searching in Directories	<ul style="list-style-type: none"> <li>• CoRE Resource Directory [123]</li> <li>• HyperCat [124]</li> <li>• Advanced Message Queuing Protocol [125]</li> </ul>	Remote within directory	×	×	✓
Accessing Thing Metadata	<ul style="list-style-type: none"> <li>• CoRE Link Format [126]</li> <li>• Open Geospatial Consortium's Sensor Observation Service (<i>OGC SOS</i>) [69]</li> </ul>	Remote within directory	×	×	✓

From the Table 2.2, it is observed that most of the existing resource discovery approaches on *IoT* have not provided means of indexing and ranking. The researchers have suggested various indexing and ranking techniques but these are not widely explored for large data sets. As *IoT* deals with heterogeneous data to index or rank, the possible effective solution is clustering [127]. It helps in linking of resources from one network to another and combines

them to treat as a single cluster that provides the continuation of resource discovery on the basis of query asked. The clustering helps in synchronizing the resources providing faster computation of search metrics asked by user. It improves the performance of network on the basis of energy to the nodes, lifetime of the network, scalable and aggregating data. It preserves bandwidth for communication and tries to avoid the redundant exchanging of data among inter-cluster interactions. Thus, the clustering for resource discovery comes out to be reality for handling data, indexing, ranking and to provide fast results due to its properties of scalability, performance and flexibility. Moreover, it can adhere to dynamic changes to the resources and clusters them on the basis of similar storage type. The major application domains for clustering are data compression, vector quantization, pattern recognition, data mining and knowledge discovery [128].

From the existing clustering techniques, k-Means, being simplest algorithm, works for compact and hyper-spherical clusters. It is a frequently used technique for clustering data without any labels where data points are precisely allocated to predefined clusters according to some distortion measure. The researchers have suggested enhanced k-means clustering algorithm to determine the clusters, which is the improved version of the existing k-means algorithm. The algorithm's performance is analyzed in terms of accuracy for selecting final clusters and computational time. It fails to provide the mechanism for the selection of initial centroid resulting into delay in selecting final clustered data points [10]. To this improvement, constrained specific algorithm is introduced which incurs low communication costs and achieve constant approximation ratios. But, its limitations is that it does not provide the method to achieve similar approximation ratio and communication complexity for k-median or center clustering problems [12]. To reduce complexity, another Lloyd's k-means based Filtering algorithm is suggested that performs data-sensitive analysis of the algorithm's running time. The efficiency in terms of time is achieved due to fixed data points. It helps in separating clusters with increase in number of data points. Its limitation is that it passes no information from one stage to the other which effects the overall performance [129]. Also, *CLARANS* iteratively perform clustering process but it has limitation to predict the count for cluster in the beginning only [130]. Due to order-sensitive property of the cluster's instances, the problem arises for selecting and generating a possible cluster. This

results into time consuming process. Further the knowledge gathering in this proposed method has heuristic knowledge rather than exact details [131].

The k-means, though, helps in selecting final clustered data points but are not widely used due to its limitations like more time consumption, not removing empty clusters, not feasible in selecting data points and is highly effected with surroundings factors like noise, outliers etc. Therefore, the k-means is further distinguished among hard and soft clustering by combining it with fuzzy sets.

***Hard Clustering-Fuzzy k Means:*** The fuzzy partitions the clusters on the basis of probability of 0 and 1 to get the best selection of clustered data points. It performs both information retrieval and visualization [6]. The approach is useful in pruning the formal context for reducing attribute implication base. It has the limitations for manually suggesting the number of clusters to find the righteous data points and thus desired output is not received.

***Soft Clustering - Fuzzy c Means:*** The soft clustering considers the membership matrix for selecting clusters using k Means and fuzzy. The approach that combines the two methods is *FCM* that targets to find a partition for set of data points while minimizing the cost function. Based on *FCM*, another model, namely, possibilistic-fuzzy c-means (*PFCM*) is introduced that solves the sensitivity to noise effect and mitigates the problem of similar clusters [128]. As conventional methods of *FCM* does not retain information related to spatial domain for nodes selection, therefore, spatial *FCM* is proposed that considers membership matrix for storing spatial information of clusters. It helps in improving segmentation results and obtains the log for cluster distribution. It fails to deal with robustness to noise, and outliers and more time consuming [4]. Therefore, a robust learning-based *FCM* framework is suggested that automatically find the best number of clusters. Its advantages are that it is free from initialization parameters and selecting the clusters. It has limitations in handling high-dimensional data sets and iteration time is decreased rapidly after certain iterations [16]. To meet the drawbacks, few authors have suggested novel intuitionist *FCM* clustering method. It considers erred parameter as hesitation degree while describing the membership matrix. This results into clusters centers that converge to desired location [9]. The obtained

results suggest that membership matrix can not be measured accurately due to personal error of inputting data points or selecting clusters. Its limitations are due to effect of noise, outliers and the difficulty to identify the initial partitions [39].

Another feature of clustering is to efficiently store the clustered data. For the purpose, similarity indexes of *VSM* are highly acceptable as they use characteristic vector to represent every object and to calculate its similarity in dimensional space. It uses various coefficient like *Cosine*, *Jaccard*, *Euclidean* and *Dice* that helps in forming disjoint sets of similar objects [18]. These models are used for information filtering, data retrieval, indexing and relevancy rankings. Another suitable example is Sensor Semantic Overlay Networks (*SSONs*) which is an approach for sensor search on the basis of context information. It performs cluster formation using an ant colony algorithm and has discussed various text retrieval techniques based on vector space model for maintaining the optimal balance between processing efficiency and retrieval effectiveness for the documents. Moreover, it tends to eliminate network challenges like the limited energy efficiency, lifetime of a network, scalability, data aggregation capabilities, communication bandwidth and time consumption [8]. Google Similarity Distance is used for clustering *IoT* services in collaboration with skewness-aware clustering tree. This approach does not work well with increase in the number of resources [132].

Based on the literature survey on clustering, Table 2.3 summarizes various hard and soft clustering based algorithms applied on different application domains with several parameters.

**Table 2.3: Literature Survey on Clustering**

Author	Year	Clustering Approach	Platform or Simulator	Dataset	Parameters	Advantages	Disadvantages
Cai et al. [30]	2007	Fuzzy c-means	MATLAB	Synthetic images dataset	<ul style="list-style-type: none"> <li>• Segmentation.</li> <li>• Computation time.</li> </ul>	<ul style="list-style-type: none"> <li>• Enhances clustering performance.</li> <li>• Provides fast segmentation.</li> <li>• Preserves the detail of the image.</li> </ul>	<ul style="list-style-type: none"> <li>• Robustness to noise.</li> <li>• Time consuming.</li> <li>• Difficulty in computation for <i>RGB</i> images.</li> </ul>
Kannan et al. [33]	2013	Noise distance based fuzzy c-means	MATLAB	Random datasets	<ul style="list-style-type: none"> <li>• Total Iterations.</li> <li>• Number of clusters.</li> <li>• Running time.</li> <li>• Clustering accuracy.</li> </ul>	<ul style="list-style-type: none"> <li>• It helps in reformulation and interpretation of clusters in same space.</li> <li>• It eradicates noise.</li> <li>• Provides effective segmentation.</li> </ul>	<ul style="list-style-type: none"> <li>• Noise affects</li> <li>• The approaches euclidean distance method is replaced by kernel induced</li> </ul>
Amini et al. [29]	2014	Density based	Java	DARPA Intrusion Detection 1998	<ul style="list-style-type: none"> <li>• Execution time.</li> <li>• Memory usage.</li> <li>• Clustering quality.</li> <li>• Mutual information.</li> </ul>	<ul style="list-style-type: none"> <li>• Data points are mapped to grids.</li> <li>• Low computation.</li> <li>• <math>O(mi+g)</math> storage space.</li> <li>• High quality clustering.</li> </ul>	<ul style="list-style-type: none"> <li>• Limited time for evaluation.</li> <li>• Clustering quality is less if value for lambda is too small.</li> <li>• Not suitable for distributed environments.</li> </ul>

to be cont'd on next page

**Table 2.3: Literature Survey on Clustering (contd.)**

Author	Year	Clustering Approach	Platform or Simulator	Dataset	Parameters	Advantages	Disadvantages
Bangoria [10]	2014	K Means Clustering	MATLAB	Random dataset	<ul style="list-style-type: none"> <li>• Accuracy.</li> <li>• Time complexity.</li> </ul>	<ul style="list-style-type: none"> <li>• Clusters are formed by considering nearest mean.</li> <li>• Simple and flexible.</li> </ul>	<ul style="list-style-type: none"> <li>• User need to specify number of cluster in advance.</li> <li>• It is fully dependent on selection of initial centroids.</li> <li>• High computation.</li> </ul>
Zhou et al. [32]	2015	Heuristic based	MATLAB	Simulation of random nodes	<ul style="list-style-type: none"> <li>• Computation time.</li> <li>• Hop Count.</li> <li>• Deployment process.</li> </ul>	<ul style="list-style-type: none"> <li>• Models the multiple traveling salesman problem.</li> <li>• Reduces computation cost.</li> <li>• Monitors the route with each cluster.</li> </ul>	<ul style="list-style-type: none"> <li>• Long computation time.</li> <li>• Fault recovery.</li> </ul>
Chang et al. [44]	2015	Neuro-fuzzy clustering	MATLAB	Sample Set	<ul style="list-style-type: none"> <li>• Service reliability.</li> <li>• Fault tolerance.</li> <li>• Error analysis (<i>MSE</i>).</li> </ul>	<ul style="list-style-type: none"> <li>• Visualize potential knowledge granules.</li> <li>• Structural analysis of large scale clustered data.</li> </ul>	<ul style="list-style-type: none"> <li>• Semantic representation of knowledge is missing.</li> </ul>
Malki et al. [133]	2016	K means with Genetic Algorithm	MATLAB	Random data points	<ul style="list-style-type: none"> <li>• Data points.</li> <li>• Total clusters.</li> </ul>	<ul style="list-style-type: none"> <li>• Eliminates empty clusters.</li> <li>• Fast method of clustering.</li> <li>• No overlapping of clusters.</li> </ul>	<ul style="list-style-type: none"> <li>• User-specific for clusters.</li> <li>• Limited data points.</li> </ul>

to be cont'd on next page

**Table 2.3:** Literature Survey on Clustering (contd.)

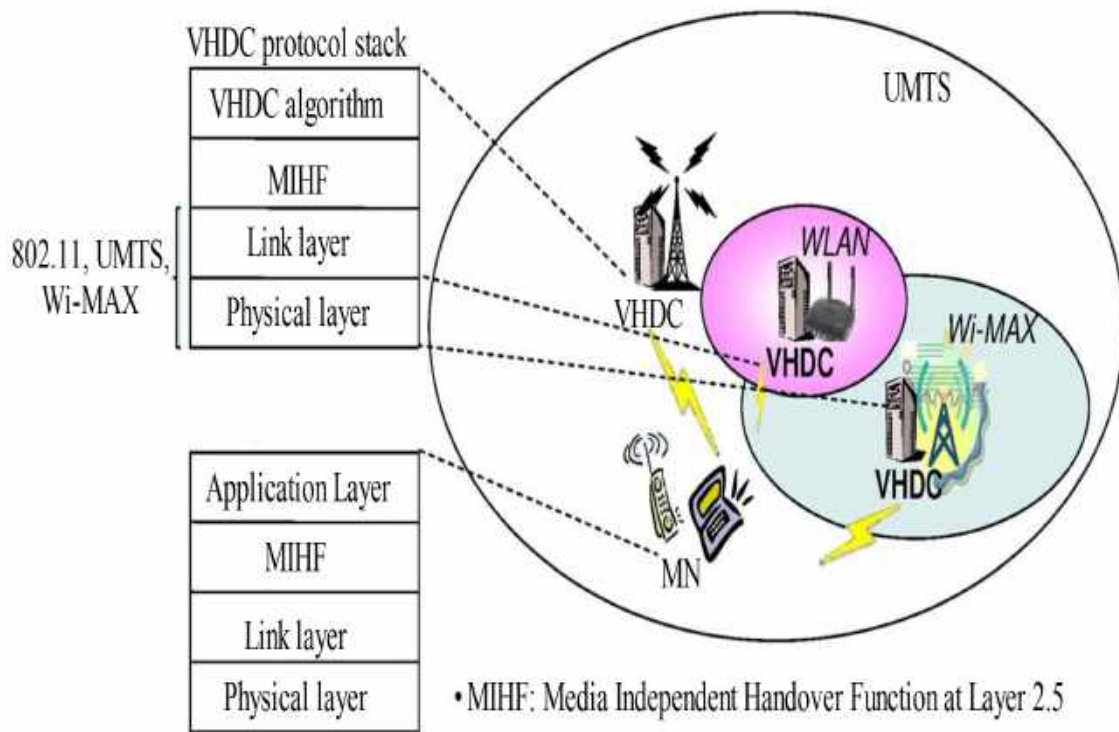
<b>Author</b>	<b>Year</b>	<b>Clustering Approach</b>	<b>Platform or Simulator</b>	<b>Dataset</b>	<b>Parameters</b>	<b>Advantages</b>	<b>Disadvantages</b>
Zhang et al. [31]	2017	CFS Algorithm	MATLAB	UCI Datasets	<ul style="list-style-type: none"> <li>• Accuracy.</li> <li>• Information complexity.</li> </ul>	<ul style="list-style-type: none"> <li>• New data points are updated in generated clusters.</li> <li>• Computation of new cluster head with each iteration.</li> <li>• Helps in improving industrial services.</li> </ul>	<ul style="list-style-type: none"> <li>• Collision may occur due to more data points.</li> <li>• Limited to small cloud platform.</li> </ul>
Xie et al. [36]	2017	Fuzzy clustering	MATLAB	Intel Berkeley Laboratory	<ul style="list-style-type: none"> <li>• Loss of data rate.</li> <li>• Total number of anonymous records.</li> </ul>	<ul style="list-style-type: none"> <li>• Improves the efficiency of protection of data.</li> <li>• Enhances information loss.</li> <li>• Retains the quality of data.</li> </ul>	<ul style="list-style-type: none"> <li>• Time sampling is continuous period.</li> <li>• Sampling time interval is very short.</li> </ul>

Clustering faces few limitations in terms of overheads during the process of selection, addressing and forming of clusters. The key challenge is related to time consumption as it handles trillions of request in parallel and demand fast response. Moreover, a time constraint application raise priority issues which needs to be resolved to prevent deadlock condition. The limitations of clustering are dynamic size of the clusters, high complexity, they fail to provide effective hard clustering solutions in real time applications and clustering methods do not consider the previously updated clusters.

### **2.3.2 Resource Selection**

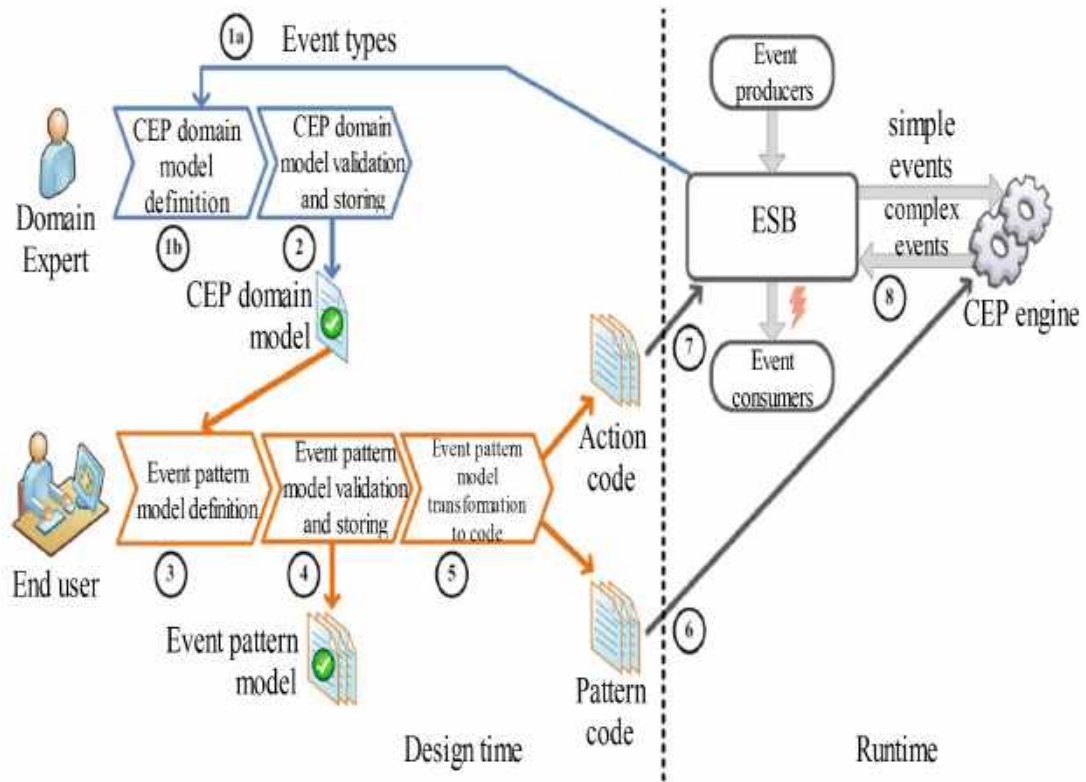
There are various techniques existed for resource selection. These techniques are broadly categorized into pattern, nature inspired, supervised learning, optimization and trust based methodologies and are discussed below.

***Pattern based Selection:*** Patterns are a popular method for helping to define data movement, integration, processing problems and solutions. A number of approaches are suggested by researchers: for example, T. G. Kolda *et al.* have provided a unifying framework that lends itself to a variety of convergence results considering the special properties of problems of direct search methods [134]. The framework allows generalization to handle bound constraints and linear constraints. Also, Y. Tao *et al.* have suggested a method to capture the performance of nearest neighbor queries using approximation [135]. The effectiveness of the method is demonstrated to solve several optimization problems related to nearest neighbor search. However, the method does not provide effectiveness to real world entity search problems. Therefore, S. Sergi *et al.* have derived an optimal solution to the problem of cooperative data transmission based on distributed selection diversity in an Adhoc wireless network consisting of selfish nodes [136]. The strategy allows to coordinate the transmissions among the potential relays without any explicit information exchange between them to avoid the transmission overheads. To balance the overall load, a Vertical Hand-Off (*VHO*) decision algorithm has been developed to balance the overall load among all attachment points (*e.g.*, base stations and access points) and maximizes the collective battery lifetime of Mobile Nodes (*MN*) [137] (see Figure 2.17).



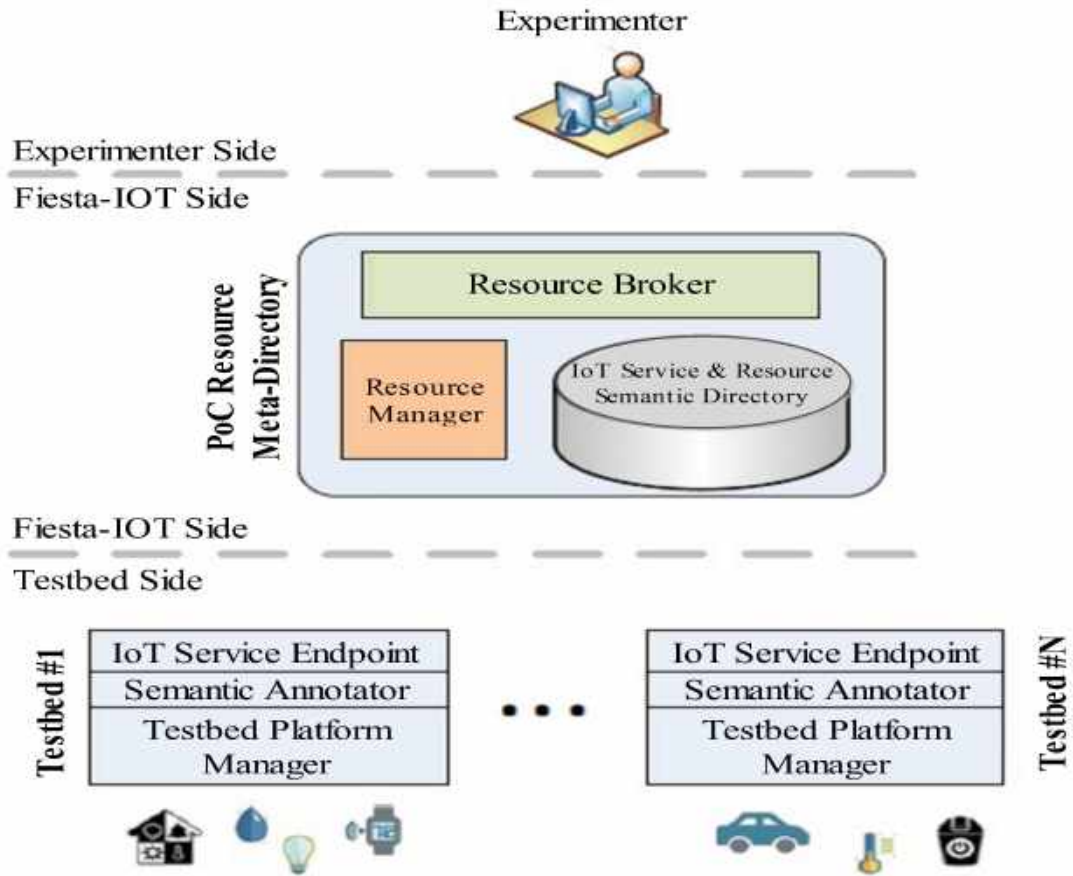
**Figure 2.17:** Vertical Hand-off Decision Controller Implementation based on *IEEE 802.21* [137]

The algorithm has its limitations that it does not help in minimizing battery consumption if the *MN* are distributed in nature. A resource discovery algorithm based on preference and movement pattern similarity in disconnected and delay-tolerant Social Internet of Things (*SIoT*) has been postulated [139]. The algorithm implements a three dimensional cartesian coordinate system with the aim of enhancing the search efficiency over the *SIoT*. It lacks in modeling effective behavior prediction model to reduce the wait time of the nodes in mobile *SIoT*. To reduce wait time, J. Boubeta *et al.* have postulated a model-driven solution for real-time decision making in event-driven *SOA* [140]. This approach allows the integration of Complex Event Processing (*CEP*) and event pattern through a graphical and intuitive editor, which also permits automatic code generation (see Figure 2.18).



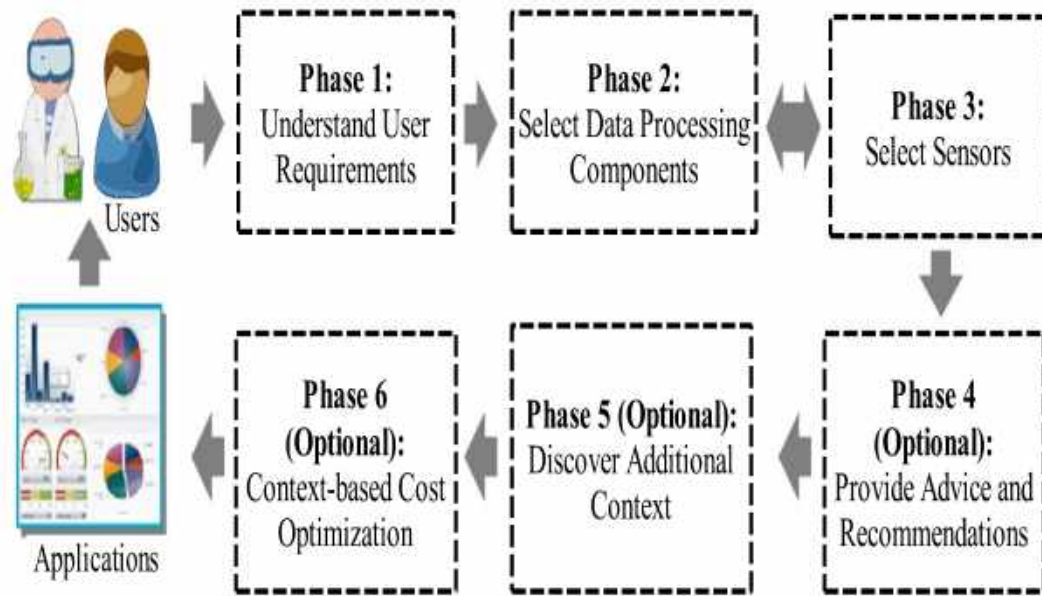
**Figure 2.18:** Model-Driven Approach for *CEP* in *SOA 2.0* [140]

The solution has its disadvantages that it has not recommended event patterns in a semi-automatic way which leads to delayed interpretation of existing pattern usage statistics. To provide event patterns, J. Lanza *et al.* have described a proof-of-concept implementation that federates two different *IoT* experimentation facilities by means of semantic-based technologies (*see* Figure 2.19).



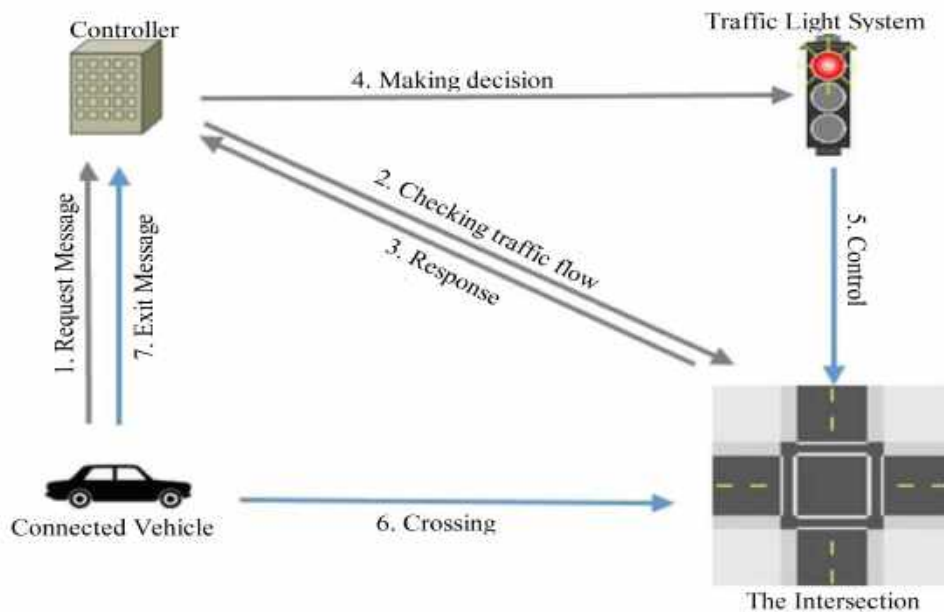
**Figure 2.19:** Proof-of-Concept System Functional Architecture [141]

The method transcends the (silo) boundaries of individual deployments, based on the semantic interconnection and interoperability of diverse *IoT* platforms and testbeds [141]. To address the gap of interoperability, a knowledge driven approach called Context Aware Sensor Configuration Model (*CASCoM*) has been formulated [41]. It simplify the process of configuring *IoT* middleware platforms, so the data consumers, specifically non-technical personnel, can easily retrieve the required data as shown in Figure 2.20.



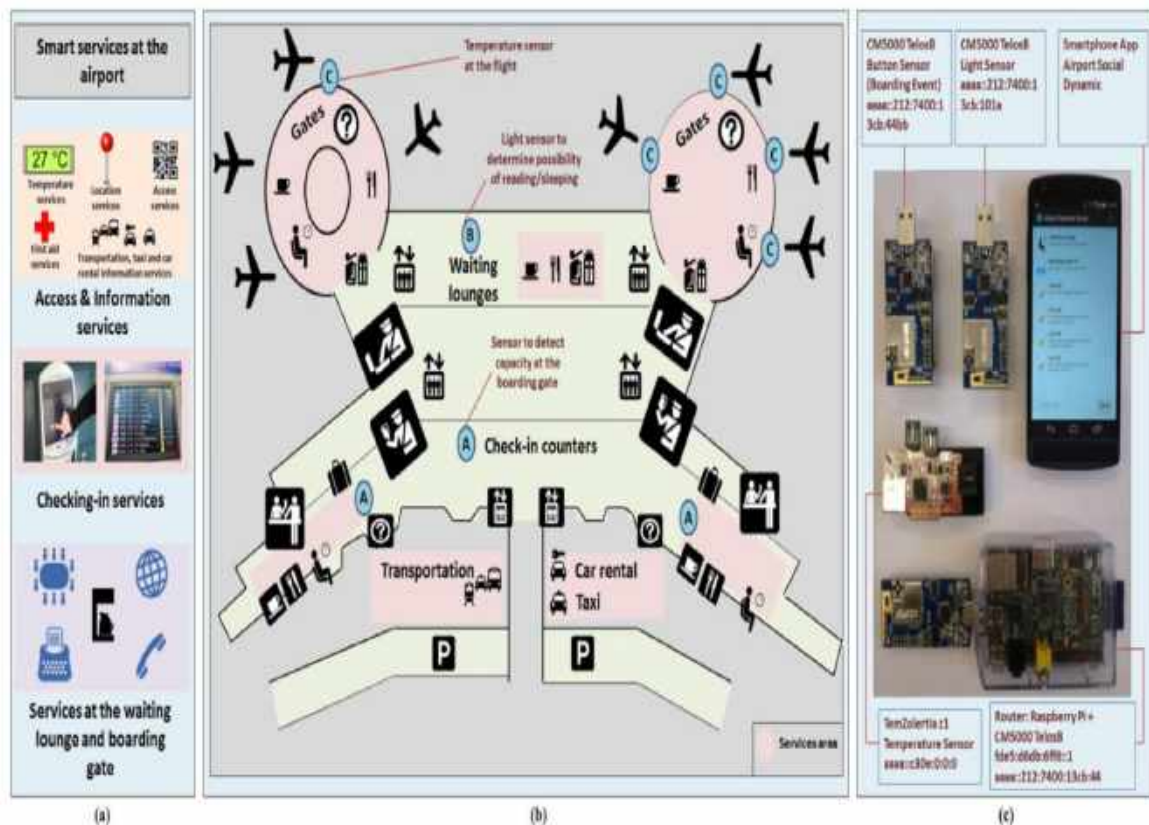
**Figure 2.20:** Context-Aware Sensor Configuration Model (CASCoM) [41]

It faces certain drawbacks with respect to privacy violation that may occur when data processing components and sensors are composed together. The researchers have suggested a connected intersection system where every objects such as vehicles, sensors, and traffic lights will be connected and sharing information to one another as shown in Figure 2.21.



**Figure 2.21:** System Model [143]

The optimization algorithms for traffic lights by applying algorithmic game theory are also suggested, to deal with different scenarios of traffic flow. The approach has its disadvantages that it takes into account the priority of vehicles which are at same level of emergency. To provide privacy to the data over the Internet, D. Hussein *et al.* have formulated a novel service framework based on a cognitive reasoning approach for dynamic *SIoT* services discovery in smart spaces [142]. The reasoning about users' situational needs, preferences, and other social aspects along with users' surrounding environment is suggested for generating a list of situation-aware services which matches users' needs. This reasoning is implemented as a proof-of-concept prototype, *namely*, Airport Dynamic Social, within a smart airport as shown in Figure 2.22.

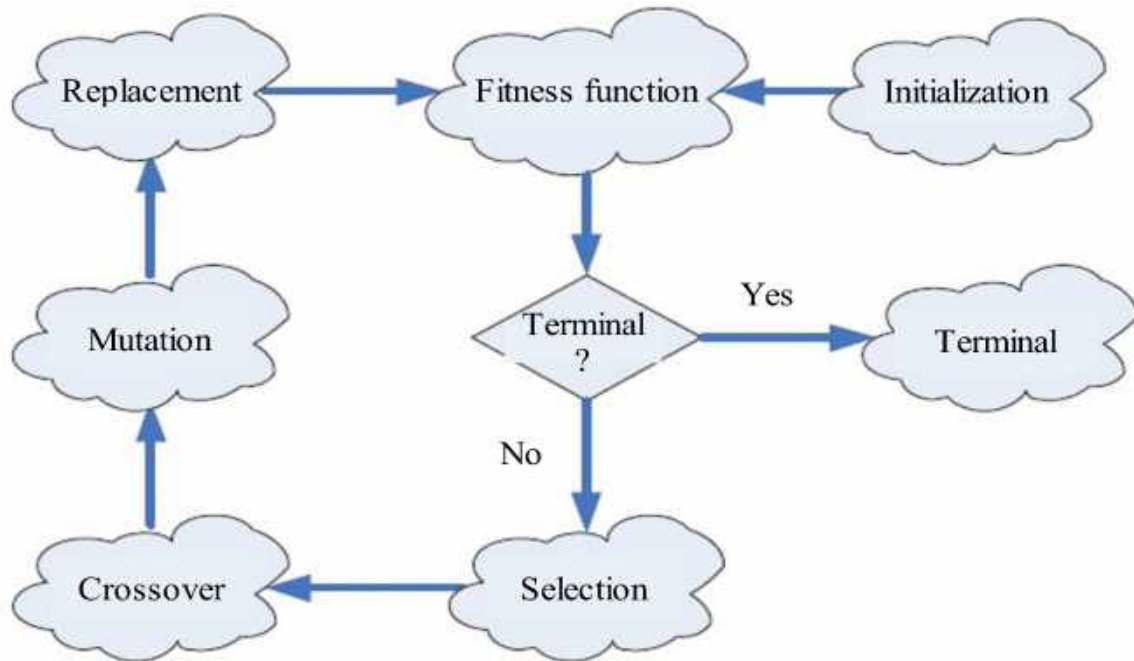


**Figure 2.22:** Airport Dynamic Social Application Environment and Equipment: (a) Example of Smart Services at the Airport, (b) Airport Dynamic Social Environment, (c) Airport Dynamic Social Equipment [142]

K. H. N. Bui *et al.* have introduced a new approach for smart traffic light control at intersection [143]. To consider priority at different level, A. G. de Prado *et al.* have

postulated a COLLaborative ConText aware service oriented architecture (*COLLECT*), which facilitates both the integration of *IoT* heterogeneous domain context data through the use of a light message broker, easy data delivery among several agents and collaborative participants in the system, and making use of an enterprise service bus [144]. The approach helps to avoid additional resource consumption to edge devices and saving costs in cloud hosting. The architecture is not feasible for real time prediction using contextual information to improve intelligent decision making in the domain.

***Nature Inspired based Selection:*** The nature-inspired computing paradigm is fairly vast and is used where (i) problem is complex, nonlinear and involves a large number of variables or potential solutions or has multiple objectives, (ii) problem to be solved cannot be suitably modeled using conventional approaches such as complex pattern recognition, (iii) finding an optimal solution is not possible or cannot be guaranteed and (iv) problem lends itself to a diversity of solutions or a diversity of solutions is desirable. Nature-inspired computing refers to a class of meta-heuristic algorithms that imitate or are inspired by some natural phenomena. For example, J. Arunadevi *et al.* have implemented a Parallel Genetic Algorithm (*PGA*) using High Performance Cluster (*HPC*) to address the problem of selecting route to a given destination on an actual map under a static environment [145]. The algorithm has no adaptability to dynamic environments. To provide adaptation to dynamic surroundings, some researchers have formulated to use the Genetic Algorithm (*GA*) to alleviate the rising computational cost and to find the shortest time in driving with diverse scenarios of real traffic conditions and varying speeds of vehicles (*see* Figure 2.23).



**Figure 2.23:** Schematic Genetic Algorithm [146]

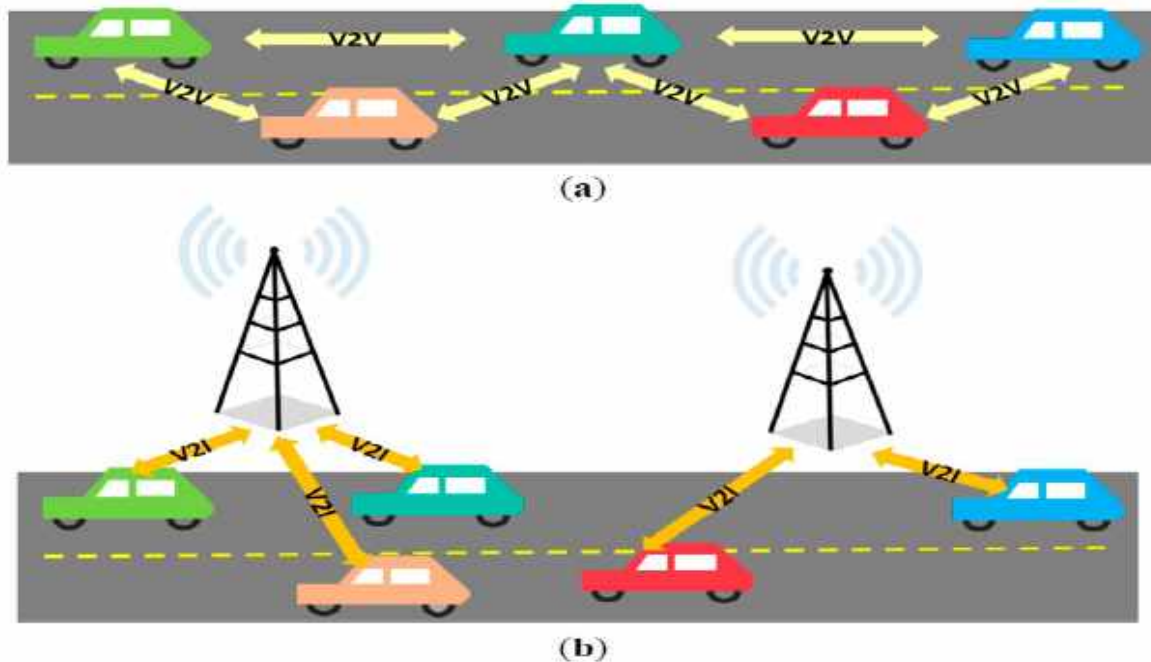
The evaluation results suggest that genetic algorithm is feasible to be used in the route guidance system but due to limited computation power and memory space on the hand-held devices, *GA* is not practical to search for the exact optimal solution on hand-held devices with huge traffic data [146]. To deal with huge traffic, L. Mourelle *et al.* have investigated migration strategies using *GA* for the execution of *PGA* in a Multi-Processor System-on Chip (*MPSoC*) [147]. The researchers use a network-on-chip platform to implement these algorithms, which provides the interconnection network required for the communication between processors.

***Supervised Learning based Selection:*** Supervised learning is the machine learning that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples. For example, J. Tang and Y. Hu have implemented a new non-monotone line search which is a novel form of the Armijo rule to choose a larger step size at each iteration [148]. The method possibly reduces the function evaluation and a new memory gradient method is investigated to prove the suggested method's global convergence. Also, S. M. Asad and A. Zerguine have used

Armijo rule for the selection of the learning rate to introduce the Armijo rule Learning Rate Least Mean Fourth (*ALRLMF*) algorithm [149]. The algorithm exhibits excellent tracking capability of the time-varying statistics of highly correlated input signals by achieving faster convergence. To improve the performance of system for providing optimal solution in less time, P. Henning and M. Kiefel have shown that the Quasi-Newton algorithms can be interpreted as approximations to Bayesian regression under Gaussian and other priors [150]. They use a kernel model to learn from all observations in each line-search, explicitly track uncertainty, and thus achieve faster convergence towards the true Hessian. For achieving faster convergence, various reduction techniques to overcome the weakness of slow convergence of Stochastic Gradient Descent (*SGD*) have been postulated [151]. These methods are either based on computations of full gradients at pivot points, or on keeping per data point corrections in memory. Therefore, speed-ups relative to *SGD* may need a minimal number of epochs in order to materialize. These algorithms exploit neighborhood structure in the training data to share and re-use information about past stochastic gradients across data points, which offers advantages in the transient optimization phase. A Regularized Newton Method (*RNM*) has been postulated without line search [152]. The method controls a regularization parameter instead of a step size in order to guarantee the global convergence. It is shown that by using the suggested method, the tightness of the global complexity bounds are easily constructed which solves subproblem inexactly.

***Optimization Techniques:*** Optimization is defined as the process of finding the conditions that give the minimum or maximum value of a function, where the function represents the effort required or the desired benefit. The optimization techniques include data collection, problem formulation, model development, validation and interpretation. As an example, R. M. Lewis *et al.* have discussed direct search methods for unconstrained optimization characterized by the absence of the construction of a model of the objective [153]. The direct search methods are based on straight forward heuristics to ensure global convergence of sequence of iterates to a first-order stationary point of the objective function. Also, J. Barcelo *et al.* have presented two methodologies to microscopic traffic simulation to support traffic management decisions [154]. The first methodology is an extension of

the traditional microscopic simulation paradigm in which, instead of a pure stochastic emulation of traffic flows on a road network, vehicles are assigned to paths from origins to destinations according to stochastic route choice models. The second methodology aims for the analysis of traffic scenarios for conflicting situations using the microscopic simulation and is implemented in a new software platform Intermodal Strategy Manager (*ISM*). T. E. Abrudan has formulated a Riemannian geometry approach for optimization of a real-valued cost function of complex-valued matrix argument [155]. The authors have derived Steepest Descent (*SD*) algorithms on the Lie group of unitary matrices. The algorithm is applied to blind source separation in *MIMO* systems by using the joint diagonalization approach and is evaluated for its computational complexity. The algorithm has lower computational complexity and refines estimates of gradient obtained by classical means. N. Parrado and Y. Donoso have formulated an optimization model over the load balancing in the congestion percentage of the streets [156]. A fully congestion-oriented route discovery mechanism is explored and Vehicle-to-Vehicle (*V2V*) and Vehicle-to-Infrastructure (*V2I*) supportive communication infrastructure are suggested in which nodes are able to interact with vehicles as shown in Figure 2.24.



**Figure 2.24:** Representation of *V2V* and *V2I* Technologies (a) *V2V* Communication, (b) *V2I* Communication [156]

The model is robust and applied in a real context for its performance measurements in terms of both efficiency and fairness.

**Trust based Selection:** Trust calculations become an important issue to be resolved in the *IoT* because number of connected resources are dynamically changed or they can move to different networks freely. To make data exchange among objects can be done safely as it requires appropriate recommendations [157]. As an example, a trust model of choosing trusted source peers in *P2P* system has been presented which is based on the fuzzy relation theory in fuzzy mathematics [158]. The simulation results show that this fuzzy relation trust model has good performance on choosing trusted source peers and filtering unfair recommendations. The model has its disadvantages that the current maximum and product operators cannot show the different weight of different acquaintance and different recommending levels that leads to false selection of trusted source peers in *P2P* system. To select trusted peers, J. Luo *et al.* have postulated *RFSTrust*, a trust model based on fuzzy recommendation similarity, to quantify and to evaluate the trustworthiness of nodes [159]. The model deals with uncertainty and the tolerance of imprecise data inputs for the subjective tasks of trust evaluation, packet forwarding review and credibility adjustment. The model is needed to improve to protect the security of *MANETs* more effectively. Also, S. S. Babu *et al.* have postulated a new geometric mean based trust management system by evaluating direct trust from the *QoS* characteristics (trust metrics) and indirect trust from recommendations by neighbor nodes, which allows for trusted nodes only to participate in routing [161]. It is presented that the trusted relations change by changing the number of trust metrics, direct trust threshold and total trust threshold. The system has its disadvantages that it is adhered to various attacks when it considers the randomly deployed *WSN* with live trust metric parameter values. Also, D.Chen *et al.* have presented a Trust and Reputation Model (*TRM-IoT*) to enforce the cooperation between things in a network based on their behaviors and have considered various security challenges such as detection of malicious attacks [162]. The model facilitates the detection of untrustworthy entities, and assist the decision-making process of various protocols. The model does not provide efficient local trust as it is not adhered to dynamic addition of new nodes into the network. Also, S. E. A. Rafey *et al.* have

presented a Context-based Social Trust model for the Internet of Things (*CBSTM-IoT*) to integrate numerous factors such as direct and indirect trust, transaction context, owner trust, and social modeling of trust [165]. Performance evaluation results validate that *CBSTM-IoT* benefits from using both direct observations and indirect recommendations to counter attacks and achieves satisfactory results in isolating malicious nodes in the network. The model has its limitations as it has not considered different types of trust such as cooperativeness, other errors generated without malevolent intentions and to eradicate in direct trust based observations. The literature survey is summarized in Table 2.4.

**Table 2.4:** Summary of Literature Survey

Authors	Approaches	Strength	Weaknesses
N. Koshizuka and K. Sakamura [4]	Spatial <i>FCM</i>	<ul style="list-style-type: none"> <li>•Considers membership matrix</li> <li>•Stores spatial information of clusters</li> <li>•Helps in improving segmentation results</li> <li>•Obtains the log for cluster distribution</li> </ul>	<ul style="list-style-type: none"> <li>•Fails to deal with robustness to noise</li> <li>•More time consuming</li> </ul>
K. Finkenzeller [8]	Sensor Semantic Overlay Networks	<ul style="list-style-type: none"> <li>•Based on context information</li> <li>•Performs cluster formation</li> <li>•Maintains the optimal balance between processing efficiency and retrieval effectiveness</li> <li>•Eliminates network challenges like the limited energy efficiency, lifetime of a network, scalability</li> </ul>	<ul style="list-style-type: none"> <li>•Does not work well with increase in the number of resources</li> </ul>
G. V. Crosby and F. Vafa [9]	Intuitionist <i>FCM</i> clustering	<ul style="list-style-type: none"> <li>•Considers erred parameter</li> <li>•Describes the membership matrix</li> <li>•Results into clusters centers that converge to desired location</li> </ul>	<ul style="list-style-type: none"> <li>•Effect of noise is more</li> <li>•Difficult to identify the initial partitions</li> </ul>

to be cont'd on next page

**Table 2.4:** Summary of Literature Survey (contd.)

Authors	Approaches	Strength	Weaknesses
A. Dunkels <i>et al.</i> [10]	k-Means Clustering	<ul style="list-style-type: none"> <li>•Works for compact and hyper-spherical clusters</li> <li>•Clusters data without any labels</li> <li>•Minimizes an objective function using squared error function</li> </ul>	<ul style="list-style-type: none"> <li>•Fails to provide the mechanism for the selection of initial centroid</li> <li>•Results into delay in selecting final clustered data points</li> </ul>
Q. Cao <i>et al.</i> [12]	Constrained specific algorithm	<ul style="list-style-type: none"> <li>•Incurs low communication costs</li> <li>•Achieves constant approximation ratios</li> </ul>	<ul style="list-style-type: none"> <li>•Does not provide the method to achieve similar approximation ratio</li> <li>•Communication complexity is high</li> </ul>
L. M. Xing Xiaojiang and Wang Jianli [16]	Robust learning-based <i>FCM</i> framework	<ul style="list-style-type: none"> <li>•Automatically finds the best number of clusters</li> <li>•Free from initialization parameters and selecting the clusters</li> <li>•Rejects the erroneous or noisy points</li> </ul>	<ul style="list-style-type: none"> <li>•Does not handle high-dimensional data sets</li> <li>•Iteration time is decreased rapidly</li> </ul>
S. K. Datta and C. Bonnet [37]	Thing Description ( <i>TD</i> )	<ul style="list-style-type: none"> <li>•Semantic based descriptions <i>via</i> CoRE Link Format</li> <li>•Enable sensors semantically</li> <li>•Provides thing to thing communication</li> </ul>	<ul style="list-style-type: none"> <li>•Does not provide extensibility for connected vehicles</li> </ul>
C. Perera and A. V. Vasilakos [41]	Context Aware Sensor Configuration Model	<ul style="list-style-type: none"> <li>•Simplifies the process of configuring <i>IoT</i> middleware platforms</li> <li>•Easily retrieves the required data</li> <li>•Highly flexible and scalable</li> </ul>	<ul style="list-style-type: none"> <li>•Privacy violation</li> <li>•Error in composing data processing components and sensors together</li> </ul>

to be cont'd on next page

**Table 2.4:** Summary of Literature Survey (contd.)

Authors	Approaches	Strength	Weaknesses
S. Lu <i>et al.</i> [57]	Designed working flow of ETC	<ul style="list-style-type: none"> <li>•Method is part of ITS</li> <li>•It is GPS based ETC</li> <li>•Widely used in internal expressway</li> <li>•Charging ports are given at various points</li> </ul>	<ul style="list-style-type: none"> <li>•RFID frequently mismatched with BankID's</li> <li>•Charging at ports worn out in less time</li> </ul>
D. Gu and H. Hu [58], A.P. Gupta and N. Goyal [59]	Distributed Gaussian Process Regression (DGPR)	<ul style="list-style-type: none"> <li>•Generates spatial function</li> <li>•Supports covariance function</li> <li>•Enables mobile nodes' network to enhance spatio-temporal functions</li> </ul>	<ul style="list-style-type: none"> <li>•Not tested practically</li> <li>•Not feasible with real-world environment</li> </ul>
C. Perera <i>et al.</i> [61]	Context-Aware Sensor search, Selection, And Ranking Model (CASSARAM)	<ul style="list-style-type: none"> <li>•Deals with challenges of sensor selection</li> <li>•Resolves issues related with overlapping of large set of sensors</li> <li>•Resolves issues of generation of redundant information from sensors</li> </ul>	<ul style="list-style-type: none"> <li>•Does not achieve efficient results</li> <li>•Does not provide more functionality due to closely related semantic and quantitative reasoning</li> </ul>
A. Sheth <i>et al.</i> [63], Y. Qian <i>et al.</i> [64]	Semantic Computing (SC), Cognitive Computing (CC) and Perceptual Computing (PC)	<ul style="list-style-type: none"> <li>•SC deals with challenges of big data</li> <li>•CC helps to improve the understanding of data</li> <li>•CC provides the ability to utilize relevant knowledge</li> <li>•PC helps to generate massive amounts of data from the physical, cyber, and social domains</li> </ul>	<ul style="list-style-type: none"> <li>•Maintaining database is not possible</li> <li>•Data leakages are common due to tremendous amount of data</li> </ul>

to be cont'd on next page

**Table 2.4:** Summary of Literature Survey (contd.)

Authors	Approaches	Strength	Weaknesses
J. Pradilla <i>et al.</i> [65]	Sensor Observation Service ( <i>SOS</i> )	<ul style="list-style-type: none"> <li>• Fits small sensors in network environments</li> <li>• Do not need robust system for computation</li> <li>• <i>SOS</i> increases interoperability by providing interdependency of heterogeneous data</li> </ul>	<ul style="list-style-type: none"> <li>• Few sensors provide erred data</li> <li>• Computation cost is high</li> </ul>
E. Aguirre <i>et al.</i> [66]	Wireless body area network	<ul style="list-style-type: none"> <li>• Transmits medical information</li> <li>• Works within complex indoor e-health scenarios</li> <li>• Easy hand-held android device application</li> </ul>	<ul style="list-style-type: none"> <li>• Regular monitoring process is slow</li> <li>• Designed only for health-care services</li> </ul>
L. Shen <i>et al.</i> [67], V. Sharma <i>et al.</i> [68]	Modified CLIQUE Algorithm	<ul style="list-style-type: none"> <li>• Discovers web services</li> <li>• Indexes web services using <i>KPI</i></li> </ul>	<ul style="list-style-type: none"> <li>• Issue of embedding dynamic environment</li> </ul>
Z. Ding <i>et al.</i> [69]	Hybrid Search Engine technique	<ul style="list-style-type: none"> <li>• Provide dynamic searches</li> <li>• Based on Spatial-Temporal, Value-based, and Keyword-based conditions</li> <li>• Good solution for real time searches</li> </ul>	<ul style="list-style-type: none"> <li>• Search domain is limited</li> <li>• Issue of redundant data</li> </ul>
B. Djamaa and A. Yachir [70]	Proactive Resource Directory	<ul style="list-style-type: none"> <li>• Increases network performance</li> <li>• Advertises <i>RD</i> in an efficient manner</li> <li>• Implemented on Contiki Operating System</li> </ul>	<ul style="list-style-type: none"> <li>• Does not provide optimizability towards multiple directories</li> </ul>
M. A. Quddus <i>et al.</i> [71]	Map Matching algorithm	<ul style="list-style-type: none"> <li>• Uses fuzzy logic</li> <li>• Takes <i>GPS</i> augmented data of sensors</li> <li>• Provides continuous navigation</li> </ul>	<ul style="list-style-type: none"> <li>• Not evaluated for urban condition</li> <li>• Less accuracy</li> </ul>

to be cont'd on next page

**Table 2.4:** Summary of Literature Survey (contd.)

Authors	Approaches	Strength	Weaknesses
J. Cao <i>et al.</i> [72]	Type-2 fuzzy controller architecture	<ul style="list-style-type: none"> <li>•Evaluates positioning performance</li> <li>•Resolves non-linear problems</li> <li>•Provides solution to vehicle suspension</li> <li>•Constructs route switching between generated linear model control surfaces</li> </ul>	<ul style="list-style-type: none"> <li>•Not implemented for electric vehicle suspension system</li> </ul>
S.K. Das and S. Tripathi [74], M. Collotta and G. Pau [75]	Smart energy management solution	<ul style="list-style-type: none"> <li>•Based on fuzzy</li> <li>•Provides dynamic inputs</li> <li>•Manages consumer's feedback</li> <li>•Reduces peak load demand</li> </ul>	<ul style="list-style-type: none"> <li>•Not implemented in real-time</li> <li>•Generates overlapped resource's identification</li> </ul>
X. Su [76]	Text categorization	<ul style="list-style-type: none"> <li>•Based on description logic</li> <li>•Automates assigning of documents</li> <li>•Mitigates the problem of ontology mapping</li> </ul>	<ul style="list-style-type: none"> <li>•Rules are not fully described</li> <li>•Text mapping of different domains leading to complex information</li> </ul>
I. Tachmazidis <i>et al.</i> [77]	Semantic enrichment of the Hypercat Data Hub	<ul style="list-style-type: none"> <li>•Provides a center point to share wide range of resources</li> <li>•Translates existing data into <i>RDF</i> schema</li> </ul>	<ul style="list-style-type: none"> <li>•Does not provide direct access to spatial information</li> </ul>
C. Intanagonwivat <i>et al.</i> [80], D. Wijaya sekara <i>et al.</i> [81]	Directed diffusion paradigm	<ul style="list-style-type: none"> <li>•Performs sensing of environmental phenomena</li> <li>•Significant energy efficiency</li> </ul>	<ul style="list-style-type: none"> <li>•Not explored for entire space of alternative designs •features like data aggregation and caching are missing</li> </ul>

to be cont'd on next page

**Table 2.4:** Summary of Literature Survey (contd.)

Authors	Approaches	Strength	Weaknesses
R. C. Shah and J. M. Rabaey [84], J. H. Lee <i>et al.</i> [85]	Energy aware routing	<ul style="list-style-type: none"> <li>●Reduces energy consumption</li> <li>●Uses sub-optimal paths occasionally</li> <li>●Sends traffic through different routes</li> </ul>	<ul style="list-style-type: none"> <li>●Survivability of network protocols is not estimated</li> <li>●Process of computing the performance for network protocols is very slow</li> </ul>
S. Lindsey and C. S. Raghavend [87]	Power-Efficient GATHERING in Sensor Information Systems	<ul style="list-style-type: none"> <li>●Optimal chain-based protocol</li> <li>●Reduces the amount of energy spent per round</li> </ul>	<ul style="list-style-type: none"> <li>●Dependency on the platform</li> <li>●Slows down the performance of the system</li> </ul>
W. Liu <i>et al.</i> [88]	Improved algorithm based on <i>LEACH</i>	<ul style="list-style-type: none"> <li>●Balances regional energy</li> <li>●Increases system's performance</li> <li>●Effective in balancing the network energy consumption</li> </ul>	<ul style="list-style-type: none"> <li>●Consumes extra energy</li> <li>●Balancing of regional energy is random</li> </ul>
J. Singh <i>et al.</i> [89]	Hierarchical <i>LEACH</i> based routing strategy	<ul style="list-style-type: none"> <li>●Refreshes clusters periodically</li> <li>●Computes residual energy and distance</li> <li>●Enhances network's lifetime</li> </ul>	<ul style="list-style-type: none"> <li>●Does not find an optimal value for the dynamic threshold</li> <li>●Does not conserve energy at cluster head</li> </ul>
W. Twayej and H. S. Al Raweshidy [90]	Multi-Level Clustering Multiple Sink	<ul style="list-style-type: none"> <li>●Works with IPv6 protocol over Low Wireless Personal Area Networks</li> <li>●Enhances network's lifetime</li> </ul>	<ul style="list-style-type: none"> <li>●Issue of data conversion</li> <li>●Frequent connection losses to <i>IP</i> network</li> </ul>
H. Qu <i>et al.</i> [92]	Vector location-based algorithm	<ul style="list-style-type: none"> <li>●Based on Distributed Hash Tables</li> <li>●Decreases computational cost</li> <li>●Determines optimal route</li> </ul>	<ul style="list-style-type: none"> <li>●Works with limited nodes</li> <li>●Does not specify shortest route for data transmission</li> </ul>
J. Han and X.-Y. Li [94]	Pick-up game	<ul style="list-style-type: none"> <li>●Finds set of neighbors</li> <li>●Collects IDs of neighboring nodes</li> <li>●Minimizes the latency of neighbor-discovery process</li> </ul>	<ul style="list-style-type: none"> <li>●Does not account for multi-hop neighbor discovery</li> <li>●Consumes more energy</li> </ul>

to be cont'd on next page

**Table 2.4:** Summary of Literature Survey (contd.)

Authors	Approaches	Strength	Weaknesses
M. Kodialam and T. Nandagopal [95]	collision-based estimator, probabilistic estimators and two unified estimation algorithms	<ul style="list-style-type: none"> <li>•Enable the identification of <i>RFID</i> tags</li> <li>•Less consumption time</li> <li>•Eradicates neighborhood estimation problem</li> </ul>	<ul style="list-style-type: none"> <li>•BankID's mismatching takes place at some ports</li> <li>•Multiple RFID reader problem</li> </ul>
D. Luddecke <i>et al.</i> [96]	Model contextual information	<ul style="list-style-type: none"> <li>•Uses the concept of context aware in-car infotainment system</li> <li>•Beneficiary for modeling calculation of reliabilities</li> </ul>	<ul style="list-style-type: none"> <li>•Issue of interoperability</li> <li>•More complexity</li> </ul>
D. Shin <i>et al.</i> [97], R. Pozza <i>et al.</i> [98]	Context Aware Resource Discovery framework	<ul style="list-style-type: none"> <li>•Leverages Q-Learning techniques</li> <li>•Reduces wastage of energy</li> <li>•Decreases delays in resource search</li> </ul>	<ul style="list-style-type: none"> <li>•Limited processing capabilities</li> <li>•Less secure</li> </ul>
P. Gomes <i>et al.</i> [99]	Federated Discovery Service	<ul style="list-style-type: none"> <li>•Provides resource search</li> <li>•Describes resources and services semantically</li> </ul>	<ul style="list-style-type: none"> <li>•Trust and security issues</li> <li>•Computation time is high</li> </ul>
A. Zamanifar <i>et al.</i> [102]	Distributed Self-Healing Movement Prediction scheme	<ul style="list-style-type: none"> <li>•Predicts movement direction</li> <li>•Multi-user environment</li> <li>•Potential power of a <i>DDoS</i> attack has been determined</li> </ul>	<ul style="list-style-type: none"> <li>•One or more static sensors fail(s)</li> <li>•False positive movement direction prediction</li> </ul>
M. Z. Shafiq <i>et al.</i> [129]	Lloyd's k-means based Filtering algorithm	<ul style="list-style-type: none"> <li>•Reduces complexity</li> <li>•Performs data-sensitive analysis of the algorithm's running time</li> <li>•Helps in separating clusters with increase in number of data points</li> </ul>	<ul style="list-style-type: none"> <li>•Passes no information from one stage to the other</li> <li>•Overall performance is degraded</li> </ul>
J. Gantz and D. Reinsel [130]	<i>CLARANS</i>	<ul style="list-style-type: none"> <li>•Iteratively perform clustering process</li> <li>•Contains heuristic knowledge</li> <li>•Order-sensitive property</li> </ul>	<ul style="list-style-type: none"> <li>•Time consuming process</li> <li>•Problem arises for selecting and generating a possible cluster</li> </ul>

to be cont'd on next page

**Table 2.4:** Summary of Literature Survey (contd.)

<b>Authors</b>	<b>Approaches</b>	<b>Strength</b>	<b>Weaknesses</b>
T. G. Kolda <i>et al.</i> [134]	Unifying framework	<ul style="list-style-type: none"> <li>●Provides convergence results</li> <li>●Considers direct search methods</li> <li>●Allows generalization to handle bound constraints and linear constraints</li> </ul>	<ul style="list-style-type: none"> <li>●Handling bound constraints are complex</li> <li>●Manual constraints are provided</li> </ul>
Y. Tao <i>et al.</i> [135]	Approximation method	<ul style="list-style-type: none"> <li>●Captures the performance of nearest neighbor queries</li> <li>●Efficient to compute and accurate for up to 10 dimensions</li> </ul>	<ul style="list-style-type: none"> <li>●Does not provide effectiveness to real world entity search problems</li> </ul>
S. Lee <i>et al.</i> [137]	Vertical Hand-Off decision algorithm	<ul style="list-style-type: none"> <li>●Balances the overall load</li> <li>●Allows to coordinate the transmissions among the potential relays</li> <li>●Cooperative data transmission</li> </ul>	<ul style="list-style-type: none"> <li>●Does not help in minimizing battery consumption</li> <li>●Does not incorporate distributed nature of mobile node</li> </ul>
Z. Li <i>et al.</i> [139]	Resource discovery algorithm	<ul style="list-style-type: none"> <li>●Based on preference and movement pattern similarity</li> <li>●Delay-tolerant Social Internet of Things</li> <li>●Enhances the search efficiency</li> <li>●Reduces the system overheads</li> </ul>	<ul style="list-style-type: none"> <li>●Lacks in modeling effective behavior prediction model</li> <li>●Does not encounter to reduce the wait time of the nodes</li> </ul>
J. Boubeta Puig <i>et al.</i> [140]	Model-driven solution	<ul style="list-style-type: none"> <li>●Real-time decision making in event-driven <i>SOA</i></li> <li>●Allows the integration of Complex Event Processing (CEP) and event pattern</li> <li>●Model patterns about situations of interest</li> </ul>	<ul style="list-style-type: none"> <li>●It has not recommended event patterns in a semi-automatic way</li> <li>●Delayed interpretation of existing pattern usage statistics</li> </ul>

to be cont'd on next page

**Table 2.4:** Summary of Literature Survey (contd.)

Authors	Approaches	Strength	Weaknesses
D. Hussein <i>et al.</i> [142]	Service framework	<ul style="list-style-type: none"> <li>●Based on a cognitive reasoning approach</li> <li>●Generates a list of situation-aware services</li> <li>●Implemented on prototype, <i>namely</i> Airport Dynamic Social</li> </ul>	<ul style="list-style-type: none"> <li>●Sensor detection at doors may fail sometimes</li> <li>●Less secure</li> </ul>
K.-H. N. Bui <i>et al.</i> [143]	Connected intersection system	<ul style="list-style-type: none"> <li>●Used for smart traffic light control</li> <li>●Traffic lights will be connected and sharing information to one another</li> <li>●Collects effectively and mobility traffic flow at intersection in real-time</li> </ul>	<ul style="list-style-type: none"> <li>●It takes into account the priority of vehicles</li> <li>●Does not cater to deal with huge traffic</li> </ul>
A. G. de Prado <i>et al.</i> [144]	COLLaborativE ConText aware service oriented architecture	<ul style="list-style-type: none"> <li>●Facilitates the integration of <i>IoT</i> heterogeneous domain context data</li> <li>●Provides real-time data processing</li> <li>●Helps to avoid additional resource consumption</li> </ul>	<ul style="list-style-type: none"> <li>●Not feasible for real time prediction</li> <li>●Does not improve intelligent decision making in the domain</li> </ul>
J. Arunadevi <i>et al.</i> [145]	Parallel Genetic Algorithm	<ul style="list-style-type: none"> <li>●Addresses the problem of selecting route to a given destination</li> <li>●Most significant for network analysis</li> </ul>	<ul style="list-style-type: none"> <li>●No adaptability to dynamic environments</li> </ul>
C.-H. Lin <i>et al.</i> [146]	Genetic Algorithm	<ul style="list-style-type: none"> <li>●Alleviates the rising computational cost</li> <li>●Finds the shortest time in driving with diverse scenarios</li> </ul>	<ul style="list-style-type: none"> <li>●Not practical to search for the exact optimal solution</li> <li>●Does not deal with huge traffic data</li> </ul>

to be cont'd on next page

**Table 2.4:** Summary of Literature Survey (contd.)

Authors	Approaches	Strength	Weaknesses
L. Mourelle <i>et al.</i> [147]	Multi-Processor System-on Chip	<ul style="list-style-type: none"> <li>•Provides the interconnection network for the communication between processors</li> <li>•Analyzes the speedup and efficiency</li> </ul>	<ul style="list-style-type: none"> <li>•Computation cost is high</li> <li>•Works well with limited nodes</li> </ul>
S. M. Asad and A. Zerguine [149]	Armijo rule Learning Rate Least Mean Fourth	<ul style="list-style-type: none"> <li>•Minimizes the mean fourth error</li> <li>•Exhibits excellent tracking capability</li> <li>•Reduces the function evaluation</li> </ul>	<ul style="list-style-type: none"> <li>•Low system's performance</li> <li>•Complex non-monotone line search</li> </ul>
P. Hennig and M. Kiefel [150], T. Hofmann <i>et al.</i> [151]	Quasi-Newton algorithms and Stochastic Gradient Descent	<ul style="list-style-type: none"> <li>•Provides optimal solution in less time</li> <li>•Interpreted as approximations to Bayesian regression under Gaussian</li> <li>•Uses a kernel model to learn from all observations</li> </ul>	<ul style="list-style-type: none"> <li>•Slow convergence to optimal solution</li> <li>•Uncertainty in path tracking</li> </ul>
K. Ueda and N. Yamash [152], R. M. Lewis <i>et al.</i> [153]	Regularized Newton Method and Direct search methods	<ul style="list-style-type: none"> <li>•Deals with unconstrained optimization problems</li> <li>•Guarantees the global convergence</li> <li>•Easy construction of global complexity bounds</li> </ul>	<ul style="list-style-type: none"> <li>•Manual data input and output constraints</li> <li>•Time consumption is more</li> </ul>
T. E. Abrudan <i>et al.</i> [155]	Riemannian geometry approach	<ul style="list-style-type: none"> <li>•Considers real-valued cost function</li> <li>•Provides optimum solution along the geodesics and the local parametrization</li> <li>•Applied to blind source separation in MIMO systems</li> </ul>	<ul style="list-style-type: none"> <li>•Computational complexity is high</li> <li>•Slow processing</li> </ul>

to be cont'd on next page

**Table 2.4:** Summary of Literature Survey (contd.)

Authors	Approaches	Strength	Weaknesses
N. Parrado and Y. Donoso [156]	Congestion-oriented route discovery mechanism	<ul style="list-style-type: none"> <li>•Provides load balancing in the congestion percentage of the streets</li> <li>•Supportive communication infrastructure</li> <li>•Achieves lower congestion</li> </ul>	<ul style="list-style-type: none"> <li>•Less efficient</li> <li>•Congestion issue due to traffic</li> </ul>
F. Yu <i>et al.</i> [158]	Trust model	<ul style="list-style-type: none"> <li>•Chooses trusted source peers in <i>P2P</i> system</li> <li>•Based on the fuzzy relation theory</li> <li>•Good performance on choosing trusted source peers</li> </ul>	<ul style="list-style-type: none"> <li>•Current maximum and product operators cannot show the different weight</li> <li>•Leads to false selection of trusted source peers</li> </ul>
J. Luo <i>et al.</i> [159]	<i>RFSTrust</i>	<ul style="list-style-type: none"> <li>•Based on fuzzy recommendation similarity</li> <li>•Evaluates the trustworthiness of nodes</li> <li>•Deals with uncertainty and the tolerance of imprecise data inputs</li> </ul>	<ul style="list-style-type: none"> <li>•Security of <i>MANET's</i> is missing</li> <li>•Less efficient</li> </ul>
S. S. Babu <i>et al.</i> [161]	Geometric mean based trust management system	<ul style="list-style-type: none"> <li>•Evaluates direct trust using <i>QoS</i> characteristics</li> <li>•Allows for trusted nodes only to participate in routing</li> </ul>	<ul style="list-style-type: none"> <li>•It is adhered to various attacks</li> <li>•Considers only the randomly deployed <i>WSN</i></li> </ul>
D. Chen <i>et al.</i> [162]	Trust and Reputation Model	<ul style="list-style-type: none"> <li>•Enforces the cooperation between things in a network</li> <li>•Facilitates the detection of untrustworthy entities</li> <li>•Accuracy, robustness and lightness of the model is validated</li> </ul>	<ul style="list-style-type: none"> <li>•Does not provide efficient local trust</li> <li>•Not adhered to dynamic addition of new nodes</li> </ul>

to be cont'd on next page

**Table 2.4:** Summary of Literature Survey (contd.)

Authors	Approaches	Strength	Weaknesses
S. E. A. Rafey <i>et al.</i> [165]	Context-based Social Trust model	<ul style="list-style-type: none"> <li>●Integrates numerous factors such as direct and indirect trust, transaction context, owner trust</li> <li>●Benefits from using both direct observations and indirect recommendations to counter attacks</li> <li>●Achieves satisfactory results in isolating malicious nodes in the network</li> </ul>	<ul style="list-style-type: none"> <li>●Not considered different types of trust such as cooperativeness and malevolent intentions</li> <li>●Not provided method to eradicate in direct trust based observations</li> </ul>

## 2.4 Issues in Resource Discovery and Selection

- (i) With the exponential growth of resources on *IoT*, enormous amount of data is generated that varies in formats, domains and types, *i.e.*, heterogeneity. Such data poses different challenges for machines with respect to its storage, representation, interpretation, therefore, it becomes extremely difficult to mine and dig out the hidden information. In short, heterogeneity in data leads to the difficulty in communication and computation among resource across *IoT*' application areas. Hence, there is challenge for developing efficient indexing methods to deal with the challenges due to heterogeneity.
- (ii) With changing operating system, language, platform across different application areas on *IoT* paradigm, exponentially growing resources on *IoT* faces high level of dynamism in the network and results into unpredictable patterns. These patterns raise the issue of interoperability at various levels like radio access, protocol, semantic level, semantic and context level. Thus there is need for developing ontology's and semantic based techniques to support *IoT*.
- (iii) The scale of experiment considered to envision *IoT* leads to another basic challenge, *i.e.*, scalability. It puts issues while discovering location of the resources both locally

and globally.

- (iv) All the security goals must be achievable like confidentiality, non-repudiation, authorization, integrity and authentication. The challenge is to provide security into these goals.
- (v) Energy conservation determines the lifetime of *IoT* platform. Since smart devices use battery power as an energy source for sensing, processing data and communicating information. Battery-life is limited and also as sensors are placed in a hostile environment, replacing the battery is impractical. So, issue is to conserve or minimize the energy consumption or harvest energy from natural resources.
- (vi) Smart devices are prone to failure because of unattended environment. A smart device may fail due to software problem or energy exhaustion. If a few of smart nodes fail, *IoT* gateway should handle this type of fault tolerance. The issue is to provide more functionality to gateway in order to deal with fault tolerance in a way to replace the failed node.
- (vii) Research is expected to add to a thorough methodology in recognizing and determining conditions crosswise over applications. Hence decision making algorithm needs to be developed aiming to make context aware and optimized usage of offered resources.

## 2.5 Problem Formulation

Resource Discovery and Selection is found to be a promising field of research for the realization of vision of *IoT*. It will be definitely a major step to facilitate interaction, communication and computation across different application areas through the resources. The resource is basically a smart object that has some capabilities, properties, registration, configuration, and accessing methods. Their domains are either local or global having both physical and digital representations. They are expected to grow exponentially and would produce complex data. Such sheer volume of data due to issues like heterogeneity, interoperability, lack of standardization, periodic evaluations would act as a bottleneck for the mechanism of resource discovery and selection. These issues limit the resource discovery on *IoT* paradigm leading to challenges like (i) to provide answers to knowledge-based

queries, (ii) to develop models to exchange applications' context, (iii) to bridge the gap between device level and across applications, (iv) indexing and splitting of multiple parameters, and (v) sophisticated techniques for managing metadata. Thus, it concludes resource discovery as a fundamental challenge for the realization of *IoT* vision and has gained the researcher's interest all over the world.

To address the challenge, the various parameters that need to be focused are (i) to automate and access, (ii) local and remote aspects, (iii) several dimensions related to query, (iv) scope of query, and (v) to understand the search evaluation metrics. Keeping the perspective, the major focus of this research proposal is to investigate the basic operations such as data acquisition, knowledge representation, interpretation with appropriate reasoning and decision making for the selection. These processes together define the mechanism of resource discovery and selection and are needed to run an intelligent system on *IoT*.

## **2.6 Thesis Objectives**

Main objectives of this work are:

- (i) To study and analyze existing *IoT* technologies.
- (ii) To design/improve an algorithm for discovery of resources.
- (iii) To design/improve a decision making algorithm for resource selection.
- (iv) To verify the proposed technique.



# Clustering based Resource Discovery and Selection

---

## 3.1 Introduction

A resource with some capabilities, properties, registration, configuration and accessing methods require to be discovered in order to facilitate interaction, communication and computation among them on *IoT*. For this purpose, resources have to undergo both retrieval and ranking processes, *i.e.*, to capture information and to prioritize the resources, respectively. With the perspective, resource discovery require organization and analysis of the gathered complex descriptions as data<sup>1</sup>. The big volume of data would slow down the process of discovery due to issues like heterogeneity, interoperability, scalability, lack of standardization, periodic evaluations, *etc.* These issues lead to challenges like (i) to provide answers to knowledge-based queries, (ii) to develop models to exchange applications' context, (iii) to bridge the gap between device level and across applications, (iv) indexing and splitting of multiple parameters, and (v) sophisticated techniques for managing meta data. Thus, it concludes resource discovery as a fundamental challenge for the realization of *IoT* vision.

To meet these challenges, the various functions that need to be focused on are (i) to automate

---

<sup>1</sup>The contents are published as M. Bharti, R. Kumar, S. Saxena, "Clustering based Resource Discovery on Internet-of-Things ", *International Journal of Communication Systems*, Vol. 31, No. 5, pp. 1099-1131, 2018. SCI Indexed (IF: 1.278)

and access, (ii) local and remote aspects, (iii) several dimensions related to the query, (iv) scope of query, (v) to understand the search evaluation metrics. Moreover, *IoT* has multiple parameters to be focused, therefore, resource discovery demands techniques or algorithms that eliminate limitations for indexing and ranking of the resources. Also, the gathered data from unreliable physical measurements gets corrupted due to noise factor, hence, are not reliable. Due to limited memory, processing power and opportunistic presence of the resources, availability becomes another drawback that affects data integrity. Further, the enormous data generated by increasing *IoT* resources is sufficient to choke the entire underlying network infrastructure, resulting into slow down of searching process or halting user's request in frequent time.

Considering the parameters, the prerequisite to resource discovery is self-organization of resources so that they group themselves by creating homogeneous descriptive data that helps to understand and explore internal relations of resources. The requirement is achieved through *clustering*. The chapter, therefore, outlines a novel clustering technique, *namely*, Iterative K-means Clustering Algorithm (*IKm-CA*) to resolve the problems of resource discovery and selection. It groups contextual information into clusters using similarity measures coefficients of Vector Space Model (*VSM*) for efficient search. It performs knowledge representation and reduces search complexity with minimum time. The performance of the proposed technique is compared with Fuzzy k-means (*FKM*) and Fuzzy c-means (*FCM*) in collaboration with *VSM* to provide fair evaluation. It is experimentally verified with similarity measurements on the basis of various input queries.

## **3.2 Resource Clustering**

Clustering is indexed as basic principle applied after discovery of resources in *IoT* as it extracts the relationship among resources that results in knowledge formation. For the purpose, it uses unsupervised classification of patterns to form clusters of resources virtually with no prior knowledge. It results in effective knowledge representation, learning and planning with minimal overheads. Moreover, it helps with the linking of resources from one network to another and combines them to treat as a single cluster that provides

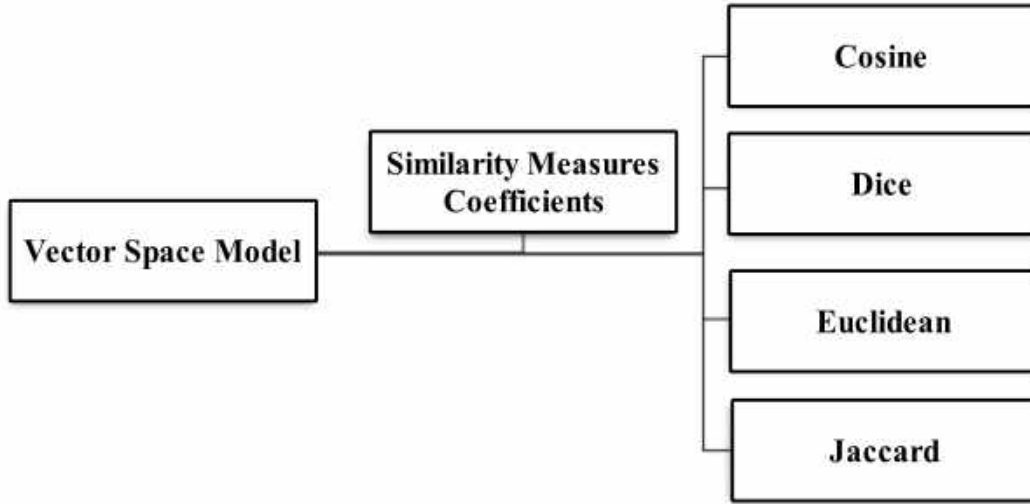
the continuation of resource discovery on the basis of query asked. Clustering helps synchronizing the resources to provide faster computation of search metrics required by the user. It improves the performance of the network on the basis of nodes' energy, lifetime of the network, scalability and aggregating data. It preserves bandwidth for communication and tries to avoid the redundant exchange of data among inter-cluster interactions. Thus, the clustering for resource discovery comes out to be reality for data handling, indexing, ranking and to provide fast results due to its properties of scalability, performance and flexibility. Moreover, it can adhere to dynamic changes to the resources and clusters them on the basis of similar storage type. But they have their own limitations of overheads during the process of selection, addressing and forming clusters. The key challenge is related to time consumption as it handles trillions of request in parallel and demand fast response. Moreover, a time constraint application raise priority issues which needs to be resolved to prevent deadlock condition. The limitations with clustering are dynamic size of the clusters, high complexity, fails to provide effective hard clustering solutions in real time applications and clustering methods do not consider the previously updated clusters. In summary, the major challenges to clustering that are needed to be addressed are computation time, erroneous data points, existence of empty clusters and noise factor.

### **3.2.1 Vector Space Model**

One major feature of clustering is to enable more efficient storing of data. For the purpose, similarity indexes of *VSM* are highly acceptable as it use characteristic vector to represent every object and to calculate its similarity in dimensional space. Various coefficients like *Cosine*, *Jaccard*, *Euclidean Distance* and *Dice* are used that helps in forming disjoint sets of similar objects [18] and performs information filtering, data retrieval, indexing and relevancy rankings.

In order to present each object of the cluster in multi-dimensional space along with its estimated similarity, *VSM* is suggested. Due to limited usage of binary weights, the *VSM* make use of partial matching for assigning non-binary weights to the indexed terms of queries that are asked and stored in the documents. Here, weight is defined as *similarity degree of text* and *script* as stored in the query or system. The model considers matching

queries of text and sort them in the descending order according to its degree of similarity using various coefficient like Cosine, Jaccard, Euclidean Distance and Dice [123]. These coefficients (see Figure 3.1) are discussed below.



**Figure 3.1:** Similarity Measures Coefficients

- *Cosine*: It represents occurrence of words of an object as vector and it checks whether two objects are similar or not. Objects are considered to be similar, if their vectors are confused, and they are not similar, if their vectors create an angle whose cosine gives the similarity value. The cosine similarity ( $simCos(G, H)$ ) between two objects  $G$  and  $H$  is given as Eq. 3.1.

$$simCos(G, H) = Cos(G, H) = \frac{g \times h}{\|g\|^2 + \|h\|^2} \quad (3.1)$$

Here,  $g$  and  $h$  are vectors that are retrieved from  $G$  and  $H$ .  $\|g\| = \Omega_g m = 1$  which tells that vector normalizes  $G$ .

- *Dice*: Dice similarity ( $simDice(G, H)$ ) defines the multiplication for  $G$  and  $H$  by two and is given in Eq. 3.2.

$$simDice(G, H) = \frac{2 \times g \times h}{\|g\|_2^2 + \|h\|_2^2} \quad (3.2)$$

- *Euclidean Distance*: Euclidean Distance ( $dEuc$ ) is defined as straight line distance between two points  $g, h$  retrieved from two objects  $G, H$  in metric space and is given by Eq. 3.3.

$$dEuc = \| g - h \|^2 \quad (3.3)$$

Euclidean Similarity ( $simEuc(G, H)$ ) is the ratio of the Euclidean Distance increased by  $I$  and is given as Eq. 3.4.

$$simEuc(G, H) = \frac{I}{I + dEuc} \quad (3.4)$$

- *Jaccard*: Jaccard Similarity  $simJac(G, H)$  presents a statistic to compare similarities and dissimilarities between  $G, H$ . It is given by Eq. 3.5.

$$simJac(G, H) = \frac{g \times h}{\| g \|^2 + \| h \|^2 - (g \times h)} \quad (3.5)$$

The similarity coefficients help in forming disjoint sets of similar objects.

### 3.2.2 k-means Algorithm

Clustering algorithms partition data into a certain number of clusters. There is no universally agreed definition. Most researchers describe a cluster by considering the internal homogeneity and the external separation, *i.e.*, patterns in the same cluster should be similar to each other, while patterns in different clusters should not. Both the similarity and the dissimilarity should be examinable in a clear and meaningful way. The number of algorithms that exist are distance and similarity measures, vector quantization, graph theory based, *etc.* These algorithms face issues such as scalability leading to increased data volume, high dimensionality and computational complexities with respect to execution time and storage space.

As one possible solution, k-means is suggested that is effective in clustering large scale data sets. It is the being simplest algorithm that works for compact and hyper-spherical clusters. It is a frequently used technique for clustering data without any labels where data points are precisely allocated to predefined clusters according to some distortion measure. This algorithm aims at minimizing an objective function using squared error function.

k-means algorithm [120] is considered for partitioning the set of objects  $O$  into clusters  $c$

that minimizes the sum of squared error. Mathematically, it is formulated as [121]:

$$\text{Minimize: } Prob(Y, Z) = \sum_{b=1}^c \sum_{a=1}^m (y_{a,b} \times sed(O_a, Z_b)) \quad (3.6)$$

$$\text{subject to: } \begin{cases} \sum_{b=1}^c y_{a,b} = 1, & 1 \leq a \leq m \\ y_{a,b} \in \{0, 1\}, & 1 \leq a \leq m, 1 \leq b \leq c \end{cases} \quad (3.7)$$

where, *Prob* stands for the Problem, *Y* is a partition matrix  $m \times c$  and *Z* is a set of objects *m* in the same *O* domain and *sed* represents the Squared Euclidean Distance between two objects.

Problem *Prob* is solved iteratively considering  $\check{Y}$  as a partition matrix and  $\check{Z}$  as a set of objects *m* in the same *O* in the following steps:

(i) Fix ( $Z = \check{Z}$ ) and solve the reduced *Prob*( $Y, \check{Z}$ ). It is solved as:

$$Prob(Y, \check{Z}) = \begin{cases} y_{a,b} = 1 \text{ if } sed(O_a, Z_b) \leq sed(O_a, Z_h), & \text{for } 1 \leq h \leq c \\ y_{a,h} = 0, & h \neq b \end{cases} \quad (3.8)$$

(ii) Fix ( $Y = \check{Y}$ ) and solve the reduced *Prob*( $\check{Y}, Z$ ). It is solved as:

$$z_{b,k} = \frac{\sum_{a=1}^m y_{a,b} \times O_{a,k}}{\sum_{a=1}^m y_{a,b}} \quad (3.9)$$

for ( $1 \leq b \leq c$ ) and ( $1 \leq k \leq m$ ). Here, *z* is the element of *Z*.

The basic algorithm to solve *Prob* is as follows:

- (i) Choose an initial element  $Z_0$  and solve *Prob*( $Y, Z_0$ ) to obtain  $Y_0$ . Set  $h = 0$ .
- (ii) Assume  $\check{Y} = Y_h$  and solve *Prob*( $\check{Y}, Z$ ) to obtain  $Z_{h+1}$ . If  $Prob(\check{Y}, Z_h) = Prob(\check{Y}, Z_{h+1})$ , then output ( $\check{Y}, Z_h$ ), else, go to step 3.
- (iii) Assume  $\check{Z} = Z_{h+1}$  and solve *Prob*( $Y, \check{Z}$ ) to obtain  $Y_{h+1}$ . If  $Prob(Y_h, \check{Z}) = Prob(Y_{h+1}, \check{Z})$ , output  $Y_h, \check{Y}$  and stop; otherwise, assume  $h = h + 1$  and go to step 2.

The k-means, though, helps in selecting final clustered data points but are not widely used

due to its limitations like more time consumption, not removing empty clusters, not feasible in selecting data points and is highly effected with surroundings factors like noise, outliers *etc.* Therefore, the k-means is further classified as *hard* and *soft* clustering by combining it with fuzzy sets. Fuzzy Logic partitions the clusters on the basis of probability of 0 and 1 to get the best selection of clustered data points.

- (i) **Hard Clustering:** It clusters on the basis of probability of 0 and 1 to get the best selection of clustered data points. The suggested approach is *FKM* which is the combination of *k*-means and fuzzy. It reduces the size of the concept lattices and is computationally advantageous than the matrix decomposition technique, *i.e.*, Singular Value Decomposition (*SVD*). It performs both information retrieval and visualization [6].

*FKM* performs the partition step iteratively to generate new clusters until the process's convergence. It first calculates the means of clusters, *i.e.*,  $M_{FKM} = \{M_1, M_2, \dots, M_i\}$ , where, 'i' denotes total clusters. For the estimated means, *B*, it computes the degree of membership,  $\mu(z, i)$  for  $X_z$  in cluster *i*, *i.e.*,  $B(z, i) = \exp(- \|X_z - M_{FKM}\|^2)$ , one uses,

$$\mu(z, i) = \frac{B(z, i)}{\sum_z B(z, i)} \quad (3.10)$$

It afterwards replaces  $M_{FKM}$  with fuzzy means given as -

$$M_F = \frac{\sum_z \mu(z, i)^2 \times X_z}{\sum_z \mu(z, i)^2} \quad (3.11)$$

The process works until its convergence. The working is summarized in Algorithm 3.1.

---

**Algorithm 3.1** Fuzzy k-means Clustering Algorithm (*FKM*)

---

**begin**

Initialize  $M_{FKM} = \{M_1, M_2, \dots, M_i\}$

**for** (*i* = 1 to *k*) **do**

    compute  $B(z, i) = \exp(- \|X_z - M_{FKM}\|^2)$

    compute  $\mu(z, i) = \frac{B(z, i)}{\sum_z B(z, i)}$

    compute  $M_F = \frac{\sum_z \mu(z, i)^2 \times X_z}{\sum_z \mu(z, i)^2}$

**end for**

**end**

---

In the algorithm, initially, the means for all clusters, *i.e.*,  $M_{FKM}$  are computed. A loop begins for computing estimated means,  $B(z, i)$  and degree of membership,  $\mu(z, i)$ . The process replaces the  $M_{FKM}$  with new fuzzy means as  $M_F$ , that helps in optimizing the objective function and clusters the resources.

The limitations for this algorithm are convergence, slow process, difficulty in the selection of centroids and trouble in handling the data of wider range. The approach is useful in pruning the formal context for reducing attribute implication base. It has the limitations for manually suggesting the number of clusters to find the righteous data points and thus desired output is not received.

- (ii) **Soft Clustering:** It considers the membership matrix for selecting clusters using k-means and Fuzzy. The approach which combines these two methods is *FCM* that targets to find a partition for set of data points while minimizing the cost function. *FCM* assigns degree of membership ( $0$  and  $1$ ) to each resource of each cluster [166]. The value tells the probability of every object to different groups [124]. The *FCM* helps in reducing the clustered matrices of the objective functions by considering the projected centroids of clusters for better communication. It partitions set of  $z$  dimensional vectors,  $L = L_1, L_2, \dots, L_{E_d}$  in  $K_m$  clusters, where  $L_j = \{L_{1j}, L_{2j}, \dots, L_{zj}\}$  represents the  $j$ th sample for  $j = 1, 2, \dots, E_d$ . For the  $j$ th sample,  $L_j$  and the  $i$ th cluster center  $cc_i$ , here is a membership degree  $m_{d_{ji}}$  that indicates the degree of sample  $L_j$  belonging to the centroid cluster, *i.e.*,  $cc_i = cc_{1i}, cc_{2i}, \dots, cc_{zi}$  for  $i = 1, 2, \dots, k$ . This results into fuzzy partitioned matrix  $\hat{U} = (m_{d_{ji}})_{dlk*}$ . The *FCM* minimizes the objective function as

$$FCM = \sum_{l_j=1}^{E_d} \sum_{i=1}^k (m_{d_{ji}})^s \times E_{d_{ji}}^2 \quad (3.12)$$

where,  $E_{d_{ji}}$  is the Euclidean Distance of  $L_j$  to centroid cluster  $cc_i$  and is given by Eq. 3.13.

$$E_{d_{ji}} = \sqrt{\sum_{v=1}^{E_d} (cc_{vi} - l_{vj})^2} \quad (3.13)$$

and exponent  $s$  describes the fuzziness of clustering and is called as fuzzifier parameter.

The  $m_{d_{ji}}$  and  $cc_i$  are solved by Eqs. 3.12 and 3.13 as:

$$m_{d_{ji}} = \frac{1}{\sum_{v=1}^k \left( \frac{E_{d_{ji}}}{E_{d_{jv}}} \right)^{\frac{2}{s-1}}} \quad (3.14)$$

where, ( $s \neq 1$ ) and

$$cc_i = \frac{\sum_{j=1}^{E_d} m_{d_{ji}}^s \times L_j}{\sum_{j=1}^{E_d} m_{d_{ji}}^s}; \text{for}(i = 1, \dots, k) \quad (3.15)$$

The Eq. 3.12 with linear combination of vectors results in to better approximations.

The working of *FCM* is described in Algorithm 3.2. In this algorithm, the number of

---

**Algorithm 3.2** Fuzzy c-means Clustering Algorithm (*FCM*)

---

**begin**

Initialize  $k \leftarrow 3, s \leftarrow 1, \hat{U} \leftarrow \infty, \check{U} \leftarrow \infty$

**for** ( $i = 1$  to  $k$ ) **do**

**if** ( $\check{U} - \hat{U} \leq \text{threshold}$ ) **then**

**for** ( $j = 1$  to  $E_d$ ) **do**

      compute  $cc_i \leftarrow \frac{\sum_{j=1}^{E_d} m_{d_{ji}}^s \times L_j}{\sum_{j=1}^{E_d} m_{d_{ji}}^s}$

      Compute  $E_{d_{ji}} \leftarrow \sqrt{\sum_{v=1}^{E_d} (cc_{vi} - l_{vj})^2}$

      Compute  $m_{d_{ji}} \leftarrow \frac{1}{\sum_{v=1}^k \left( \frac{E_{d_{ji}}}{E_{d_{jv}}} \right)^{\frac{2}{s-1}}}$

**end for**

      update  $\check{U} \leftarrow \hat{U}$

**end if**

**end for**

**end**

---

clusters  $k$ , exponent  $s$  and fuzzy matrices  $\hat{U}$  and  $\check{U}$  are initialized to  $3, 1, \infty$ . The loop begins with comparison between  $\hat{U}$  and  $\check{U}$ . If their difference is less than threshold,

then centroid cluster  $cc_i$  is computed. For each  $cc_i$ , euclidean distance among clusters and partitioned vectors are calculated. It, finally, finds the membership degree which helps in minimizing the objective function and updates the fuzzy matrices. The process keeps on repeating unless the value of difference between  $\check{U}$  and  $\acute{U}$  is less than threshold.

The limitations for this algorithm are sensitivity to noise, difficulty in selection of appropriate parameters, easy to get stuck in the local minimal and fails to reach global minimal.

### **3.3 Proposed Iterative K-means Clustering Algorithm**

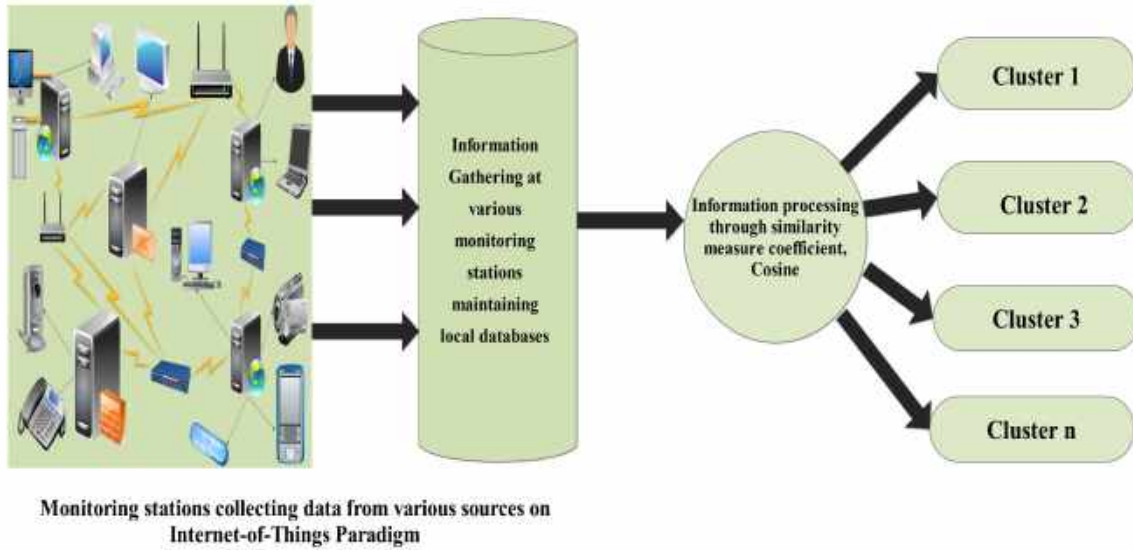
*IKm-CA* targets to cluster the meta-data that define an object identification, availability and computation capability for the resource discovery. The information is collected through gadgets, media, applications, and hardware connected to the *IoT* paradigm. It works in three steps, *namely*, Cluster formation, applying *IKm-CA* algorithm and Matching Conditions for Discovery. Each step is detailed below.

#### **3.3.1 Cluster Formation**

It targets to form the disjoint sets of similar objects as clusters using *VSM* described in Section 3.2.1. *VSM* helps to compute similarity through its similarity measures, *i.e.*, Jaccard, Euclidean Distance, Dice and Cosine.

#### **3.3.2 Iterative K-means Clustering Algorithm**

The formed disjoint sets of similar objects as clusters  $c_1, c_2, \dots, c_n$  (*see* Figure 3.2) undergoes *IKm-CA* process (*see* Algorithm 3.3).



**Figure 3.2: Clustering Methodology**

---

**Algorithm 3.3** Iterative K-means Clustering Algorithm (*IKm-CA*)

---

**Require:** (i) Clusters ( $c_1, c_2, \dots, c_n$ ) (ii) Array of vectors ( $V_a$ ) (iii) Centroid of clusters ( $cent_i[m]$ ) (iv) Associated cluster for array of vectors ( $A_c, A_{c_i}$ ) (v) similarity function ( $\text{sim}(A, B)$ ), (vi) Updated vector of clusters ( $best_n[m]$ ) (vii) Reassigned new clusters ( $L_{final}$ ) (viii) Final vector of cluster formed ( $total_{new}$ )

**begin**

*//Initialize clusters*

**for** ( $m = 1$  to  $c_n$ ) **do**

$V_a[n] = c_n$  *// initialize clusters as array vectors*

**end for**

*Initialize*  $n \leftarrow \text{rand}(\text{size}(V_a))$

**for** ( $n = 1$  to  $c_n$ ) **do**

*Calculate*  $cent_i[n] \leftarrow c_n[n]$  *// calculating centroids*

**end for**

**for** ( $m = 1$  to  $n$ ) **do**

**for** ( $k = 1$  to  $c_n$ ) **do**

$A_c[m] = \text{sim}(V_a[m], c_n[m])$  *// calculating distances between clusters and array vectors via similarity coefficient measures*

$A_{c1}[m] = \text{sim}(V_a[m], cent_i[m])$  *// calculating distances between centroids and array vectors via similarity coefficient measures*

**end for**

**end for**

---

---

```

for ( $m = 1$  to  $n$ ) do
  for ( $k = 1$  to  $c_n$ ) do
    if ( $(A_c[m] < c_n[m]) \&\& (A_{c_1}[m] < cent_i[m])$ ) then
       $best[m] = A_c[m]$ 
    else
       $best[m] = A_{c_1}[m]$ 
    end if
  end for
end for
// Recalculating cluster's location
for ( $m = 1$  to  $c_n$ ) do
  for ( $n = 1$  to  $V_a$ ) do
    Initialize mean  $\leftarrow 0$ , count  $\leftarrow 0$ 
    if ( $A_c[n] == m$ ) then
       $mean = mean + V_a[n]$ 
       $count = count + 1$ 
       $best_n[m] = \frac{mean}{count}$ 
    end if
  end for
end for
Output  $best_n[m]$ 
// Reassign array of vectors to clusters
for ( $m = 1$  to  $n$ ) do
  for ( $best = 1$  to  $m$ ) do
     $L = sim(best_n[m], [A_c[m] || A_{c_1}[m]])$  // comparing and reassigning clusters of array
vectors via similarity coefficient measures
     $L_{new} = A_c[m] - best_n[m]$ 
     $L_{new1} = A_{c_1}[m] - best_n[m]$ 
  end for
end for
 $L_{final} = merge(L, L_{new}, L_{new1})$ 
Output  $L_{final}$ 

```

---

---

*// Updating and finding number of clusters*

*Initialize total*  $\leftarrow 0$ , *threshold*  $\leftarrow 1$

**for** (*m* = 1 to  $c_n$ ) **do**

**if** ( $L_{final} < threshold$ ) **then**

$total_{new} = \max(L_{final}) - 1$

**else**

$total_{new} = \max(L_{final})$

**end if**

**end for**

**Output**  $total_{new}$

**end**

---

Here,  $c_n$  are initialized as array of vectors  $V_a[n]$ , where,  $n$  is the maximum number of clusters with  $V_a$ . For each cluster, its centroid  $cent_i[m]$  is calculated. The k-means clustering reforms new clusters taking into consideration array of vectors  $V_a$ , clusters  $c_n[m]$  and centroid array  $cent_i[m]$ , using similarity coefficients.

The loop is initialized again to select the best matched solution against the query with its top priority saved in sequence order in  $best[m]$ . The best value is selected between  $A_c[m]$  and  $A_{c1}[m]$  with condition that holds the maximum value from clusters of array  $c_n[m]$  or centroid array  $cent_i[m]$ . It recalculates the cluster's location in order to validate the process and the output is shown in  $best_n[m]$ , which is the average value of updated clusters. Further, the process then reassigns array of vectors to clusters by comparing the matched solution with either array of similarity measures of  $V_a$  and  $c_n[m]$ , *i.e.*,  $A_c[m]$  or array of similarity measures of  $V_a$  and  $cent_i[m]$ , *i.e.*,  $A_{c1}[m]$ . The process then updates the clusters with the difference of best matched array and save these updated clusters as  $L_{new}$ ,  $L_{new1}$ . The value of flag is incremented by 1 for each updated clusters. It merges the newly formed clusters and provides it as an output. Finally, it updates and finds the total number of clusters by comparing the updated clusters against threshold, providing the newly formed cluster as  $total_{new}$ .

### 3.3.3 Matching Conditions for Discovery

Here, the final clusters  $total_{new}$  with the arrival of query undergoes search using matching function (*see* Figure 3.3) against query parameters provided by the end user. The main target

is to check whether the relation exist for object or not by the means of matching function which should be less than the threshold value *Th* i.e.,  $F_m(I, J) \leq Th$ .

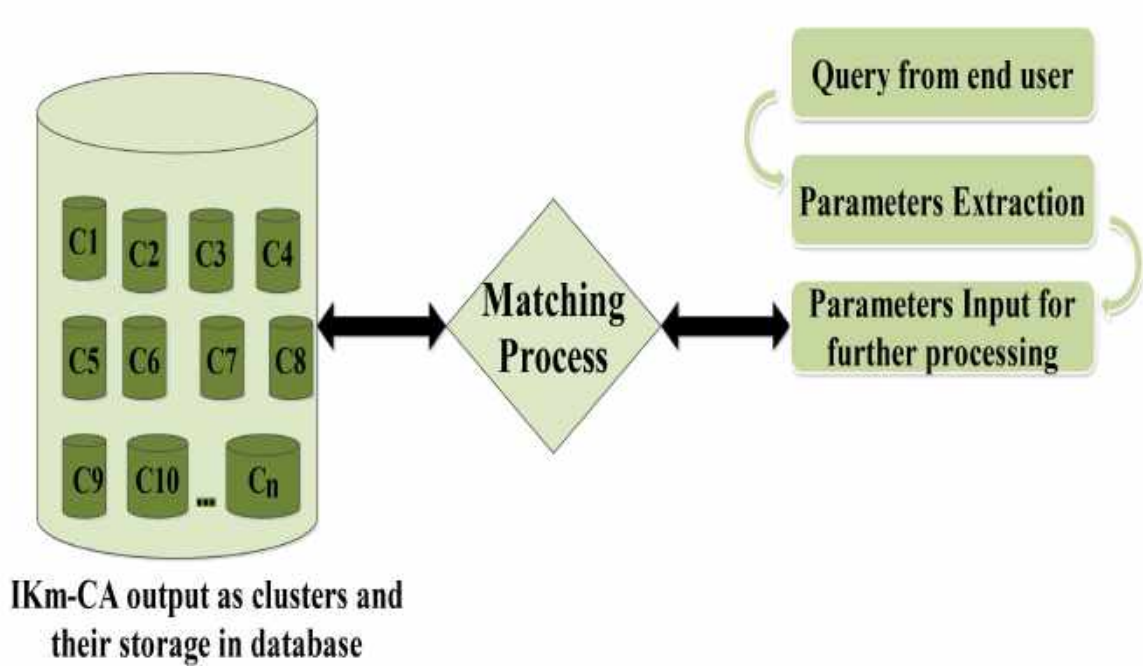
A query  $Q_a$  for discovery is given as

$$Q_a = S_K \times S_L \longrightarrow \mathbb{R} \quad (3.16)$$

and

$$(K, L) \longrightarrow Q_a(K, L) = \sum_{m=1}^P \vartheta_m \times q_a(\vec{K}, \vec{L}) \quad (3.17)$$

where,  $\vartheta_m, m \in \{1, \dots, P\}$  is the weights calculated among similar objects and  $q_a : \{0, 1\} \times \{0, 1\} \mapsto [0, 1]$  with the following matching conditions:



**Figure 3.3: Matching Conditions**

- $q_a(0,0) = 0$
- $q_a(0,1) = \eta_1 \in [0, 1]$
- $q_a(1,0) = \eta_2$  with  $\eta_2 \in [0, 1 - \eta_1]$
- $q_a(1,1) = 1$

Here,  $K \in S_K$  and  $L \in S_L$ . The  $Q_a(K, L)$  helps in finding the matching of query from asked set to array vector of similar clusters.

Considering  $Th \in \mathbb{R}$  and  $A_c[m]$ , there is a matching function  $F_m$  among them and is given by:

$$F_m = S_K \times S_L \longrightarrow \{0, 1\} \quad (3.18)$$

$$(K, L) \longrightarrow F_m(K, L) = 1 \Leftrightarrow (A_c[m])(K, L) \leq Th \quad (3.19)$$

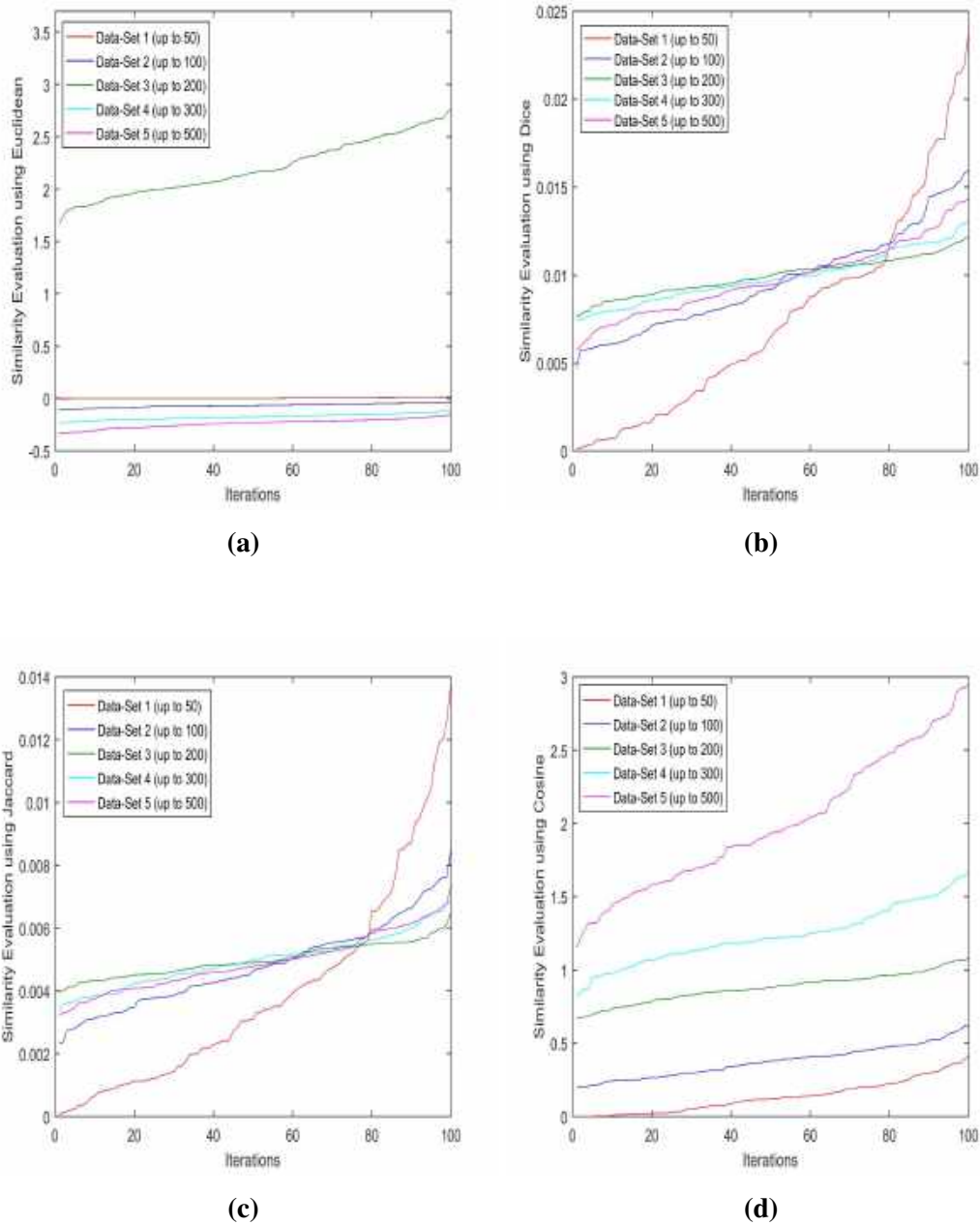
The  $F_m$  is the required possible solution of searched query  $Q_a$ . Using bounded intervals of  $\eta_1$  and  $\eta_2$  in a way that they are not empty and are included in subsets of  $S_K$  and  $S_L$  describing the two conditions:

- (i) If  $K$  matches  $L$ , *i.e.*,  $\vec{K} \subset \vec{L}$  then  $\forall m \in \{1, 2, \dots, P\}$ ,  $q_a(\vec{K}, \vec{L}) \leq \eta_1$  and  $F_m(K, L) = 1$ .
- (ii) If  $K$  does not match with  $L$ , *i.e.*,  $\vec{K} \not\subset \vec{L}$ ; this means that there is at least a single  $Q_a(K)$  with respect to asked text but having no similar object in array of clusters. Thus,  $q_a(\vec{K}, \vec{L}) \geq \eta_2$  and  $F_m(K, L) = 1$ .

## 3.4 Experimental Evaluations

### 3.4.1 Similarity Computation

The various similarity measure coefficients are discussed in subsection 3.2.1 for *VSM*. They are simulated and implemented for random data sets with 50, 100, 200, 300 and 500 nodes. The graph generated with the data sets are shown in Figure 3.4. The Euclidean Distance graph (*see* Figure 3.4 (a)) depicts the similarity evaluation with increasing number of nodes. It is observed that if the nodes are limited, *i.e.*, maximum up to 300, it will not provide any relevant searches but if the resources are more than 500, then it provides better resource searches in less time. The Dice coefficient (*see* Figure 3.4(b)) depicts that the number of searches are decreased with increasing number of nodes and is showing better results of search space with less number of nodes present in the set. In comparison with Euclidean Distance, the number of search space is increased with 20%. As observed from Figure 3.4 (c), the Jaccard similarity comes out to be almost similar with Dice coefficient and is providing best search with less number of nodes. It remains constant and is increasing consistently with increasing number of nodes whereas it provide search space rapidly with lesser nodes.



**Figure 3.4:** Similarity Computation using (a) Euclidean Distance, (b) Dice, (c) Jaccard, (d) Cosine

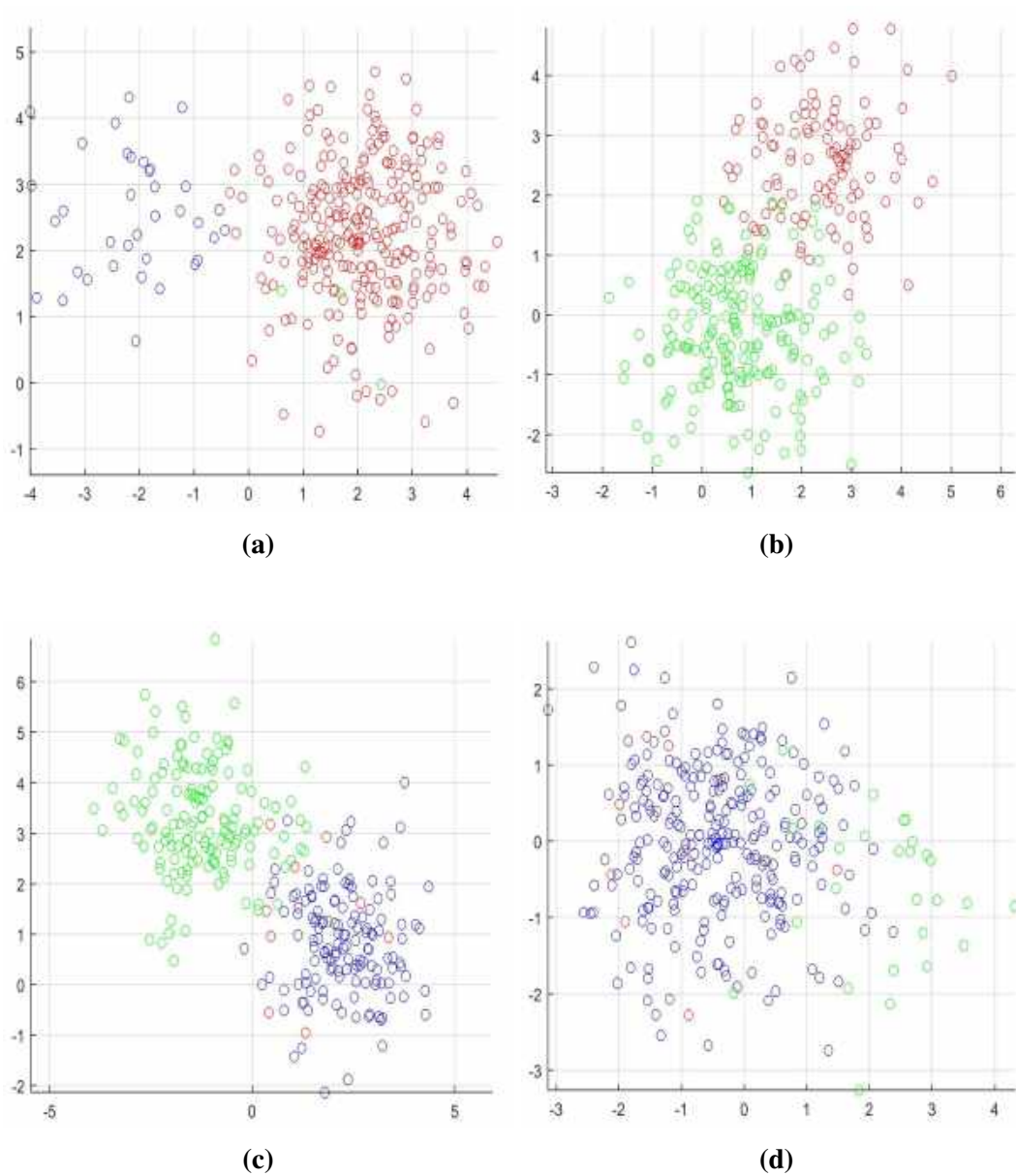
In comparison, it provides 25% increase search space than Dice and 45% than Euclidean Distance. The Cosine graph (*see* Figure 3.4(d)) shows efficient search with increased number of nodes. It has best match as compared with other similarity coefficients.

In comparison, the search space increases as 33%, 71% and 89% of Jaccard, Dice and Euclidean Distance, respectively. Among these similarity search approaches, it is observed

that cosine provides the best matches from the search space set with an increase in the number of nodes.

### 3.4.2 Clustering Analysis

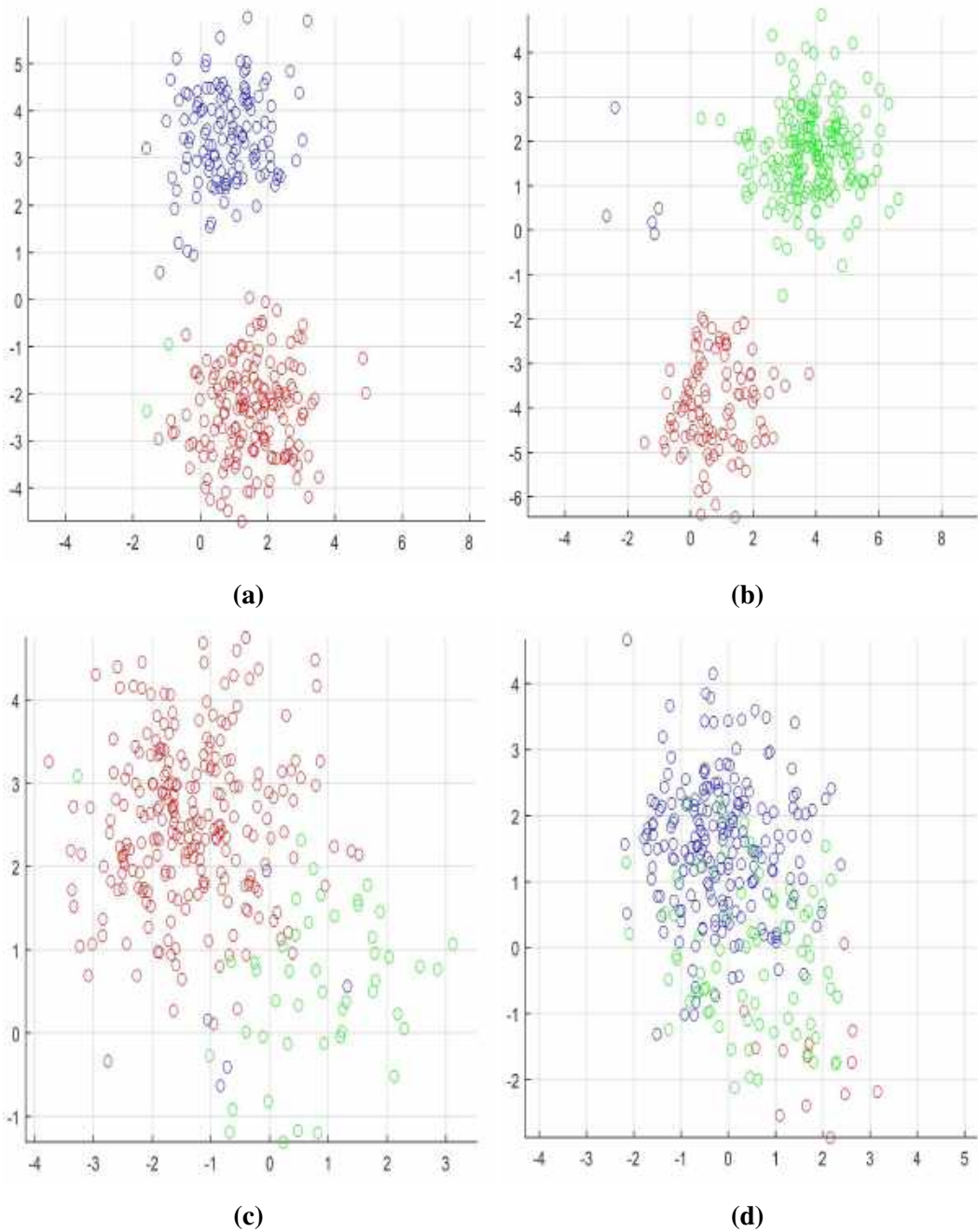
- (i) *FKM*: The *FKM* is analyzed with similarity coefficients like Jaccard, Dice, Euclidean Distance and Cosine; against the input query,  $Q$ . The formed clusters are of size 3 with its random populations of nodes as 300.



**Figure 3.5:** Fuzzy k-means Clustering using (a) Euclidean Distance, (b) Dice, (c) Jaccard, (d) Cosine

The clusters in the results are distributed normally with distance fuzzification of objects. The formed clusters are shown in Figure 3.5. From Figure 3.5(a) it is depicted that the Euclidean Distance approach helps in generating clusters of similar objects with distances measured among centroids by minimizing the objective function. It takes the geometric mean of a prototype matrix belonging to the cluster. Euclidean Distance considers the high density regions for making clusters. Figure 3.5(b) is produced by Dice coefficient similarity measurement with *FKM* providing the results by clustering the neighboring object mean-distances. It helps in finding the objects in range. It considers only the objects whose fuzzy distance is minimum. Figure 3.5(c) shows that the *FKM* in combination of Jaccard approach makes clusters of dispersed data present only in small area. It normalizes the searched objects and clusters them. Figure 3.5(d) shows the clusters formed with cosine coefficient in wide area taking the maximum angle between two similar objects. It clusters the dispersed data at the end of the process. From the discussion, it is concluded that efficient search is performed by Euclidean coefficient in collaboration with *FKM*, *i.e.*, more search spaces are provided in less time.

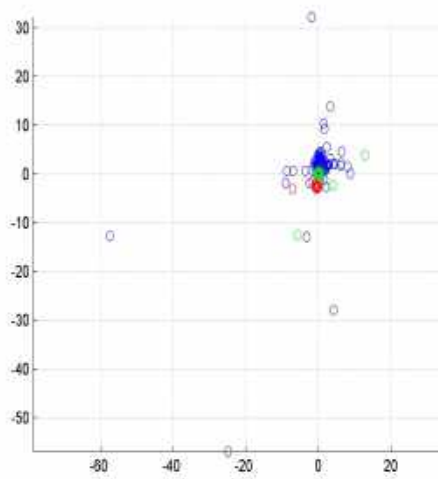
- (ii) *FCM*: The similarity coefficients like Jaccard, Dice, Euclidean Distance and Cosine are simulated with *FCM* against the input query,  $Q$ . it is processed for 300 nodes with formed clusters as 3. The results are shown in Figure 3.6, distributed in  $2D$  area. Figure 3.6(a) shows the clusters having minimum distances with the centroids. Thus, it provides the clusters by taking geometric mean between nodes and centroid. Figure 3.6(b) suggests that Dice coefficient provides clusters by retaining maximum nodes in the same cluster. It takes in to account the coverage of maximum area. Figure 3.6(c) is produced by Jaccard similarity that considers the centroid to be selected near to each other and selecting the nodes for clusters by finding suited match against the query  $Q$ . It is observed from Figure 3.6(d) that cosine generates clusters by considering the angular distances between nodes and centroid. It clusters maximum data into one cluster. Thus, it is concluded that the search efficiency by *FCM* is produced with *Jaccard* approach, providing maximum coverage of search area. It uses the fuzzy coefficient to measure the distance.



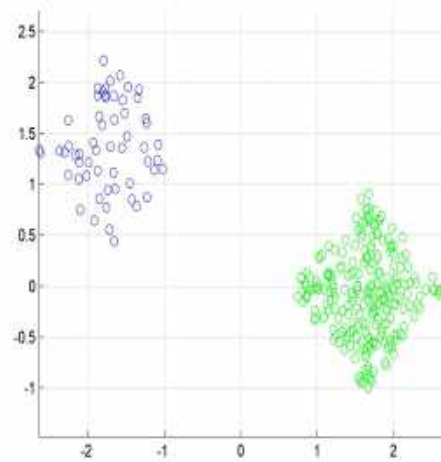
**Figure 3.6:** FCM Clustering using (a) Euclidean Distance, (b) Dice, (c) Jaccard, (d) Cosine

(iii) *IKm-CA*: The proposed *IKm-CA* (see section 3.3), is simulated using *MATLAB* environment. According to the matching criteria of similarity against the input query  $Q$ , the similarities among clusters are analyzed on the basis of Jaccard, Dice, Euclidean Distance and Cosine. The inputs to the *IKm-CA* are 2-D area as dimensions, number of clusters are 3 and number of random objects are 300. The process uses distribution

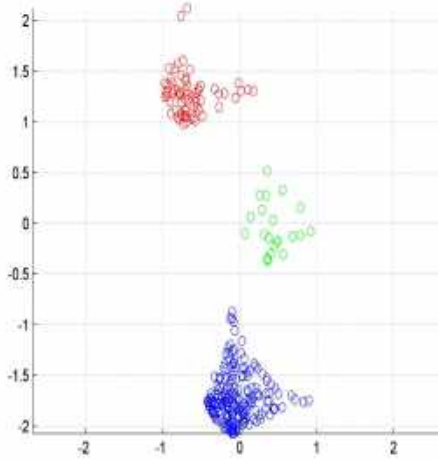
function with common variances, generates matrix for data and response variable by finding the point distance among them. The results are distributed in 2-D area and clusters are formed with minimum mean of objects. The clusters thus formed are shown in Figure 3.7. The graph in Figure 3.7(a) shows that Euclidean coefficient is helpful among the similar objects with minimum distance's mean. It calculates the averaging distances for generating clusters among similar objects. Thus, it is helpful for those objects which are at least once in relation, with one of the other objects.



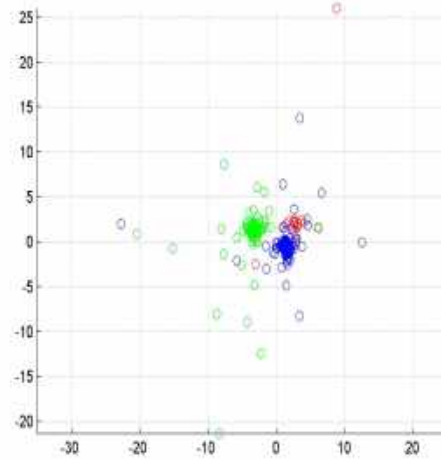
(a)



(b)



(c)



(d)

**Figure 3.7:** *IKm-CA* Clustering using (a) Euclidean Distance, (b) Dice, (c) Jaccard, (d) Cosine

The graph in Figure 3.7(b) is produced by Dice coefficient which compresses the area of the search. It tells that its search efficiency is compatible with less number of objects or smaller area. The Jaccard similarity performs similar like Dice and from graph in Figure 3.7(c), it is found that Jaccard normalizes the search clusters with the best suited matched query  $Q$ . The produced results are dispersed in whole area. This shows its efficiency in large sets of data or covering large area. From the graph in Figure 3.7(d), it is observed that formation of clusters using Cosine coefficient is done by finding the minimum angle among the vectors of similar objects, hence, creates clusters accordingly. The cosine similarity is benefited in small as well as for larger area. Thus, it comes out to be best similarity with less time consumption. From the obtained results, it is concluded that cosine coefficient with *IKm-CA* provides better search results in comparison with other coefficients like *Euclidean Distance*, *Jaccard* and *Dice*. It takes less time to perform searching that results in to maximum search coverage.

### 3.4.3 System Performance

The clustering algorithms, *i.e.*, *FKM*, *FCM* and *IKm-CA* are evaluated using open source simulator “*Clust 3.0*”, considering two evaluation metrics, *namely*, similarity search and search time [126]. The former results for maximum matching (in %) whereas latter provides the time (in seconds), against query, respectively. For the purpose, input to the simulator are number of characters, blank spaces, total iterations and number of partitions into which query is divided. The values calculated are shown in Table 3.1.

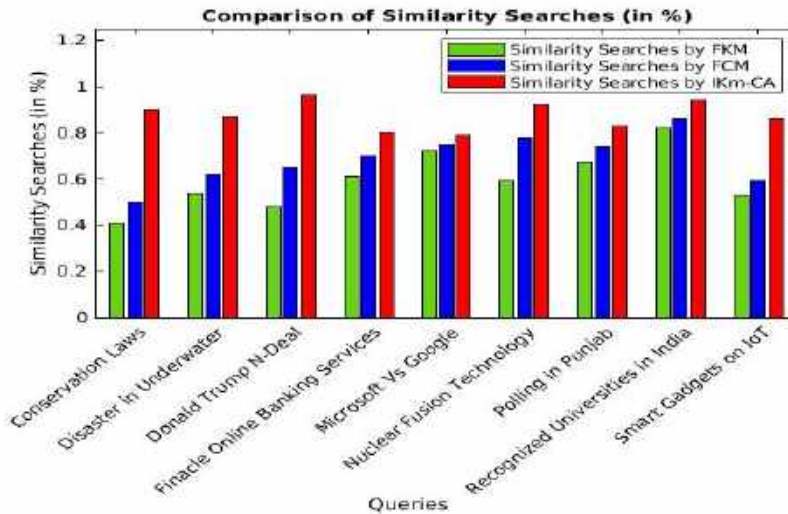
As observed from Table, various queries with its input threshold as 5 are iterated for 100 times. The queries using simulator provides the similarity measures (in %) for *FKM*, *FCM* and *IKm-CA*. It also processes the queries with its search efficiencies.

**Table 3.1:** Experiment for Queries at Threshold = 5

<b>Queries</b>	<b>Number of Partitions</b>	<b>Threshold</b>	<b>Number of Iterations</b>	<b>Similarity Searches by FKM (in %)</b>	<b>Search Time by FKM (in seconds)</b>	<b>Similarity Searches by FCM (in %)</b>	<b>Search Time by FCM (in seconds)</b>	<b>Similarity Searches by IKm-CA (in %)</b>	<b>Search Time by IKm-Ca (in seconds)</b>
<b>Conservation Laws</b>	2	5	100	0.41	0.35	0.5	0.27	0.9	0.12
<b>Disaster in Underwater</b>	2	5	100	0.54	0.5	0.62	0.31	0.87	0.23
<b>Donald Trump N-Deal</b>	3	5	100	0.48	0.24	0.65	0.24	0.96	0.07
<b>Finacle Online Banking Services</b>	3	5	100	0.61	0.29	0.7	0.15	0.8	0.1
<b>Microsoft Vs Google</b>	3	5	100	0.72	0.31	0.75	0.32	0.79	0.14
<b>Nuclear Fusion Technology</b>	2	5	100	0.59	0.18	0.78	0.2	0.92	0.05
<b>Polling in Punjab</b>	3	5	100	0.67	0.25	0.74	0.17	0.83	0.08
<b>Recognized Universities in India</b>	2	5	100	0.82	0.68	0.86	0.32	0.94	0.27
<b>Smart Gadgets on IoT</b>	3	5	100	0.53	0.43	0.59	0.22	0.86	0.16

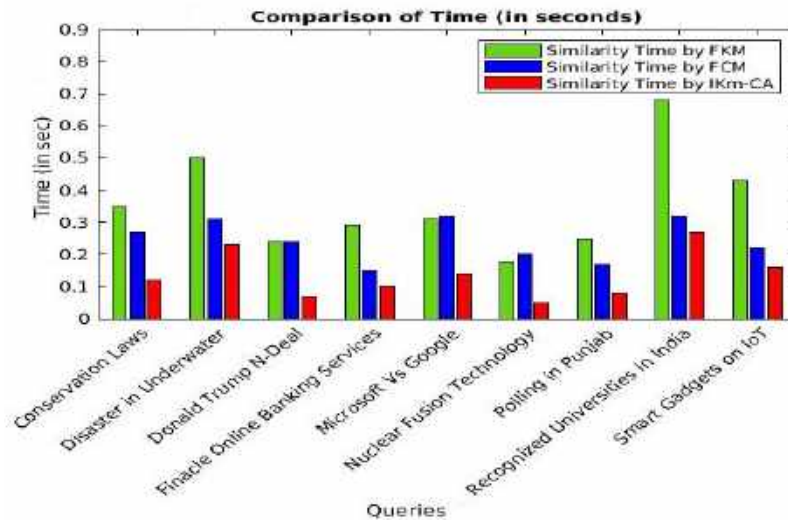
The similarity measures and search time of *FKM*, *FCM* and *IKm-CA* is shown in Figure 3.8 and Figure 3.9.

- (i) Similarity Searches: From the graph, it is suggested that *IKm-CA* outperforms well in similarity searches with respect to *FKM*, *FCM* and is providing 1.5 times and 1.3 times better similarity searches.



**Figure 3.8:** Similarity Searches

It is possible due to angular distance based similarity computation among centroids of clusters rather than considering membership matrix of similarity. The *IKm-CA* provides shortlisted resources with sufficient discovered results.



**Figure 3.9:** Searching Time

- (ii) Search Time: The *IKm-CA* has faster processing time with respect to requirements of the system, based on which it is implemented to test its feasibility. On the other hand, *FKM* and *FCM* takes more computational time and provides less scalable results. It depicts that *IKm-CA* gain a search efficiency by 1.63 times and 1.45 times better than *FKM* and *FCM*.

### 3.4.4 Complexity Analysis

- (i) Fuzzy k-means: It uses degree of membership matrix to arrange data sets of centroids in clusters. It chooses  $v_l$  centroids and assigns membership matrix  $F$  to centroid. It identifies  $m$  samples with  $r$  attributes in data sets. The *FKM* performance is checked on the basis of its computation time in terms of time and space complexity, described as under.

- (a) Time Complexity: the number of operations are given by

$$Operations = M \times N + dist + cent_{cal} = O(M \times N \times K \times L) = O(N_z \times K \times L) \quad (3.20)$$

where,  $M \times N$  are the total number of resources,  $dist$  is the distance among the centroid of membership matrix and  $cent_{cal}$  is the total iterations of centroids.

- (b) Space Complexity: All the iterations and membership matrix are stored in modest space, therefore it requires only  $O(N_z + K)$  storage. Here,  $N_z$  is number of resources and  $K$  is the distance among attributes of centroids.

The *FKM* is measured with similarity coefficients their complexity on their basis are discussed as under:

- (a) Euclidean Distance: It considers distance among two centroid's membership matrix. Its complexity is given by:

$$Complexity = \frac{O(N_z)}{O(N_z \times K \times L) + 1} = O(N_z) \quad (3.21)$$

- (b) Dice: It considers common objects of membership matrix of centroids with the

complexity of sender and receiver as  $O(N_Z)$ . Its complexity is

$$Complexity = \frac{2 \times O(N_Z) \times O(N_Z)}{\|O(N_Z)\|^2 + \|O(N_Z)\|^2} = O(N_Z^2) \quad (3.22)$$

- (c) Jaccard: It considers pairwise similarity among prototype matrix and membership matrix for all neighboring resources. Its complexity is given by

$$Complexity = \frac{O(N_Z) \times O(N_Z)}{\|O(N_Z)\|^2 + \|O(N_Z)\|^2 - (O(N_Z) \times O(N_Z))} = O(N_Z^2) + 2 \times N_Z \quad (3.23)$$

- (d) Cosine: It takes angular distances among membership matrix of centroids of clusters. Its complexity is given by

$$Complexity = \frac{O(N_Z) \times O(N_Z)}{\|O(N_Z)\|^2 + \|O(N_Z)\|^2} = O(N_Z^2) \quad (3.24)$$

It is concluded that Euclidean Distance in combination with *FKM* provides solution with minimum complexity, *i.e.*, it takes minimum computation time and storage.

- (ii) Fuzzy c-means: It is iterative in nature. It stops when maximum iterations are processed. The query  $Q$  data is grouped in to  $C$ -clusters with  $N$  data points in each dimension  $D$ . These data points are linked to every other cluster and is processed for new cluster formation with  $I$  iterations. As the clusters are updated instantaneously and therefore, time complexity for *FCM* is given by:

- (a) Time Complexity: It is dependent on the number of iterations and cluster formation. It is given by:

$$\begin{aligned} T_c &= \text{number of clusters} \times \text{iterations} \times \text{dimension} \times \text{datapoints} \\ &= C^2 \times N \times D \times I = O(C^2 N D I) \end{aligned} \quad (3.25)$$

where,  $C, N, D, I$  are the number of clusters, data points, dimension and iterations, respectively.

- (b) Space Complexity: The *FCM* requires  $O(N \times D + N \times C)$  space to store the

centroids, clusters in the modest area. Here,  $N$  are the data points that are distributed among  $c$ -clusters in  $D$ -dimension. It uses *VSM* similarity coefficients like Dice, Cosine, Jaccard and Euclidean Distance for making clusters. Therefore, the complexity against similarity coefficients are described as under:

- i. *Euclidean Distance*: Its complexity is defined by number of iterations for selecting centroids in each cluster. It is denoted as:

$$Complexity = \frac{O(C)}{O(N \times d \times C^2 \times I) + 1} = \frac{O(C)}{1 + O(C^2)} = O(C) \quad (3.26)$$

- ii. *Dice*: It takes in to account the equal complexity of  $O(N)$  at sender and receiver after each iteration of common objects. Therefore,

$$Complexity = \frac{2 \times O(C) \times O(C)}{\|O(C)\|^2 + \|O(C)\|^2} = O(C^2) \quad (3.27)$$

- iii. *Jaccard*: It considers the similar objects for its neighboring clusters by finding distances among clusters. Its complexity is given by:

$$Complexity = \frac{O(C) \times O(C)}{\|O(C)\|^2 + \|O(C)\|^2 - (O(C) \times O(C))} = O(N^2) + 2 \times C \quad (3.28)$$

- iv. *Cosine*: It describes the angular distances among the centroids of clusters and is denoted as:

$$Complexity = \frac{O(C) \times O(C)}{\|O(C)\|^2 + \|O(C)\|^2} = O(C^2) \quad (3.29)$$

It is deduced that Euclidean Distance in combination with *FCM* results in to low complexity, *i.e.*, minimum time and space is required with Euclidean Distance similarity measure.

- (iii) *IKm-CA*: It uses proximity function to separate data-sets arranged in various clusters and considers centroid calculation for rearranging them. For the purpose, it chooses  $N_{pt}$  points as initial centroids and assigns each point to cluster closest to centroid. For

analyzing algorithm's complexity, it identifies operations performed at each step and keeps information for every iteration. For  $N_{nd}$  nodes with  $N_{att}$  attributes in each data set, the  $IKm - CA$  is analyzed on its performance. The performance is measured with respect to its computation time, *i.e.*, complexity, as it measures the time and storage evaluation on the basis of algorithm's working. The complexity is categorized into time and space complexity. Therefore, the complexity of  $IKm - CA$  in terms of time and space are described as under.

- (a) Time Complexity: It measures the total time computation to form clusters iteratively. Therefore, its number of operations are given by Eq. 3.30.

$$\begin{aligned} Operations = dist_{cal} + comp + cent_{cal} = [N_{pt} \times N_{nd} \times N_{att}] \\ + [(N_{pt} - 1) \times N_{nd} \times N_{att}] + [N_{pt} \times ((N_{nd} - 1) + 1) \times N_{att}] \end{aligned} \quad (3.30)$$

where,  $dist_{cal}$  is distance calculated among clusters,  $comp$  is total number of comparison and  $cent_{cal}$  is calculated centroids of clusters.  $IKm - CA$  performs after  $J_e$  iteration that results into Eq. 3.31.

$$\begin{aligned} Operations = [J_e \times N_{pt} \times N_{nd} \times N_{att}] + [J_e \times (N_{pt} - 1) \times N_{nd} \times N_{att}] \\ + [J_e \times N_{pt} \times ((N_{nd} - 1) + 1) \times N_{att}] = O(J_e \times N_{pt} \times N_{nd} \times N_{att}) \end{aligned} \quad (3.31)$$

Here, it has been assumed that same amount of time is consumed by each operation. Since,  $N_{pt} \ll N_{nd}$  and  $N_{att} \ll N_{nd}$ , k-means outcasts as linear graph with respect to total nodes  $N_{nd}$  and the number of points  $N_{pt}$ .

- (b) Space-Complexity: Since, data points and centroids need to be stored, modest space is required for  $IKm - CA$ . In particular, it needs  $O((N_{nd} + N_{pt}) \times N_{att})$  storage, where  $N_{nd}$  is total nodes and  $N_{att}$  is their attributes.

Since,  $IKm - CA$  uses similarity coefficients like Euclidean Distance, Dice, Jaccard and Cosine for cluster formation, it is required to calculate complexity on their basis. It is discussed below.

- (a) Euclidean Distance: Its complexity considers the distance estimation among two resources with partial change in their positions. Therefore its complexity using

Eq. 3.5 is given in Eq. 3.32.

$$Complexity = \frac{O(N_w)}{O(J_e \times N_{pt} \times N_{nd} \times N_{att}) + 1} = \frac{O(N_w)}{1 + O(N_w)} = O(N_w) \quad (3.32)$$

Here,  $N_{pt}$  and  $N_{att}$  are assumed to be 1,  $J_e$  be the number of iterations and  $\tilde{N}$  as the total iterations. Therefore, it implies the complexity to be  $O(N_w)$ .

- (b) Dice: Its complexity measurement takes in to consideration common objects  $N_{nd}$  of resources with complexities of initiator and receptor as  $O(N_w)$  then total complexity is given in Eq. 3.33.

$$Complexity = \frac{2 \times O(N_w) \times O(N_w)}{\|O(N_w)\|^2 + \|O(N_w)\|^2} = O(N_w^2) \quad (3.33)$$

where  $N_w$  is the total comparisons among resource's distances.

- (c) Jaccard: It selects the common number from the resources which follows one to one communication and is further divided by the number of neighboring resource's distances. Moreover, it calculates pairwise similarities for all neighboring resources. The output complexity with  $IKm - CA$  is given by Eq. 3.34.

$$Complexity = \frac{O(N_w) \times O(N_w)}{\|O(N_w)\|^2 + \|O(N_w)\|^2 - (O(N_w) \times O(N_w))} = \frac{O(N_w^2)}{\|O(N_w)\|^2 - O(N_w)^2} = O(N_w^2) \quad (3.34)$$

- (d) Cosine: It is the angular distance among two resources and is considered to be fast coefficient for cluster formation. Its complexity is given by Eq. 3.35.

$$Complexity = \frac{O(N_w) \times O(N_w)}{\|O(N_w)\|^2 + \|O(N_w)\|^2} = \frac{O(N_w^2)}{\|O(N_w)\|^2} = O(N_w) \quad (3.35)$$

where  $N_w$  is angular centroid distances among resources.

From the above discussion, it is concluded that cosine coefficient in collaboration

with *IKm-CA* provides the best solution for clustering and is proved to be an efficient method on the basis of minimum complexity.

### 3.5 Conclusion

The proposed technique, *IKm-CA*, facilitates resource discovery on the *IoT* through clustering using matching criteria and addresses heterogeneity in data. Clustering provides self-organization of resources and group them by creating homogeneous descriptive data. The clustered metadata provides similarity measures through *VSM*, performs knowledge representation and mitigates the problem of selecting erroneous or empty data points to clusters. *IKm-CA* is simulated using *MATLAB* against exponentially growing nodes to analyze the search efficiency. *IKm-CA* is compared with existing methods, *i.e.*, *FKM* and *FCM*, for providing clusters with respect to various inputted number of queries. These are measured on the basis of similarity searches and search time. It is observed that *IKm-CA* provides 1.5 times, 1.3 times better similarity searches and 1.63 times, 1.45 times better search time in comparison with *FKM* and *FCM*, respectively. The results support algorithm's efficiency in collaboration with Cosine similarity measure having minimum complexity of  $O(N)$ .



# Context-Aware Resource Discovery and Selection using Convex Optimization

---

## 4.1 Introduction

The previous chapter has discussed proposed technique, *namely*, *IKm-CA* that eliminates the heterogeneity in data for resource discovery and selection on *IoT*. But, it has not accounted for issues such as context of the gathered data, scalability and search metrics in the *IoT*. These issues, if not resolved, would lead resource discovery and selection to emerge as a non-linear constrained specific problems. Such problems experience higher search complexity in terms of time. Therefore, it is required to optimize them to ease the inter-communication and interaction<sup>1</sup>. To address these issues, the chapter has proposed a Context-Aware Search Optimization Framework on Internet-of-Things which targets knowledge presentation through schema, discovery *via* multi-modal search algorithm and its optimization through an Iterative Gradient Descent algorithm. The multi-modal search algorithm through keywords, value or spatial-temporal indices perform resource discovery by finding the suited matches as search set from search-space. The search set is further evaluated *via* Iterative Gradient Descent Algorithm for optimization through the usage of iterative and convergence properties of Gradient Descent. It is tested using various objective

---

<sup>1</sup>The contents are published as M. Bharti, R. Kumar, S. Saxena, “Context-Aware Search Optimization Framework on Internet-of-Things”, *Concurrency and Computation: Practice and Experience*, Vol. 30, no. 14, pp. e4426, 2018 (Wiley). SCI Indexed (IF: 1.167)

functions and is compared with *Newton* and *Quasi-Newton* methods. The obtained results depict the efficiency of the algorithm graphically with reference to the searching time and validate the system performance.

## **4.2 Context Awareness and Internet-of-Things**

Any processed data that helps to characterize the circumstance of an entity is termed as context. Here, the entity could be a person, place, or object that is related to communication and interactions among end users. The system is considered to be context-aware if uses context data to provide services to the end users on the basis of relevancy with respect to user's task. Context Awareness as fundamental property for both ubiquitous and pervasive computing systems permits context storage to facilitate interpretation easily and more meaningfully. Moreover, it performs *M2M* communication and supports actions such as data collection, representation, delivery, and reaction through three main context model, *namely*, No application-level, Implicit, and Explicit. In No application-level context model, applications perform all the actions within its boundaries. In Implicit context model, applications use libraries, frameworks, and toolkits to execute the actions and provide a standard design that makes it easier to build the applications quickly. However, still the context is hard bound to the applications. In Explicit context model, applications use a context management infrastructure or middleware solution and therefore, lie outside the application boundaries. Using these context models, Context-Aware Life Cycle is defined that works through four steps, *i.e.*, to collect, to model, to reason and to disseminate the context. Till date, the real time implementation for Context-Aware Life Cycle could be possible through various application areas such as Intelligent Transportation System (*ITS*), Healthcare, Logistics, *etc.*

## **4.3 Non-Linear Programming**

Nonlinear programming solves an optimization problem which is defined as set of constraints that includes equalities and inequalities against the set of unknown real variables having an

objective function. The function is to be minimized or maximized on the basis of constraints or functions which are non-linear. On *IoT*, with the exponential growth of the resources, there will be increase in the number of fetched resources against query's parameters and stored as search set,  $z$ . Considering  $s(z)$  as objective function that varies over  $R^n$  with respect to functional constrained like access type, data format, *etc.*, *i.e.*,  $s(z)$  turns out to be a non-linear constrained specific problem of type

$$\begin{aligned}
 & \text{Minimize } s(z) \\
 & \text{such that } k_a(z) \leq 0, a = 1, \dots, r \\
 & \quad \quad \quad l_b(z) = 0, b = 1, \dots, p \\
 & \quad \quad \quad m_c(z) = 0, c = 1, \dots, q
 \end{aligned} \tag{4.1}$$

where,  $k_a(z)$ ,  $l_b(z)$ ,  $m_c(z)$  are the functional constraints. Here,  $s(z)$  require to be minimized to reduce the search complexity in terms of time. In order to deal with these problems, non-linear programming as the sub-field of mathematical optimization provides number of methods such as Gradient Descent, Newton and Quasi-Newton. Each of them is discussed below.

### 4.3.1 Gradient Descent Method

Gradient descent or steepest descent is a first-order iterative optimization algorithm that intends to minimize the objective function *via* taking steps in descent direction from the current position. Formally, let objective function,  $OF(m)$ , is defined and differentiable in a neighborhood of a point,  $b$ , then  $OF(m)$  starts decreasing in a faster manner if one goes from  $b$  in the direction of negative gradient of  $OF$  at  $b$ ,  $-\nabla OF(b)$ . It follows that, if

$$b_{n+1} = b_n - \gamma \times \nabla OF(b_n) \tag{4.2}$$

where,  $\gamma$  defines step size and is very small. Thus,  $OF(b_n) \geq OF(b_{n+1})$ . With this observation, one starts with a guess  $m_0$  for a local minimum of  $F$  and considers the sequence  $m_0, m_1, m_2, \dots, m_n$  such that

$$m_{n+1} = m_n - \gamma_n \times \nabla OF(m_n), n \geq 0 \quad (4.3)$$

and

$$OF(m_0) \geq OF(m_1) \geq OF(m_2) \geq \dots \geq OF(m_n) \quad (4.4)$$

The sequence,  $m_n$ , converges to the desired local minima.

### 4.3.2 Newton Method

The Newton Method uses first and second order derivatives for finding successively better approximations to the roots of a real-valued function. It helps to minimize the non-linear unconstrained problem in to linear constrained [152]. It considers  $s(z)$  to be differentiable up to second order (*see* Eq. 4.1). Considering the iteration for  $s(z)$ , gradient  $\nabla s(z)$  and Hessian  $\nabla^2 s(z)$  for the approximation of Taylor Expansion,  $s(q)$ , and Eq. 4.5 is given as:

$$s(q) \approx s(z) + (q - z)^T \times \nabla s(z) + \frac{1}{2} \times (q - z)^T [\nabla^2 s(z)] \times (q - z) \quad (4.5)$$

where,  $T$  is transpose,  $q$  is the minimizer of  $s(z)$  on  $R^n$ .

For minimization,  $\nabla s(z)$  is set to zero. It implies,

$$[\nabla^2 s(z)] \times (q - z) = -\nabla s(z) \quad (4.6)$$

Assuming,  $\nabla^2 s(z)$  to be non-zero, Eq. 4.6 becomes,

$$q = z - [\nabla^2 s(z)]^{-1} \times \nabla s(z) \quad (4.7)$$

For any starting point, say,  $z_0$ , the relation would be

$$z_t = z_{t-1} - [\nabla^2 s(z_{t-1})]^{-1} \nabla s(z_{t-1}) \quad (4.8)$$

where,  $z_t$  and  $z_{t-1}$  are the singular values at points  $t$  and  $t - 1$ , respectively. This method is not efficient if  $\nabla^2 s(z_{t-1}) > 0$ . Therefore, it is considered to be minimum and positive definite

which converges to all gradient  $z^*$  quadratically. The method selects positive step size,  $z_{t_r}$ , as:

$$z_{t_r+1} = z_{t_r} + l_s(z_{t_r}) \quad (4.9)$$

where, Newton direction,  $l_s$ , is given as:

$$l_s = [-\nabla^2 s(z)]^{-1} \nabla s(z) \quad (4.10)$$

The direction converges globally in the method but not the step size. Therefore, step size is selected using *Armijo* based line [148] in Newton (with its iteration  $t_r$  and step size  $\delta_{t_r}$ ) as:

$$\begin{aligned} z_{t_r+1} &= z_{t_r} + \delta_{t_r+1} \times l_s(z_{t_r}), \\ l_s(z_{t_r}) &= [\nabla^2 s(z_{t_r})]^{-1} \nabla s(z_{t_r}) \end{aligned} \quad (4.11)$$

with step size  $\delta_{t_r+1} \geq 0$ .

The property provides local minimizer of  $s(z)$  to make a descent direction such that

$$\nabla s(z) \neq 0 \Rightarrow l_s^T(z) \nabla s(z) \equiv -(\nabla s(z))^T [\nabla^2 s(z)]^{-1} \nabla s(z) < 0 \quad (4.12)$$

The *Newton Armijo* never satisfy the convergence of non-critical solution if  $s(z + \delta \times l_s(z))$  comes out as non-decreasing set at  $\delta > 0$ . To satisfy the convergence, it should have property, *i.e.*,

$$\nabla s(z) \neq 0 \Rightarrow (\nabla s(z))^T [\nabla^2 s(z)]^{-1} \nabla s(z) > 0 \quad (4.13)$$

The best way to impose this property is that  $z$  is convex with non-zero Hessian as:

$$(\nabla^2 s(z) > 0) \forall z [\equiv h^T [\nabla^2 s(z)] h > -\forall z \forall h \neq 0] \quad (4.14)$$

where,  $h = \nabla s(z)$ . This property possess the global convergence of Newton Method and its process is summarized as Algorithm 4.1.

The algorithm begins with initialization of point  $z_{t_r}$  for search set  $s(z) \leftarrow \infty$ , starting point,  $x_0 \leftarrow 0$ , total iteration,  $maxiter \leftarrow 2000$ , change in objective function,  $\delta \times s(z) \leftarrow \infty$ , tolerance,  $tolr = 1 * e^{-6}$ , change in step size,  $d_{xmin}$  to  $1 * e^{-6}$ . The loop initiates by checking

---

**Algorithm 4.1** Newton Algorithm

---

**Require:** (i) Search Set ( $s(z)$ ) (ii) starting point ( $x_0$ ) (iii) total iteration ( $maxiter$ ) (iv) change in objective function ( $\delta \times s(z)$ ) (v) Change in step size ( $d_{xmin}$ ), (vi) tolerance,  $tolr = d_{xmin} = 1 * e^{-6}$  (vii) Step Size ( $\delta$ )

**begin**

*select*  $z_{tr} \leftarrow z$  // initialize starting point

*Initialize*  $s(z) \leftarrow \infty$ ,  $maxiter \leftarrow 2000$ ,  $x_0 = 0$ ,  $\delta \times s(z) \leftarrow \infty$ ,  $d_{xmin} \leftarrow \infty$

// Calculation of Newton's direction

**while** ( $z_{tr} \geq tolr$ ) **do**

**if** ( $(x_0 < maxiter) \&\& (\delta \times s(z) \geq d_{xmin})$ ) **then**

$z_{onew} \leftarrow \text{Newton}(z_{tr})$

$z_{onew1} \leftarrow \text{normalize}(z_{onew})$

$z = x_0 - \left( \frac{s(z_{tr})}{\delta \times s(z_{tr})} \right)$

$l_s(z) = \text{normalize}(z - x_0)$  //unit step size calculation

*compute*  $s(z_{tr-1})$ ,  $\nabla s(z_{tr-1})$ ,  $\nabla^2 s(z_{tr-1})$

*initialize*  $\delta = 2$

**while** ( $l_s(\delta - 1) \geq tolr$ ) && ( $\delta \leq maxiter$ ) **do**

$z(\delta) = z(\delta - 1) - \left( \frac{s(z(\delta - 1))}{\delta \times s(z(\delta - 1))} \right)$  // Armijo's Rule

$\delta \times z(\delta) = \text{normalize}(z(\delta) - z(\delta - 1))$  // Updated step size

$\delta = \delta + 1$

**end while**

**end if**

**end while**

**end**

---

the value of  $z_{tr} \geq tolr$ . The condition is checked whether the resources are empty or change in step size goes to null. It then apply the Newton line search on initial point  $z_{tr}$  and stores in  $z_{onew}$ . The  $z_{onew1}$  has the normalized value of  $z_{onew}$ . It then updates the value of search set by taking difference between initial points and updated newton line direction. It checks for unit step size by each iteration and stores it in  $l_s(z)$ . Afterwards, it computes updated search set, its first and second order differentiable as  $s(z_{tr-1})$ ,  $\nabla s(z_{tr-1})$  and  $\nabla^2 s(z_{tr-1})$ , respectively. It initializes step size  $\delta$  as 2 and condition is checked that Newton's direction is larger than  $tolr$ ,  $\delta$  is smaller than  $maxiter$ . It calculates  $z(\delta)$  using *Armijo's Rule* by taking difference of updated search set  $z(\delta - 1)$  with change in updated search set of second order and then, updates the  $\delta$ . The process goes unless the search set is minimized.

The *Newton-Armijo* method takes second order derivative with step size  $\delta > 0$  and minimize the objective function. Though, it is fast at minimization of the objective function but it require to compute second order derivative and to update search set  $z$ . As a result, it takes more time and space with respect to complex objective function or n-dimensional constrained specific problems. Moreover, it is computational expensive. Therefore, to speed up the minimization process, another approach *namely*, Quasi-Newton Method is discussed.

### 4.3.3 Quasi-Newton Method

The Quasi-Newton Method is implemented to find the global minimum of a function that is twice-differentiable. It uses Variable Metric routines [167] which are recurrence of type

$$z_{r+1} = z_r - \delta_{r+1} \times Q_{r+1} \times \nabla s(z_r) \quad (4.15)$$

where,  $Q_{r+1}$  is the symmetric positive definite matrices,  $z_r, z_{r+1}$  are non-critical points,  $\nabla s(z_r)$  is the gradient of non-critical points,  $\delta_{r+1}$  is the step-size and is given in the direction ( $d_s$ ) as:

$$d_{s_{r+1}} = -Q_{r+1} \times \nabla s(z_r) \quad (4.16)$$

The directions are descent for  $s(z_r)$  at non-critical points,  $z_r$

$$\nabla s(z_r) \neq 0 \Rightarrow d_{s_{r+1}}^T \times \nabla s(z_r) \equiv -(\nabla s(z_r))^T \times Q_{r+1} \times \nabla s(z_r) < 0 \quad (4.17)$$

where,  $Q_r$  is positive definite. In Quasi-Newton Method [150],  $Q_{r+1}$  matrices is computed recursively which defines  $Q_{r+1}$  with the inversion of the Hessian matrices,  $M_H$ , *i.e.*,

$$M_H = Q_{r+1} - [\nabla^2 s(z_r)]^{-1} \rightarrow 0, r \rightarrow \infty \quad (4.18)$$

ensures the quick convergence of the objective function.

Suppose,

$$s(z_r) = \frac{1}{2} z_r^T \times (M_H) z_r - m^T z_r \quad (4.19)$$

to be strong convex quadratic form on  $R^n$  and is symmetric positive definite. Assume  $s(z)$

could be minimized by the Variable Metric approach which intend to generate  $Q_r$  recursively. For minimization, assume certain approximation  $Q_r$  of the matrix is used while moving from step  $z_{r-1}$  to  $z_r$ , *i.e.*,  $[\nabla^2 s(z_{r-1})]^{-1}$ , in the direction  $-Q_r \times k_{r-1}$ ,  $k_r \equiv \nabla s(z_r)$ . Thus, the step  $p_r$  is given as

$$p_r \equiv z_r - z_{r-1} \equiv -\delta_r \times Q_r \times k_{r-1} \quad (4.20)$$

where,  $\delta_r$  is the step-size. At this point, the first order information on  $s(z)$  helps to define the vectors,  $v_r$  as

$$v_r \equiv k_r - k_{r-1} \quad (4.21)$$

Here, if the  $p_r$  is small and  $s(z)$  is twice continuously differentiable, then,

$$v_r \equiv k_r - k_{r-1} \equiv \nabla s(z_r) - \nabla s(z_{r-1}) \approx [\nabla^2 s(z_{r-1})](z_r - z_{r-1}) \equiv [\nabla^2 s(z_{r-1})] \times p_r \quad (4.22)$$

The  $s(z)$  is quadratic and independent from  $p_r$ . Thus, after step  $r$  the approximation for  $Q_r$  to  $M_H$  provides  $[\nabla^2 s(z_{r-1})] \times p_r$ . To update previous approximations, there is need for inverse of Hessians. Therefore, its exact line search is given by:

$$\delta_{r+1} = \min (s(z_r + \delta \times d_{s_{r+1}})) \quad (4.23)$$

Then, the residuals,  $R_d$ , in terms of the objective function is given as

$$R_d(z_r) \equiv s(z_r) - \min (s(z)) \quad (4.24)$$

that satisfy the recurrence

$$R_d(z_{r+1}) \leq R_d(z_r) \left[ \frac{C_{n_{r+1}} - I}{C_{n_{r+1}} + I} \right]^2 \quad (4.25)$$

where,  $C_{n_{r+1}}$  is the condition number of the matrix and is given as:

$$M_{H_{r+1}} = Q_{r+1}^{\frac{1}{2}} \times M_H \times Q_{r+1}^{\frac{1}{2}} \quad (4.26)$$

The process is summarized as Quasi-Newton Algorithm 4.2.

---

**Algorithm 4.2** Quasi- Newton Algorithm

---

**Require:** (i) Set  $z$  (ii) Search Set ( $s(z)$ ) (iii) Gradient ( $k$ ) (iv) Starting point ( $z_0$ ) (v) total iteration ( $maxiter$ ) (vi) Hessian ( $M_H$ ) , (vii) descent direction ( $d_s$ ) (viii) Residual matrix ( $R_d$ )

**begin**

Initialize  $z_0 \leftarrow 0, r = 0, s(z) \leftarrow \infty, maxiter \leftarrow 2000$

Compute  $k_0 \leftarrow \nabla s(z_0)$

**for**  $r = 1$  to  $maxiter$  **do**

$k_{r-1} \leftarrow \delta \times s(z_{r-1})$  //Gradient

    Compute  $M_H \leftarrow Q_{k_{r-1}}$  // describing Hessian

**if** ( $k_{r-1} == 0$ ) **then**

        Terminate  $z_{r-1}$  // critical point

**else**

        set  $d_{s_{r+1}} \leftarrow -Q_{r+1} \times \nabla s(z_r)$  // describing descent direction using Variable Matrix

**end if**

**for** ( $d_{s_r} \leftarrow 1$  to  $z_{r-1}$ ) **do**

$z_r \leftarrow z_{r-1} + \delta_r \times d_{s_r}$  // updated sets

**end for**

    Compute  $k_r \leftarrow \delta \times s(z_r)$

    Set  $p_r \leftarrow z_r - z_{r-1}$  // decreasing step size

    Set  $v_r \leftarrow k_r - k_{r-1}$  // decrease in Gradient

**end for**

**for**  $r = 1$  to  $maxiter$  **do**

        Compute  $S_{r+1} \times v_r \leftarrow p_r$  // new direction

        Update  $S_{r+1}, s(z)$  // updating search sets

**end for**

    Print  $R_d(z_r) \leftarrow s(z_r) - mins(z)$  // display Residuals

**end**

---

It initializes starting point  $z_0$  at zero, search set  $s(z)$  at infinity, total iteration *maxiter* at 2000 and  $r$  to zero. It computes the first order differential  $\nabla s(z_0)$  at the starting point and stores in  $k_0$ . The loop is initiated for  $r$  up to *maxiter* in which gradient is computed and Hessian  $M_H$  is described. The condition is checked if the gradient value,  $k_{r-1} = 0$ . If it is null, then it terminates the critical point as  $z_{r-1}$  otherwise it finds the descent direction,  $d_{s_{r+1}}$  using variable metric. Another loop begins for descent direction and updates the set  $z_r$ . It calculates the gradient for updated  $k_r$ . Afterwards, it checks for new step size and gradient. The last loop finds the new direction and updates the search set. Finally, it prints the obtained residual matrix and the process provides the minimized set.

Unlike *Newton Method*, the *Quasi-Newton* method needs not to compute a second derivative and has faster computation. But it requires more convergence steps to reach the optimal solution. Therefore, to overcome the time constraint, another approach, *namely*, *IGD* is introduced that is implemented for *ITS* applications.

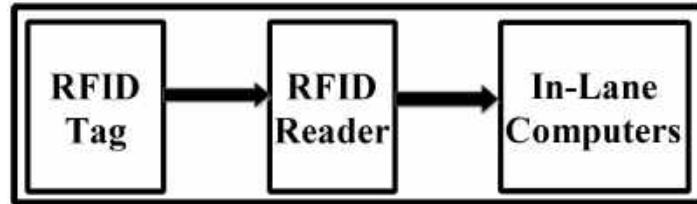
## 4.4 Intelligent Transportation System

*ITS* refers to adding information and communication technologies to transport infrastructure in order to enhance safety, reduce traffic congestion, transportation time and fuel consumption. Its major applications are traffic light control, weather information system, bus-stop information, navigation information, Electronic Toll Collection (*ETC*), parking management, *etc.* Among them, *ETC* is focused on for the proposed work.

### 4.4.1 Electronic Toll Collection

*ETC* is a supervening application that helps in identifying resources, *say*, vehicles, to enhance convenience for drivers using cashless toll collection, eradicating traffic congestion, reducing waiting time at toll queue and illegal toll-gate entry. For example, India, a developing country, is facing traffic congestion problems at National Highways due to public transport infrastructures and lack of formal national road safety policies. This congestion leads to obvious delays, jams; resulting into traffic chaos with increase in air and noise pollution. Moreover, it has negative impact on the operational efficiency of the network and thus,

eliminating the benefits of the infrastructure. In order to ease this inconvenience, *ETC* helps in removing all these problems to a greater extent. The main components of *ETC* are *RFID* reader, *RFID* tag, and in-lane computer as shown in Figure 4.1.

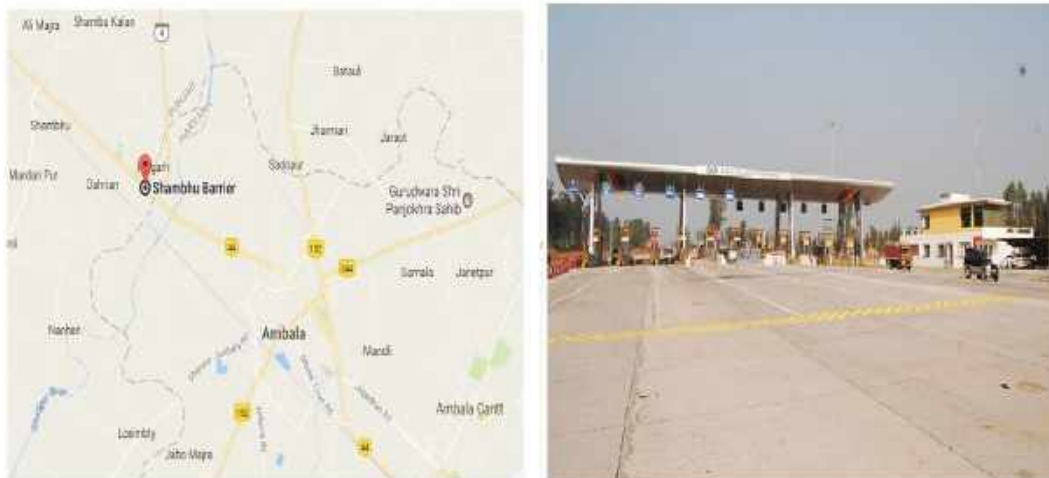


**Figure 4.1:** Components for *ETC* [8]

An *RFID* Tag consists of a chip that transmits the identification data including time, date, toll charge for vehicle, vehicular details to the reader *via* a antenna. The reader with the help of the inbuilt antenna then, transmits this data to in-lane computers, where the data is checked in the databases for valid tags and active accounts.

#### 4.4.2 Study Area: Shambhu Toll Plaza

The National Highway Authorities (*NHAI*), India targets to extend six lanes facility from 96 kms to 387.10 kms on toll basis through various Toll Plazas such as Ladowal/Shambhu, *etc.* Among them, proposed work has considered Shambhu Toll Plaza, National Highway (*NH-1*), India that has six counters to deal with the heavy traffic (*see* Figure 4.2).



(a)

(b)

**Figure 4.2:** Scenario (a) Application Area (b) Shambhu Toll Plaza, NH-1, India

These counters witness massive traffic snarls at all entry and exit points of toll plaza and are inadequate to meet the heavy traffic at rush hours. Therefore, it requires automated techniques to handle the rush in order to avoid a traffic bottleneck which leads to congestion and road accidents. To eradicate the problems, one possible solution is to provide automated optimized techniques for car reading, over speeding, vehicle out of control, sharp or slight curves, vehicle details, *etc.* that helps in minimizing traffic load at Toll Plazas.

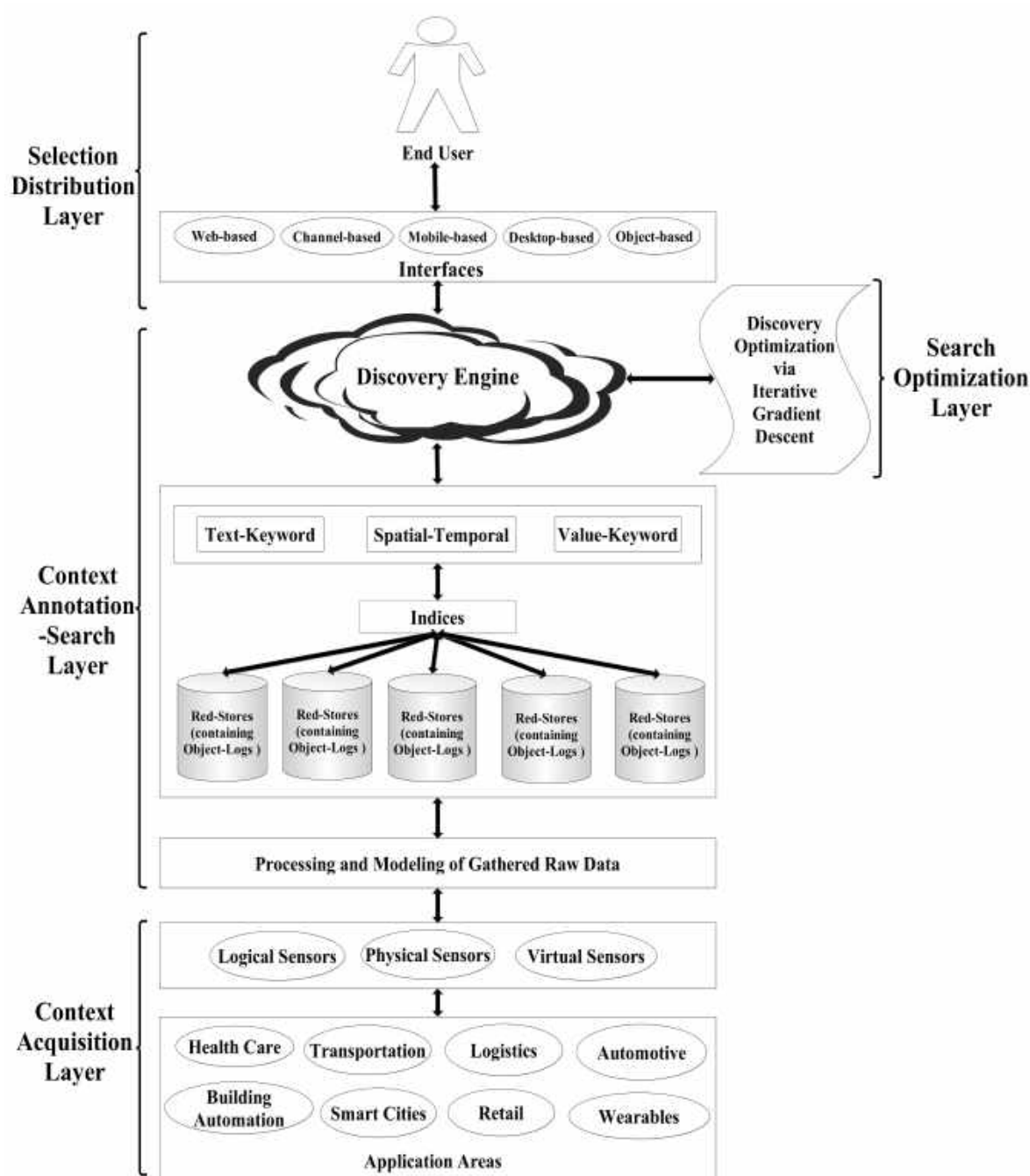
From the Shambhu Toll Plaza, the data is collected that includes details such as transaction identification (*transID*), vehicle registration number as its identification (*vehicleID*), *etc.*, as shown in Figure 4.3.

Transaction ID	Date/Time	Lane L.State	P.Type	P.Mean	Barcode No.	Collector	Vsh. Reg. No.	Col.	AVC	Rev.	Col.
010106JMA201609012359088212	01/09/2016 21:59:15	06	Open Payment	Daily Pass (I)	QM7ET22030A	00090	HR45A2730	LCV/Mini Bus	Truck/Bus	LCV/Mini Bus	175
010105JMA201609012356352482	01/09/2016 21:58:30	05	Open Payment	Daily Pass (I)	JTBA272202G5	00120	HR63B4788	Truck/Bus	LCV/Mini Bus	Truck/Bus	349
010113PMA201609012357588332	01/09/2016 21:58:15	13	Open Payment	Daily Pass (I)	UOVNET2206GT	00111	PB07BA0812	Car/jeep	Car/jeep	Car/jeep	100
020102JMA201609012357303482	01/09/2016 21:57:47	02	Open Payment	Daily Pass (R)	T7BA9K2206GD	00015	PB10E22121	Car/jeep	Car/jeep	Car/jeep	0
010113PMA201609012357579202	01/09/2016 21:58:02	13	Open Payment	Daily Pass (I)	VCEF67206GN	00111	PB08CG7007	Car/jeep	Car/jeep	Car/jeep	100
010113PMA201609012356506392	01/09/2016 21:57:57	15	Open Payment	Daily Pass (R)	Q2HT2P2201G8	00122	HR02AE3281	Car/jeep	Car/jeep	Car/jeep	0
010102JMA201609012357145882	01/09/2016 21:57:25	02	Open Payment	Monthly Pass	311111200000311892394618	00015	PB11BU9288	Truck/Bus	Truck/Bus	Truck/Bus	0
010117PMA201609012357129212	01/09/2016 21:57:15	17	Open Payment	thly Pass Local	311111200000637978014137	00096	HR01AE0E10	Car/jeep	Car/jeep	Car/jeep	0
010114PMA201609012356184742	01/09/2016 21:56:45	14	Open Payment	Daily Pass (I)	ND4N1T22070P	00087	HR26AY2828	Car/jeep	Car/jeep	Car/jeep	100
010113PMA201609012356572802	01/09/2016 21:56:58	13	Open Payment	Daily Pass (I)	KCEF272206G1	00111	PB08DDA286	Car/jeep	Car/jeep	Car/jeep	100
010117PMA201609012355287232	01/09/2016 21:56:19	17	Open Payment	Daily Pass (R)	HML1LJ22030W	00096	HR01AG8831	Car/jeep	Car/jeep	Car/jeep	0
010113PMA201609012354570682	01/09/2016 21:56:04	13	Open Payment	Daily Pass (R)	UJ3UJER2201GU	00111	HR01AG7776	Car/jeep	Car/jeep	Car/jeep	0
010114PMA201609012355079352	01/09/2016 21:56:07	14	Open Payment	Daily Pass (R)	R13GBP220204	00087	DLL1T9310	LCV/Mini Bus	LCV/Mini Bus	LCV/Mini Bus	0
010113PMA201609012354462282	01/09/2016 21:55:26	13	Open Payment	Daily Pass (I)	J1BMR72206GU	00111	PB11BF4751	Car/jeep	Car/jeep	Car/jeep	100
010114PMA201609012354269372	01/09/2016 21:55:08	14	Open Payment	Cash	-	00087	PB10CJ0869	MAV	MAV	MAV	374
010117PMA201609012355248012	01/09/2016 21:55:45	17	Open	Exempt No payment	-	00096	0000	Agriculture	MAV	Agriculture	0
010103JMA201609012355052842	01/09/2016 21:55:05	03	Open Payment	Cash	-	00100	HR19C4811	Car/jeep	Car/jeep	Car/jeep	66
010117PMA201609012354470392	01/09/2016 21:54:52	17	Open Payment	Cash	-	00096	HR628988	Car/jeep	LCV/Mini Bus	Car/jeep	66
010106JMA201609012354278972	01/09/2016 21:54:46	06	Open Payment	Cash	-	00090	HR678095	MAV	MAV	MAV	374
010102JMA201609012354339042	01/09/2016 21:54:57	02	Open Payment	Daily Pass (R)	T75GA42206G4	00015	PB1807443	Car/jeep	Car/jeep	Car/jeep	0
010102JMA201609012354079022	01/09/2016 21:54:16	02	Open Payment	thly Pass Local	311111200000920782731898	00015	PB11BA1689	Car/jeep	Car/jeep	Car/jeep	0
010117PMA201609012353038762	01/09/2016 21:53:45	17	Open Payment	Daily Pass (I)	UMDMVT2208GQ	00096	PB09L1233	Car/jeep	Car/jeep	Car/jeep	100

Figure 4.3: Dataset collected from Shambhu Toll Plaza, NH-1, India

## 4.5 Proposed Context-Aware Search Optimization Framework on Internet-of-Things

The Context-Aware Search Optimization Framework on Internet-of-Things (*CASOF-IoT*) intends to perform optimal resource discovery (*see* Figure 4.4) through its four layers, *namely*, Context Acquisition Layer (*CAL*), Context Annotation-Search Layer (*CASL*), Search Optimization Layer (*SOL*) and Selection-Distribution Layer (*SDL*).



**Figure 4.4:** Context-Aware Search Optimization Framework on Internet-of-Things

These layers are responsible for gathering the data, modeling it for storage and processing, efficient searching with appropriate reasoning, optimizing the searched output which is accessed through various channels and interfaces, respectively. Each layer with its complete functionality is detailed below.

### 4.5.1 Context Acquisition Layer

*CAL* is responsible for data collection which depends on context originating source, sensor type, frequency, responsibility and the acquisition process itself. Here, the source of context can be context servers, hardware, and can be acquired through middle-ware or infrastructure; sensor types are physical, virtual and logical; frequency means how data is captured, *i.e.*, either instantly or periodically; responsibility means either to push data to the software or pull request by the software; and the acquisition processes means to sense, derive or provide data manually.

### 4.5.2 Context Annotation-Search Layer

*CASL* models the gathered data as storage layer to provide efficient storage through the usage of indexed tags or labels with few specific constraints in the record. The layer has provisions to store raw data, its originating source, meta-data, resource type and field linked with the resource. It is achieved through the resource data stores (*Red-Stores*) which are organized as object records. Each object record is composed of series of time-stamped sampling values which contain its corresponding sampling values for objects. The sampled values are stored together to form a sampling sequence which is further indexed using three indices, *namely*, text-keyword, spatial-temporal and value-keyword. The indices follow the heap storage that provides faster access to the stored data as sampling sequence at various locations [69]. The schema for the storage layer is detailed below.

***Instance Value:*** The data is generated from the heterogeneous sources and is in uninformed manner, therefore, it requires to be indexed as per its instance. It is defined as:

$$\text{Instance Value} = (T, Loc, Ptrn, Val)$$

where, *T, Loc, Ptrn, Val* are time of sampling, location of sampled value, format of sampling and actual value, respectively.

***Sequence of Instance:*** It is composed of the sampled values of various objects which are monitored by one equipment at a given period of time. For example, values of similar objects have similar pattern. Therefore, sequence of instance is defined as:

$$\text{Instance Sequence} = (\text{Ptrn} \sum_{j=1}^n (T_j, \text{Loc}_j, \text{Val}_j))$$

where,  $\text{Ptrn}, T_j, \text{Loc}_j, \text{Val}_j$  are the format of sample, time, position and actual value of the  $j^{\text{th}}$  sample, respectively.

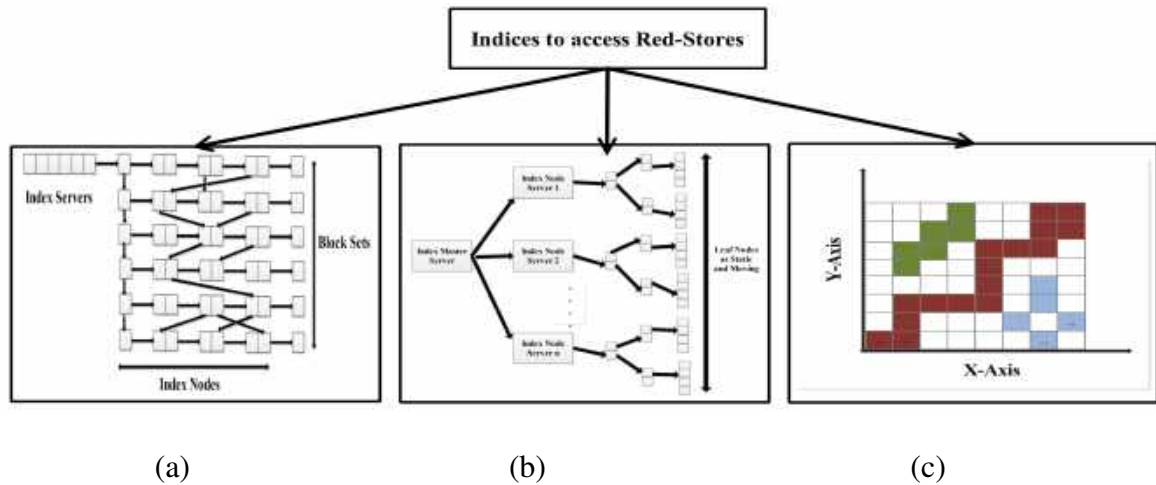
**Object Log:** It denotes the log of certain objects that are monitored and is given by:

$$\text{ObjectLog} = (\text{objId}, \text{objDesc}, \text{Instances}, \text{PhyAddr})$$

where,  $\text{objId}, \text{objDesc}, \text{Instances}$  are the identification, description, sequence of the object, respectively and  $\text{PhyAddr}$  is the physical address of the equipment which gather values from the object.

In summary, there are multiple *Red-Stores* containing number of Object Logs and are distributed among each other in accordance to the  $\text{objId}$  and  $\text{Loc}$ . These *Red-Stores* are indexed through three indices, *namely*, text-keyword, spatial-temporal and value-keyword (see Figure 4.5) and are explained below.

**Text-Keyword Index:** It is a distributed index built on multiple index servers. Its internal and root node have index records having form  $\langle k, \text{Idxnode}, \text{Ptr} \rangle$  where  $k, \text{Idxnode}, \text{Ptr}$  represents keyword, address of the child node and pointer to detailed page address of child node (inside  $\text{Idxnode}$  server), respectively (see Figure 4.5(a)). The leaf nodes have form  $\langle k, \text{set}(\text{Red - Stores}), \text{set}(\text{objId}) \rangle$ , where,  $k$  is the keyword,  $\text{set}(\text{Red-Stores}), \text{set}(\text{objId})$  are the identifiers specifying detailed logs of objects. The detailed logs are stored in blocks whereas leaf nodes contain pointers only leading to them.



**Figure 4.5:** Indices to Red-Stores (a) Text-Keyword, (b) Spatial-Temporal, (c) Value-Keyword

**Spatial-Temporal Index:** It is a two-layered index, *namely*, Partition Table (*PT*) stored at master server and Instance Sequence Time (*IST*) (*see* Figure 4.5(b)). Assume that there are  $k$  index nodes denoted as  $Idx_1, Idx_2, \dots, Idx_n$ . Each node server ( $Idx_p$ ) has geographical area which is denoted as  $\beta(Idx_p)$ , then it should meet two conditions:

- (i)  $\forall p, q (p \neq q) : \beta(Idx_p) \cap \beta(Idx_q) = \phi$
- (ii)  $\cup_{p=1}^k \beta(Idx_p) = G$

The records in *PT* has form  $(G_A, Idxnode)$ , where,  $G_A$  is the geographical area and  $Idxnode$  is the index node server corresponding to that area. *IST* resides at index node server and has two subtrees, *namely*, Spatial Tree and Grid Sketched Tree. The spatial tree indexes the location of stationary objects which has form  $\langle Loc, Red - Stores, objId \rangle$ , where *Loc*, *Red-Stores*, *objId* are the position, detailed log and identification of the objects, respectively. Grid Sketched Tree index the trajectories of moving objects and are in form exactly the same as of spatial tree. More precisely, the logs of the leaf node has form  $\langle G_sJ, set(Red - Stores), set(objId) \rangle$ . Here,  $G_sJ$  is the grid-sketched by the dynamic object and other attributes, *i.e.*,  $set(Red - Stores), set(objId)$  specifies the detailed log of the objects. The trajectory of the dynamic objects changes frequently and is not the same as observed. Hence, it leads to reduction in the frequency updation of the indexed objects.

**Value-Keyword Index:** It is based on Grid Symbolization method and follows Instance Value based search. The grid cells are adjoined on the basis of value (*V*), Time (*T*) and are partitioned in to equal sized  $V \times T$  grid cells (*see* Figure 4.5(c)). The Value at time *T* is mapped as  $(Value, T)$  pair and uses cell identifier for the symbolizing keyword. Thus, all instances are symbolized in to keyword and on the basis of keywords, value-keyword trees are build up.

The complete *CASL* process ease the knowledge representation for the gathered data which is accessed through its calling. The calling process is executed as Multi-modal Search Algorithm 4.3.

Initially, a query is input to the system as  $Q_a$ . The system stores the content of  $Q_a$  as boolean in order to search the context. If the query search is text-based, then it searches the context using the Text-Keyword Index. If the searched query is either spatial or temporal, then it

---

**Algorithm 4.3** Multi-modal Search Algorithm

---

**Require:** Query ( $Q_a$ ), Storing atomic searches ( $Q_{bool}$ ), Search Text (text-keyword), Search Spatial (spatial or temporal), Searching Instance (instance value), output ( $z$ )

```
begin
Input  $Q_a \leftarrow query$ 
for  $i = 1$  to  $n$  do
     $Q_{bool}[n] \leftarrow q_a[n]$ , where  $q_a[1], q_a[2], \dots, q_a[n]$  are the atomic searches
end for
for  $i = 1$  to  $n$  do
    if ( $q_a[i] \leftarrow keyword\ search$ ) then
         $z(q_a[i]) = search(Text\text{-}keyword, q_a[i])$ 
    else if ( $q_a[i] \leftarrow spatial\ or\ temporal$ ) then
         $z(q_a[i]) = search(spatial\ or\ temporal\ index, q_a[i])$ 
    else if ( $q_a[i] \leftarrow value\ search$ ) then
        grid cells  $\leftarrow Grid\ Symbolization\ q_a[i]$ 
         $z(q_a[i]) = search(instance\ value, grid\ cells)$ 
    end if
end for
 $z = evaluate(Q_{bool}[n], z(q_a[1]), z(q_a[2]), \dots, z(q_a[n]))$ 
Print  $z$ 
end
```

---

process the results using Spatial-Temporal Index. If the query searched is value-based, then it symbolizes the context with the grid cells, which is further searched against cells using Instance-Value search. The search matches with keyword, spatial or temporal and value based are finally combined as output  $z$  for the query.

### 4.5.3 Search Optimization Layer

In order to optimize  $z$ , an optimization algorithm, *namely*, Iterative Gradient Descent *IGD* is proposed that uses *Armijo's Rule* to optimize the search complexity by minimizing functional constrained. *IGD* follows two properties, *i.e.*, *iterative* and *convergence*. The former helps

in finding solution to the problems that require finite number of computations whereas latter is the property that validates the qualified method for solving non-linear problems. *IGD* uses first order derivative to construct the local approximations of the objective function.

Consider  $s(z)$  to be continuously differentiable and is defined in a neighborhood of a point  $g_0$  in search-space set  $z$ . Assuming that at point  $g_0$ ,  $s(z)$  and  $\nabla s(z)$  are computed. Considering that  $g_0$  is not a critical point of  $s$ :  $\nabla s(z) \neq 0$ . In this scenario, Gradient Descent  $G_d = -\nabla s(z)$ , which is a descent direction of  $s$  at  $g_0$ . It implies

$$\frac{d}{d\delta} \Big|_{\delta=0} s(g_0 - \delta \times \nabla s(z)) = - |\nabla s(z)|^2 < 0 \quad (4.27)$$

For any other direction, say,  $g_1$ ,  $|g_1| = |G_d|$  of  $s(z)$ , it comes out as:

$$\frac{d}{d\delta} \Big|_{\delta=0} s(g_0 + \delta \times g_1) = g_1^T \times \nabla s(g_0) \geq - |g_1| \times |\nabla s(g_0)| = - |\nabla s(z)|^2 \quad (4.28)$$

The above equation's equality holds if  $g_1 = G_d$ . It suggests that to perform a step from predecessor point to successor with smaller value of the objective function, it requires

$$g_0 \mapsto g_0 + \delta \times G_d \equiv g_0 - \delta \times \nabla s(z) \quad (4.29)$$

with step-size  $\delta > 0$  for decrease in  $s(z)$ . For any positive step size, say,  $\delta_{t_r}$ , the above equation is formalized as:

$$g_{0_{t_r}} = g_{0_{t_r-1}} - \delta_{t_r} \times \nabla s(g_{0_{t_r-1}}) \quad (4.30)$$

In order to perform line search more faster, it implements Gradient Descent with *Armijo's Rule* with its iteration,  $t_r$  and step size,  $\delta_{t_r} > 0$  such that  $\nabla g_{0_{t_r-1}} \neq 0$  and is given as:

$$s(g_{0_{t_r-1}} - \delta_{t_r} \times \nabla s(g_{0_{t_r-1}})) \leq s(g_{0_{t_r-1}}) - \varepsilon^T \delta_{t_r} |\nabla s(g_{0_{t_r-1}})|^2 \quad (4.31)$$

$$s(g_{0_{t_r-1}} - \vartheta \delta_{t_r} \times \nabla s(g_{0_{t_r-1}})) \geq s(g_{0_{t_r-1}}) - \varepsilon^T \vartheta \delta_{t_r} |\nabla s(g_{0_{t_r-1}})|^2 \quad (4.32)$$

where  $\varepsilon \in (0, 1)$  and  $\vartheta > 1$  are the parameters of method. Here, if  $s(g_{0_{t_r-1}}) = 0$ , then  $\delta_{t_r} > 0$

becomes independent and Eq. 4.30 results as:

$$g_{0_{t_r}} = g_{0_{t_r-1}} \quad (4.33)$$

The value of critical points need to be set for the convergence of the objective function  $s(z)$  globally such that it follows:

$$G_d^{**} = \{g_0 \in R^n \mid \nabla s(z) = 0\} \quad (4.34)$$

and it needs to satisfy following statements.

- (i) The critical points of  $s(z)$  are the limiting points for the trajectory,  $t_r$  and are bounded.
- (ii) The level set  $L_s$  is

$$L_s = \{g_0 \in R^n \mid s(z) \leq s(z_0)\} \quad (4.35)$$

holds previous statement of the trajectory  $t_r$  and provides  $G_d$  only if:

$$g_{0_{t_r}} \neq g_{0_{t_r-1}} \Rightarrow s(g_{0_{t_r}}) < s(g_{0_{t_r-1}}) \quad (4.36)$$

which minimizes the  $s(z)$ . The equation implies that the trajectory of *Armijo's Rule* is contained in  $s(z)$  and is bounded.

The *Armijo's Rule*  $I_A$  is memory less, iterative and is given as:

$$I_A = \{G_d \in R^n, \theta(\cdot)\} \quad (4.37)$$

where,  $\theta$  is point-to-set mapping of critical points and is defined as:

$$\theta(g_0) = \begin{cases} g_{0_1} \mid g_{0_1} = g_0 - \delta \nabla s(z) \text{ satisfying the Armijo's Rule,} & \nabla s(z) \neq 0 \\ g_0, & \nabla s(z) = 0 \end{cases} \quad (4.38)$$

Also,  $\theta(\cdot)$  is composed of two mapping process, *i.e.*,

$$\theta(\cdot) = M_{in} \times M_{out} \quad (4.39)$$

where,  $M_{in}$  is the inner mapping (point-to-point) from  $R^n$  to  $(R^n \times R^n)$  and is given as:

$$M_{in}(g_0) = (g_0 - \nabla s(g_0)) \quad (4.40)$$

which is continuous and has direct descent direction of  $s(z)$  at  $g_0 : d_s^T \nabla s(z) < 0$ . On the other hand,  $M_{out}$  is the outer mapping (point-to-set) from  $(R^n \times R^n)$  to  $R^n$  and is given as:

$$M_{out}(g_0, d_s) = \begin{cases} g_{0_1} = g_0 + \delta \nabla s(z) \text{ with } \delta \text{ satisfying the Armijo's Rule,} & d_s^T \nabla s(z) < 0 \\ \{g_0\}, & d_s^T \nabla s(z) \geq 0 \end{cases} \quad (4.41)$$

which is closed at point  $M_{in}(z)$ . This ensures the global convergence of the  $G_d$ . Its limiting points are calculated by continuous approximation method with trajectory  $g_0(t_r)$  which is given by:

$$\frac{d}{dt_r} g_0(t_r) = -\nabla s(g_0(t_r)) \quad (4.42)$$

$\nabla s(g_0(t_r))$  is finite difference approximation solution providing positive definite local minimizer of  $s(z)$  with trajectory of limiting points at  $g_0$  and  $g_0^*$  having function  $P(g_0)$ , which is given as:

$$P(g_0) = |g_0 - g_0^*|^2 \quad (4.43)$$

Therefore, the continuous approximation time becomes

$$\frac{d}{dt_r} P(g_0(t_r)) = 2(g_0(t_r) - g_0^*)^T \frac{d}{dt_r} g_0(t_r) = -(g_0(t_r) - g_0^*)^T \times \nabla s(g_0(t_r)) \quad (4.44)$$

However, the trajectory convergence to local minimizer depends upon the choice of the starting point. The entire process of *IGD* is summarized in Algorithm 4.4.

Firstly, it initializes the starting point  $g_0$  from the search set  $z$ , Gradient Descent  $G_d$  to  $\infty$ , maximum Iteration *maxiter* to 2000, step size  $\delta$  as 0.2, minimum change in  $\delta$  as  $d_{xmin}$ , tolerance *tolr* to  $1 \times e^{-6}$  and change in step size as  $d_x$  to  $\infty$ . The objective function  $s(z)$  is minimized to find the neighbor using  $G_d$  from the  $z$ . The loop performs for  $g_0 = g_{norm}$  unless its value is greater than *tolr*. If the iterations do not exceed the defined limit for *niter* and change in step size  $d_x$ , then it finds the gradient and stores in  $g_{onew}$ . Further,  $g_{onew}$  is normalized and stored again in  $g_{onew1}$ . Finally, it computes new search set of  $g_{onew}$ , i.e.,

---

**Algorithm 4.4** Iterative Gradient Descent Algorithm (*IGD*)

---

**Require:** (i) Search Space ( $z$ ) (ii) Initial Critical point ( $g_0$ ) (iii) Gradient Descent  $G_d$  (iv) Maximum Iteration ( $maxitr$ ) (v) number of iterations ( $niter$ ) (vi) Step Size ( $\delta$ ) (vii) Change in step size ( $d_{xmin}, d_x$ ) and (viii) tolerance ( $tolr$ ).

```
begin
  // initialize starting point
  Select  $g_0 \leftarrow z$ 
  Initialize  $G_d \leftarrow \infty, maxitr \leftarrow 2000, niter \leftarrow 0, \delta \leftarrow 0.2, d_{xmin} = tolr = 1 \times e^{-6}, d_x \leftarrow \infty, g_{norm} = g_0$ 
  // Gradient Descent Calculation
  while ( $g_{norm} \geq tolr$ ) do
    if ( $nitr < maxitr$ ) && ( $d_x \geq d_{xmin}$ ) then
       $g_{onew} \leftarrow Gradient(g_0)$ 
       $g_{norm1} \leftarrow normalize(g_{onew})$ 
      Compute ( $s(g_{onew})$  and  $\nabla s(g_{onew})$ )
       $g_{oxnew} \leftarrow g_0 - \delta \times g_{onew}$  // Armijo's Rule
       $niter++$ 
       $d_x \leftarrow normalize(g_{oxnew} - g_{onew})$ 
       $g_0 \leftarrow g_{oxnew}$ 
    end if
  end while
end
```

---

$s(g_{onew})$  and change in set as  $\nabla s(g_{onew})$ . Now, the gradient and new line using *Armijo's Rule* is recalculated and stores in  $g_{oxnew}$ . The iteration is incremented by 1. The difference between  $g_{oxnew}$  and  $g_0$  is normalized and stored as updated  $d_x$ . At last,  $g_0$  is updated with new gradient  $g_{oxnew}$ . The process is repeated unless gradient point is found.

#### 4.5.4 Selection-Distribution Layer

The *SDL* provides the computed and optimized output to the end-user *via* various modes like Web-based, Channel-based, Mobile-based, Object-based and Desktop-based. These modes depends on the interface being used by the user in accordance to the application domains such as health-care, transportation, retail, *etc.*

## 4.6 Experimental Results and Discussion

In the experimental study, *ETC* at Shambhu Toll Plaza, *NH-1*, India is considered.

### 4.6.1 Framework Implementation

**Context Acquisition Layer:** The data is gathered from Shambhu Toll Plaza, *NH-1*, India (see Figure 4.3).

**Context Annotation-Search Layer:** According to the schema definition as explained in Section 4.5.2, the gathered data is modeled as shown in Figure 4.6.

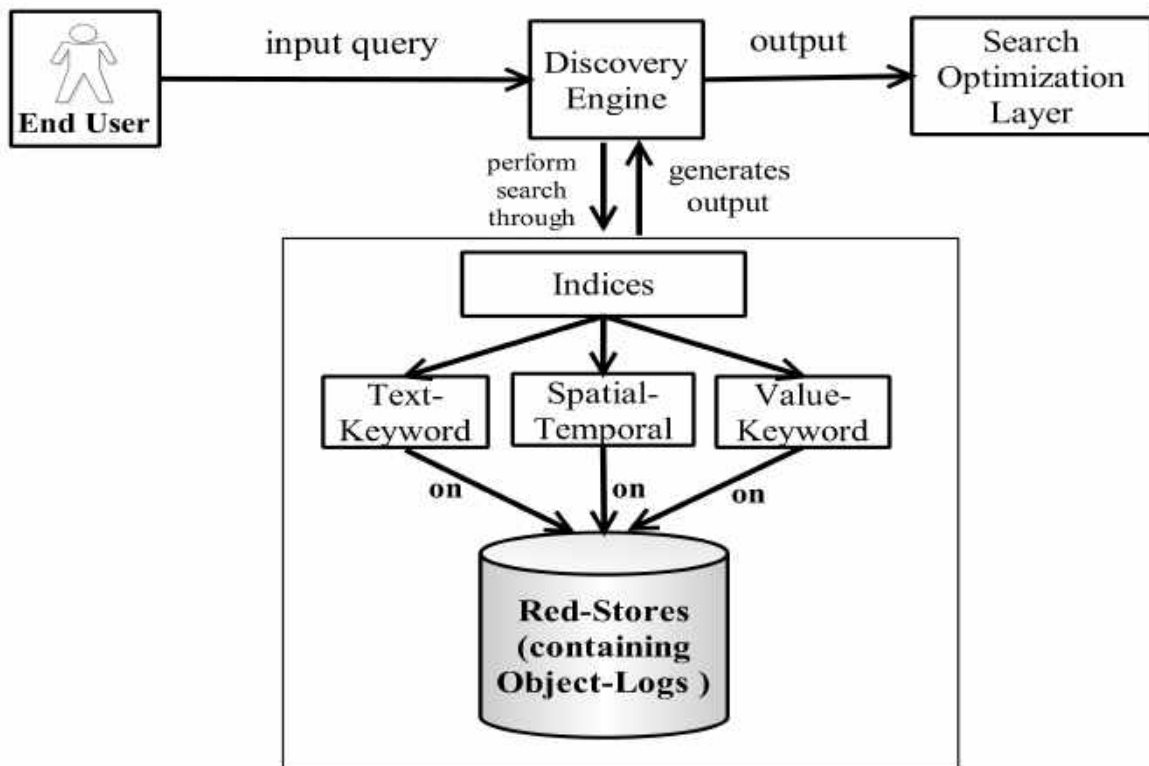


Figure 4.6: Data Modeling

For example, data like *transID*, *VehicleID*, *BankID* are collected and considered as  $x_1$ ,  $x_2$  and  $x_1.x_2$ , respectively. Mathematically, it is formulated as  $(x_1 + x_2 + x_1x_2)$ . Using this formulation, various objective functions (*OF*) are considered as shown in Table 4.1.

**Table 4.1:** Objective Functions(*OF*)

<b>OFs</b>	<b>Equations</b>
<i>OF1</i>	$x_1^2 - 2 \times x_1x_2 + x_2^2$
<i>OF2</i>	$5x_1^2 + 90 \times x_1x_2 + 9x_2^2$
<i>OF3</i>	$83x_1^2 - 121 \times x_1x_2 + 2x_2^2$
<i>OF4</i>	$0.4x_1^2 + 0.9 \times x_1x_2 + 0.9x_2^2$
<i>OF5</i>	$-9x_1^2 - 10 \times x_1x_2 - 8x_2^2$
<i>OF6</i>	$0.5x_1^2 + 0.2 \times x_1x_2 - 0.1x_2^2$

The *OFs* undergo further processing through the search algorithm (*see* Algorithm 4.3) which needs to be optimized with respect to the search metrics.

**Search Optimization Layer:** For optimizing the search output, the proposed *IGD* is performed to minimize the objective function by considering first order derivative (*see* Section 4.5.3). Initially, the search begins against query's parameters in the *CASL* layer (*see* Section 4.5.2), through which number of matched searches are retrieved as output  $z$ . The output  $s(z)$  undergoes the optimization process *via IGD* Algorithm as discussed in Section 4.5.3. Finally, the optimized solution against query is forwarded to the end user.

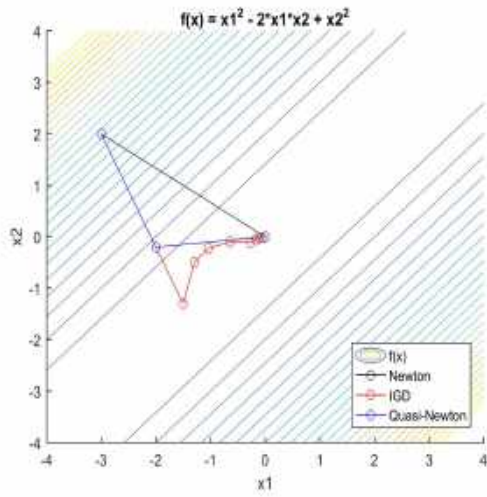
**Selection-Distribution Layer:** An end user through this layer accesses his or her account, authorized by the manager of the Electronic Toll Plaza. The interface channel for the same are either web, object or desktop based depending on the provision approved by the organization.

## 4.6.2 System Performance

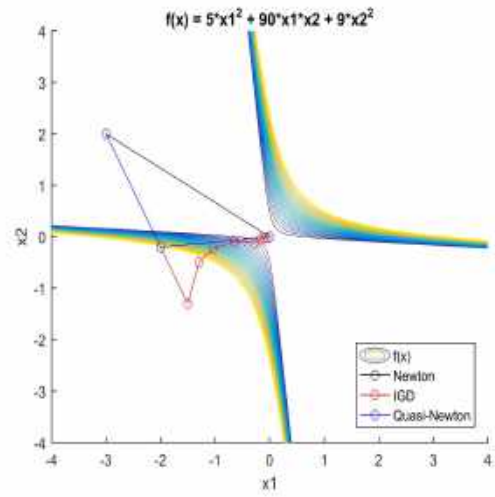
**Evaluations against Objective Functions:** To estimate the step size and gradient point, existing Newton, Quasi-Newton and proposed *IGD* methods are simulated for various non-linear quadratic equation as *OFs* using *MATLAB*. Figure 4.7, presents the simulated implementation of the *OFs*. In Figure 4.7(a), the graph plot is the simulation for the *OF1*:

$x_1^2 - 2 \times x_1x_2 + x_2^2$ , to be distributed in 2-D plane of  $x_1$  and  $x_2$  axes. Here, black color is for the *Newton* method, blue color shows the Quasi-Newton and red color represents *IGD*. The *OF1* is processed against 2000 number of iterations to converge it to global optimal solution. The elapsed time consumed by Newton, Quasi-Newton and *IGD* methods are 0.7322, 0.276 and 0.1804 s, respectively. In Figure 4.7(b), *OF2*:  $5x_1^2 + 90 \times x_1x_2 + 9x_2^2$ , is simulated upon which the various methods converge to get optimal solution. *OF2* is shown as hyperbola lines within 2-D plane of  $x_1$  and  $x_2$  axes. For 2000 iterations, the elapsed time for Newton, Quasi-Newton and *IGD* methods are 0.4652, 0.5353 and 0.1769 s, respectively. *OF3*:  $83x_1^2 - 121 \times x_1x_2 + 2x_2^2$ , is simulated to estimate step-size and gradient point for Newton, Quasi-Newton and *IGD* methods, as shown in Figure 4.7(c). The plot for *OF3* is shown in 2-D plane of  $x_1$  and  $x_2$  axes as hyperbola lines. The elapsed time for Newton, Quasi-Newton and *IGD* methods are 0.4372, 0.4147 and 0.1409 s, respectively, for 2000 iterations. In Figure 4.7(d), the graph plot is the simulation for the *OF4*:  $0.4x_1^2 + 0.9 \times x_1x_2 + 0.9x_2^2$ , to be distributed in 2-D plane of  $x_1$  and  $x_2$  axes. Here, black color is for the *Newton* method, blue color shows the Quasi-Newton and red color represents *IGD*. The *OF4* is processed against 2000 number of iterations to converge it to global optimal solution. The elapsed time consumed by Newton, Quasi-Newton and *IGD* methods are 0.5081, 0.4941 and 0.0233 s, respectively. In Figure 4.7(e), *OF5*:  $-9x_1^2 - 10 \times x_1x_2 - 8x_2^2$ , is simulated upon which the various methods converge to get optimal solution. *OF5* is shown as set of ellipses within 2-D plane of  $x_1$  and  $x_2$  axes. For 2000 iterations, the elapsed time for Newton, Quasi-Newton and *IGD* methods are 0.4752, 0.6350 and 0.0186 s, respectively. *OF6*:  $0.5x_1^2 + 0.2 \times x_1x_2 - 0.1x_2^2$ , is simulated to estimate step-size and gradient point for Newton, Quasi-Newton and *IGD* methods, as shown in Figure 4.7(c). The plot for *OF6* is shown in 2-D plane of  $x_1$  and  $x_2$  axes as hyperbola lines. The elapsed time for Newton, Quasi-Newton and *IGD* methods are 0.4179, 0.3409 and 0.2500 s, respectively, for 2000 iterations.

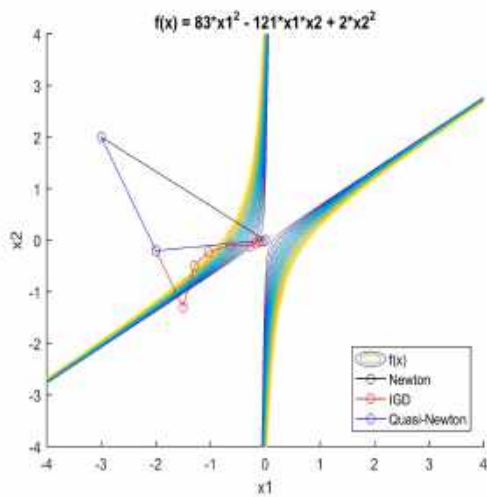
The elapsed time for Newton, Quasi-Newton and *IGD* methods to estimate step sizes and converging to an optimal solution is summarized in Table 4.2.



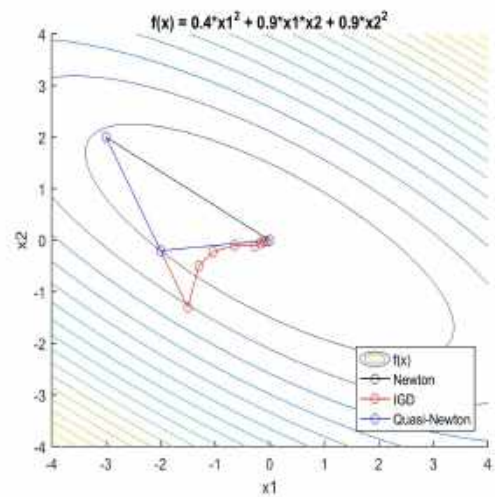
(a)



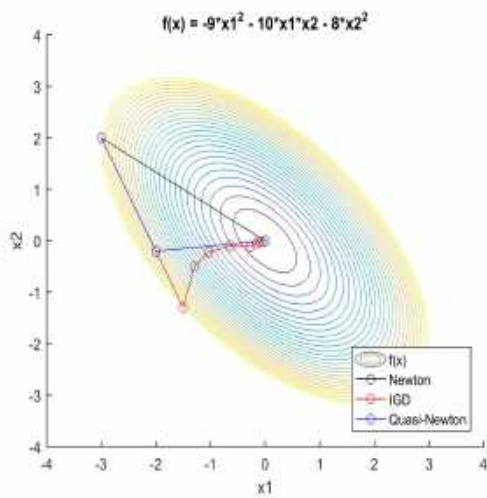
(b)



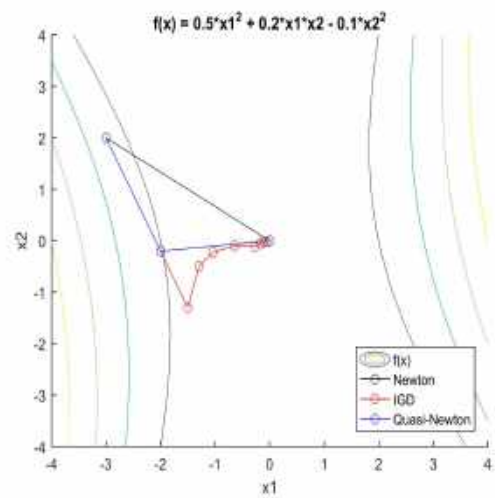
(c)



(d)



(e)



(f)

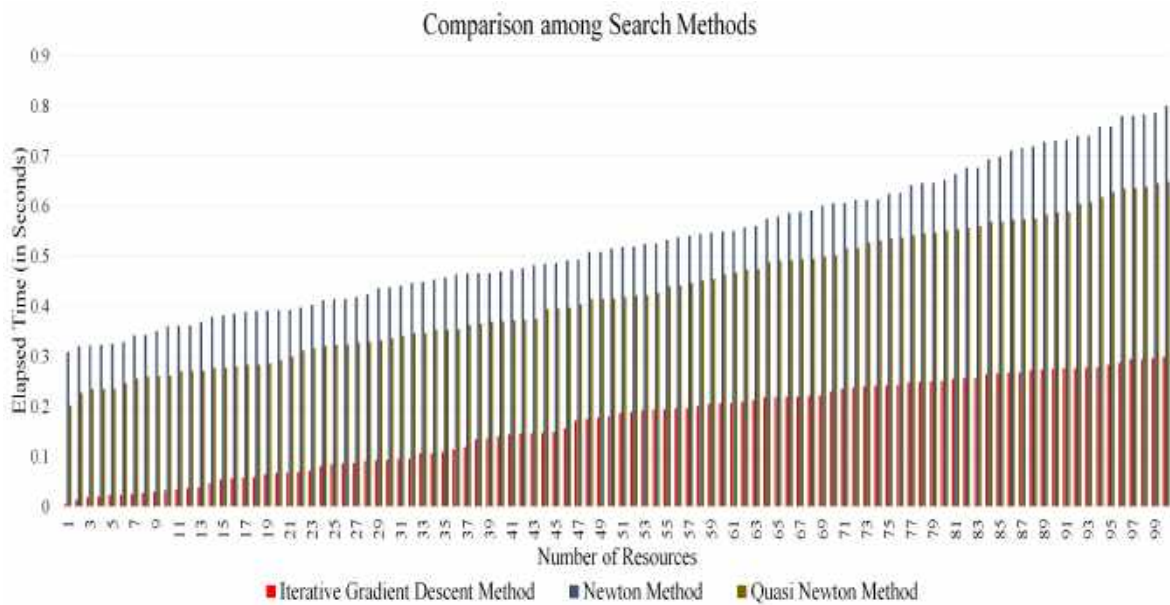
**Figure 4.7: Evaluation Results for Distinct Objective Functions**

**Table 4.2:** Objective Functions(*OF*), Elapsed Time (*ET*)

<b>OFs</b>	<b>Newton <i>ET</i> (s)</b>	<b>Quasi-Newton <i>ET</i> (s)</b>	<b><i>IGD ET</i> (s)</b>
<i>OF1</i>	0.7322	0.276	0.1804
<i>OF2</i>	0.4652	0.5353	0.1769
<i>OF3</i>	0.4372	0.4147	0.1409
<i>OF4</i>	0.5081	0.4941	0.0233
<i>OF5</i>	0.4752	0.6350	0.0186
<i>OF6</i>	0.4179	0.3409	0.2500

The table also suggest that the resources are exponentially growing with respect to its line search in a plane. The *Armijo's Rule* helps in describing gradient and decreasing the step size by every iteration in order to get the best gradient and matching resource in a set which is the neighboring resource.

**Evaluation against 100 resources:** The comparison among *IGD*, *Newton* and *Quasi-Newton* methods for searching neighbors is analyzed on the basis of its search time (see Figure 4.8) for 100 non-linear constrained equations as resources.



**Figure 4.8:** Comparison between *IGD*, *Newton* and *Quasi-Newton*

The *Newton method* varies considerably between 0.3 - 0.8s for searching. The method provides delay in searching if the number of resources are increasing. The *Quasi-Newton* method takes search time between 0.2 - 0.65s whereas the *IGD* provides an optimal search taking time among 0.1 - 0.35s for resources. Also, there is no delay in searching with increase in number of resources.

Thus, it is concluded that *IGD* is better and optimized neighbor search technique as compared with *Newton* and *Quasi-Newton* methods, for 100 resources.

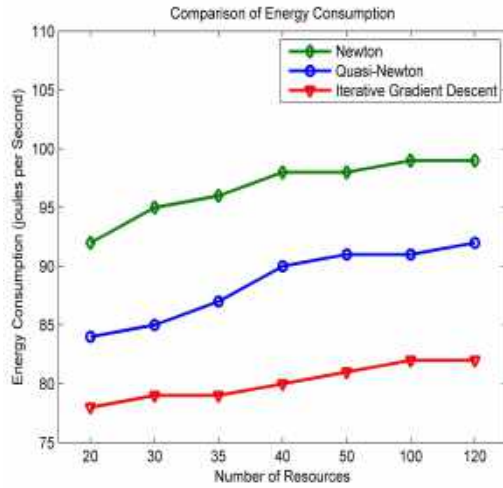
### 4.6.3 Performance Metrics

The *CASOF-IoT* against the discussed techniques, *i.e.*, *Newton*, *Quasi-Newton* and *IGD* is analyzed with respect to performance metrics like energy consumption, delivery of packet, data loss, throughput, latency and quality of data. It is done for application, *Electronic Toll Plaza* (see Section 4.4.2) to compute the efficiency of the proposed technique. The input for the same are *UserID*, *VehicleID* and *BankID* as  $x_1$ ,  $x_2$  and  $x_1 \times x_2$ , respectively. The metrics are implemented on *AIMSUN* simulator which provides the Traffic Network Interface Environment [154]. The simulation against metrics is beneficial for evaluating the network performance and to compare it with *Newton* and *Quasi-Newton* methods.

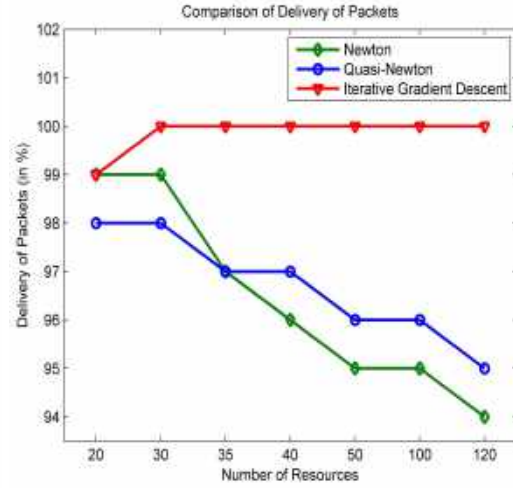
**Energy Consumption:** It is the amount of energy consumed during transmission of resource among end terminals and is measured in Joules per second (J/s).

$$E_{con} = (N_H + 1)(z_r) \times E_R + N_R \times E_{Nd} \quad (4.45)$$

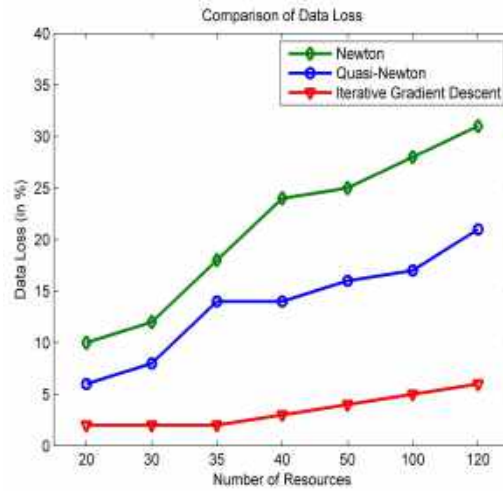
where,  $N_H$ ,  $E_R$ ,  $E_{Nd}$  are the total number of hops among terminals, energy consumed by network and energy consumption by terminals, respectively. It is observed from graph (see Figure 4.9 (a)) that *IGD* consumed less energy than *Newton* and *Quasi-Newton* methods.



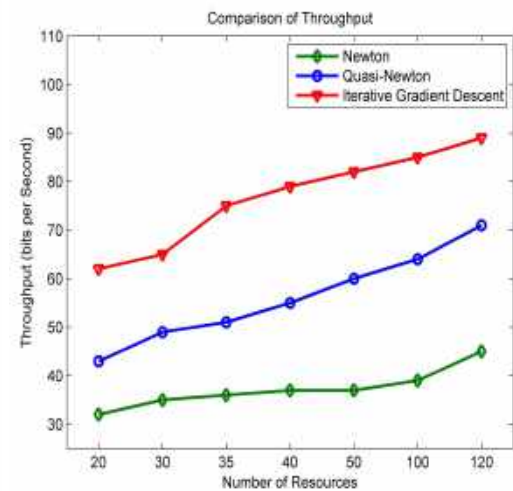
(a)



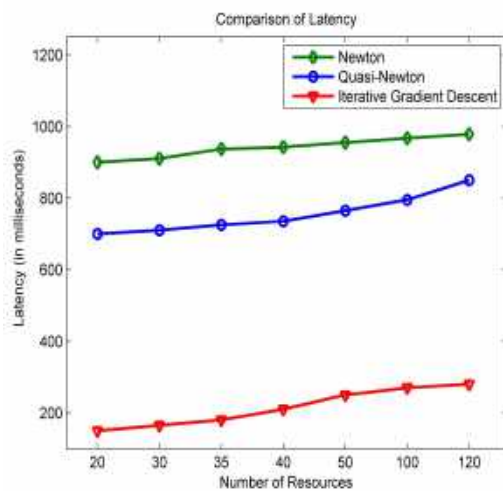
(b)



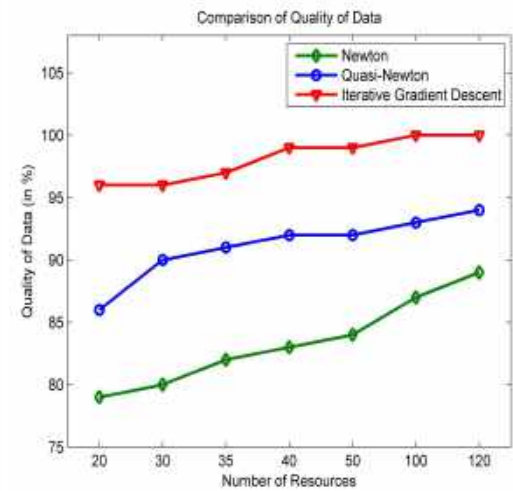
(c)



(d)



(e)



(f)

Figure 4.9: Performance Metrics

**Delivery of packet:** It is defined as the ratio of delivered packets to the send packets. More is the delivery of packet, better will be the performance of the system. It is measured in percentage of total delivered packets, *i.e.*,

$$DL_{pkt} = \frac{\sum_{i=1}^n K_{RD}}{\sum_{i=1}^n K_S} \quad (4.46)$$

where,  $DL_{pkt}$ ,  $K_{RD}$  and  $K_S$  presents the total delivered packets, total packets received at destination and total packet sent by the source, respectively. Figure 4.9 (b) illustrates that Quasi-Newton least delivery, Newton provides less delivery rate whereas *IGD* delivers maximum packets.

**Data Loss:** It is the rate of loss that occurs when packet fails to reach the destination due to traffic congestion on network and is measured in percentage. The system performance is considered to be good if it has less data loss, *i.e.*, it is antonym of  $DL_{pkt}$  and is given as:

$$D_{loss} = D_{gen} - D_{rec} \quad (4.47)$$

where,  $D_{loss}$ ,  $D_{gen}$  and  $D_{rec}$  are the data loss, packets generated at the source and packets received at the destination. It is found from the Figure 4.9 (c) that Newton Method provides higher data loss in comparison to the Quasi-Newton and *IGD* with increase in the number of resources.

**Throughput:** It is the defined as the total packets delivered successfully in a given period of time and is measured in bits per second. It is calculated in bandwidth ( $B$ ).

$$Throughput = \frac{D_{psize}}{Totaltime} \quad (4.48)$$

Here,  $D_{psize}$  is the total size of the data packets. As observed from Figure 4.9 (d), the throughput of *IGD* is increasing consistently with increase in the number of resources. On the other hand, Quasi-Newton has averaged and Newton has decreased throughput. Thus, *IGD* provides the best throughput for resource search.

**Latency:** Latency is the maximum time in which data is transmitted over the network. It is

measured in milliseconds and helps to find the bandwidth and speed of the data transmission.

$$Latency = \frac{D_{psize}}{Bandwidth} \quad (4.49)$$

Figure 4.9 (e) depicts that Newton Method takes more time, Quasi-Newton has moderate latency and *IGD* takes less time for delivering the packets.

**Quality Of Data:** The rate of the successful delivery of packets which are managed and processed among end terminals. Figure 4.9 (f) shows that the quality of data retrieved through *IGD* is better than Newton and Quasi-Newton methods.

#### 4.6.4 Complexity Analysis

The system performance is evaluated on the complexity measures with respect to *Newton*, *Quasi-Newton* and *IGD* methods and are detailed as under.

- (i) **Newton Method:** It considers second order derivative with Hessian Matrix ( $\nabla^2 s(z)$ ) for minimizing the constrained problem and its complexity is deduced using Eq. 4.5 as:

$$\begin{aligned} s(q) &\approx s(z) + (q-z)^T \times \nabla s(z) + \frac{1}{2} \times (q-z)^T [\nabla^2 s(z)] \times (q-z) \\ &= O(N) + \varepsilon^T \times \log_2(N) + \frac{1}{2} \times \varepsilon^T |\log_2(N)|^2 \times \varepsilon \quad (4.50) \\ &= O(N) + \log_2(N) + \frac{1}{2} |\log_2(N)|^2 = O(N) + \log_2\left(\frac{N}{2}\right) = O(N) \end{aligned}$$

where,  $O(N)$  defines the total number of iterations for the objective function,  $\log\left(\frac{N}{2}\right)$  tells the updated Hessian Matrix and  $\varepsilon = (q-z)$  calculates the distance among resources. The computed time ( $T$ ) for each trajectory ( $t_r$ ) is:

$$T(t_r) = O\left(d^2 \times \log_2\left(\log_2\left(\frac{1}{t_r}\right)\right)\right) \quad (4.51)$$

where,  $d$  is the distance among resources and  $(q-z) = \varepsilon$  is to find the minimum distance.

It is noted that *Newton* method takes more computational time for resource search. It

helps in decreasing the traffic snarls at toll collection points but is not suitable at peak time for traffic, *i.e.*, exceed up to 300. Therefore, another technique, *Quasi-Newton Method*: is suggested.

- (ii) ***Quasi-Newton Method***: It eliminates the limitation of higher time complexity of *Newton Method* by considering the single order derivative with Hessian Matrix,  $M_H$ . Therefore, its complexity is derived as:

$$s(z) = \frac{1}{2}z^T \delta s(z) \times z - m^T z = \frac{1}{2}O(N) \times O(N) \times O(N) - O(N) = O(N^3) - O(N) = O(N^3) \quad (4.52)$$

where,  $O(N^3)$  defines the total number to update the search set for minimizing the objective function. The time taken for decreasing each step size in descent direction is:

$$T(d_{s_r}) = -Q_r \times k_{r-1} = O(N_s \times \log_2(N_r)) \quad (4.53)$$

where,  $N_s$  is the optimized set and  $N_r$  is the decrease in the gradient of search set. It outperforms better than *Newton* method for resource discovery but its computational complexity is high due to its adaptive filtering and has partial derivative which worsen the process of minimizing the search set. Therefore, the proposed technique, *IGD* is preferred where resources are growing faster, *i.e.* up to 500.

- (iii) ***IGD Method***: Through the usage of starting point  $g_0$  and gradient descent  $G_D = -\nabla s(z)$ , *IGD* calculates the resource sets with linear descent direction to minimize the constrained problem. In such case, the complexity for *IGD* is computed as:

$$\frac{d}{d\delta} |_{\delta=0} s(g_0 + \delta \times G_D) = s(g_0 - \delta \times \nabla s(z)) = s(O(N) - \delta \times O(\log_2 N)), \quad (4.54)$$

In order to optimize the efficiency of *IGD*, it is collaborated with *Armijo's Rule* (see Eq. 4.32) and this equation with respect to complexity is formulated as:

$$\begin{aligned} s(g_{0_{r-1}} - \vartheta \delta_{t_r} \times \nabla s(g_{0_{r-1}})) &\geq s(g_{0_{r-1}}) - \epsilon \vartheta \delta_{t_r} | \nabla s(g_{0_{r-1}}) |^2 \\ &= O(N) - \epsilon^T \times \vartheta \times O(N) \times | \log_2(N) |^2 \quad (4.55) \\ &= O(N) - 2 \times \epsilon^T \times \vartheta \times O(N) \times \log_2(N) = O(N) - \log_2(N) = O(N) \end{aligned}$$

where,  $O(N)$  defines the total trajectories inward for minimizing the objective function and  $\log_2(N)$  tells the decrease in the number of resources with each iteration of the trajectory. To perform this, the optimized time  $T$  for each trajectory  $t_r$  is given as:

$$T_{t_r} = O\left(N_k \times \log_2\left(\frac{I}{t_r}\right)\right) \quad (4.56)$$

where,  $N_k$  is the minimized optimal set.

From the above discussion, it is concluded that *IGD* provides less complexity for minimizing the constrained problem and takes less time for resource search when compared to the Newton as well as Quasi-Newton methods.

## 4.7 Conclusion

The proposed framework performs resource discovery and selection in *IoT* and resolves the issues related with context of the gathered data, scalability and search metrics. It ease intercommunication and interaction efficiently through the schema for the knowledge representation, multi-modal approach for searching and optimization with iterative and convergence properties of Gradient Descent. The multi-modal approach uses three indices, *namely*, text-keyword, spatial-temporal and value-keyword to provide faster access to stored data as Red-Stores and output search set. The search algorithm provides concise view and best matches against query from Red-Stores. The *IGD* algorithm reduces the search complexity through Gradient Descent using Armijo's Rule which eradicates unpredictable step size. The algorithm enables the user to decrease and update the step size for finding gradient points. When compared to Newton and Quasi-Newton methods for searching neighbors with respect to time against 100 resources, it is found that *IGD* provides an optimal neighbor search. The time taken by *IGD* varies from 0.1 - 0.35 (seconds) whereas Newton and Quasi-Newton methods took 0.3 - 0.8 (seconds) and 0.2 - 0.65 (seconds), respectively. Thus, *IGD* is better and optimal neighbor search technique having less computational complexity. In future, *CASOF-IoT* could be extended to incorporate interoperability challenge and to discover resources continuously. Also, it can be implemented to trace resource's localization position for optimal communication.

# Semantic based Resource Discovery and Selection in the Internet-of-Things

---

## 5.1 Introduction

The framework in Chapter 4 has addressed issues such as context of the gathered data, scalability and search metrics on *IoT* but has not incorporated the interoperability challenge at various levels like semantic, radio access and context with respect to resource discovery<sup>1</sup>. Further, to discover resources continuously and tracing its localization position in a network, there is a need to metadata and semantic tagging of information. To meet this challenge, an Intelligent Resource Inquisition Framework on Internet-of-Things (*IRIF-IoT*) has been proposed. It solves the challenge through its three layers, *namely*, perception, discovery, and application. Its main features are to link resources through usage of semantic description and ontology, to discover them with Semantic Matchmaking Engine using Bipartite Graph (*SMEBG*) and to access information *via* web terminal for users.

---

<sup>1</sup> The contents are published as M. Bharti, S. Saxena, R. Kumar, “Intelligent Resource Inquisition Framework on Internet-of-Things ”, *Computers and Electrical Engineering*, Vol. 58, pp. 265–281, 2017. SCI Indexed (IF: 2.189)

## 5.2 Semantics and Internet-of-Things

*IoT* envision integration among various disciplines such as healthcare, telecommunication, agriculture, semantic web, *etc.* which raise interoperability as a key challenge due to the heterogeneity of the resources. Moreover, the volume, velocity and volatility of the data generated by highly distributed and heterogeneous resources add complexity to the interoperability challenge. This implies that providing interoperability among the interconnected resources is the prerequisite to support addressing, information representation, storage, and exchange. To address the challenges, semantic technologies are the promising solution that describe and store resources to infer new knowledge. The following scenarios demonstrate the importance of semantics to the research and development of *IoT*.

***Semantics for interoperability:*** It target machine-interpretable descriptions of data, originating source, provider, its relation with surroundings and attributes with respect to quality, technical, and non-technical.

***IoT data integration:*** Semantic integration helps to enable interoperability among various resources across different networks, though, mapping between various semantic models is needed to support data integration.

***IoT Data Abstraction and Access:*** Data abstraction is provided by semantic reasoning at various levels of knowledge representation and management. Data access in *IoT* is executed at various levels through low-level programming languages and operating systems. However, heterogeneity of the resources makes data access across the networks is a challenging task.

***Resource Discovery:*** Discovery is the core functionality for implementing the *IoT*' vision as it plays an important role for searching resources. It is supported by semantic annotation of the dynamic resources with different attributes and functionalities.

***Semantic reasoning and interpretation:*** The knowledge representation formalism used in the suite of Semantic Web Technologies allows logical reasoning that is able to infer new information or knowledge from existing assertions and rules. Semantic reasoning is an important instrument in the domain of *IoT* for various purposes such as resource discovery, data abstraction, and knowledge extraction. It is important to note that providing semantic

descriptions alone does not provide semantic interoperability and will not resolve all the issues regarding discovery, management of data, and supporting autonomous interactions. The semantic description still needs to be shared, processed, and interpreted by various methods and services across different domains.

### 5.3 Ontology

*IoT* uses intelligent components such as sensors, actuators, robots, *etc*, that require a vocabulary with meaning to describe the gathered data. The data could be either time or location dependent. Also, it could be noisy and its quality may vary depending upon its type, *i.e.*, continuous streaming data. Therefore, it raise issues with respect to its management for handling device, command and control. To address the concern, a mechanism to define vocabulary for the data with its meaning is generally referred as Ontology. Ontology is the meta-knowledge that describes everything known about problem domain. To model Ontology, the major challenges are due to heterogeneity, multi-modality and volume of data. To eliminate the challenges, techniques are required which use the meaning and information about the context of request to semantically match it with the meaning of the offered services. Here, service requesters and providers utilize ontology to discover similarity between two concepts or services and determine semantic distance between concepts. Some of the semantic matchmaking techniques are discussed below.

**Genetic Algorithm:** The basic *GA* is a four step process which includes initial random population, crossover, mutation and selection through fitness function, *say*, Poisson equation. Here, crossover helps in exploration of the datasets whereas mutation does exploitation using fitness function.

**Fuzzy Logic:** Fuzzy Logic introduces the concept of partial truth, where the truth value may range between completely true, *1* and completely false, *0*. It works with membership values in a way that mimics Boolean logic. To this end, replacements for basic operators *AND*, *OR*, *NOT* must be available.

For TRUE/*1* and FALSE/*0*, the fuzzy expressions produce the same result as the Boolean expressions and are given in Table 5.1.

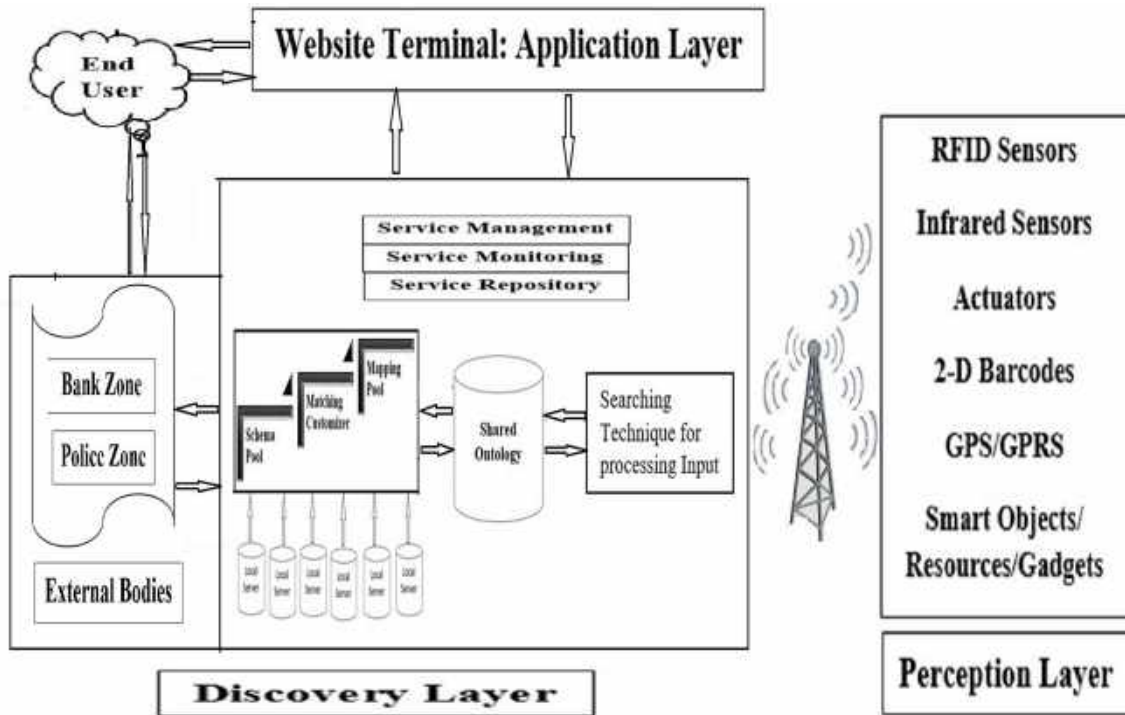
**Table 5.1:** Fuzzy Operations

<b>Boolean</b>	<b>Fuzzy</b>
AND(x,y)	MIN(x,y)
OR(x,y)	MAX(x,y)
NOT(x)	$1 - x$

**Bipartite Graph:** A bipartite graph, also called a bi-graph, comes under Conceptual Graph which is a logical formalism that includes classes, relations, individuals and quantifiers. It is based on semantic networks but has direct translation to the language of first-order predicate logic from which it takes its semantics. Bipartite Graph display instances of concepts as rectangle and conceptual relations are displayed as ellipse. It is used to capture a relationship between two types of objects where the distinction between the types of objects is important. In bi-graph ( $G$ ), vertices' are divided into two disjoint and independent sets,  $U$  and  $V$ , such that every edge connects a vertex in  $U$  to one in  $V$  and are usually called the parts of the  $G$ . Considering  $G = (U, V, E)$  denotes a bigraph whose partition has the parts  $U$  and  $V$ , with  $E$  denoting the edges of the graph. If  $|U| = |V|$ , that is, if the two subsets have equal cardinality, then  $G$  is called a balanced bipartite graph. If all vertices are on the same side of the bipartition having the same degree, then  $G$  is called bi-regular. If a bipartite graph is not connected, it may have more than one bipartition; in this case, the  $(U, V, E)$  notation is helpful in specifying one particular bipartition that may be of importance in an application.

## **5.4 Intelligent Resource Inquisition Framework for the Internet-of-Things**

This section describes *IRIF-IoT* (see Figure 5.1), which is composed of three layers, *i.e.*, perception, discovery, and application. It facilitates the service *request* at user terminal and *provisioning* at the provider end. It eradicates the syntax limitation with the use of semantic descriptions using ontology, widening the scope of the search with the semantic matchmaking. Hence, the framework overcomes the challenges arising in resource discovery. Each layer is discussed below.



**Figure 5.1:** Intelligent Resource Inquisition Framework on Internet-of-Things (*IRIF-IoT*)

### 5.4.1 Perception Layer

This layer is responsible for collecting data from the active nodes based on application. The active nodes are sensors, actuators like *RFID*, infrared and smart objects; as resources. These resources collect and transfer data to the connected server using wireless connection.

### 5.4.2 Discovery Layer

This layer executes two processes: *mapping of local databases* and *searching process* that are discussed below.

#### 5.4.2.1 Mapping of Local Databases

The mapping process creates shared ontology by linking various databases situated at different locations. The Shared Ontology is basically the centralized storage repository wherein data is compared on the basis of few parameters, axioms, rules and constraints to give optimal results. The mapping process is divided into three steps: *firstly*, developing schema pool that contains replicas of all the related databases; *secondly*, finding out

relationships, axioms and signatures between these databases; and thirdly, having a mapping pool which is shared ontology. The formation of shared ontology is discussed as under:

The ontology  $O_n$  is described as a Conceptual Sets  $C_s$  (*individuals of rdfs:Class*), having consequent hierarchy  $H_{C_s}$ ; defining binary relation with respect to *rdfs:subClassOf*. The relation that exists between each concepts is an instance  $\check{I}$  of *rdfs:Property* and is represented as  $\check{R}$ . These relations are arranged more like hierarchy  $H_{C_s, \check{R}}$  as *rdfs:subPropertyOf*. The fillers like  $(i, j, k) \in \check{I}$  can be an entity, as *rdf:type*, for a relation  $\check{r}$  from  $\check{R}$ . In order to infer knowledge from existing ontologies, axioms  $\check{A}$  are also defined. Mathematically, ontology is expressed as [168]:

$$O_n = \{ C_s, H_{C_s}, \check{R}, H_{C_s, \check{R}}, \check{I}, \check{A} \} \quad (5.1)$$

where,  $O_n$  : ontology

$C_s$  : Conceptual Sets

$H_{C_s}$  : corresponding hierarchy *w.r.t.* subclasses

$H_{C_s, \check{R}}$  : hierarchy for relations

$\check{I}$  : instance

$\check{R}$  : relations between each concept

$\check{A}$  : axioms to infer knowledge

Since, the knowledge representation works on the basis of number of ontologies that need to be mapped properly such that correct mapped schema is prepared, hence, a partial mapping function is defined to provide correct mapping process. Ontology Partial Mapping Function *ParMap* [76] is defined as mapping of all terms  $\check{t} \in \nu$  on the basis of set of all possible ontologies  $O_n$  and Vocabulary  $\nu$ . It is expressed as:

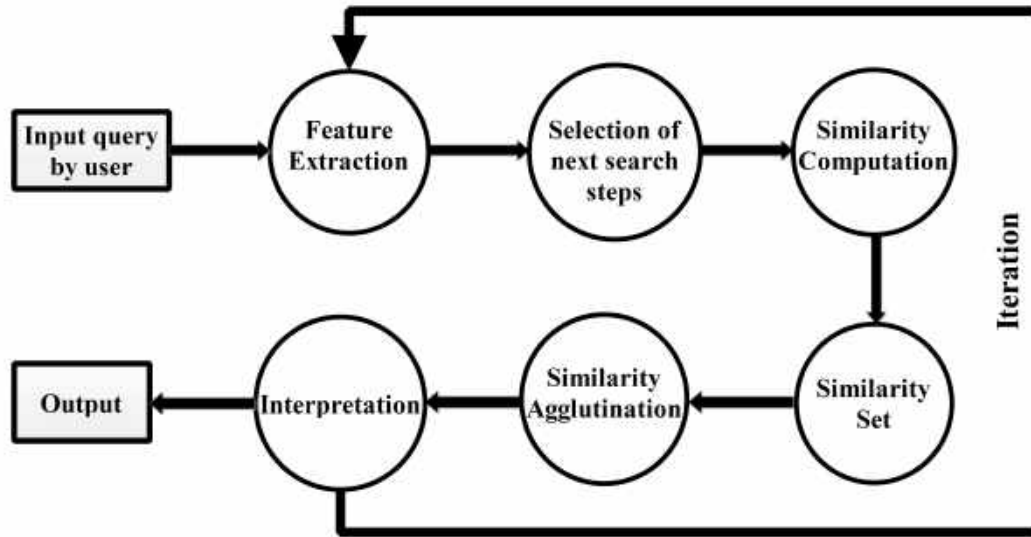
$$ParMap : \nu * O_{n1} * O_{n2} \longrightarrow \nu \quad (5.2)$$

$$\forall \check{t} \in O_{n1} (\exists k \in O_{n2} : ParMap(\check{t}, O_{n1}, O_{n2}) = k \vee ParMap(\check{t}, O_{n1}, O_{n2}) = \perp) \quad (5.3)$$

$O_n$  has term  $\check{t}$ , which is a concept/relation/instance; denoting context of the writing. The mapping of  $(\check{t}, O_{n1}, O_{n2})$ , is written as  $ParMap_{O_{n1}, O_{n2}}(\check{t})$ . The relation for  $ParMap_{O_{n1}, O_{n2}}$  is defined by

$$ParMap_{O_{n1}, O_{n2}}(\check{t}, k) \Leftrightarrow ParMap_{O_{n1}, O_{n2}}(\check{t}, k) = k \quad (5.4)$$

giving partial mapped function as  $ParMap(\check{i}) = k$  and  $ParMap(\check{i}, k)$ . The partial mapping between  $O_{n1}, O_{n2}$  is developed iff  $ParMap(\check{i}, k)$ . The pair entities  $(\check{i}, k)$  which are not mapped and appropriate mapping criteria needed to be tested. Thus, the efficient mechanism for defining  $ParMap(O_{n1}, O_{n2})$  includes six steps that are shown in Figure 5.2.



**Figure 5.2:** Sequence Steps for Partial Mapping

**Feature Extraction:** The structure of ontologies undergoes complex transformations to produce an initial digestible representation on the basis of Resource Description Frameworks *RDFS* primitives subset such that mapping process can be carried out. For the purpose, the key focus is on the features of the different ontologies, which can be either identical or non-identical. Hence, feature extraction needs to be handled by expert understanding the encoding knowledge in ontologies. Some of the parameters used for feature extraction are identifiers, relation between classes and subclasses, features like basic or derived, Web Ontology Language (*OWL*) and domain specific.

**Selection of Next Search Steps:** This similarity of restricted subsets is computed from extracted features of ontology mappings.

**Similarity Computation:** As the input given in Feature Extraction section, the features are needed to process for similarity computation  $S_c$ . This implies that similarity measures

function for comparing of two ontologies [91], *i.e.*:

$$S_c = \mathbf{v} * \mathbf{v} * O_{n1} * O_{n2} \rightarrow [0, 1] \quad (5.5)$$

where,

$$S_c(\check{i}, k) = 1 \Leftrightarrow \check{i} = k : \text{let say, two objects are similar} \quad (5.6)$$

$$S_c(\check{i}, k) = 0 \Leftrightarrow \check{i} \neq k : \text{let say, two objects are not similar and are disjoint in terms of features} \quad (5.7)$$

$$S_c(\check{i}, k) = S_c(k, \check{i}) : \text{symmetric} \quad (5.8)$$

$$S_c(\check{i}, \check{i}) = S_c(k, k) = 1 : \text{reflexive, two objects having maximum similarity and they coincide} \quad (5.9)$$

$$S_c \text{ is inversely proportional to distance} \quad (5.10)$$

The similarity measures required to find out the similarities between two ontologies at iteration ( $t$ ) with two concepts  $G, H$  from which  $g, h$  are retrieved and is given below.

- *Entity Equatability*: Entity Equatability  $S_{cEntity}(g, y)$  is based on existing logical assertions.

$$S_{cEntity}(g, y) = \begin{cases} 1, & \text{ParMap}_{t-1}(g) = y \\ 0, & \text{otherwise} \end{cases} \quad (5.11)$$

- *Definite Equatability*: Definite Equatability  $S_{cDef}(g, y)$  verifies whether a logic assertion forces two entities to be equal or not.

$$S_{cDef}(g, y) = \begin{cases} 1, & \exists \text{ statement } (g, \text{"sameAs"}, y) \\ 0, & \text{otherwise} \end{cases} \quad (5.12)$$

- *String Equatability*: String Equatability  $S_{cStrEqu}(j, e)$  performs the comparison of two or more strings  $j, e$ .

$$S_{cStrEqu}(j, e) = \begin{cases} 1, & j.char(r) = e.char(r) \forall r \in [0, |j|] \text{ with } |j| = |e| \\ 0, & \text{otherwise} \end{cases} \quad (5.13)$$

- *Dice Coefficient*: Dice Coefficient  $S_{cDice}(T, K)$  compares two sets of entities [169].

$$S_{cDice}(T, K) = \frac{|z \in (T \cap K)|}{|z \in (T \cup K)|} \quad (5.14)$$

**Similarity Set:** To determine the extent of similarity between two sets of entities, their features are identified. As each entity set has different features, therefore, creating a vector to represent all sets of individuals are difficult. To rectify this problem, multidimensional scaling technique is used where each entity is described using a vector to represent similar features of entities present in two sets. It is assumed that there is similar distance between entities, and can be achieved easily as it relies on other measures having computation of similarity values, say,  $[0, 1]$ . Now, for both sets, a representative vector is created to determine average vector of all individuals. At last, cosine between two sets of vector  $S_{cSet}(T, K)$  is determined with scalar product as similarity value.

$$S_{cSet}(T, K) = \frac{\sum_{i \in T} i \cdot \sum_{k \in K} k}{|T| \cdot |K|} \quad (5.15)$$

**Identity Agglutination:** Identity Agglutination  $S_{cAgg}(i, k)$  is the sum total of all the input given by as similarity computation output and is calculated by:

$$S_{cAgg}(i, k) = \frac{\sum_{k=1..l} w_s \cdot adj(S_c(i, k))}{\sum_{k=1..l} w_s} \quad (5.16)$$

**Interpretation:** The mapping values derived from  $S_c$  are assigned on the basis of semantic matching of the  $C_s$ . The query as the input from the client and the output is a Matching Set  $M_s$ , according to the Degree of Match  $D_m$ . The algorithm iterates in its repository in order to determine a match between input and output. The matching of output is defined as:

$$\forall C_s \in Q_{out}, \exists D_m \in M_{sout} \text{ s.t } match(C_s, D_m) \neq Failure \quad (5.17)$$

Let  $Q_{in}, M_s$  be the list of input concepts, then matching of inputs is calculated by:

$$\forall C_s \in M_{sin} \text{ and } \exists D_m \in Q_{in}, \text{ s.t. } match(C_s, D_m) \neq Failure \quad (5.18)$$

Thus, the match  $C_s$  and  $D_m$  returns the degree of match between two concepts. For conceptual sets,  $q_{out} \in Q_{out}$  and  $c_{out} \in C_s$  and  $d_{out} \in D_m$  the match function is described as shown in Table 5.2.

**Table 5.2:** Degree of Match

Condition	Match ( $q_{out}, c_{out}, d_{out}$ )
$c_{out}, d_{out}$ equivalent to $q_{out}$	Correct
$c_{out}, d_{out}$ SuperClass of $q_{out}$	Correct
$c_{out}, d_{out}$ Subsumes $q_{out}$	Segment
$c_{out}, d_{out}$ Subclass $q_{out}$	ContainIn
None of the above	Failure

The degrees of match are ranked as: Correct < Segment < Contain In < Failure where  $m < n$  indicates that  $m$  is ranked higher than  $n$ .

**Iteration:** The algorithmic process undergoes a number of iterations in order to bootstrap the structural knowledge or it may stop when no mappings are needed.

#### 5.4.2.2 Searching Process

At this layer, resource discovery is done by *SMEBG*. It has components such as crawling, reasoning and matchmaking as shown in Figure 5.3. Firstly, it takes input query as  $M_s$  (input from perception layer) to retrieve large set of data from the shared ontology, as the set of crawling. Secondly, the consolidate component finds the best possible similar identifiers and prioritize it. Thirdly, it uses Bipartite Graphs (*BG*) for ranking and optimized semantic matchmaking where ranking is done on the basis of indexing of the crawled data in vertices lists, say,  $List_i$  and  $List_j$ . These lists are semantically matched with the searched input query. The output is stored as the combined result of indexing and matching set; as updated edge

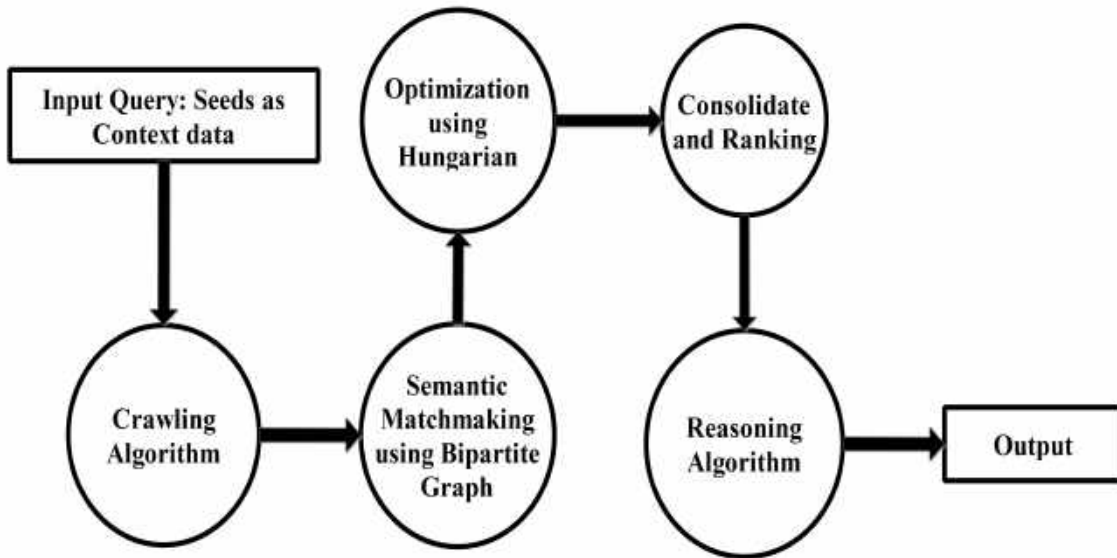


Figure 5.3: Searching Process

in a graph. The reasoning method is done by providing Axiomatic triplets on the basis of matched edge from  $BG$ , and finally, matching those triplets using  $S_c$  with the defined roles of input query. The matched set on the basis of rules and triplets are binded and  $IDs$  are confirmed; output is displayed on screen and it opens the process of payment gateway. The entire process is discussed in detail step by step.

**Crawling Algorithm:** In this algorithm, there is a queue named *frontal* in which Entities  $E$  are put on. From the *frontal* queue, a new queue is made of  $C_s$ , having similar  $M_s$ . For this purpose, a counter *ctr* is set and the process goes till the match is found. At this stage,  $C_s$  is matched on the basis of  $S_c$  and the process will continue to prioritize the computations. A timer is started for both  $C_s$  and  $S_c$ . If the randomized probability of time-sets is between 0 and 1, it picks and prints the matched set. For those  $M_s$  which have not been matched using  $S_c$  are considered as unseen/blacklisted sets ( $M_{s_{defer}}$ ) and are put in *frontal*, and the process is repeated with updated time-sets of new queues. At last, the process calculates the total time in finding the results and performs the output according to the Algorithm 5.1.

**Semantic Matchmaking Algorithm using Bipartite Graph:** The algorithm takes input of Entities  $E$  as vertices which are processed using  $M_s$  (see Algorithm 5.2). The entities  $E$

---

**Algorithm 5.1** Crawling Algorithm

---

**Require:** Conceptual Sets ( $C_s$ ), Entities ( $E$ ), Matching Set ( $M_s$ )

```
begin
frontal  $\leftarrow E$ 
 $C_{s_0, \dots, s_n} \leftarrow$  new queue
ctr  $\leftarrow$  new ctr
while ( $n+1 < n$ ) do
  put frontal into  $C_{s_0, \dots, s_n}$ 
  while ( $depth+1 < C_s$  &&  $S_c == 1$ ) do
    put frontal into  $C_{s_0, \dots, s_n}$ 
    for ( $i = 1$  to  $n$ ) do
      prioritise  $S_c \leftarrow (C_{s_i}, ctr)$ 
    end for
    start  $\leftarrow$  time_ctr()
    for ( $i = 0$  to  $n$ ) do
       $t_i \leftarrow$  calculate time( $S_c$ )
       $t_j \leftarrow$  calculate time( $C_{s_i}, ctr$ )
      if ( $t_i > \text{random}([0,1])$  &&  $t_j > \text{random}([0,1])$ ) then
        get  $M_s$  from  $C_s$ 
         $M_{s_{deref}} = deref(M_s)$ 
        if ( $M_{s_{deref}} = deref$ ) then
           $G_B = \text{get}(M_s)$ 
          output  $M_s$ 
           $M \leftarrow M_s \in$  in  $G_B$ 
           $\overline{M_G} \leftarrow$  prune blacklist from  $M_G$ 
          add unseen  $M_s$  in  $\overline{M_G}$  to frontal
          update ctr wrt.  $\overline{M_G}$ 
        else
          if ( $M_{s_{deref}}$  is unseen) then
            add  $M_{s_{deref}}$  to frontal
            update ctr for  $M_{s_{deref}}$ 
          end if
        end if
      end if
    end for
    elapsed  $\leftarrow$  time_ctr() - start
    if ( $elapsed < delay_{min}$ ) then
      wait( $delay_{min} - elapsed$ )
    end if
  end while
end while
end while
end
```

---

are put in graph  $G$  and search process initiates using two queues, where  $V_{x0}$  and  $V_{x1}$  denote vertices for  $List_i$  and  $List_j$ , respectively. The weights are computed for these queues as a function of the degree of match between  $p$  and  $q$ . If found *Correct*,  $M_s$  is matched and assign weight  $W_a$ ; *Segment*, if half of  $M_s$  is matching and assign weight  $W_b$ ; *ContainIn*, if some part of it is matched and assign weight  $W_c$ ; *Failure*, if  $M_s$  does not match. These weights are ranked in the following order:  $Correct < Segment < ContainIn$ . The process ends up showing the matched  $M_s$  from both ends. Since, the process generates a number of complete  $M_s$ , therefore, there is requirement for the optimizing the solution for the cases where multiple matching sets exist for an edge. In such cases, degree of match for an edge is given as maximum weighted edge, say,  $max(wt_i)$  and is assumed to be worst degree of match that needs to be minimized. In order to minimize total sum of weights of edges, Hungarian algorithm is chosen to compute the optimal matching due to its strong polynomial time bound complexity, i.e.,  $O(|V_x|^3)$  for  $V_x$  vertices [170, 171]. To assign weights to the edges, following lemma is provided which must hold true for Hungarian algorithm to find optimal matching.

*Lemma 1:* The total weights in matching sets are minimized  $\sum wt_i$  and are equivalent to sets of matching where  $max(wt_i)$  is minimized.

In the list of matching sets,  $G_B \leftarrow \{V_{x0} + V_{x1}\}E$ , various assigned weights of edges ( $W_a, W_b, W_c$ ) are calculated as under (see Table 5.3):

**Table 5.3: Weight Calculation**

Degree of match	Weight
Correct	$W_a = 1$
Segment	$W_b = (W_a \times  V_x ) + 1$
ContainIn	$W_c = (W_b \times  V_x ) + 1$

To prove the lemma, following properties need to be fulfilled, i.e.,

- (i) The matched number of edges from  $G_B = |V_x|$ .
- (ii) The relation  $W_a < W_b < W_c$  holds true, where  $W_a$  has only one complete matched edge;  $W_b$  gives few of the matching edges and  $W_c$  is providing more similar matched edges.
- (iii) By computing the weights, it is found that an edge with higher weight is greater than

---

**Algorithm 5.2** Semantic Matchmaking Using Bipartite Graph

---

**Require:** List of matching sets ( $G_B$ ) (input from crawling), Vertices of entities ( $V_x$ ), Entities ( $E$ )

```
begin
 $G_B \leftarrow \{V_{x0} + V_{x1}\} E$ 
 $V_{x0} \leftarrow List_i, V_{x1} \leftarrow List_j$ 
compute Weights ( $-V_x-$ )  $\leftarrow (W_a, W_b, W_c)$ 
for  $p \in M_s$  in  $V_{x0}$  do
  for  $q \in M_s$  in  $V_{x1}$  do
    frontal = match (p, q)
    if frontal  $\neq$  Failure then
      add connecting edge of (p, q) in  $G_B$ 
    end if
    if frontal = Correct then
       $W_a = W_c(p, q)$ 
    end if
    if frontal = Segment then
       $W_b = W_s(p, q)$ 
    end if
    if frontal = ContainIn then
       $W_c = W_{CI}(p, q)$ 
    end if
  end for
end for
end
```

---

$V_{x0}$  edges of lower weights *i.e.*  $wt_i > wt_j \times |V_{x0}|, \forall_i > j$ .

*Proof:* In graph  $G_B$ ,  $M_s$  are the sets in which  $\sum wt_i$  is minimum. Consider two matching sets ( $M_{s1}, M_{s2}$ ) having edges as  $(a_1, a_2, a_3, \dots, a_m)$  and  $(b_1, b_2, b_3, \dots, b_n)$ , respectively. Both sets have maximum weighted edge, *i.e.*,  $\max(e)$ . If *Lemma 1* does not hold true, it implies that  $M_{s1} \neq M_{s2}$  and no weighted edge is minimum. Therefore, it concludes atleast one maximum edge in  $M_{s1}$  and  $M_{s2}$ , thus, contradicting the lemma, if  $\sigma(wt(a_i)) > \sigma(wt(b_i))$ . Hence, it is found that  $M_{s1}$  and  $M_{s2}$  will be equivalent if weights of matched edges are assigned according to Table 5.3. The entire process works as explained in Algorithm 5.3. Here, the crawling and *SMEBG* algorithms are invoked. It takes  $D_m, G_B$  as input and provides optimal matching *via* Hungarian match  $H_M$ . It uses Hungarian for computing optimal complete matching of input query from  $M_s$ . Here, the matching is processed twice, *i.e.*, *firstly* in *SMEBG* and *secondly*,

---

**Algorithm 5.3** Optimized *SMEBG* Algorithm

---

**Require:** Hungarian match ( $H_M$ ), Bipartite graph ( $G_B$ ), Degree of match ( $D_m$ )

---

*begin**Call* procedure as Crawling // calling Algorithm 1*Call* procedure as Semantic matching using bipartite graph // calling Algorithm 2Graph  $H_M$  = Hungarian-matchmaking ( $G_B$ )**if**  $H_M == \text{null}$  **then** $M_s \leftarrow \text{empty}$ 

Return Failure

**end if**Let  $(p, q) = \max(\text{wedge})$  in  $G_B$  $D_m \leftarrow \text{match}(p, q)$ Return  $D_m$ *end*

---

in Hungarian. However, the input query and matching sets are swapped. Finally,  $D_m$  is returned.

**Reasoning Algorithm:** As  $M_s$  are matched and are put in the list having Axiomatic triplets. This process is used for semantic matchmaking and ranking. For each entity and matching Axiomatic Triplets, the process starts to put entities in frontal queue. In this queue, the rules are applied in order to have the best and accurate  $M_s$ . For each rule applied, a new triplet is generated with valid terms. The search is successful and the matching id is stored in the frontal (*see* Algorithm 5.4). From this process, the  $M_s$  is confirmed and it undergoes further processing of payments/deductions, *etc.*

### 5.4.3 Application Layer

It is the web portal or terminal embedded with *RFID* technology that provide services to the user for enabling direct toll payments from moving vehicles. These web terminal comes with login facility, logs of transactions and details of all crossing of tolls. Moreover, there is a provision for third party linking with users' bank accounts for automatic deductions by virtue of which waiting time at tolls could be minimized.

---

**Algorithm 5.4** Reasoning Algorithm

---

**Require:** Input File ( $E$ ), Rules ( $R_c$ ), Axiomatic Triples  $A_{xi}$ 

```
begin
 $A_{xi} \leftarrow$  Axiomatic Triples
output( $A_{xi}$ )
output( $S_c$ )  $\leftarrow$  Similarity Computation
frontal  $\leftarrow$  {}
for  $\check{t} \in E \cup A_{xi}$  do
  for  $r \in R_c$  do
    if  $r.\check{A}$  needs  $\check{t}$  then
      frontal  $\leftarrow$  frontal  $\cup E$ 
    end if
  end for
end for
 $S_c^{frontal} \leftarrow$  {  $r \in R_c \mid r.\check{A} \neq \phi, r.\check{I} \neq \phi$  }
 $frontal^{new} \leftarrow$  apply Rules (frontal,  $S_c^{frontal}$ )
output  $frontal^{new}$ 
frontal  $\leftarrow$  frontal  $\cup frontal^{new}$ 
 $R^E \leftarrow$   $r \in R \mid r.\check{I} \neq \phi$ 
for  $\check{t} \in E \cup (A_{xi}) \cup frontal^{new}$  do
   $E_{\check{t}} \leftarrow$   $\check{t}$ 
  for (new triple  $\check{t}_n \in E_{\check{t}}$ ) do
    for  $r \in R^E$  do
      if  $\exists$  binding  $b_E \mid b_E(r.\check{I}) = \check{t}_n$  then
        for  $\forall b_{frontal} \mid b_{frontal}(b_E(r.\check{A})) \subseteq$  frontal do
           $E_{\check{t}} \in E_{\check{t}} \cup b_{frontal}(b_E(r.C.on))$ 
        end for
      end if
    end for
  end for
  end for
  output ( $E_{\check{t}}, \check{t}$ )
end for
frontal  $\leftarrow$  match (p, q)
return frontal
end
```

---

## 5.5 Experimental Results and Discussion

### 5.5.1 Framework Analysis

The various layers of *IRIF-IoT* framework is implemented on *ETC*. The working through its each layer is explained below.

**Perception Layer:** This layer is responsible for gathering data from the active devices. Here, *RFID* tags are embedded on the polls which scan the details from the embarked *RFID* tags on vehicles. Each *RFID* tag carries basic information related to vehicle like the reader's location, date and time of vehicle arrival. The scanner works and collects vehicle details. Once, these details are scanned, hamper gets open for the next step, wherein, *RFID* card reader is checked on check post for authorization. The card reader fetches the information of end user and forward it to the connected computer *via* wireless connection.

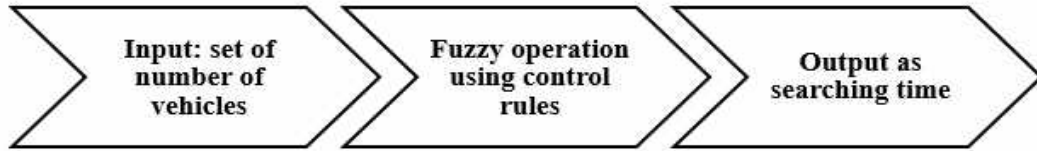
**Discovery Layer:** This layer is responsible for resource discovery through the network. On the basis of search time, the various discovery techniques *FCL*, *GA* and *SMEBG* are analyzed and compared on the basis of datasets having 100 resources, *say*, vehicle; entering at *ETC*, which is validated and authenticated on the basis of  $M_s$ . This dataset is the combination of vehicle numbers and card numbers which is searched through the discovery techniques of *FCL*, *GA* and proposed method *SMEBG*. These are discussed below.

- **Fuzzy Control Logic (FCL):** The input to the *FCL* is given by the active devices of perception layer in *IRIF-IoT* framework. In general, at *ETC*, the *RFID* readers have combined details of vehicle and card (end users). The input is then processed using control rules under Fuzzy operation to generate fuzzy value as shown in Table 5.4. Here, output is calculated as total search time for processing the input. Finally, accurate value is generated by fuzzy estimation. As an input, three sets are required: i) comprised set of number of vehicles, ii) fuzzy control rules set and iii) set of search time (*see* Figure 5.4). The input, as set of number of vehicles helps to generate a fuzzy value, *e.g.*, value '1' is transformed into "very few", while '10' is mapped to "too many".

This input then undergoes fuzzy operation where control logic represent machine

**Table 5.4:** Fuzzy Control Rules

Condition	Match (Q, I)
$Q_{in}$ : if $M_s == S_c$	$Q_{out}$ : 1 (True)
$Q_{in}$ : if $M_s \leq S_c$	$Q_{out}$ : 1 (True)
$Q_{in}$ : if $M_s < S_c$	$Q_{out}$ : 0/1
$Q_{in}$ : if $M_s \geq S_c$	$Q_{out}$ : 1/0
$Q_{in}$ : if $M_s > S_c$	$Q_{out}$ : 0/1
$Q_{in}$ : if $M_s \neq S_c$	$Q_{out}$ : 0



**Figure 5.4:** Fuzzy Operation

reasoning process and generates one output for every two inputs, *e.g.*, if the congestion state of present and successive phase are “very few” and “many”, then it generates output as “short”. The set of searching time is used to transform the fuzzy value of delay time into accurate value, for example “short” refers to ‘*I*’s approximately. The various rules for finding the query set  $Q_s$  using fuzzy control rules. The rule set is taken on  $M_s$  according to the asked query, *i.e.*, Q:  $Q_{in}$  for input and  $Q_{out}$  for output, where  $M_s$  is matched according to  $S_c$  schema described in section 5.4.2. Here, searching time is calculated for every given two inputs according to fuzzy operation and it is processed for all possible inputs, to avoid repetition in computations as system is initialized. Finally, query sets are derived from other sets and search time is calculated.

- **Genetic Algorithm** The working of *GA* is explained as:
  - Initially, the random population is generated, calculated and encapsulated in array list. Out of the population, one gene with its membership values is specified within a range and rest are generated randomly.
  - Here, calculation for the  $M_s$  of searched queries, set of searching and fuzzy rules are used as a complete set to model random generation using Poisson process. This helps in calculating the searching time and managing traffic in more optimal way.
  - The evolution procedure is carried with crossover and mutation. In former

process, operator chooses parent genes using probability set (80%) and generates two child from the parents, randomly. Whereas in mutation, operator exploit genes, each with representative chromosome, at random points using mutation probability set (20%). Here, cost function is considered as average searching time of each  $M_s$ .

- The genes are sorted using fitness function, Poisson process and then genes with less average time are chosen and forwarded to the next generation.

The Poisson distribution with searching time  $T^*$  is defined as Eq. 5.19 [172]:

$$P_{T^*}(n) = \frac{\lambda^n}{n!} e^{-\lambda}, n > 0 \quad (5.19)$$

assuming  $n$  is number of vehicles,  $\lambda$  is the total time which tells the searching number of vehicles. The process is simulated for the Poisson function. Let,

$$\lambda = \beta.T^* \quad (5.20)$$

here,  $\beta$  is the approximate searching time which is less than  $1s$ . When  $n=1$  and keeping  $T$  as a variable;  $P_1(T^*)$  is probability density function of interval time between any two successive searches.

$$P_1(T^*) = \beta.T^* . e^{-\beta T^*}, T^* > 0 \quad (5.21)$$

or

$$P_1(\lambda) = \lambda . e^{-\lambda}, \lambda > 0, \lambda = \beta.T^* \quad (5.22)$$

The probability density function  $F^*(\lambda)$  which is monotonic increasing and positive definite, as defined above is integrated and given as Algorithm 5.5:

$$F^*(\lambda) = (-\lambda - 1) e^{-\lambda} + 1, \lambda > 0 \quad (5.23)$$

- **Semantic Matchmaking Engine using Bipartite Graph** The process and algorithms are discussed in detail in Section 5.4.2 and results are generated. The Input seeds as datasets undergoes the Crawling Algorithm (*see* Algorithm 5.1) for the retrieval

---

**Algorithm 5.5** Poisson Searching Algorithm

---

**Input:** Number of vehicles ( $V_h$ )**Output:** Search time set ( $S_T$ ),  $s \in S_T$  where each  $v \in V_h$ *begin* $k \leftarrow 1$ **for** ( $k < V_h$ ) **do**

Prob = random(0.0,1.0)

    let Prob = Fitness function,  $F^*(\lambda)$ , derive searching time,  $\lambda$      $s_k = \lambda / \beta$      $k++$ **end for***end*

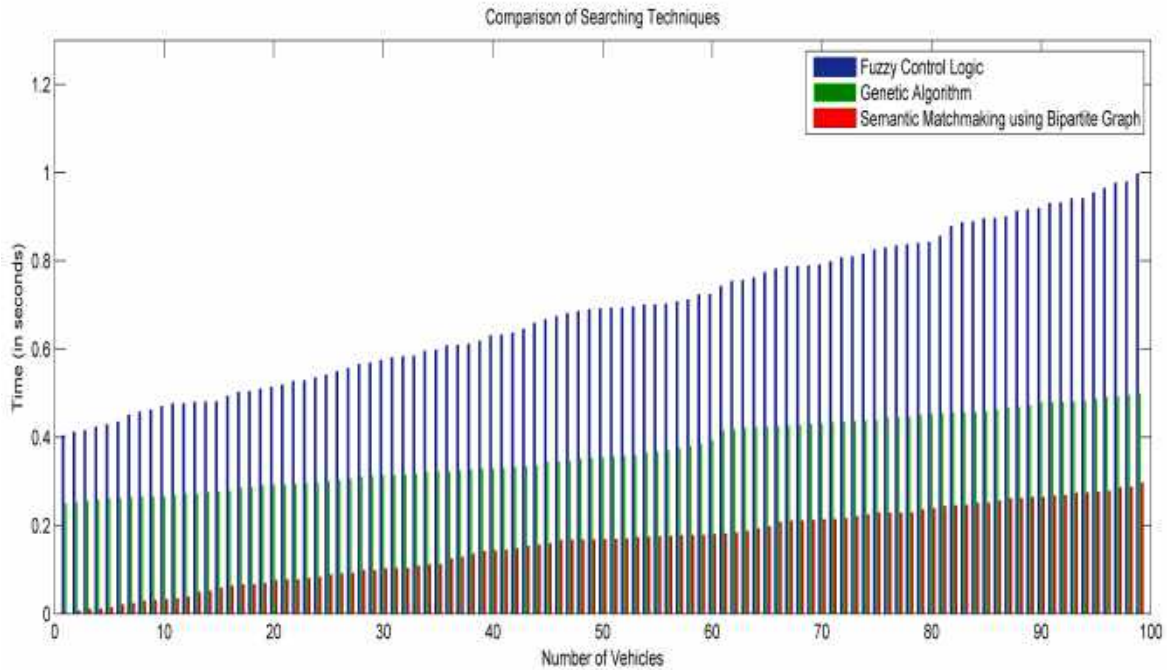
---

of large dataset from the shared ontology. These datasets are then checked using similarity computation, and prioritized using Semantic Matchmaking Engine (*see* Algorithm 5.2), where ranking is done. The optimal matching is processed using Hungarian algorithm (*see* Algorithm 5.3). Finally, using Reasoning Algorithm (*see* Algorithm 5.4), matched data sets are validated using rules and axioms.

**Application Layer:** Through this layer, end user can access his/her account online, including transactions, daily record, available amount, toll location details, *etc.*

### 5.5.2 System Performance

From the comparison (*see* Figure 5.5), system performance is analyzed for three approaches, namely, *FCL*, *GA* and *SMEBG*, respectively. In *FCL*, the time taken for the searching is less than 1 second but is not optimized to satisfaction level. This type of algorithm is not fully optimized as it leads to delay in search process. It also leads to collision at peak hours due to increase of traffic. In *GA*, a search is possible using matchmaking algorithms with Fuzzy sets and Poisson process. Here the search time is varying between 0.3 to 0.5s. Thus, it is providing far better results than fuzzy sets but there is an issue that it is still giving rise to delayed searching if the search sets would be increased to 1000. In comparison with *FCL* and *GA*, the *SMEBG* provides an optimal search taking time among 0.1 to 0.35s and there is no delay in searching with increased number of resources.

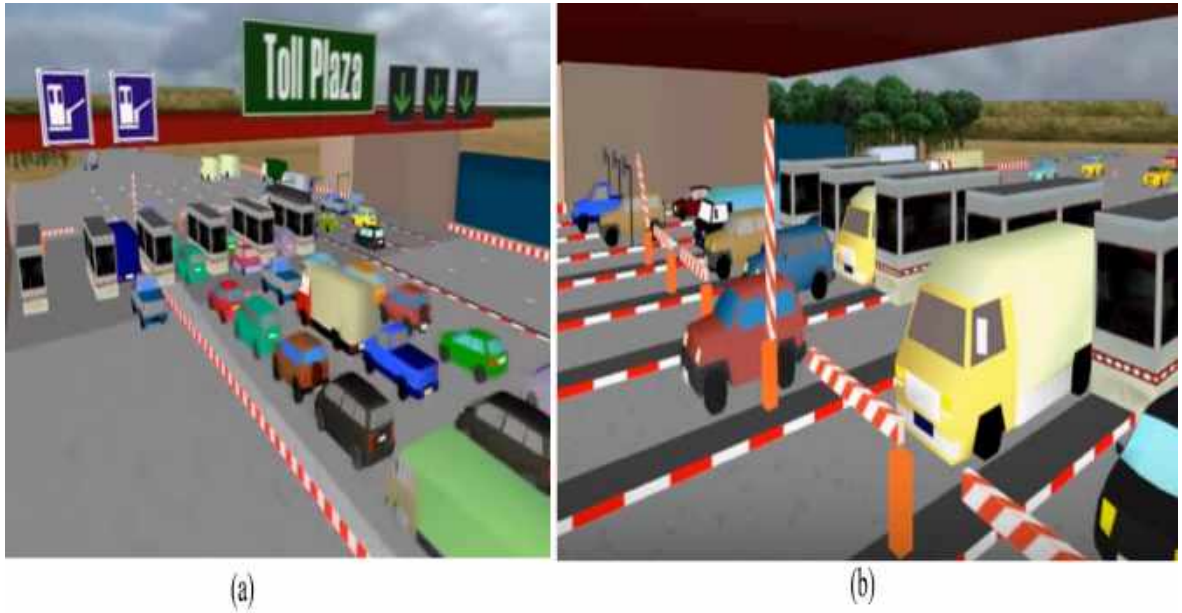


**Figure 5.5:** Comparison of Various Approaches

Thus, it is found that *SMEBG* is a better and an optimized searching technique having 1.47 and 1.57 times better search time compared with *FCL* and *GA*, respectively. Therefore, it is observed that *SMEBG* works well with the prescribed application and is indeed a fast, reliable and efficient technique for resource inquisition.

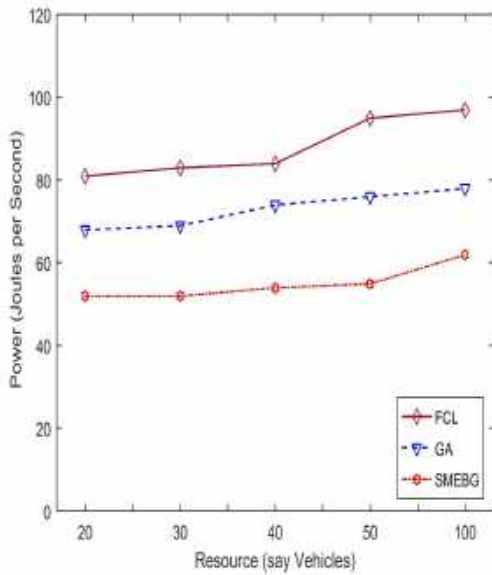
### 5.5.3 Performance Metrics

The approaches discussed above are analyzed for its efficiency and optimizability on various performance metrics like power consumption, packet delivery rate, packet loss, throughput, latency and data quality for the *ETC* application (explained in Section 4.4.1). The input parameter is resources, *i.e.*, vehicles approaching to the *ETC*. In regard to its implementation, each time *BG* has to update the information and it overcomes the limitations of performance metrics. These metrics are implemented on ‘*AIMSUN*’, that provides environment for Traffic Network Editor (*TEDI*). It is composed of a network database, Application Programming Interface (*API*) and modular (for simulation) [173]. It models each vehicle on traffic network continuously in a detailed manner like distinct types of vehicles, traffic flows, route selection, *etc.* The simulation environment view is shown in Figure 5.6.

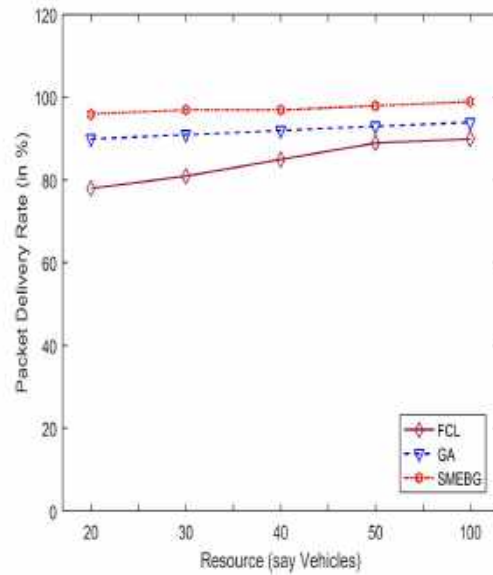


**Figure 5.6:** Simulation Environment of Toll Plaza (a) 3D Side View (b) Front View

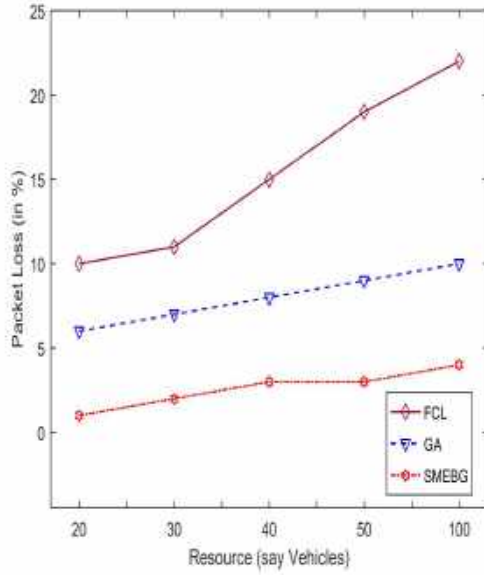
The simulation results are analyzed on the basis of various performance metrics which are helpful in assessing networks performance and evaluating it in comparison to other approaches and are discussed as below.



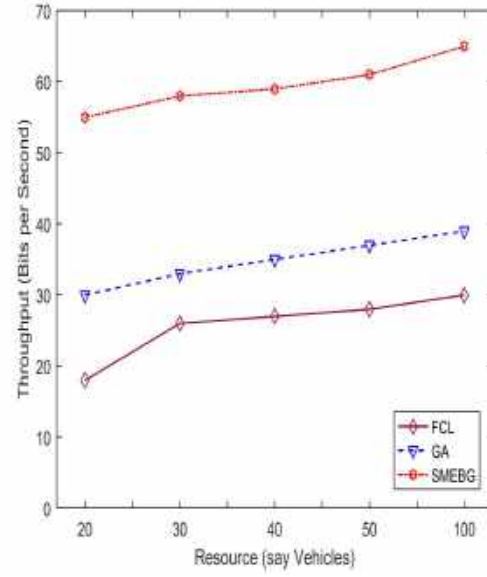
**(a)**



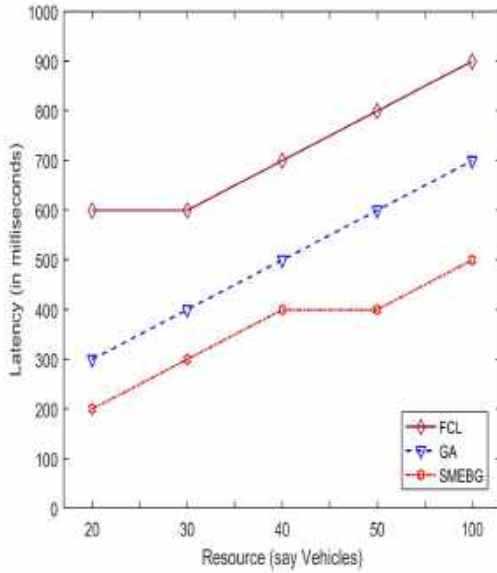
**(b)**



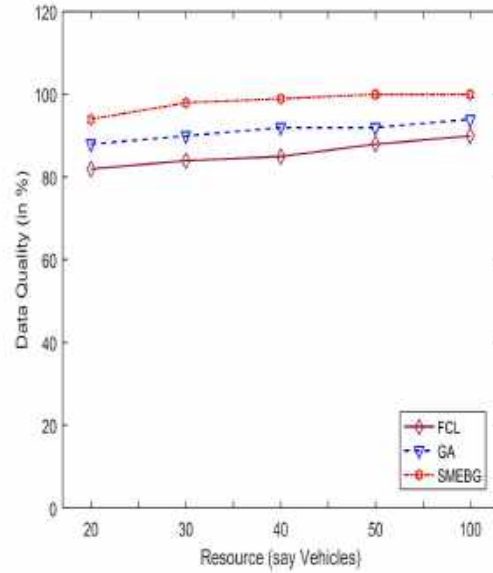
(c)



(d)



(e)



(f)

**Figure 5.7: Performance Metrics**

**Power Consumption:** It is described as the rate by which power is consumed by a resource transmission among end users and is measured in Joules per second.

$$P_{cbit} = (Y_{i,j} + 1) \times Energy_{resource} + Y_{i,j} \times Energy_{nodes} \quad (5.24)$$

where,  $P_{cbit}$  is the Power Consumption,  $Y_{i,j}$  is the number of hops among end points,

$Energy_{resource}$  is the energy consumed in network and  $Energy_{nodes}$  is the power consumption at the end points. Thus, from the Figure 5.7(a), it is observed that power consumed by *SMEBG* is very less as compared with *FCL* and *GA*. The *FCL* provides the maximum consumption whereas *GA* is showing average consumption.

**Packet Delivery Rate:** It is the ratio of the number of data packets delivered and the number of data packets send. The more the value of packet delivery ratio, better is the performance and is usually measured in percentage of total packets delivered.

$$PDR = \frac{S_1}{S_2} \quad (5.25)$$

where, *PDR* is the Packet Delivery Rate,  $S_1$  is the sum of data packet received by destination and  $S_2$  is the sum of data packets generated by the source. Therefore, Figure 5.7(b) depicts that *FCL* and *GA* provide less packet delivery whereas in comparison *SMEBG* guarantees to deliver 98% data.

**Packet Loss:** It occurs when one or more transmitted packets traveling among end users fails to reach due to network congestion and collision. The network performance is better if there is less packet loss. Its calculation is just opposite to packet delivery rate and is also measured in percentage (number of packet loss).

$$P_{loss} = P_{gen} - P_{rec} \quad (5.26)$$

where,  $P_{loss}$  is the packet loss,  $P_{gen}$  are the generated packets and  $P_{rec}$  are the received packets. It is observed from Figure 5.7(c) that there is highest degree of packet loss in *FCL* whenever number of resources is increasing and, in comparison, *GA* has average packet loss, *SMEBG* has lowest packet loss, respectively.

**Throughput:** It tells the total number of the packets that are successfully delivered within the required time. It can be calculated with network bandwidth and packet delivery rate. It is measured in bits per second (*bps*).

$$\text{Throughput} = \frac{P_{size}}{\text{total time to send packets (in bps)}} \quad (5.27)$$

where,  $P_{size}$  is the total size of packets. Figure 5.7(d) shows the comparison of throughput among the approaches *FCL*, *GA* and *SMEBG*. It is observed that the throughput of *SMEBG* is increasing consistently if the number of resources are increasing whereas *FCL* and *GA* have lowest and average throughput, respectively.

**Latency:** In two-way communication process, the latency is described as the maximum time interval in which information is transmitted among end users. It is measured as unit of time (*milli-seconds*).

$$\text{Latency} = \frac{P_{size}}{\text{network bandwidth (sec)}} \quad (5.28)$$

This constraint is more beneficial to find networks bandwidth and speed of data transmission. Therefore, Figure 5.7(e), depicts that *GA* is provided fluctuating results with increasing number of resources whereas *FCL* takes more time in delivering packets. The *SMEBG* provides data delivery in a short span of time.

**Data Quality:** The data quality tells the rate of data delivered between end locations satisfying the properties of precision, timeliness and requirements according to the situations. It is the rate of data delivered, processed and managed by end users. Thus, Figure 5.7(f), tells that quality of data retrieved by *SMEBG* is better than *FCL* and *GA*.

#### 5.5.4 Complexity Analysis

Assuming average of the total number of concepts in query and matching sets be  $Q_{in}$ ,  $Q_{out}$  and  $M_{sin}$ ,  $M_{sout}$ , respectively. Then, complexity is analyzed on the basis: (i) number of search iterations say,  $\tilde{N}$  for matching sets; (ii) weights of vertices's are computed as  $O(1)$ , and (iii) time complexity for comparing each pair of  $(p, q)$ ,  $p \in Q_{out}$  and  $q \in M_{sout}$  with operations complexity,  $O(|Q_{out}| \times |M_{sout}|)$  is  $|Q_{out}|^3$ . Therefore, time complexity for overall search is:

$$\tilde{N} \times (|Q_{out}| \times |M_{sout}| + |Q_{out}|^3) + (|M_{sin}| \times |Q_{in}| + |M_{sin}|^3) \quad (5.29)$$

Considering  $|Q_{out}| = |M_{sout}| = |Q_{in}| = |Q_{in}| = \alpha$ , where,  $\alpha$  takes smaller values and is independent of  $M_s$ . The time complexity is calculated as:

$$O(\tilde{N} \times 2 \times \alpha^2 + \alpha^3) = O(\tilde{N}) \quad (5.30)$$

The simplified complexity of *FCL* is computed using Table 5.4, where  $Q_{in}$  is the input query,  $M_{sin}$  are the matching sets,  $S_c$  is the similarity computation based on fuzzy control rules. Therefore, time complexity is given by:

$$((88 \times Q_{in}) + (37 \times M_{sin}) + 20)S_c + 6 = (\tilde{N} + \tilde{N} + 20)\tilde{N} + 6 = O(\tilde{N} \times \tilde{N}) = O(\tilde{N}^2) \quad (5.31)$$

and the complexity of *GA* using section 5.5.1 is computed as (Poisson process):

$$O(\tilde{N} \times \tilde{N} \times O(P_{T^*}(\alpha)) \times F^*(\alpha)) = O(\tilde{N}^2 \log(\tilde{N})) \quad (5.32)$$

The multiplying coefficients of *FCL* and *GA* are quite higher, thus, decreasing system performance and on the other hand, *SMEBG* improves search time by virtue system performance increases. It is observed that the search complexity of *FCL* and *GA* is higher and also it is found from section 5.5.3 that both has more packet losses, less throughput (efficiency), providing less latency with decrease in data quality. On the other hand, *SMEBG* guarantees to deliver maximum data with less packet loss, high throughput, good latency maintaining data quality. Also, the optimal matching using Hungarian is consuming less energy in *SMEBG* as compared with *FCL* and *GA*. Thus, proposed approach, *i.e.*, *SMEBG* improves system performance with effectual search time.

## 5.6 Conclusion

*IRIF-IoT* incorporated the interoperability challenge at various levels like semantic, radio access and context with respect to resource discovery and selection in *IoT*. It uses Ontology to eradicate the syntax limitation with the use of semantic descriptions, widening the

scope of the search with the semantic matchmaking. It performs an optimized searching using *SMEBG* algorithm that guarantees to deliver maximum data with less packet loss, high throughput, good latency maintaining data quality. Also, the optimal matching using Hungarian is consuming less energy in *SMEBG*. In comparison with *FCL* and *GA*, the *SMEBG* provides an optimal search taking time among *0.1 to 0.35s* and there is no delay in searching with increased number of resources. Thus, it is found that *SMEBG* is a better and an optimized searching technique having 1.47 and 1.57 times better search time compared with *FCL* and *GA*, respectively. Hence, the framework is indeed a fast, reliable and efficient technique for resource inquisition.

In future, *IRIF-IoT* could be extended for trust building among resources in terms of security and privacy. Also, it could be automated for restricted optimal decision making.



# Trust based Decision Making for Optimal Resource Selection in the Internet-of-Things

---

## 6.1 Introduction

Although, the previous chapters have addressed the various challenges to resource discovery and selection but they have not accounted the trust issue. This issue arises on a global network infrastructure like the *IoT* due to continuously changing mobility patterns, heterogeneity, interoperability and scalability on the network that imposes restriction to process an optimal decision <sup>1</sup>. The relevance of decision making on *IoT* platform with respect to solving complicated problems is to provide an optimal solution for resource selection that should concatenate two parameters, *i.e.*, related to human intelligence and computing power. The former demands some computerized decision rules to automate the process which adds intelligence to the resources, whereas, latter asks for network prototype or topology for communication. Since, resources on *IoT* have different attributes, identities, capabilities; their integration demands decision making to enable automatic inference of

---

<sup>1</sup>The contents are published as M. Bharti, S. Saxena, R. Kumar, “A Middleware Approach for Reliable Resource Selection on Internet-of-Things ”, *International Journal of Communication Systems*, 2019.(<https://doi.org/10.1002/dac.4278>). Title for framework, *i.e.*, Autonomous Decision Making Framework on Internet-of-Things (*ADMF-IoT*) is modified to Middleware Approach for Reliable Resource Selection on Internet-of-Things (*MARRS-IoT*), and Figure 6.1 Autonomous Decision Making Framework on Internet-of-Things (*ADMF-IoT*) is modified and renamed to *MARRS-IoT*, as per the revised-accepted manuscript in *International Journal of Communication Systems*, 2019. (<https://doi.org/10.1002/dac.4278>). SCI Indexed (IF : 1.278)

appropriate information such that resources become self-manageable.

The self-manageability among resources on *IoT* requires system to be either reactive or proactive. The reactive system would be required to continuously respond to changes in the user's situation whereas proactive demands to evaluate behavior based on predicted situations computed with some level of confidence and probability. Such requirement for self-manageability of resources face difficulty due to inaccurate analysis, flaws in the data that leads to distinct research challenges with respect to knowledge acquisition, discovery, trust evaluation measurement, security, communication and computation. Some of the challenges are (i) to discover appropriate devices and information, (ii) to effectively and efficiently aggregate and calculate trustworthiness of the information, (iii) to model provenance of information, (iv) to safeguard the privacy and security of resources leading to possible incorrect inferences and (v) to provide transparency among processes to assess their impact. These challenges motivate researchers to investigate for intelligent and automated approaches for decision making with changing organizations, applications, locations, type of resource, *etc.*, across *IoT*. In order to address these challenges, various parameters that need to be focused are: (i) to automate resources and access them, (ii) local and global aspects of resources' parameters, (iii) query's dimension and its scope, (iv) understanding search evaluation metrics and (v) trust-building issue among interconnected resources. These challenges arise on *IoT* due to continuous addition of data from connecting resource dynamically that impose restriction for optimal resource selection.

Considering the parameters, the chapter focuses on optimal decision making on *IoT* paradigm. For the purpose, "Middleware Approach for Reliable Resource Selection on Internet-of-Things (*MARRS-IoT*)" is proposed. It works through three layers, *i.e.*, *SAL*, Middleware that works through its two building components, *i.e.*, *KA*, *DM* and *HMIL*. *SAL* has number of active resources that targets to gather, process and transmits data to the base server through the wireless connection. In Middleware, *KA* is responsible for the discovery of the resources by adding to their self-manageability properties like self-configuration and self-optimization. Once the resource is able to discover itself and its neighbors, *DM* in Middleware, initiates to calculate trust score of participant resources against initiator resource using Fuzzy based trust model. The trust model helps to detect malfunctions

and to provide qualified services to enhance user privacy and information security. It helps in reliable collaboration among resources and ensures optimal selection of resource. The last layer, *HMIL*, is responsible for communication and computation among resources through Hybrid M-Gear protocol. The proposed solution allows resource discovery through Neighbor Discovery Algorithm, evaluates trust score using Fuzzy-trust model and reputation to perform optimal decision making on the basis of the availability of resources.

## 6.2 Neighbor Discovery in the Internet-of-Things

The discovery of the resource as neighbor requires to understand the acquired knowledge with respect to its availability patterns. It helps in the selection of the resource effectively in the neighborhood and helps to minimize energy wastage. For the same, following concepts such as cognition, reasoning, knowledge representation, learning and planning need to be supported that together raise few issues that are discussed below.

***Recognition of Presence:*** The foremost requirement for resources to communicate is to recognize the presence of other radio resources within effective range of finite temporal window. Another requirement is related to setting a bound for latency to discover and to exploit communication time.

***Mobility Patterns Features:*** It is necessary to understand the dynamic key features of mobility patterns of the resources.

***Knowledge Acquisition:*** For knowledge acquisition, it is needed to learn resources' storage with respect to information about their recurrence and reproducibility is a further challenge.

### 6.2.1 Trust Evaluation for Decision Making

Since, the entire infrastructure of *IoT* is distributed and virtual; communication among resources is highly dimmed by malfunctioning and attacks with dynamic addition of the resources. Therefore, trust mechanism as a key component is needed to check

trustworthiness of the resources for selecting reliable node for communication. Its basic tasks are to locate the resources, check their availability and authenticate them using scalar range from 0 to 1; where 0 stands for no trust and 1 stands for complete trust. Considering the requirement, some of the decision making algorithms are discussed in detail.

**Particle Swarm Optimization:** Particle Swarm Optimization (*PSO*) is a meta-heuristic approach and considers population or swarm of candidate solutions or particles which are moved around in the search space [191]. Formally, let cost function,  $CF : CF^n \rightarrow CF$  is to be minimized and it considers a particle as an argument in the form of a vector of real numbers. As an outcome, it generates a real number that presents the value of the objective function of given particle solution. The gradient of  $CF$  is not known. The goal is to find a solution  $p$  for which  $CF(a) \leq CF(b)$  for all  $q$  in the search-space, which would mean  $p$  is the global minimum. Since, *PSO* does not use the gradient of the problem being optimized, which means *PSO* does not require that optimization problem be differentiable as is required by classic optimization methods such as Gradient Descent and Quasi-Newton methods.

**Ant Colony Optimization:** Ant Colony Optimization (*ACO*) is most popular and successful optimization technique inspired from the natural behavior of ants [191]. The heuristic method identifies optimal routes *via* graphs for population of computational ants. These ants passes through a node for each variable and paths' length defines the value of objective function. Over a set of successive iterations or ant generations respectively, highly fit paths emerge and the objective function value is minimized using fitness function. Due to the stochastic nature of the algorithm, however, the search can continue to explore new sections of the decision space, avoiding local optima, as an edge with a low pheromone concentration may be randomly selected on occasion.

**Binary Genetic Algorithm:** Binary Genetic Algorithms (*BGA*) are designed for gray code representation of two corresponding binary strings with large difference in their bitwise representations [191]. It has defined dead zones or no search zones for loss of specific bit and fails to achieve the optimal solution if global optima is located inside this zone. In

order to obtain optimal solution, a sufficiently large design population and high mutation probabilities are natural remedies. But this is not possible due to high computational cost. Hence, it becomes a challenging task to select a number of appropriate operation parameters for the *BGA* to solve a new optimization problem.

## 6.2.2 Routing

From the point of computation and communication among resources, number of routing protocols exist that help to turn *IoT* vision into reality. Some of them are detailed below.

***Low-Energy Adaptive Clustering Hierarchy Protocol:*** Low-Energy Adaptive Clustering Hierarchy protocol (*LEACH*) is self-manageable which distribute energy uniformly in a randomized manner among the resources [192] [193]. Few resources choose themselves as the cluster-head at any time with certain probability as threshold and broadcast their status to other resources in the network. The resources check for minimum energy consumption to the cluster-head before joining it and form the clusters. Here, cluster-head performs the scheduling for its connected resources as cluster. It helps cluster-head to keep track of all the active or dead resources which protect sensors from battery drainage. Moreover, it minimizes the energy dissipation and increase the lifetime of the infrastructure by performing the local data fusion and data aggregation. The *LEACH* is good if there is single hop communication among resources in ad-hoc networks. Its major drawback is energy depletion if the distance among resources is large. Moreover, it follows the probability criteria for the selection of the resources as the cluster-head. Therefore, another approach, *namely, Mod-LEACH* is suggested for better communication.

***Modified Low-Energy Adaptive Clustering Hierarchy Protocol:*** Modified Low-Energy Adaptive Clustering Hierarchy Protocol *Mod-LEACH* is a variant of *LEACH* protocol having efficient head replacement scheme and dual transmission power level that minimizes energy consumption [194]. Unlike *LEACH* algorithm, where communication depends on energy, the *Mod-LEACH* algorithm is more beneficial which constantly checks for cluster-head energy and has provision to select another cluster-head for continuous communication and sustains

for longer time. *Mod-LEACH* has limitation that it does not store the updated location of cluster-head and their minimum distances with the resources. As a result, it provides irrelevant data with least matches.

***Gateway based Energy Aware Multi-Hop Routing Protocol:*** Gateway based Energy Aware Multi-Hop Routing Protocol (*M-GEAR*) divides the sensor nodes into four logical regions on the basis of their location in the sensing field. In *WSN*, direct transmission of data packets from nodes to the base station is very uneconomical as it costs a lot of energy consumption. This can be improved by the aggregation of data packets from all sensor nodes at a single node called the gateway node. This node can be placed at the center of the network and is used for transmitting the aggregating data signals to the sink node located outside the network or far away from the network. Moreover, the node can be rechargeable so that there will be less burden of being energy efficient, as the expenses for recharging a gateway node are much less than the replacement of a sensor node. Also, the data transmitted from the sensor nodes can be collected at the cluster heads before being transmitted to the gateway node, again improving the energy efficiency of the network.

### **6.3 Middleware Approach for Reliable Resource Selection on Internet-of-Things**

*MARRS-IoT* considers two major perspectives, *i.e.*, *service request* and *service provision* (*see* Figure 6.1). Both the perspective shares bidirectional relationship in terms of self-manageability and authentication of the resources. *MARRS-IoT* has provided distinct layers, *say*, *SAL*, Middleware that works through its building blocks (*KA*, *DM*) and *HMIL*. Each layer has defined functionalities, methodology and its working, as discussed below.

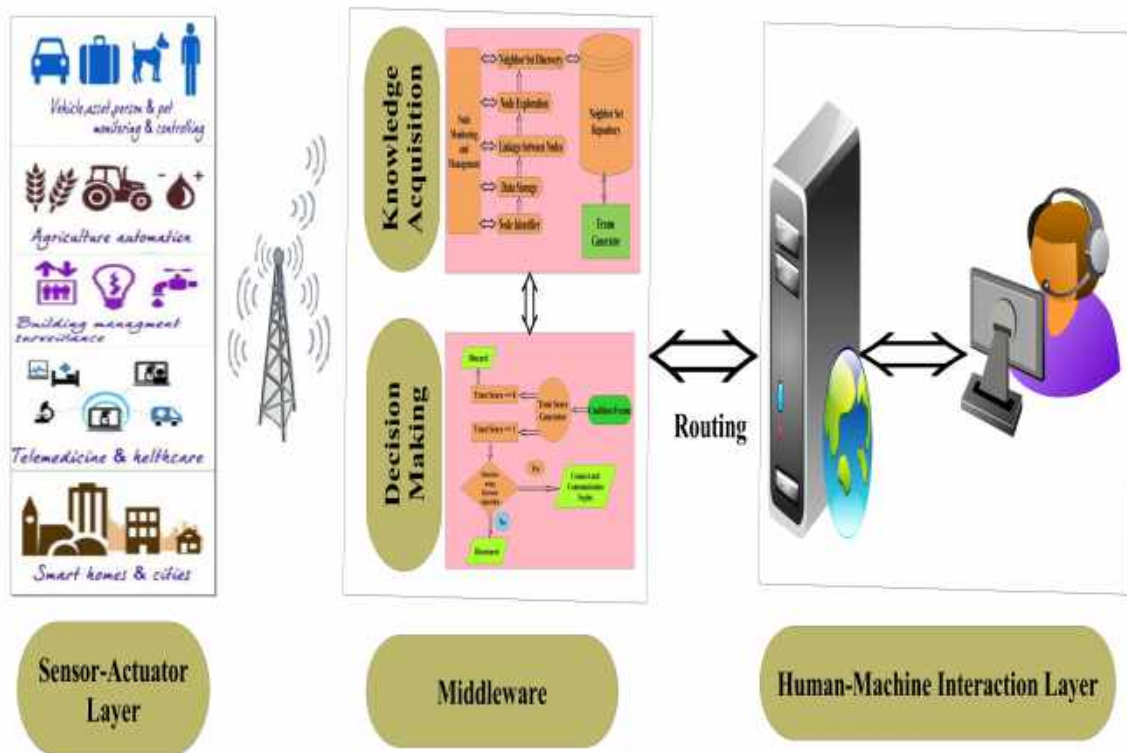


Figure 6.1: Middleware Approach for Reliable Resource Selection on Internet-of-Things

### 6.3.1 Sensor-Actuator Layer

SAL has interconnected set of resources, say,  $N = \{n_1, n_2, \dots, n_k, \dots, n_n\}$  belonging to different application areas, such as Agriculture Automotive, Building Management Surveillance, Tele-medicine Health-care, etc., that interacts *via* wireless connection. The resources gather, process and forward data to the middleware that works through KA and DM for further computation and communication.

### 6.3.2 Middleware

Middleware is the system software layer that controls, organizes and manages the functionalities *via* its building blocks, *i.e.*, KA and DM. Each of the component is detailed below.

## Knowledge Acquisition

It is responsible to fulfill service request by the end users, which may contain context, content and attributes. The objectives of the layer are creating meta-data for the storage of resources, linkage between various resources, knowledge representation, coordination among resources and continuous monitoring of the environment to ease the discovery of the resources.

For the purpose, a 2-D area 'A' that contain set of resources,  $N = \{n_1, n_2, \dots, n_k, \dots, n_n\}$  is considered. The resource 'k' is located at 2-D position,  $p_{k,l}$ , with an assumption that it is found itself at 'lth' step with self-discoverable properties. Also, each resource has tendency to make point observation,  $Y_{k,l}$ , of a spatial function,  $F_{p_{k,l}}$ , at 'lth' step and the distribution is taken as Gaussian as in Eq. 6.1:

$$Y_{k,l} = F_{p_{k,l}} + \varepsilon_{k,l} \quad (6.1)$$

where,  $\varepsilon_{k,l}$  is a Gaussian noise with zero mean and covariance,  $\sigma_{k,l}^2$ , i.e.  $\varepsilon_{k,l} = \text{Noise}(0, \sigma_{k,l}^2)$ . The process is conducted for collecting information based on position and point observation. The gathered information includes input vectors and the corresponding observations. Moreover, the prior distribution of latent variable,  $F_{k,l} = F_{p_{k,l}}$ , is assumed to have zero mean value. Therefore, the prior distribution about multiple latent variables is given by covariance matrix,  $CV_{NN} = [\text{cv}(p_{k,l}, p_{j,l})]$ ,  $k = 1 \dots N$  and  $j = 1 \dots N$ . Taking a positive definite covariance matrix, the complete prior distribution is defined below:

$$p(F_{N,l}) = \text{Noise}(0, CV_{NN}) \quad (6.2)$$

Similarly, the prior distribution,  $pd$ , for the observation vector is given as:

$$pd(Y_{N,l}|F_{N,l}) = \text{Noise}(F_{N,l}, R_{N,l}) \quad (6.3)$$

where, Gaussian process regression,  $R_{N,l} = \text{diag}(\sigma_{1,l}^2, \sigma_{2,l}^2, \dots, \sigma_{N,l}^2)$ .

$R_{N,l}$  infers  $F_{*,l} = F_{p_{*,l}}$  for a test point  $p_{*,l} \in A$ , using  $p(Y_{N,l}|F_{N,l})$  giving a training data set  $(p_{k,l}, Y_{N,l})$ . At a particular point, calculation of the latent predictive distribution is provided below by solving maximum posteriori problem [180].

$$pd(Y_{N,l}|F_{N,l}) = Noise(\mu_{*,l}, \Sigma_{*,l}) \quad (6.4)$$

where, predictive mean function,  $\mu_{*,l}$  is:

$$\mu_{*,l} = CV_{*N}(CV_{NN} + R_{N,l})^{-1}(Y_{N,l}) \quad (6.5)$$

and predictive covariance function,  $\Sigma_{*,l}$  is:

$$\Sigma_{*,l} = CV_{NN} - CV_{*N}(CV_{NN} + R_{N,l})^{-1}CV_{N*} \quad (6.6)$$

where,  $CV_{*N} = [cv(p_{*,l}, p_{1,l}, \dots, p_{j,l})]$  and  $CV_{**} = cv(p_{*,l}, p_{*,l})$ .  $CV_{N*}^l = CV_{*N}^l$  for symmetrical covariance functions at step  $l$ .

The prior knowledge about  $cv(p_{k,l}, p_{j,l})$  is the crucial ingredient that analyze the behavior of the process. Let ' $CR_k$ ' and  $NS_k$  be the communication range and neighbor set, respectively, for the resource ' $k$ '. The  $CR_k$  for  $NS_k$  is defined as:

$$NS_k = \{j \in N \mid \|p_{k,l} - p_{j,l}\| \leq CR_k\} \quad (6.7)$$

The compact covariance function [181] is defined for each local resource ' $k$ ' and has local hyper-parameters,  $r_{k,l}, s_{k,l}$ . An effective range,  $ER_k$ , for  $NS_k$  is

$$ER_k = \{j \in N \mid \|p_{k,l} - p_{j,l}\| \leq r_{k,l}\} \quad (6.8)$$

A predictive neighbor set,  $B_k$ , for resource ' $k$ ' is needed and is represented as:

$$B_k = \sum_{j=1}^{|N_k|} ER_j \quad (6.9)$$

where,  $N_k$  is the cardinality of set  $ER_k$ . Now using Wu's polynomial function [182] [183],  $cv(p_{k,l}, p_{j,l})$  is:

$$cv(p_{k,l}, p_{j,l}) = cv(g_l) = s_l \left(1 - \frac{g_l}{r_l}\right)_+^4 \left(1 + \frac{4g_l}{r_l} + \frac{3g_l^2}{r_l^2} + \frac{3g_l^3}{4r_l^3}\right) \quad (6.10)$$

where,

$$\left(1 - \frac{g_l}{r_l}\right)_+ = \begin{cases} \left(1 - \frac{g_l}{r_l}\right), & \text{if}(g_l < r_l) \\ 0, & \text{otherwise} \end{cases} \quad (6.11)$$

As the increase in the number of resources joining the network randomly with their registered *IDs*, it faces challenge related to discovery of neighbors from a group of *NS* on distributed network. To meet the challenge, a logic with multi-criteria is required and is defined on the basis of resources as *initiator* and *participants* [184]. The resource that begins the process is called *initiator* whereas resources ready to be discovered are *participants*. At the first, the main task is to determine the number of valid potential neighbors from *NS* on the network. The initiator sends a *CALCULATE* message carrying its own *ID*, the probability,  $P_{ack}$ , with which a participant sends an acknowledgment and the frame size which is used for start message. The next step deals with collecting *IDs* of the obtained list of the responded resources by sending a *SEARCH* message. The *SEARCH* message contains new frame size of searching and the participants send acknowledgment messages within frame to report their *IDs*. The following assumptions are made for the execution of the process.

- (a) The static time domain slots are defined with frames having entries of continuous time slots. The receiver has the ability to detect collisions as well as idle slots on the basis of packets. A resource is defined on the basis of active and nonactive states, where an *active* state is responsible for transmitting or receiving else *non-active*. If a resource  $k$  is at slot  $s$  and is able to obtain *ID* successfully then it is at active state,  $st(k) = 1$  else non-active state,  $st(k) = 0$ .
- (b) The neighbor of a resource ' $k$ ' from *NS* depends on both distances, *i.e.*, physical and interest. The former is considered to be short if they are within the  $CR_k$  and share similar interests. The latter is specified by interest vector between two users and is calculated using Euclidean distance, or vector similarity between two interest vectors.
- (c) The active participants are not supposed to leave during the process, the initiators follow the Poisson process with an unknown rate,  $\lambda$ , and the system starts at a global slot  $0$ . The number of participants joining system at slot ' $l$ ' is denoted as  $j_l$  and total

number of initiators in the system at slot ' $l$ ' as  $J_l$ , i.e.,  $J_l = J_0 + \sum_{k=1}^l (j_k)$ .

On the basis of these assumptions, the estimate for number of resources and their obtained  $IDs$  is discussed below.

- (a) **Estimating the number of resources in the neighbor set:** The  $NS$  has to undergo the processing like wait frame, acknowledging and sending  $CALCULATE$  message. Assuming  $k_i$  be the number of resources in the  $NS$  that will send acknowledgment message within frame of size,  $F_s$ , randomly. The load factor,  $\alpha$ , is defined as ratio of number of resources in the neighbor sets sending acknowledgment message within frame to the total size of the frame, i.e.,  $\alpha = \frac{k_i}{F_s}$ . Considering  $rv_0$ ,  $rv_1$  and  $rv_2$  are the number of slots which have received acknowledgment message as 0, 1 and multiple, respectively, in a frame. Then, the expected value,  $EXP$ , is calculated using [185] as:

$$\begin{aligned} EXP[rv_0] &= F_s e^{-\alpha} \\ EXP[rv_1] &= F_s \alpha e^{-\alpha} \\ EXP[rv_2] &= F_s (1 - (1 + \alpha) e^{-\alpha}) \end{aligned} \quad (6.12)$$

On the basis of these equations, estimate on  $k_i$  is obtained with the observed values of  $rv_0$ ,  $rv_1$  and  $rv_2$ .  $rv_1$  and  $rv_2$  cannot be used to calculate the value of  $k$ , as it is observed to be non-monotone function. Value of  $rv_0$  is used for computation of  $k_i$  as:

$$\hat{k}_i = F_s \log(F_s / rv_0) \quad (6.13)$$

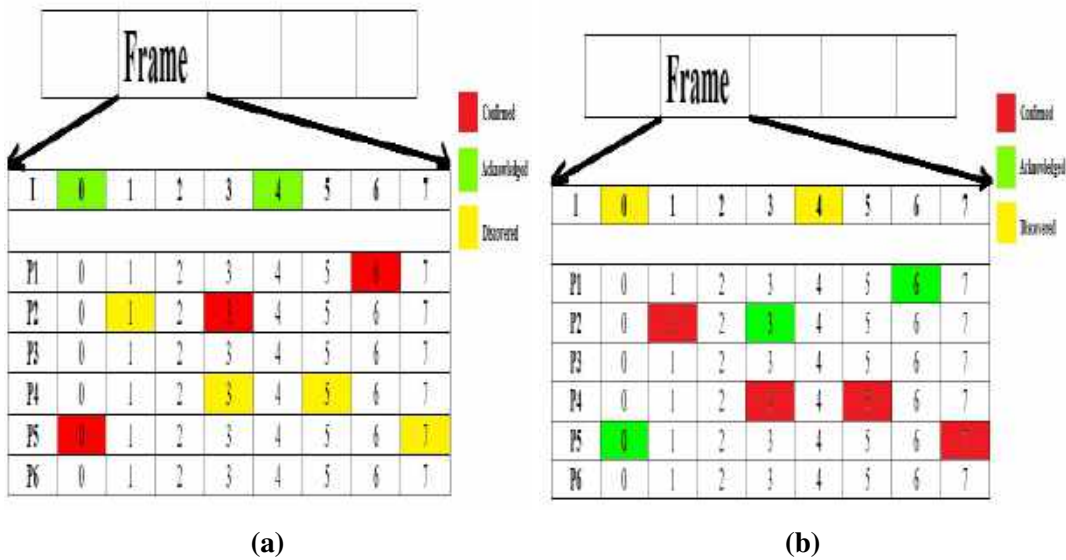
Its variance,  $\sigma_0$ , is defined as:

$$\sigma_0 = k_i \frac{e^{\alpha} - (1 + \alpha)}{\alpha} \quad (6.14)$$

The collision could take place, if value of  $\alpha$  is large with high probability and  $k_i$  in the neighbor set estimate; which will result in to larger variance as  $rv_0$  is being used. For the situation, where number of resources as participants, say,  $k_p > k_i$ , the participants send acknowledgment messages with certain probability,  $p_{ack}$ , such that participants sending acknowledgment messages are reduced to  $J_l$ .  $p_{ack}$ , where,  $J_l = J_0 + \sum_{k=1}^l (j_k)$

are the resources joining network at slot, say,  $l$ . The participants ' $J_l$ ' are computed by initiator with desired confidence as:

- (i) The *CALCULATE* message is published by initiator containing frame of size,  $F_s$ , and response probability,  $p_{ack} = 1$ .
  - (ii) All participant response with an acknowledgment message to one of the  $F_s$  slots randomly with probability  $p_{ack}$ , whereas the initiator calculates remaining participants with idle slots  $rv_0$ .
  - (iii) Another *CALCULATE* message is published with reply probability by initiator when it fails to achieve the expected confidence.
- (b) **Obtaining IDs of k Participants:** At this stage, the initiator is responsible for sending *CALCULATE* message with  $F_s$ -size frame, to which participants respond by sending *IDs* among  $F_s$ -slots in the frame. As a result, initiator will receive acknowledgment messages when there is no collision. Also the participants who have sent acknowledgment messages, say,  $k_i$  in the previous stage, will participate. There are two strategies through which *IDs* are obtained and are discussed below:
- (i) **Static fraction of participants:** Here, initiator discovers ' $q$ ' participants out of  $k_i$  with  $j_1q \leq k_i \leq j_2q$  for constants  $1 < j_1 < j_2$  (see Figure 6.2(a)).



**Figure 6.2:** Static fraction of participants (a) Broadcasting Message (b) Acknowledgment Message

As a result, participant will be active at the successive slot and wait for the confirmation message. Once confirmation message is received, participant sends no more acknowledgment message in any slots (*see* Figure 6.2(b)).

- (ii) **All participants:** The initiator wishes to discover all participants, *i.e.*,  $q = k_j$ . Here, the initiator sends only *END* message to terminate the process once it has discovered maximum number of resources from the neighbor set.

The entire process of *KA* for the discovery of the *NS* is summarized in Algorithm 6.1.

The process initializes sets of resources as '*N*'. The point observation,  $Y_{k,l}$ , for resource '*k*' is described using its position,  $p_{k,l}$ , at '*l*<sup>th</sup>' step using spatial function,  $F_{p_{k,l}}$ . It collects information by initializing prior distribution,  $pd$ , and calculates the observation vector. It further computes communication range,  $CR_k$ , effective range,  $ER_k$  and predictive neighbor set,  $B_k$ , for *k*. The knowledge for '*N*' is acquired which is taken as input to the system for search evaluation. Here, the resources are distinguished as initiator '*i<sub>t</sub>*' and participants '*p<sub>n</sub>*'. The *i<sub>t</sub>* sends *CALCULATE* message for finding the number of *p<sub>n</sub>* and the *p<sub>n</sub>* sends their *ID*'s to *i<sub>t</sub>*. The *i<sub>t</sub>* further estimates the number of resources as *p<sub>n</sub>* that sends acknowledgment message within frame size,  $F_s$ , *i.e.*, within defined time stamp, and the *ID*'s are stored in  $F_s$ . The load factor,  $\alpha$ , is described for each  $F_s$  using which expected value of resources as  $EXP[rv_0]$ ,  $EXP[rv_1]$ ,  $EXP[rv_2]$  are estimated. It depends upon  $\theta$ ,  $I$ , *multiple* acknowledgment messages. Further,  $F_s$  is indexed and stored in parent index service, *MyIndex*. The parameters present in *MyIndex* are matched with input query, '*Q*'. If it matches completely then the user is alarmed with matched discovered resources set as *NS*. The output of the algorithm as *NS* is further processed for selecting rightful resources by calculating trust scores, *locally* and *globally*, as explained in next section.

## Decision Making

With the increase in the number of resources present in the *NS*, large sensitive data are exposed that demands to consider security attacks as malicious attacks. The attacks steal personal and private important information which hamper the trust of the system. Therefore, to provide reliable communication among resources, a trust mechanism should be considered to maintain the authenticity of the resources. The trust in itself is a complicated concept to

---

**Algorithm 6.1 Neighbor Discovery Algorithm (NDA)**

---

Require: Set of resources  $N = \{n_1, n_2, \dots, n_i, n_k, \dots, n_n\}$ , Position  $(p_{k,l})$ , Point Observation  $(Y_{k,l})$ , Spatial function  $F_{p_{k,l}}$ , Prior Distribution  $(pd)$ , predictive mean function  $(\mu)$ , predictive covariance function  $(\Sigma)$ , Parent Index Service (*MyIndex*)

```
1: initialize  $N \leftarrow \{n_1, n_2, \dots, n_i, n_k, \dots, n_n\}$ 
2: for ( $N = 1$  to  $n$ ) do
3:   for ( $s = 1$  to  $n$ ) do
4:     describe  $Y_{k,l} = F_{p_{k,l}} + \epsilon_{k,l}$  // point observation using Eq. 6.1 for resource  $k$  at  $l$ th step
5:   end for
6: end for
7: initialize  $p(F_{N,l}) = \text{Noise}(0, CV_{NN})$  // prior distribution using Eq. 6.2
8: for ( $j = 1$  to  $N$ ) do
9:   for ( $l = 1$  to  $j$ ) do
10:     $pd(Y_{N,l}|F_{N,l}) = \text{Noise}(F_{N,l}, R_{N,l}) = \text{Noise}(\mu_{*,l}, \Sigma_{*,l})$  //using Eq. 6.3 and 6.4
11:   end for
12: end for
13: for ( $k = 1$  to  $N$ ) do
14:   describe  $NS_k = \{j \in N \mid \|p_{k,l} - p_{j,l}\| \leq CR_k\}$  //
    communication range  $CR_k$  using Eq. 6.7
15:   describe  $ER_k = \{j \in N \mid \|p_{k,l} - p_{j,l}\| \leq r_{k,l}\}$  // effective range  $ER_k$  using Eq. 6.8
16:   describe  $B_k = \sum_{j=1}^{|N_k|} ER_j$  // predictive neighbor set  $B_k$  using Eq. 6.9
17: end for
18: // Search Evaluation
19: for ( $\text{input} \leftarrow Q$ ) do
20:   initialize initiator  $\leftarrow i_t$ , participants  $\leftarrow p_n$ 
21:   while ( $i_t == \text{true}$ ) do
22:     send CALCULATE  $\leftarrow (i_t, p_n)$ 
23:     collect  $ID(p_n) \leftarrow \text{SEARCH}(p_n)$ 
24:      $F_s \leftarrow ID(p_n)$ 
25:   end while
26:   describe  $\alpha = \frac{k}{F_s}$ 
27:   for ( $\text{each } F_s == 1$ ) do
28:     calculate  $EXP[rv_0] = F_s e^{-\alpha}$ 
29:     calculate  $EXP[rv_1] = F_s \alpha e^{-\alpha}$ 
30:     calculate  $EXP[rv_2] = F_s (1 - (1 + \alpha) e^{-\alpha})$ 
31:   end for
32:   forward  $\text{MyIndex} \leftarrow F_s$  // indexing frame size
33:   if ( $\text{match}(Q, \text{MyIndex}) == \text{true}$ ) then // matching parameter
34:     send sender  $\leftarrow F_s(N)$  // discovered resources
35:   end if
36:   store  $NS \leftarrow F_s(N)$ 
37:   print  $NS$ 
38: end for
```

---

measure and depends on various security parameters as discussed below.

- (A) *Heterogeneity*: The resources with distinct capabilities related to hardware, software, operating system or performance require to be discoverable and addressable. For the purpose, it needs various routing protocols, network topology or prototype for communication and computation. Thus, the *IoT* platform being dynamic, demands reliable communication and resources are to be light-weighted.
- (B) *Scalability*: With increasing resources on *IoT*, the platform faces countless communication and transactions among resources. It results into increase in the scale of the area that further require trust maintenance. Moreover, it asks for maintaining robustness to handle increasing data load on the system, eliminating the congestion problem on the network.
- (C) *Data and Privacy*: The generated data by the resources keep on increasing with increase in the number of joining resources. The data being sensitive is needed to be secured before other active resources access it without prior permissions. Hence, a trust mechanism should have capability to select the appropriate resource for communication.
- (D) *Confidentiality and governance*: The trust system should help *IoT* to find better trusted resource that maintains the confidentiality of the resource. It would lead to communication for longer period.

Keeping in view the various security constraints, the Fuzzy based trust model is suggested that dynamically add new resources for reliable communication where each node develops a direct reputation with each other. It helps to build trust among them *locally* and *globally via* evaluating the direct and indirect trust schemes. It considers the communication among neighbors' resources and maintaining confidentiality, privacy and security. Further, trust relations are based on reputations,  $R$ , *i.e.*, created by communications among resources *via* protocol that help a resource for evaluating the trustworthiness with other resources. The reputation metrics are packet delivery, packet loss and energy consumption. The methodology for suggested approach is explained.

**Fuzzy-based Trust Model System:** Trust is defined as the probability by which a service provider is shortlisted through participants by initiator to perform the desired operation as expected by the service requester. Its domain lies between 0 and 1 which indicates its success and failure to provide the service. For value 1, it is considered to be trustworthy to rely upon for future transactions whereas for value 0, it is rejected.

In the proposed Fuzzy based system, membership functions using mathematical model needs to be established for calculating trust score,  $T_s$ , for a resource. Assuming  $NS_{sub} = \{NS_1, NS_2, NS_3, \dots, NS_n\}$  be the resources shortlisted for a resource in a corresponding range. Then, its mapping function is defined as:

$$\begin{aligned} \text{MappingFunction} : NS * NS &\rightarrow [0, 1] \\ (NS_i, NS_j) &\rightarrow T_s(NS_i, NS_j) \in [0, 1] \end{aligned} \quad (6.15)$$

where,  $T_s$  represents the trust score value between  $NS_i$  and  $NS_j$ . Also, each resource from the  $NS$  maintains a transaction table  $TT_{i,j}$  containing successful transaction ' $ST_{i,j}$ ', positive transactions ' $PT_{i,j}$ ', time-to-live ' $TTL$ ', initiator and the participant. In short,  $TT_{i,j}$  is represented as:

$$TT_{i,j} = \langle \text{initiator}, \text{participant}, ST_{i,j}, PT_{i,j}, TTL \rangle \quad (6.16)$$

$TT_{i,j}$  is maintained by each resource of  $NS$  and  $TS$  is calculated on the basis of trust metrics. A resource from neighbor set as participant or initiator performs various jobs such as sensing and actuating to collect data about the packet, transmitting data to one-another, transactions, packet forwarding activities, etc. These jobs may get affected due to network bandwidth and data transfer rate. Hence, trust metrics are required to evaluate the performance of proposed *MARRS-IoT* and to validate it. These are discussed below on the basis of security and trust.

- (i) **Destination to Destination Data Delivery:** It is described as ratio of data delivered to data send. The performance is directly proportional to data delivery rate and is measured in Joules per second.
- (ii) **Data Loss:** The network congestion often leads to collision among transmitted

packets, hence, they do not reach the targeted destination. It effects the overall network's performance and is measured in percentage.

The reputation,  $R$ , between  $NS_i$  and  $NS_j$  is computed as:

$$R = \frac{PT_{i,j}}{ST_{i,j}} \varepsilon [0, 1] \quad (6.17)$$

The trust score  $TS$  is evaluated either *locally* or *globally* and is explained below [186] [187] [188].

**A Trust Evaluation Locally:** If there is successful transaction between  $NS_i$  and  $NS_j$ , then they are considered to be trustworthy for each other and have direct trust,  $DT_{i,j}$ , connection between them. Hence, as per  $TT_{i,j}$ , their fuzzy based trust model is defined as:

$$DT_{i,j} = \frac{R}{R + \gamma(1 - R) + \frac{\chi}{ST_{i,j}}} \quad (6.18)$$

where,  $\gamma$  and  $\chi$  are behavior of resource against malicious acts and uncertainty trust, respectively. Let,  $DT_{i,j}^{e-1}$  and  $DT_{i,j}^{\Delta e}$  be the direct trust at current and previous time duration,  $e$ , then the updated evaluation is given by combining all the results as:

$$\begin{aligned} DT_{i,j}^e &= \tau_1 * DT_{i,j}^{e-1} + \tau_2 * DT_{i,j}^{\Delta e}, \\ \tau_1 &= \left[ \left( 1 - \frac{1}{2} \right) * v \right], \forall v \varepsilon [0, 1] \\ \tau_1 + \tau_2 &= 1 \end{aligned} \quad (6.19)$$

In summary,  $DT_{i,j}$  depends on  $DT_{i,j}^{e-1}$ ,  $DT_{i,j}^{\Delta e}$  and  $v$ . Therefore, direct trust updating equation is:

$$DT_{i,j}^e = \left( 1 - \frac{1}{2} * v \right) * DT_{i,j}^{e-1} + \frac{1}{2} * v * DT_{i,j}^{\Delta e} \quad (6.20)$$

Since, it is not possible to calculate the good or bad behavior of resources based on communication, it is required to set a threshold value of communication time,  $C_{th}$ . On

the basis of threshold value, direct trust evaluation  $DT_{i,j}^e$  is calculated as:

$$DT_{i,j}^e = \begin{cases} \frac{1}{2} * (1 + \frac{R}{C_{th}}), ST_{i,j} < C_{th} \\ \frac{R}{R+\gamma(1-R)+\frac{\chi}{ST_{i,j}}}, ST_{i,j} \geq C_{th} \end{cases} \quad (6.21)$$

Now, considering the scenario when there is no direct connection between ‘ith’ and ‘jth’ resources. In such case, concept of third party solicitation is introduced.

Assuming,  $RS_{i,j}$  be the number of request solicits,  $PS_{i,j}$  be the number of positive solicits, and  $C_{th}$  is the threshold. Indirect trust,  $IT_{i,j}$ , based on fuzzy solicit trust model is given as:

$$IT_{i,j} = \begin{cases} \frac{1}{2} * (1 + \frac{v}{C_{th}}), RS_{i,j} < C_{th} \\ \frac{v}{v+\gamma(1-v)+\frac{\chi}{RS_{i,j}}}, RS_{i,j} \geq C_{th} \end{cases} \quad (6.22)$$

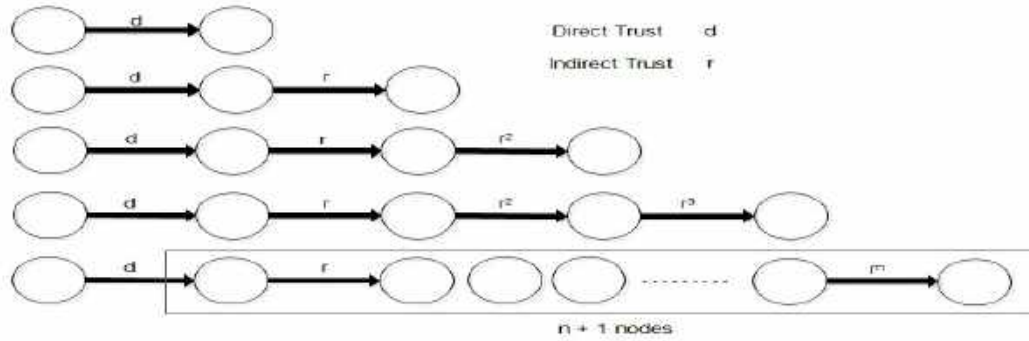
where,  $v = \frac{PS_{i,j}}{RS_{i,j}} \in [0, 1]$ . It is understood that distinct resources may give distinct solicitation for one resource. Let,  $NS_l$  and  $NS_m$  give trust value as  $IT_{l,j}$  and  $IT_{m,j}$ , respectively. On combining these two trust evaluations,  $IT_{i,j}$  can be calculated as:

$$\begin{aligned} IT_{i,j} &= (d(NS_i, NS_l) \wedge r(NS_l, NS_j)) \\ &\cup (d(NS_i, NS_m) \wedge r(NS_m, NS_j)), \\ &\forall NS_l, NS_m \in NS \end{aligned} \quad (6.23)$$

Using Eqs. 6.21, 6.23 and weight values, fuzzy trust model for local is defined as:

$$T_{i,j} = (r.d) \cup (r.d) \cup \dots \dots \dots (r.d) \quad (6.24)$$

In summary, direct trust relationship in collaboration with both direct and indirect cases is computed as Eq. 6.25:



**Figure 6.3:** Global Trust Evaluation

$$T_{i,j} = W_d \cdot DT_{i,j} + W_{ind} \cdot \sum_{\sigma} (DT_{i,l} \cdot IT_{l,j}),$$

$$1 < W_{ind} < W_d < 0 \quad (6.25)$$

$$W_d + W_{ind} = 1$$

where,  $W_d$  and  $W_{ind}$  represent the value for weight of direct and indirect trust.

**B Trust Evaluation Globally:** The  $NS_i$  have both direct and indirect observations with  $NS_j$  through its associate linkages and it follows both trust models, *i.e.*, direct trust and indirect trust. For obtaining accurate trust values among nodes, it integrates both trust experiences. Since, direct trust value changes with time, therefore, there is a need to develop indirect trust set.

In summary, Fuzzy based trust relation is defined as the union of fuzzy direct trust model with 1-level, 2-level, ..., n-level fuzzy indirect trust models, *i.e.*,  $DT_{i,j} \cup 1\text{-level } IT_{i,j} \cup 2\text{-level } IT_{i,j} \cup \dots \cup n\text{-level } IT_{i,j}$ . Taking an example for evaluating trust relation between  $NS_i$  and  $NS_j$  in a group of resources with an assumption that  $NS_i$  has multiple routes to reach  $NS_j$  as shown in Figure 6.3.

In order to calculate trust relation with more accuracy, each route need to be calculated separately and is given as:

$$T_{i,j} = d + (r * d) + (r^2 * d) + ..... * (r^n * d)$$

$$T_{i,j} = (1 + r + r^2 + ... + r^n).d \quad (6.26)$$

$$T_{i,j} = \lim_{n \rightarrow \infty} [(1 + r + r^2 + ... + r^n).d]$$

The trust evaluation algorithm is given by Algorithm 6.2:

---

**Algorithm 6.2 Trust Evaluation Algorithm**

---

Require: Neighbor Sets, ( $NS_{i,l,m,\dots,j}$ ), initiator, participants, Successive transactions ( $ST_{i,j}$ ), Positive transactions ( $PT_{i,j}$ ), Time to live ( $TTL$ )

- 1: initialize  $\leftarrow 0$
  - 2: **for** (NS = 1 to n) **do**
  - 3:     Mapping (NS, NS)  $\in [0, 1]$
  - 4:     **if** ( $\exists T_s \forall NS$  in i to j) **then**
  - 5:          $NS_i, NS_j \leftarrow T_s (NS_i, NS_j) \in [0, 1]$
  - 6:     **end if**
  - 7: **end for**
  - 8: Set  $TTL \leftarrow 0$ , initiator  $\leftarrow 0$ , participants  $\leftarrow 0$
  - 9: select initiator = Rand ( $NS_{i,\dots,n}$ )
  - 10: **for** ( $NS_i \neq NS_j$ ) **do**
  - 11:     Initiate communication among initiator and participants
  - 12:     Calculate  $ST_{i,j}$  and  $PT_{i,j}$
  - 13:     Maintain  $TT_{i,j}$  as relative function of  $\langle Initiator, Participant, ST_{i,j}, PT_{i,j}, TTL \rangle$
  - 14: **end for**
  - 15: Calculate TTL
  - 16: **for** ( $\forall ST_{i,j} \in (NS_i, NS_j) == localtrust$ ) **do**
  - 17:      $IT_{i,j} = W_d \cdot DT_{i,j} + W_{ind} \cdot \sum_{\sigma} (DT_{i,l} \cdot IT_{l,j}), 1 < W_{ind} < W_d < 0$
  - 18:      $W_d + W_{ind} = 1$
  - 19: **end for**
  - 20: **for** ( $\forall ST_{i,j} \in (NS_i, NS_j) == globaltrust$ ) **do**
  - 21:      $T_{i,j} = \lim_{n \rightarrow \infty} [(1 + r + r^2 + ... + r^n).d]$
  - 22: **end for**
  - 23: Calculate update TTL
- 

In this algorithm, at an instance, the neighbor set is empty. It is updated with the matched neighbors using trust evaluation, *locally* and *globally*, leading to the updation of the *NS*. A loop is initiated to map among updated *NS* and previously stored *NS* for mapping. It calculates trust score among them as 0 and 1. The time to live, *TTL*, initiator and participants are set to zero which are dynamically added to the *NS*.

The initiator is selected as random neighbor from the  $NS$ . The communication begins among the  $NS$  as initiator and participants, if the neighbors are not present in either sets, *i.e.*, previous and updated. The loop calculates its successive transactions,  $ST_{i,j}$ , positive transactions,  $PT_{i,j}$ , and maintains its transaction Table,  $TT$ . For  $ST_{i,j}$  of  $NS$  in local trust, the process calculates indirect trust,  $IT$  and its weights,  $W_d + W_{ind}$ . For  $ST_{i,j}$  of  $NS$  in global trust, it finds the accurate route with its trust score. The process ends with updating  $TTL$  and calculating trust, both *locally* and *globally*.

In summary, each resource is assigned a trust score which is either  $0$  or  $1$  on the basis of trust evaluation. Here,  $0$  indicates no trust and  $1$  represents trust. The trustworthy resources undergoes decision making process using proposed Fuzzy-Decision Algorithm,  $FDA$ , (*see* Algorithm 6.3).

In this algorithm, an array is considered with trust score value equal to  $1$ ,  $a[T_s]_1$ . The resources as participants,  $ns_p$ , is initialized to  $0$ . A counter ‘ $i$ ’ is taken which will start from resource with index ( $1$  to  $n$ ) till final decision is taken. Each  $ns_p$  is checked for its presence with two conditions; either present *locally* or *globally*. For both the resultant array, *i.e.*,  $ns_{lp}(i)$  and  $ns_{gp}(i)$ , sorting is done that generates two new array,  $ns_{newlp}(i)$  and  $ns_{newgp}(i)$ , for both *local* and *global* evaluations, respectively. Using these array, further distance between initiator and participating resources is calculated that provides  $p(i)$  and  $q(i)$  arrays. Using  $p(i)$  and  $q(i)$  in combination with  $ns_{newlp}(i)$  and  $ns_{newgp}(i)$ , respectively, new 1-D arrays  $p_{snew}(i)$  and  $q_{snew}(i)$ , are generated. Further, these shortlisted resources from both arrays are checked against their availability that output  $p_{avail}(i)$  and  $q_{avail}(i)$  lists. Using  $p_{snew}(i)$  with  $p_{avail}(i)$  and  $q_{snew}(i)$  with  $q_{avail}(i)$ , respectively;  $p_{final}(i)$  and  $q_{final}(i)$  lists are generated.

### 6.3.3 Human-Machine Interaction Layer

For the interaction among human and machine, the formed neighbor sets after calculating trust score undergoes routing process through *Hybrid M-Gear*. Like hub, the *Hybrid M-Gear* through base station,  $BS$ , broadcasts ‘*HELLO*’ packet to the number of similar sensor resources distributed in to area randomly for collecting their response. The response includes locations using which  $BS$  computes the distance of each resource and maintains data table

---

**Algorithm 6.3 Fuzzy-Decision Algorithm (FDA)**

---

**Require:** Array with trust score equal to 1  $a[T_s]_1$ , participating resources  $ns_p$ , counter  $i$ , resultant array ( $ns_{lp}(i)$  and  $ns_{gp}(i)$ ), sorted array  $ns_{newlp}(i)$  and  $ns_{newgp}(i)$ , sum-arrays  $p_{snew}(i)$  and  $q_{snew}(i)$ , shortlisted available array  $p_{avail}(i)$  and  $q_{avail}(i)$

```
1: Input  $\leftarrow a[T_s]_1$ 
2:  $ns_p \leftarrow 0$ ,  $ns_{lp} \leftarrow \{\}$ ,  $ns_{gp} \leftarrow \{\}$ 
3: for ( $i = 1$  to  $n$ ) do
4:   Find  $ns_p(i)$  in  $a[T_s]_1$ 
5:   for ( $i = 1$  to length  $a[T_s]_1$ ) do
6:      $ns_{lp} \leftarrow \text{local}(ns_i)$ 
7:      $ns_{gp} \leftarrow ns_p(i) - ns_{lp}(i)$ 
8:   end for
9:   for ( $ns_{lp}(i) == \text{shortlisted} \ \&\& \ ns_{gp}(i) == \text{shortlisted}$ ) do
10:     $ns_{newlp}(i) = \text{sort}(ns_{lp}(i))$ 
11:     $ns_{newgp}(i) = \text{sort}(ns_{gp}(i))$ 
12:     $p((i)) = \text{cal}_{dis}(ns(i) - ns_{newlp}(i))$ 
13:     $q((i)) = \text{cal}_{dis}(ns(i) - ns_{newgp}(i))$ 
14:     $p_{snew}(i) = \text{sum}(ns_{lp}(i), p(i))$ 
15:     $q_{snew}(i) = \text{sum}(ns_{gp}(i), q(i))$ 
16:    while ( $p_{snew}(i) \neq ns_i(i) \ \&\& \ q_{snew}(i) \neq ns_i(i)$ ) do
17:       $p_{avail}(i) = \text{available}(p_{snew}(i))$ 
18:       $q_{avail}(i) = \text{available}(q_{snew}(i))$ 
19:       $p_{final}(i) = \text{sum}(p_{avail}(i), p_{snew}(i))$ 
20:       $q_{final}(i) = \text{sum}(q_{avail}(i), q_{snew}(i))$ 
21:    end while
22:    if ( $Score_1 == \text{maxdis}(p_{final}(i))$ ) then
23:      Communicate data ( $ns_i, Score_1$ )
24:    end if
25:    if ( $Score_2 == \text{mindis}(q_{final}(i))$ ) then
26:      Communicate data ( $ns_i, Score_2$ )
27:    end if
28:  end for
29: end for
```

---

*table*. The *table* has entries like unique identification, *ID*, location, energy ‘*egy*’, gateway resource ‘*gatewayresource*’ and their distance from *BS*. The entire process is summarized in *Hybrid M-Gear Algorithm 6.4*.

The algorithm initializes the resources, *CH*, *round* to  $\infty$ , *area* as  $100 \times 100$  and energy ‘*egy*’ for all resources to full, *i.e.*, 1 with randomly dispersing resources into the prescribed area. In the beginning, it checks for the station which needs information, *i.e.*, Base Station, *BS*. The *BS* sends *hello packet* to the other resources such that they join the network, *i.e.*,

---

**Algorithm 6.4 Hybrid M-Gear Algorithm**

---

**Require:** Cluster-Head ( $CH$ ), Area ( $area$ ), Base Station ( $BS$ ), Energy ( $egy$ ), Gateway resources ( $gatewayresource$ ), Region ( $region$ ), Data Table ( $table$ )

```
1: Begin
2: // Initialization Phase
3: Initialize resources  $\leftarrow \infty$ ,  $CH \leftarrow \infty$ ,  $area \leftarrow 100 \times 100$ ,  $egy \leftarrow 0/1$ , threshold  $\leftarrow 0/1$ ,
   round  $\leftarrow \infty$ .
4: for ( $i = 1$  to  $n$ ) do
5:    $resource_i \leftarrow area$  (rand(resources)) // dispersing resources in to the  $area$  randomly.
6: end for
7: for ( $i = 1$  to  $n$ ) do
8:   if ( $BS_i == active$ ) then // initial phase
9:     Broadcast  $resource_i \leftarrow$  hello packet //sending Hello packet to join other
     resources
10:     $BS_i \leftarrow loc_i(resource)$  // sending location of the resource
11:    Calculate  $\leftarrow$  distance ( $BS_i, resource_i$ )
12:    Update table  $\leftarrow$  information(distance)
13:    Calculate  $\leftarrow distance_1(gatewayresource_i, resource_i)$ 
14:    Update  $table_1 \leftarrow$  information(distance)
15:   else
16:      $BS_i \leftarrow$  sleep
17:      $gatewayresource_i \leftarrow$  sleep
18:   end if
19: end for
20: for ( $i = 1$  to  $n$ ) do
21:    $region_i \leftarrow$  divide ( $resource_i$ ) // based on location, dividing resources in to region
22: end for
23: for ( $region_i = 1$  to  $n$ ) do
24:   if ( $region_i(CH_i) == present$ ) then // Cluster based selection
25:      $BS_i \leftarrow$  communicate data( $region_i(CH_i)$ )
26:   else
27:     select  $\leftarrow$  rand( $CH_i$ ) // selecting other resource as Cluster-head
28:   end if
29: end for
30: for ( $round(region_i) = 1$  to  $n$ ) do
31:   if ( $egy_i(CH_i) \leq threshold$ ) then
32:     Make  $\leftarrow$  TDMA schedule( $resource_i$ ) // Scheduling Phase
33:      $CH_i \leftarrow$  collect data( $resource_i$ )
34:      $gatewayresource_i \leftarrow$  send data( $CH_i$ )
35:      $BS_i \leftarrow$  send data( $gatewayresource_i$ )
36:   else
37:     Select  $\leftarrow$  rand(CH)
38:     gatewayresource  $\leftarrow$  sleep
39:      $resource_i \leftarrow$  sleep
40:     Communication  $\leftarrow$  Halt
41:   end if
42: end for
43: End
```

it asks query and finds the results on the basis of output provided by the other resources. The location of resources are send to *BS* which provides *gatewayresource* to access the information from the connected resources. Also, it stores the information in *table* having data related to *resources, gatewayresources, BS*. If there is no connected resource to provide the information, then *BS* and *gatewayresource* goes in to sleep mode for each resource. The entire region is divided in to multiple regions on the basis of their distance with the *BS* via multi-hopping. For each region, it selects a cluster-head, *CH* which acknowledge data to send it to *BS* after checking its threshold, *egy*. If it has less *egy*, then it selects another *CH* randomly for sending data. The *CH* communicates the data to the *gatewayresource* which further forwards the aggregated data to the *BS*. Thus, *BS* has complete information in less time and the communication ends.

In summary, the discovered resources as *NS* undergo calculation of trust score which help in decision making through *FDA*. The selected resources are forwarded using *Hybrid M-Gear* protocol.

## 6.4 Results and Discussion

The proposed *MARRS-IoT* is simulated and analyzed on *MATLAB* to find and select the resource autonomously for efficient communication, searched either *locally* or *globally*. For testing purpose, the *MARRS-IoT* is tested on *ETC*, as application area described in Subsection 4.4.1.

### 6.4.1 Framework Analysis

*MARRS-IoT* is analyzed for the authentication of gathered data, *i.e.*, collected from toll plaza as shown in Figure 4.3. The gathered data includes transaction identification, *trans-ID*, and Barcode Number, *Bar-ID*, vehicle registration identification and *vehicle-ID*, which is acquired using *FastTag*. The various parameters that are taken for simulation of *MARRS-IoT* are shown in Table 6.1.

**Table 6.1: Simulation Parameters**

<b>Parameters</b>	<b>Values</b>
Connecting Time	< 3 (seconds)
Processing Time	< 5 (seconds)
Execution Time	< 75 (seconds)
Simulation Time	1000 ms
Number of Resources	20, 30, 35, 40, 50, 100, 120
Computation of Resources	<i>NDA</i> , Greedy, Description Logic
Decision Making	<i>FDA</i> , Particle Swarm, Ant Colony, Binary Genetic
Protocols	<i>Hybrid M-Gear, LEACH, Mod-LEACH</i>
CPU Throughput	10-100 (%)
Max. TX Distance (single-hop)	35 m
Max. TX Distance (multi-hop)	1000 m
Min. CH candidate range	70 %
Max. CH candidate range	90 %
Packet with error ratio (1/1000)	10

The gathered dataset at toll plaza undergoes computation of processing and connecting time both *locally* and *globally*, using Neighbor Discovery Algorithm (*NDA*) (see Algorithm 6.1). The algorithm considers gathered data as initialized set of resources. The resource's relative position (local or global) is analyzed using spatial function and its observation vector is computed, *i.e.*, path covered by resource during the journey. The knowledge is acquired on the *ETCs* system gathered *locally* or *globally*, for the resource *via* matching the *trans-ID*, *vehicle-ID* and vehicle registration. Thus, the resource discovery considers the connection and processing of resources with respect to time searched on the basis of local and global trust as shown in Table 6.2 and Figures 6.4-6.5. The time is crucial in terms of communication and with the increase in the number of resources, the latency in any sense can prove to be fatal for *MARRS-IoT*. The response time reduces if the processing time among resources increases with the complex knowledge formation in *local* whereas varies

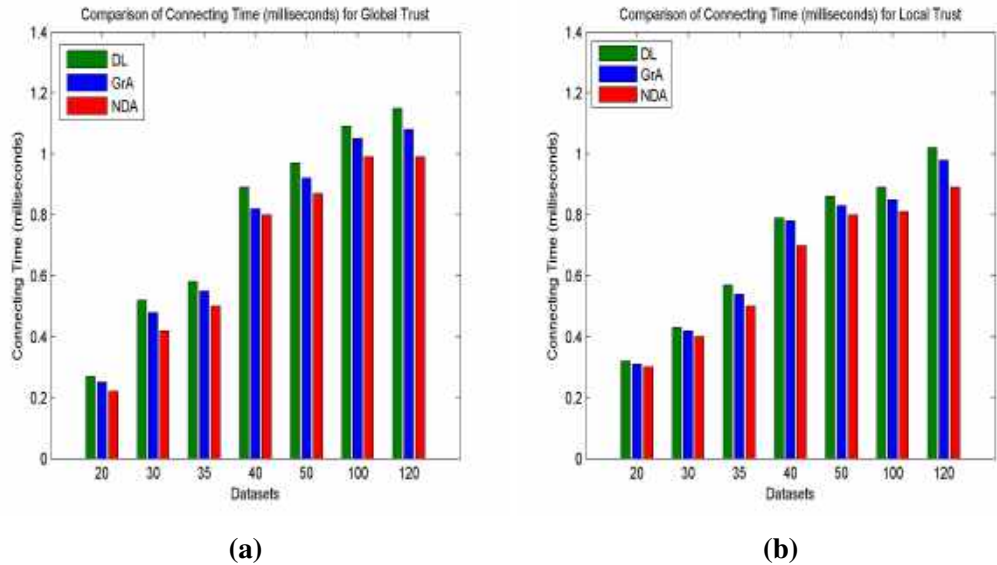
in *global* networks. Moreover, lesser is the connecting time, more robust is the framework. The obtained results are discussed.

**Table 6.2:** Comparison for Computation of Resources by Various Approaches

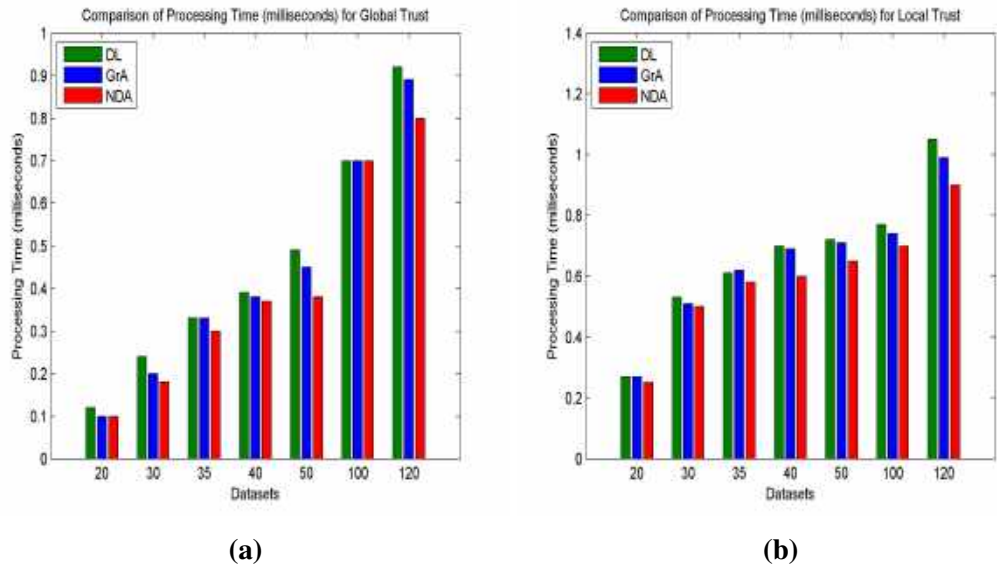
Parameter	Datasets	Global Trust			Local Trust		
		DL	GrA	NDA	DL	GrA	NDA
Connecting Time	20	0.27	0.25	0.22	0.32	0.31	0.3
	30	0.52	0.48	0.42	0.43	0.42	0.4
	35	0.58	0.55	0.5	0.57	0.54	0.5
	40	0.89	0.82	0.8	0.79	0.78	0.7
	50	0.97	0.92	0.87	0.86	0.83	0.8
	100	1.09	1.05	0.99	0.89	0.85	0.81
	120	1.15	1.08	0.99	1.02	0.98	0.89
Processing Time	20	0.12	0.1	0.1	0.27	0.27	0.25
	30	0.24	0.2	0.18	0.53	0.51	0.5
	35	0.33	0.33	0.3	0.61	0.62	0.58
	40	0.39	0.38	0.37	0.7	0.69	0.6
	50	0.49	0.45	0.38	0.72	0.71	0.65
	100	0.7	0.7	0.7	0.77	0.74	0.7
	120	0.92	0.89	0.8	1.05	0.99	0.9

To evaluate *NDA*'s performance for computing resources with respect to connecting and processing time, it is compared with existing algorithms in same domain, *i.e.*, Description Logic (*DL*) and Greedy Approach (*GrA*) [190]. These approaches are chosen for computing resources as these deal with matchmaking criteria based on input query. The *DL* helps to represent knowledge of service in a structured way and provides ranking for match. It recursively calls itself for universal quantification of resources and stores global score as degree of match. The resource having maximum degree is shortlisted. On the other hand, *GrA* matches every concept of query with the each concept of resource. It begins from all the output concepts and remove them from generated list as soon as it is matched with concept from resource.

The performance of *NDA*, *DL* and *GrA* are tested and evaluated for processing datasets of 20, 30, 35, 40, 50, 100 and 120 resources. It is examined on computation of processing and connecting time both *locally* and *globally*. The results are shown in Table 6.2 and Figures 6.4-6.5.



**Figure 6.4:** Resource Discovery on the Network for its Connecting Time during (a) Global Trust (b) Local Trust



**Figure 6.5:** Resource Discovery on the Network for its Processing Time during (a) Global Trust (b) Local Trust

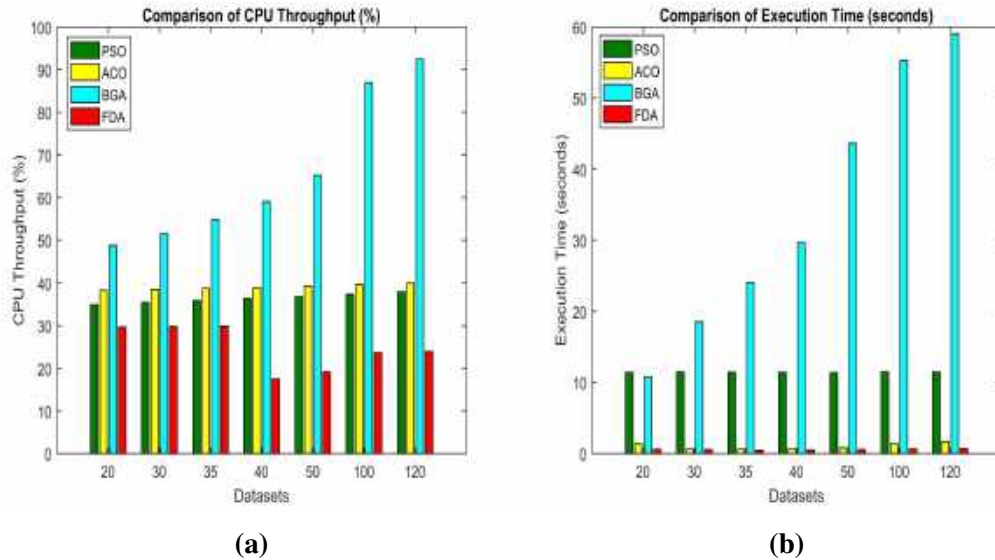
From Figure 6.4, it is found that *DL* takes maximum connecting time (both *locally* and *globally*) with increase in number of resources. In comparison, *GrA* takes lesser time and *NDA* takes minimal connecting time for computation of resources. Thus, the proposed *NDA* comes out to be an efficient resource discovery technique for connecting maximum number of resources in minimum time. It is observed from Figure 6.5 that *DL*, *GrA* take maximum

and average processing time, respectively, in computing resources. In comparison, *NDA* provides minimum processing time, both *locally* and *globally*.

Thus, it is concluded that *NDA* offers more resources by connectivity them in minimum time. This is due to the fact that trust score generator provides an array of shortlisted resources to provide sufficient options for recovering from network faults and halts that hinder the data transmission. The communication range of neighbor sets with respect to increase in resources both *locally* and *globally* shows an increase in processing time denoting that *MARRS-IoT* is capable to process large number of resources in minimum processing time using proposed *NDA*. Thus, *MARRS-IoT* proves to be highly scalable. Also, the results show an increase in confidentiality of *MARRS-IoT*. Moreover, the graphs indicate that with increase in number of resources in the network (*locally* or *globally*), collision will occur due to more resources, receiving the data. Thus, for the scenario, to minimize the collision, the decision considers the computation and evaluation of the maximum and minimum distances for *local* and *global* networks, respectively, for the proposed *FDA*.

#### **6.4.2 Performance Evaluation of Framework**

*MARRS-IoT* is tested and examined for decision making. Each gathered resource is assigned a trust score which is either *0* or *1* on the basis of trust evaluation. Here, *0* indicates no trust and *1* represents trust. The trustworthy resources undergoes decision making process using proposed Fuzzy-Decision Algorithm, *FDA*, (see Algorithm 6.3). An array is considered with trust score value equal to *1*. The resources as participants is initialized to *0*. Each participant is checked for its presence with two conditions; either present locally or present globally. For both the resultant arrays are generated after sorting for both *local* and *global* evaluations. Further, the distance between initiator and participating resources is calculated using newly generated arrays. The resources are shortlisted from both arrays and are checked against their availability. Finally, maximum and minimum distances for *local* and *global* networks are computed. To evaluate *FDA*s performance for *MARRS-IoT*, it is compared with existing algorithms, *i.e.*, *PSO*, *ACO* and *BGA* [191]. It is tested and evaluated for the given data sets of 20, 30, 35, 40, 50, 100 and 120 resources. The performance is examined on the basis of its execution time and consuming *CPU* cycles. If the *CPU* cycles is more, then it consumes



**Figure 6.6:** Comparison among algorithms for decision making with respect to (a) CPU Cycles (b) Execution time

more energy and time for selecting the best resource.

These algorithms are tested on system having configuration as *Intel i5 4th generation quad-core processor with 4 GB RAM and 2 GB NVidia card*. The objective is that once a connection is established among resources, the process is triggered repeatedly to select the best resource. The performance results are shown in Table 6.3.

From Figure 6.6(a), it is found that *BGA* takes maximum *CPU* consumption for processing the resources iteratively. In comparison, *ACO* and *PSO* take less *CPU* consumption and thus leads to resolve the congestion on server due to multiple queries processed at toll plazas. It is concluded that proposed *FDA* takes less *CPU* cycles for processing queries and increases its efficiency with less load on server. Thus, the proposed method comes out to be the efficient decision making technique to provide the best resource in less time, *i.e.*,  $< 1$  seconds.

It is observed from Figure 6.6(b) that *BGA* takes the maximum execution time that results into slow down the decision making process and thus, leading to the problems of traffic congestion. In comparison, *ACO* outperforms better than *BGA* and *PSO* takes less execution time in selecting the best resource. The proposed *FDA* takes less execution time for selecting the best resource as compared with existing benchmark algorithms.

Thus, it is concluded that *FDA* provides best resource in less time, consuming minimum *CPU* cycles.

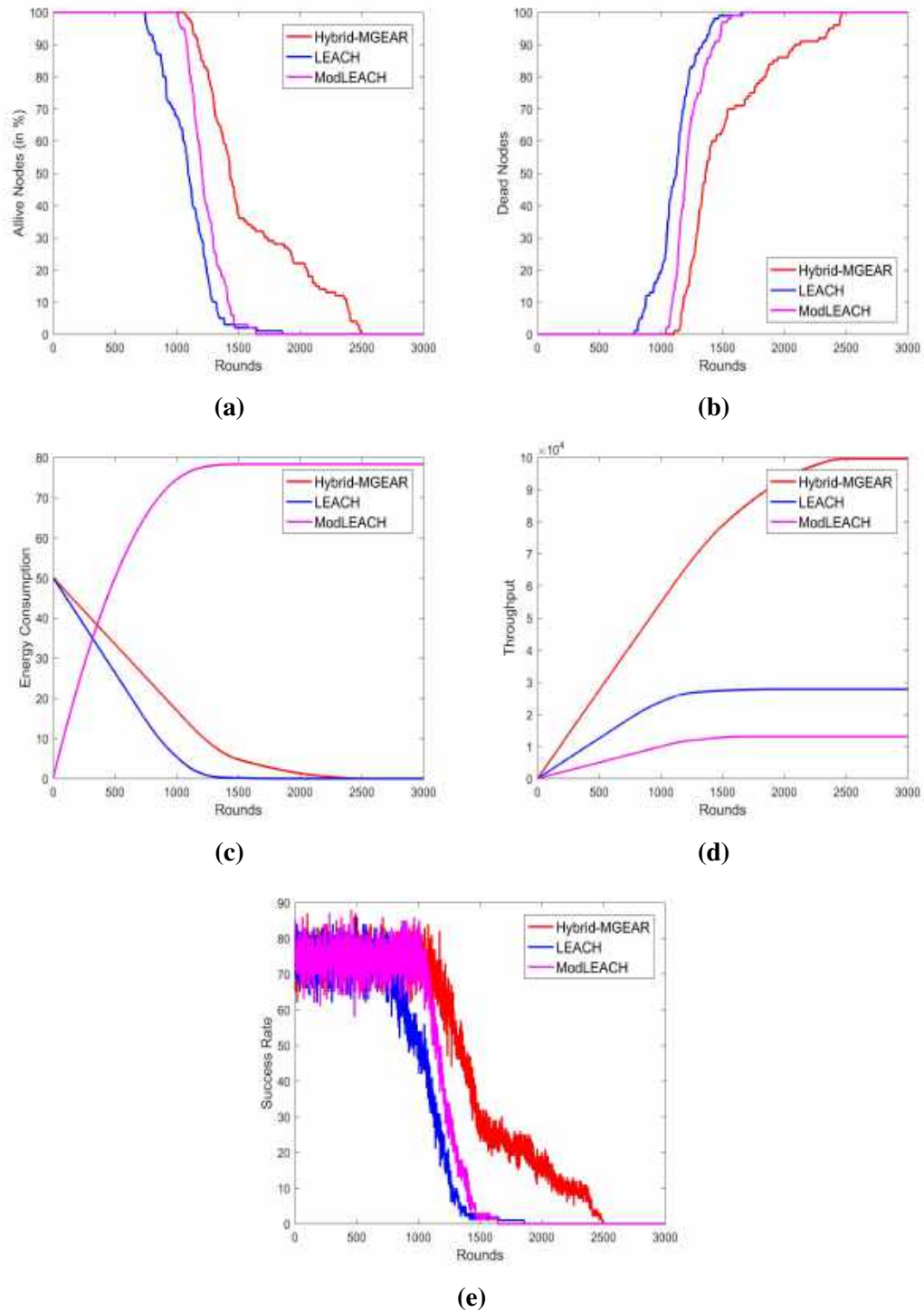
**Table 6.3:** Comparison among approaches

<b>Datasets</b>	<b>Approaches</b>	<b>Execution Time (in sec)</b>	<b>CPU Cycles (in %)</b>
20	PSO	11.43	34.92
	ACO	1.38	38.26
	BGA	10.84	48.82
	<b>FDA</b>	<b>0.59</b>	<b>29.7</b>
30	PSO	11.49	35.39
	ACO	0.6	38.5
	BGA	18.53	51.54
	<b>FDA</b>	<b>0.55</b>	<b>29.78</b>
35	PSO	11.44	35.81
	ACO	0.61	38.77
	BGA	24.1	54.94
	<b>FDA</b>	<b>0.45</b>	<b>29.87</b>
40	PSO	11.44	36.33
	ACO	0.68	38.96
	BGA	29.63	59.15
	<b>FDA</b>	<b>0.52</b>	<b>17.49</b>
50	PSO	11.38	36.80
	ACO	0.82	39.29
	BGA	43.62	65.22
	<b>FDA</b>	<b>0.56</b>	<b>19.14</b>
100	PSO	11.5	37.39
	ACO	1.4	39.6
	BGA	55.25	86.92
	<b>FDA</b>	<b>0.65</b>	<b>23.66</b>
120	PSO	11.48	37.87
	ACO	1.7	39.96
	BGA	59.02	92.46
	<b>FDA</b>	<b>0.75</b>	<b>23.9</b>

### 6.4.3 Communication

*MARRS-IoT* communicates data via *Hybrid M-Gear* (see Algorithm 6.4). The *Hybrid M-Gear* through base station, *BS*, broadcasts *hello packet* to the other resources to join the network, *i.e.*, it asks query and finds the results on the basis of output provided by the other resources. The location of resources are send to *BS* which provides gateway to access the information from the connected resources. If there is no connected resource to provide the information, then *BS* and gateway goes in to sleep mode for each resource. The entire region is divided in to multiple regions on the basis of their distance with the *BS* via multi-hopping. Each region has its own cluster head, *CH* that gathers the data from the region. The gathered data is aggregated to *BS* via gateway through *CH*. Thus, *BS* has complete information in less time and the communication ends. In order to evaluate its efficiency, it is compared to two existing protocols, *namely*, *LEACH* [192] [193] and *Mod-LEACH* [194]. These algorithms are simulated on *MATLAB* for defined *area* of  $100 \times 100$ . The simulation is processed for *3000 rounds* and its evaluated results are discussed below.

- (A) **Alive resources:** The network works for maximum period till resources are alive for communication. The number of alive resources enhance the network' performance with maximum delivery of data. Figure 6.7(a) shows that total number of alive resources during the data transmission. It is found that *Hybrid M-Gear* returns maximum number of resources for communication. In comparison, less number of alive resources are observed in *Mod-LEACH*, followed by *LEACH*. Thus, *Hybrid M-Gear* provides better utilization of resources.
- (B) **Dead resources:** The network's lifetime is defined by the number of resources that are retained during the transmission whereas dead resources highlight the network's deficiency to communicate complete data to the destination. Figure 6.7(b) presents the dead resources during communication. It is observed that energy of the resources is worn out rapidly in the *LEACH* as most of the energy is wasted in updating the table. In *Mod-LEACH*, it is comparatively less and retained up to 1200 rounds. The *Hybrid M-Gear* provides the maximum energy utilization and retaining resources for communication up to 2500 rounds. It suggests that *Hybrid M-Gear* is best among three



**Figure 6.7:** System Performance (a) Alive resources (b) Dead resources (c) Energy consumption (d) Throughput (e) Success Rate

approaches.

- (C) **Energy Consumption:** There is another aspect to find that how much energy is consumed during communication. If energy consumed is less, then the resources are retained for longer period. Figure 6.7(c) shows the energy consumption of the routing protocols. It depicts that *Mod-LEACH* consumes maximum energy and is highly reactive such that resources worn out easily whereas *Hybrid M-Gear* conserves energy for longer time. Also, *LEACH* retains the energy of the resources but as number of rounds are increasing, its energy gets depleted due to active state of cluster heads in all rounds.
- (D) **Throughput:** The throughput evaluates the efficiency of the network on the basis of maximum data delivery from source to destination and is fully dependent on the network's lifetime, *i.e.*, energy consumption of the resources. Figure 6.7(d) shows the maximum packets to be delivered to the base station. It is observed that *Mod-LEACH* provides minimum packet delivery that results in to minimum throughput. In contrast, *LEACH* has moderate throughput and *Hybrid M-Gear* guarantees to deliver maximum packet with highest throughput. Thus, *Hybrid M-Gear* sends complete data to the destination.
- (E) **Success Rate:** It suggests for successful delivery of packets to destination with accuracy to ensure network's efficiency. Figure 6.7(e) presents the success rate of the packets. It is found that *LEACH* provides the minimum delivery of packets. *Mod-LEACH* shows the average delivery of packets whereas *Hybrid M-Gear* provides maximum success rate during data transmission. Thus, *Hybrid M-Gear* does not loose the quality of data.

Thus, it is concluded that *Hybrid M-Gear* is better and energy efficient routing protocol for communication among reliable resources.

#### 6.4.4 Complexity Analysis

The set of resources,  $N = \{n_1, n_2, \dots, n_k, \dots, n_n\}$  are considered that are in 2-D area. The resource 'k' is located at 2-D position,  $p_{k,l}$ , having the tendency to make point observation,  $Y_{k,l}$ , of a spatial function,  $F_{p_{k,l}}$ ; is searched at 'lth' step due to its self-discoverable properties.

Therefore, the computational time complexity for searching a resource is deduced using Eq. 6.1 as given below:

$$Y_{k,l} = F_{p_{k,l}} + \varepsilon_{k,l} = O(N^2) + O(N) = O(N^2) \quad (6.27)$$

The computation and allocation of resources take place by taking the information from the nodes in prediction range set,  $B_k$ . Thus, the time complexity for allocating resources is deduced using Eq. 6.9, and is given as:

$$B_k = \sum_{k=1}^n \sum_{j=1}^{|N_k|} ER_j = \sum_{k=1}^n O(N) = O(N^2) \quad (6.28)$$

These resources route their data to the base station,  $BS$  using *Hybrid M-Gear*. It evaluates the shortlisted resources through which data with accuracy is selected. Based on its working methodology, its complexity is analyzed in terms of trust evaluation metrics as discussed below.

(A) *Trust Evaluation Locally*: For trust evaluation locally,  $T_{i,j}$ , it considers both direct and indirect trust and its computational time complexity is deduced using Eq. 6.25 as given below:

$$\begin{aligned} T_{i,j} &= W_d \times DT_{i,j} + W_{ind} \times \Sigma(DT_{i,l} \times IT_{l,j}) = O(N) \times O(N) + O(N) \times \Sigma O(N) \\ &= O(N^2) + O(N^2 \log N) \quad (6.29) \\ &= O(N^2 \log N) \end{aligned}$$

where,  $O(N^2 \log N)$  defines the number of iterations performed in direct and indirect trust. In terms of storage, its complexity is given by:

$$T_{storage} = R_d(DT) + R_d(IDT) = O(N) + O(N) = O(N) \quad (6.30)$$

where,  $R_d(DT)$  and  $R_d(IDT)$  are the storage repositories for the data that are required from the resources. The data is stored in the form of sorted array list and is evaluated to select best matched search parameters.

(B) *Trust Evaluation Globally*: Its computation time is computed using Eq. 6.23 and is

given as:

$$\begin{aligned} T_{i,j} &= \lim_{n \rightarrow \infty} [(1 + r + r^2 + \dots + r^n) \cdot d] \\ &= O(N) + O(N) \times O(N) = O(N^2) \end{aligned} \quad (6.31)$$

where,  $O(N^2)$  defines the number of iterations in multiple routes using indirect trust. Its storage complexity is given by:

$$GT_{storage} = R_d(IDT) + R_d(sortIDT) = O(N) + O(N) = O(N) \quad (6.32)$$

where,  $R_d(IDT)$  and  $R_d(sortIDT)$  are the storage repositories for the gathered data by indirect trust and evaluated gathered data after sorting, respectively. Here, it takes another storage for processing the data and selecting the best matched parameter.

### 6.4.5 Resolving Security Issues

The various security constraints, *i.e.*, authentication, confidentiality, integrity hamper the performance of the system due to the presence of the malicious resources. They are resolved by *MARRS-IoT* and are discussed below.

- (A) *Authentication through Communication: MARRS-IoT* enhances systems' performance by detecting the unwanted response while communicating with the malicious resources. It also authenticates the neighboring resources joining the network by direct and indirect trust computation.
- (B) *Confidentiality: In order to recognize malicious spies, MARRS-IoT* tends to separate both peer's trust value who act as service provider and confidence in their recommendations based on build reputations. Thus, it provides confidential resources to communicate data to the end user *via* routing protocol.
- (C) *Integrity: In order to avoid stable attack, MARRS-IoT* provides low trust score to the new joining resources such that malicious resources can be recognized and does not receive services. Thus, it maintains integrity for the communication among the resources.

From the above analysis, it is observed that *MARRS-IoT* helps to resolve security constraints via trust based decision making (*FDA*) and routing through *Hybrid M-Gear*.

## 6.5 Conclusion

The chapter has introduced “Middleware Approach for Reliable Resource Selection on Internet-of-Things (*MARRS-IoT*)” which is self-configurable, explorable, adaptive and efficient. It allows resource discovery through Neighbor Discovery Algorithm and trust-based *FDA*. It evaluates trust score using Fuzzy-trust model and reputation to perform optimal decision making on the basis of the availability of resources, both locally and globally. The shortlisted resources are routed using *Hybrid-MGear* protocol. The experimental results justify the proposed *MARRS-IoT* integration with service models such as *ETC* for automation and data transmission in order to minimize traffic congestion. From the experimental analysis with respect to *CPU*’ cycles and execution time, it is concluded that *MARRS-IoT* provides automated decision making with lower latency, higher accuracy and reliability. Hence, it efficiently selects the rightful resource in less time as compared with state-of-the-art algorithms. Further, it provides security with respect to constraints like authentication, confidentiality, integrity.

In future, *MARRS-IoT* could be extended for the selection of leader resource which is responsible for the synchronization of the other resources. Another possible extension could be to deal with the failure of the leader resources, *i.e.*, fault tolerance.

---

# Conclusion and Future Scope

---

## 7.1 Conclusion

The resources on *IoT* are exponentially growing in the billions and are expected to generate huge amount of data in zettabytes by 2020. In order to have efficient discovery, selection, ranking, access, integration, understanding of the data and services from these resources, it would require coordinated efforts among network, data/service provider, and core *IoT* components. It would facilitate in selection and usage of the best resources at the right time and location. Due to distribution, scale, heterogeneity, multi-modality, streaming data and dynamicity of *IoT* environments, traditional solutions for searching, accessing will not be applicable or will remain far from adequate for practical and large-scale dynamic *IoT* applications. To address the problem, the thesis has proposed various techniques for resource discovery and selection that cover the data management, knowledge representation and interpretation.

A novel discovery technique, *namely*, *IKm-CA* is proposed that intends to cluster formation iteratively to search resources through matching criteria. The criteria emphasizes on existence of relation between two points with respect to threshold value whereas similarity based clustering using Cosine coefficient of vector space model provides better evaluations than Jaccard, Dice and Euclidean, respectively. With these evaluation, *IKm-CA* in

collaboration with Cosine coefficient is simulated using *MATLAB* against exponentially growing nodes to analyze the search efficiency. The results supports algorithm's efficiency with minimum complexity of  $O(N)$ . Thus, it is an optimized and efficient approach for resource discovery.

A framework, *CASOF-IoT* is proposed that performs efficient resource discovery through its four layers, namely, *CAL*, *CASL*, *SOL* and *SDL*. The layers gather, model data *via* schema for knowledge representation and perform searching with Multi-Modal search algorithm. The searched results are further optimized with *IGD* algorithm that are accessed through interfaces. The search algorithm provides concise view and best matches against query from Red-Stores. The *IGD* algorithm reduces the search complexity through Gradient Descent using Armijo's Rule which eradicates unpredictable step size. The algorithm enables the user to decrease and update the step size for finding gradient points and to obtain best neighbor with less computational complexity. Moreover, it provides continuous update of search sets with iterative process that results into more efficient technique when compared with *Newton* and *Quasi-Newton* methods. The obtained results validate its search efficiency by taking less time, *i.e.*,  $0.1s$  to  $-0.35s$  for discovering the resources. Additionally, the system performance is analyzed through various metrics such as energy consumption, delivery of packets, data loss, throughput, quality of data and latency. The minimum operation complexity of *IGD* is  $O(N)$  providing efficient search for resources.

Another framework, *i.e.*, *IRIF-IoT* emphasizes on semantics for knowledge representation and automated reasoning. The technique implements *SMEBG* Algorithm, which is further optimized with Hungarian Algorithm having strong polynomial time bound complexity of  $O(|V_x|^3)$ . The approach is tested for application, namely, *ETC*, and is compared with *FCL* and *GA* using *AIMSUN* software. The results are analyzed on the basis of system performance, performance metrics like power consumption, throughput, packet delivery rate, packet loss, latency and data quality and complexity. From the analysis, it is concluded that the proposed approach has better search efficiency with minimal operational complexity of  $O(N)$ . Hence, it enhances system performance significantly.

The proposed *MARRS-IoT* is self-configurable, explorable, adaptive and efficient. It allows resource discovery through *NDA* and performs selection using *FDA*. It evaluates trust score

using Fuzzy-trust model and reputation to perform optimal decision making on the basis of the availability of resources, both locally and globally. The shortlisted resources are routed using proposed *Hybrid-MGear* routing algorithm. The experimental results justify the proposed framework's integration with service models such as *ETC* for automation and data transmission in order to minimize traffic congestion. From the experimental analysis with respect to *CPU*' cycles and execution time, it is concluded that *MARRS-IoT* provides automated decision making with lower latency, higher accuracy and reliability. Hence, it efficiently selects the rightful resource in less time as compared with state-of-the-art algorithms.

An analysis of different frameworks and algorithm is done using various parameters as discussed in Table 7.1.

**Table 7.1:** Comparison among Proposed Techniques

<b>Parameters</b>	<b><i>IKm-CA</i></b>	<b><i>CASOF-IoT</i></b>	<b><i>IRIF-IoT</i></b>	<b><i>MARRS-IoT</i></b>
Architecture	Distributed	Centralized	Centralized	Distributed
Data Acquisition	Request and Response	Publish and Subscribe	Request and Response	Request and Response
Data Gathering	Observational	Measurement	Measurement	Measurement
Data Modeling	Clustering	Key-Value	Ontology	Gaussian Distribution
Scalability	×	✓	✓	✓
Searching Type	Location based	Network based	Network based	Network based
Semantic Matchmaking	×	×	✓	×
Clustering	✓	×	×	×

to be cont'd on next page

**Table 7.1:** Comparison among Proposed Techniques (contd.)

<b>Parameters</b>	<b><i>IKm-CA</i></b>	<b><i>CASOF-IoT</i></b>	<b><i>IRIF-IoT</i></b>	<b><i>MARRS-IoT</i></b>
Searching Approach	Iterative k-means	Multi-Modal Search	Semantic Matchmaking Engine using Bipartite Graph	Neighbor Discovery
Indexing	✓	✓	✓	✓
Ranking	✓	✓	✓	✓
Decision Making Approach	×	Iterative Gradient Descent	Hungarian	Fuzzy based
Simulator	<i>MATLAB</i>	<i>MATLAB, AIMSUN</i>	<i>MATLAB, AIMSUN</i>	<i>MATLAB</i>
Trust Evaluation	×	×	×	✓
Routing	×	×	×	✓
Security Constraints	×	×	×	✓
Optimization	✓	✓	✓	✓
Conditions	Heterogeneity	Heterogeneity, Context of data, Scalability	Heterogeneity, Scalability, Interoperability	Heterogeneity, Scalability, Interoperability, Trust

to be cont'd on next page

**Table 7.1:** Comparison among Proposed Techniques (contd.)

<b>Parameters</b>	<b><i>IKm-CA</i></b>	<b><i>CASOF-IoT</i></b>	<b><i>IRIF-IoT</i></b>	<b><i>MARRS-IoT</i></b>
Applicability	Identification, Communication, Searching	Identification, Communication, Architecture, Network and Searching	Identification, Communication, Architecture, Network	Identification, Communication, Architecture, Network, Searching, Energy Optimization, Security and Privacy

## 7.2 Future Scope

In future, the proposed technique, *IKm-CA* could be performed for data control and resolving traffic congestion for applications like smart home, electronic toll plazas. It could be integrated with *IPv6* to ease the network management, to provide auto configuration capabilities and improved security features. It could be extended for evaluating resources' metadata that helps to rank the discovered resources. Also, it can be used for computing *RGB* images to provide robustness and provide feasibility to deal with noise. The approach of Euclidean Distance method can be replaced by Induced Kernel method to provide similarity match in less time. In *IKm-CA*, the user needs to specify number of cluster in advance, therefore, it could be improved where automated clusters can be decided in accordance with input resources. Moreover, with the distance measurement methods, the system would act independently for the selection of initial centroids. Further, *IKm-CA* can be implemented in the field of time sampling where data of resources is received in regular periods.

In *CASOF-IoT*, the respective algorithms could be extended for multiple neighbor selection with respect to various applications across distinct domains on *IoT*. Also, the privacy and security concerns need to be resolved with increase in the number of resources. Moreover,

a new indexing scheme for any web based search could be adopted on the basis of *SVK* model to provide more efficient matches against query. A ranking scheme can also be implemented in *CASOF-IoT* to minimize constrained search. It should urge a crawler to find more documents that subject to a particular resource search within described limits. This, in turn, could be ranked and indexed with the parameters of an optimal solution to provide a more categorical ranking that would help in better information retrieval.

The *IRIF-IoT* could be extended with respect to security, privacy and trust for real distributed environment where the connectivity between resources is virtual. Further, the software agents could be integrated to the framework for secure coordination and communication among various stations such that resources could be scheduled. Moreover, tracking and managing the high number of resources that would be involved in the integration of *IoT* in *ITS* remains a significant challenge. *IRIF-IoT* needs to define novel approaches for device identification and the generation of unique Identifiers (*IDs*). Moreover, a distributed set of wireless sensor nodes could be used to detect vehicular behavior (speed, direction, flow cuts, and travel times) and sends this information to processing devices (these devices could be integrated within the same environment as the sensors or within a cloud environment) to make intelligent decisions such as dynamic time for traffic light and changing the lanes available in each direction.

In *MARRS-IoT*, database should be provided where data can be stored and processed in virtual servers through cloud computing (middleware element). This information can be used in the future to model traffic and drivers' behaviors, using wheel drive sensors or diadems to read neural signals to detect mood changes, stress levels, or general body condition. It would enable the notifications to be sent to nearby vehicles about dangerous or aggressive driving patterns, allowing them to avoid accidents. Also, *MARRS-IoT* could be extended for the selection of leader resource which is responsible for the synchronization of the other resources. Another possible extension could be to deal with the failure of the leader resources, *i.e.*, fault tolerance. Further, *MARRS-IoT* can be used for the development of federated systems that allow different devices from multiple environments to seamlessly move from one network to another using the same authentication credentials.

## References

- [1] P. F. H. Sundmaeker, P. Guillemin and S. Woelffle, “*Vision and Challenges for Realizing the Internet of Things*,” European Commission, Directorate-General for the Information Society and Media, Publications Office of the European Union, pages 229, 2010.
- [2] D. Uckelmann, M. Harrison, and F. Michahelles, “*Architecting the Internet of Things*”, 1st ed. Springer Publishing Company, Incorporated, 2011.
- [3] A. Al. Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, “Internet of things: A Survey on Enabling Technologies, Protocols, and Applications, ” *IEEE Communications Surveys & Tutorials*, Vol. 17, No. 4, pp. 2347–2376, 2015.
- [4] N. Koshizuka and K. Sakamura, “Ubiquitous ID: Standards for Ubiquitous Computing and the Internet of Things,” *IEEE Pervasive Computing*, Vol. 9, No. 4, pp. 98–101, 2010.
- [5] C. P. P. Schumacher, N. Kushalnagar, and G. Montenegro, “IPv6 over Low-Power Wireless Personal Area Networks (6LOWPANS): Overview, Assumptions, Problem Statement, and Goals,” 2007.
- [6] N. Kushalnagar, G. Montenegro, D. E. Culler, and J. W. Hui, “Transmission of IPv6 Packets over IEEE 802.15.4 Networks”, *Internet Engineering Task Force*, Minneapolis, 2007. [Online]. Available: <https://tools.ietf.org/html/rfc4944>
- [7] R. Aggarwal and M. L. Das, “RFID Security in the Context of ”Internet of Things,” in *Proceedings of the First International Conference on Security of Internet of Things (SecurIT '12)*, New York, NY, USA: ACM, pp. 51–56, 2012. [Online]. Available: <http://doi.acm.org/10.1145/2490428.2490435>
- [8] K. Finkenzeller, “*RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*,” 2nd ed. Wiley Publishing, 2003.
- [9] G. V. Crosby and F. Vafa, “Wireless Sensor Networks and LTE-A Network Convergence, ” in *Proceedings of IEEE 38th Conference on Local Computer Networks (LCN)*, pp. 731–734, 2013.
- [10] A. Dunkels, B. Gronvall, and T. Voigt, “CONTIKI-A Lightweight and Flexible Operating System for Tiny Networked Sensors,” in *Proceedings of 29th Annual IEEE International Conference on Local Computer Networks*, pp. 455–462, 2004.

- [11] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer *et al.*, “TINYOS: An Operating System for Sensor Networks,” in *Ambient intelligence*, pp. 115–148, 2005.
- [12] Q. Cao, T. Abdelzaher, J. Stankovic, and T. He, “The LiteOS Operating System: Towards Unix-like Abstractions for Wireless Sensor Networks,” in *Proceedings of IEEE International Conference on Information Processing in Sensor Networks (IPSN’08)*, pp. 233–244, 2008.
- [13] E. Baccelli, O. Hahm, M. Gunes, M. Wahlsch, and T. C. Schmidt, “RIoT OS: Towards An OS for the Internet of Things,” in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPs)*, pp. 79–80, 2013.
- [14] M. M. Hassan, B. Song, and E.N. Huh, “A Framework of Sensor-Cloud Integration Opportunities and Challenges,” in *Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication (ICUIMC ’09)*, pp. 618–626, 2009. [Online]. Available: <http://doi.acm.org/10.1145/1516241.1516350>
- [15] P. Barnaghi, W. Wang, C. Henson, and K. Taylor, “Semantics for the Internet of Things: Early Progress and Back to the Future,” *International Journal on Semantic Web and Information Systems (IJSWIS)*, Vol. 8, No. 1, pp. 1–21, 2012.
- [16] L. M. Xing Xiaojiang, and Wang Jianli, “Services and Key Technologies of the Internet of Things,” *ZTE Communications*, No. 2, 2010.
- [17] O. Vermesan, P. Friess, P. Guillemin, H. Sundmaeker, M. Eisenhauer, K. Moessner, M. Arndt, M. Spirito, P. Medagliani, R. Giaffreda, S. Gusmeroli, L. Ladid, M. Serrano, M. Hauswirth and G. Baldini, “*Internet of Things Strategic Research Agenda*,” River Publisher, pp. 7-142, 2014.
- [18] P. Friess, G. Woysch, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, M. Eisenhauer and K. Moessner, “*Europe’s IoT Strategic Research Agenda*,” in IERC - Internet of Things European Research Cluster, The Internet of Things 2012 New Horizons, Chapter 2, 2012.
- [19] P. C. Evans and M. Annunziata, “*Industrial Internet: Pushing the Boundaries of Minds and Machines*,” Imagination at Work, 2012.
- [20] O. Vermesan, L.-C. Blystad, R. Zafalon, A. Moscatelli, K. Kriegel, R. Mock, R. John, M. Ottella, and P. Perlo, “Internet of Energy—Connecting Energy Anywhere Anytime,” in *Advanced Microsystems for Automotive Applications*, Springer, pp. 33–48, 2011.
- [21] T. Quang, M. A. Baharudin, and E. Kamioka, “Synergistic Approaches to Mobile Intelligent Transportation Systems Considering Low Penetration Rate,”
- [22] G. Pardo-Castellote, “OMG Data-Distribution Service: Architectural Overview,” in *Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCSW ’03)*, Washington, DC, USA: IEEE Computer Society, 2003, pp. 200–206. [Online]. Available: <http://dl.acm.org/citation.cfm?id=839280.840571>

- [23] O. Bergmann, K. T. Hillmann, and S. Gerdes, "A COAP-Gateway for Smart Homes," in *Proceedings of International Conference on Computing, Networking and Communications (ICNC)*, pp. 446–450, 2012.
- [24] R. Godfrey, and C. Vasters, "*OASIS Advanced Message Queuing Protocol (AMQP)*", OASIS Advancing Open Standards for the Information Society, 2012.
- [25] M. Chan, E. Campo, D. Estave, and J. Y. Fourniols, "Smart Homes Current Features and Future Perspectives," *Maturitas*, Vol. 64, No. 2, pp. 90 – 97, 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378512209002606>
- [26] S. Bendel, T. Springer, D. Schuster, A. Schill, R. Ackermann, and M. Ameling, "A Service Infrastructure for the Internet of Things based on XMPP," in *Proceedings of IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pp. 385–388, 2013.
- [27] R. P. V. Chander, S. Elias, S. Shivashankar, and M. P, "A REST based Design for Web of Things in Smart Environments," in *Proceedings of 2nd IEEE International Conference on Parallel Distributed and Grid Computing (PDGC)*, pp. 337–342, 2012.
- [28] A. Foster, "A Comparison between DDS, AMQP, MQTT, JMS and REST," in *Messaging Technologies for the Industrial Internet and the Internet of Things*, PrismTech Corporation, Ver. 1.2, 2013.
- [29] I. C. Ng and S. Y. Wakenshaw, "The Internet-of-Things: Review and Research Directions," *International Journal of Research in Marketing*, Vol. 34, No. 1, pp. 3–21, 2017.
- [30] B. Butler, Gartner, "Gartner's Hype Cycle Special Report for 2014," Gartner Inc. [Online]. Available: <https://www.gartner.com/newsroom/id/2819918>
- [31] Gartner, "Top Trends in the Gartner Hype Cycle for Emerging Technologies," Gartner Inc., 2017.
- [32] J. A. Stankovic, "Research Directions for the Internet of Things," *IEEE Internet of Things Journal*, Vol. 1, No. 1, pp. 3–9, 2014.
- [33] S. Hammoudi, Z. Aliouat, and S. Harous, "Challenges and Research Directions for Internet of Things," *Telecommunication Systems*, Vol. 67, No. 2, pp. 367–385, 2018.
- [34] O. Vermesan and P. Friess, "*Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems*," River Publishers Series in Communications, pages 364, 2013.
- [35] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Future Generation Computer Systems*, Vol. 29, No. 7, pp. 1645–1660, 2013.

- [36] S. K. Datta, R. P. F. Costa, and C. Bonnet, "Resource Discovery in Internet of Things: Current Trends and Future Standardization Aspects," in *Proceedings of IEEE 2nd World Forum on Internet of Things (WF-IOT)*, 2015. [Online]. Available: <http://www.eurecom.fr/publication/4716>
- [37] S. K. Datta and C. Bonnet, "Describing Things in the Internet of Things: From Core Link Format to Semantic based Descriptions," in *Proceedings of IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, pp. 1–2, 2016.
- [38] A. Sheth, P. Anantharam, and C. Henson, "Physical-Cyber-Social Computing: An Early 21st Century Approach," *IEEE Intelligent Systems*, Vol. 28, No. 1, pp. 78–82, 2013.
- [39] A. Zaslavsky and P. P. Jayaraman, "Discovery in the Internet of Things: The Internet of Things (Ubiquity Symposium)," *Ubiquity*, Vol. 2015, no. October, pp. 2:1–2:10, 2015. [Online]. Available: <http://doi.acm.org/10.1145/2822529>
- [40] F. C. Delicato, P. F. Pires, and T. Batista, "*The Activities of Resource Discovery and Resource Estimation*", Cham: Springer International Publishing, pp. 33–44, 2017. [Online]. Available: <https://doi.org/10.1007/978-3-319-54247-8-4>
- [41] C. Perera and A. V. Vasilakos, "A Knowledge-based Resource Discovery for Internet of Things," *Knowledge-Based Systems*, Vol. 109, pp. 122 – 136, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0950705116302015>
- [42] L. H. Nunes, J. C. Estrella, C. Perera, S. Reiff-Marganiec, and A. N. Delbem, "Multi-Criteria IoT Resource Discovery: A Comparative Analysis," *CoRR*, abs/1611.05172, 2016. [Online]. Available: <http://arxiv.org/abs/1611.05172>
- [43] C. Vandana and A. A. Chikkamannur, "Study of Resource Discovery Trends in Internet of Things (IoT)," *International Journal of Advanced Networking and Applications*, Vol. 8, No. 3, pp. 1–3084, 2016.
- [44] Y. Zhou, S. De, W. Wang, and K. Moessner, "Search Techniques for the Web of Things: A Taxonomy and Survey," *Sensors*, Vol. 16, No. 5, pp. 600, 2016.
- [45] D. Miorandi, S. Sicari, F. D. Pellegrini, and I. Chlamtac, "Internet of Things: Vision, Applications and Research Challenges," *Ad Hoc Networks*, Vol. 10, No. 7, pp. 1497 – 1516, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1570870512000674>
- [46] N. K. Tran, Q. Z. Sheng, M. A. Babar, and L. Yao, "Searching the Web of Things: State of the Art, Challenges, and Solutions," *ACM Computing Surveys*, Vol. 50, No. 4, pp. 55:1–55:34, 2017. [Online]. Available: <http://doi.acm.org/10.1145/3092695>
- [47] M. Weiser, "The Computer for the 21 st Century," *Scientific American*, Vol. 265, No. 3, pp. 94 –105, 1991.
- [48] K. Ashton, "That 'Internet of Things' Thing," *RFID Journal*, Vol. 22, pp. 97-114, 2009.

- [49] F. Mattern and C. Floerkemeier, “From the Internet of Computers to the Internet of Things,” in *From Active Data Management to Event-based Systems and More*, Springer, pp. 242–259, 2010.
- [50] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelffle, Eds., *Vision and Challenges for Realising the Internet of Things*, Luxembourg: Publications Office of the European Union, 2010.
- [51] J. Belissent, “Getting Clever about Smart Cities: New Opportunities Require New Business Models,” *Forrester Research*, 2010.
- [52] J. Aario, “Enabling Consumer Connectivity through Consensus Building,” *IEEE Standard Association*, 2012.
- [53] T. Kurakova, “Overview of the Internet of Things,” in *Proceedings of the Internet of Things and its Enablers (INTHITEN)*, pp. 82–94, 2013.
- [54] O. Vermesan, P. Friess, and P. Guillemin, and H. Sundmaeker, M. Eisenhauer, K. Moessner, and F. L. Gall and P. Cousin, “Internet of Things Strategic Research and Innovation Agenda,” *River Publishers Series in Communications*, pp. 1–7, 2013.
- [55] B. N. Silva, M. Khan, and K. Han, “Internet of Things: A Comprehensive Review of Enabling Technologies, Architecture, and Challenges,” *IETE Technical Review*, pp. 1–16, 2017.
- [56] S. M. Amin, A. GarcaaaOrtiz, and J. Wootton, “Network, Control, Communication and Computing Technologies for Intelligent Transportation Systems Overview of the Special Issue,” *Mathematical and Computer Modelling*, Vol. 22, No. 47, pp. 1 – 10, 1995.
- [57] S. Lu, T. He, and Z. Gao, “Electronic Toll Collection System based on Global Positioning System Technology,” in *International Conference on Challenges in Environmental Science and Computer Engineering (CESCE)* Vol. 2, pp. 11–14, 2010.
- [58] D. Gu and H. Hu, “Spatial Gaussian Process Regression with Mobile Sensor Networks,” *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 23, No. 8, pp. 1279–1290, 2012.
- [59] A.P. Gupta and N. Goyal, “Effect of Transverse Shear and Rotatory Inertia on the Forced Motion of a Plate-Strip of Linearly Varying Thickness,” *Journal of Sound and Vibration*, Vol. 174, No. 2, pp. 461-474, 1994.
- [60] V. Sharma, K. Jayakody, N. Dushantha, Y. Ilsun, R. Kumar, and L. Jun, “Secure and Efficient Context-Aware Localization of Drones in Urban Scenarios,” *IEEE Communications Magazine*, Vol. 56, No. 4, pp. 120–128, 2018.
- [61] C. Perera, A. B. Zaslavsky, P. Christen, M. Compton, and D. Georgakopoulos, “Context-Aware Sensor Search, Selection and Ranking Model for Internet of Things Middleware,” *CoRR*, abs/1303.2447, 2013.

- [62] A. Atabekov, M. Starosielsky, D. C. T. Lo, and J. S. He, "Internet of Things-based Temperature Tracking System," in *IEEE 39th Annual Computer Software and Applications Conference*, Vol. 3, pp. 493–498, 2015.
- [63] A. Sheth, P. Anantharam, and C. Henson, "Semantic, Cognitive, and Perceptual Computing: Advances Toward Computing for Human Experience," *arXiv preprint arXiv:1510.05963*, pp. 1–13, 2015.
- [64] Y. Qian, J. Yan, H. Guan, J. Li, X. Zhou, S. Guo, and K.N. Dushantha, "Design of Hybrid Wireless and Power Line Sensor Networks with Dual Interface Relay in IoT," *IEEE Internet of Things Journal*, 2017.
- [65] J. Pradilla, C. Palau, and M. Esteve, "SOSLite: Lightweight Sensor Observation Service (SOS)," *IEEE Latin America Transactions*, Vol. 13, No. 12, pp. 3758–3764, 2015.
- [66] E. Aguirre, P. Lopez-Iturri, L. Azpilicueta, C. Rivares, J. J. Astrain, JJ. Villadangos, and F. Falcone, "Design and Performance Analysis of Wireless Body Area Networks in Complex Indoor E-Health Hospital Environments for Patient Remote Monitoring," *International Journal of Distributed Sensor Networks*, Vol. 12, 2016.
- [67] L. Shen, Y. Chen, Z. Wang, W. Yu, S. He, and S. Zhang, "Web Services Dynamic Discovery based on Modified Clique Algorithm," in *International Symposium on Intelligent Information Technology Application Workshops*, pp. 379–382, 2008.
- [68] V. Sharma, K. Srinivasan, K.N. Dushantha, O. Rana and R. Kumar, "Managing service-heterogeneity using osmotic computing," *arXiv preprint arXiv:1704.04213*, 2017.
- [69] Z. Ding, J. Dai, X. Gao, and Q. Yang, "A Hybrid Search Engine Framework for the Internet of Things," in *Proceedings of Ninth Web Information Systems and Applications Conference*, pp. 57–60, 2012.
- [70] B. Djamaa and A. Yachir, "A Proactive Trickle-based Mechanism for Discovering CoRE Resource Directories," *Procedia Computer Science*, Vol. 83, pp. 115–122, 2016. [Online]. Available: <https://doi.org/10.1016/2Fj.procs.2016.04.106>
- [71] M. A. Quddus, R. B. Noland, and W. Y. Ochieng, "A High Accuracy Fuzzy Logic Based Map Matching Algorithm for Road Transport," *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, Vol. 10, No. 3, pp. 103–115, 2006.
- [72] J. Cao, P. Li, and H. Liu, "An Interval Fuzzy Controller for Vehicle Active Suspension Systems," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 11, No. 4, pp. 885–895, 2010.
- [73] O. Linda and M. Manic, "Online Spatio-Temporal Risk Assessment for Intelligent Transportation Systems," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 12, No. 1, pp. 194–200, 2011.

- [74] S.K. Das and S. Tripathi, "Energy Efficient Routing Formation Technique for Hybrid Ad-Hoc Network using Fusion of Artificial Intelligence Techniques," *International Journal of Communication Systems*, Vol. 30, No. 16, 2017.
- [75] M. Collotta and G. Pau, "A Solution based on Bluetooth Low Energy for Smart Home Energy Management," *Energies*, Vol. 8, No. 10, pp. 11 916–11 938, 2015. [Online]. Available: <http://www.mdpi.com/1996-1073/8/10/11916>
- [76] X. Su, "A Text Categorization Perspective for Ontology Mapping," Department of Computer and Information Science, Norwegian University of Science and Technology, Norway, Technical Report, 2002.
- [77] I. Tachmazidis, S. Batsakis, J. Davies, A. Duke, M. Vallati, G. Antoniou, and S. S. Clarke, "A Hypercat-enabled Semantic Internet of Things Data Hub: Technical Report," *CoRR*, 2017.
- [78] D.P. Vidyarthi, B.K. Sarker, A.K. Tripathi, L.T. Yang, "Scheduling in distributed computing systems: Analysis, design and models," *Springer Science & Business Media*, 2008.
- [79] M. Manic, D. Wijayasekara, K. Amarasinghe, and J.J.R. Andina, "Building Energy Management Systems: The Age of Intelligent and Adaptive Buildings," *IEEE Industrial Electronics Magazine*, Vol. 10, No. 1, pp. 25–39, 2016.
- [80] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pp. 56–67, 2000.
- [81] D. Wijayasekara, O. Linda, M. Manic, and C. Rieger, "Mining Building Energy Management System Data Using Fuzzy Anomaly Detection and Linguistic Descriptions," *IEEE Transactions on Industrial Informatics*, Vol. 10, No. 3, pp. 1829-1840, 2014.
- [82] E. M. Royer and C. E. Perkins, "An Implementation Study of the AODV Routing Protocol" in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1003–1008, 2000. [Online]. Available: <http://dblp.uni-trier.de/db/conf/wcnc/wcnc2000.htmlRoyerP00>
- [83] A.K. Yadav, S.K. Das, S. Tripathi, "EFMMRP: Design of Efficient Fuzzy based Multi-Constraint Multicast Routing Protocol for Wireless Ad-hoc Network, " *Computer Networks, Elsevier*, Vol. 118, pp. 15-23, May 2017.
- [84] R. C. Shah and J. M. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks," in *Proceedings of IEEE Wireless Communications and Networking Conference*, Vol. 1, pp. 350–355, 2002.

- [85] J. H. Lee, J. M. Bonnin, I. You and T.M. Chung, “Comparative Handover Performance Analysis of IPv6 Mobility Management Protocols”, *IEEE Transactions on Industrial Electronics*, Vol. 60, No. 3, pp. 1077-1088, 2013.
- [86] R. Misra and C.R. Mandal, “Performance Comparison of AODV/DSR on-Demand Routing Protocols for Adhoc Networks in Constrained Situation,” in *Proceedings of IEEE International Conference on Personal Wireless Communications (ICPWC)*, pp. 86-89, 2005.
- [87] S. Lindsey and C. S. Raghavendra, “PEGASIS: Power-Efficient Gathering in Sensor Information Dystems,” in *Aerospace Conference Proceedings*, Vol. 3, pp. 3–1125–3–1130, 2002. [Online]. Available: <http://dx.doi.org/10.1109/aero.2002.1035242>
- [88] W. Liu, Z. Ma, Z. Deng, and L. Xu, “Energy Balance Algorithm based on LEACH,” in *Proceedings of 12th International Conference on Communication Technology*, pp. 681–684, 2010.
- [89] J. Singh, B. P. Singh, and S. Shaw, “A New LEACH-based Routing Protocol for Energy Optimization in Wireless Sensor Network,” in *Proceedings of International Conference on Computer and Communication Technology (ICCCCT)*, pp. 181–186, 2014.
- [90] W. Twayej and H. S. Al-Raweshidy, “M2M Energy Efficiency Routing Protocol MLCMS by using 6LowPan based on IoE,” in *Proceedings of IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pp. 1–5, 2016.
- [91] G. Bisson, “Why and How to Define a Similarity Measure for Object based Representation Systems,” *Towards Very Large Knowledge Bases*, pp. 236-246, 1995.
- [92] H. Qu, M. Song, X. Luo, J. Tang, and J. Du, “A Resource Discovery Algorithm in Mobile Peer-to-Peer based on Vector Location,” in *Proceedings of Third International Conference on Pervasive Computing and Applications*, Vol. 2, pp. 990–994, 2008.
- [93] A. J. Jara, P. Martinez-Julia, and A. Skarmeta, “Light-Weight Multicast DNS and DNS-SD (LMDNS-SD): IPv6-based Resource and Service Discovery for the Web of Things,” in *Proceedings of Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pp. 731–738, 2012.
- [94] J. Han and X.-Y. Li, “Pickup Game: Acquainting Neighbors Quickly and Efficiently in Crowd,” in *Proceedings of IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 2014* , pp. 82–90, 2014.
- [95] M. Kodialam and T. Nandagopal, “Fast and Reliable Estimation Schemes in RFID Systems,” in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, pp. 322–333, 2006.
- [96] D. Luddecke, N. Bergmann, and I. Schaefer, *Ontology Based Modeling of Context-Aware Systems* Cham: Springer International Publishing, pp. 484–500, 2014. [Online]. Available: <https://doi.org/10.1007/978-3-319-11653-2-30>

- [97] D. Shin, V. Sharma, K. Jiyoung, S. Kwon, and I. Ilson, "Secure and Efficient Protocol for Route Optimization in PMIPv6-based Smart Home IoT Networks", *IEEE Access*, Vol. 5, pp. 11100–11117, 2017.
- [98] R. Pozza, M. Nati, S. Georgoulas, A. Gluhak, K. Moessner, and S. Krco, "CARD: Context-Aware Resource Discovery for Mobile Internet of Things Scenarios," in *Proceedings of IEEE 15th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 1–10, 2014.
- [99] P. Gomes, E. Cavalcante, T. Rodrigues, T. Batista, F. C. Delicato, and P. F. Pires, "A Federated Discovery Service for the Internet of Things," in *Proceedings of the 2nd Workshop on Middleware for Context-Aware Applications in the IoT*, pp. 25–30, 2015. [Online]. Available: <http://doi.acm.org/10.1145/2836127.2836129>
- [100] G. Fersi, "Middleware for Internet of Things: A Study," in *Proceedings of International Conference on Distributed Computing in Sensor Systems*, pp. 230–235, 2015.
- [101] K. Hengst, "DDoS through the Internet of Things," in *Proceedings of International Twente Student Conference on Analysis Determining the Potential Power of a DDoS Attack using IoT devices*, 2016.
- [102] A. Zamanifar, E. Nazemi, and M. Vahidi-Asl, "DSHMP-IoT: A Distributed Self Healing Movement Prediction Scheme for Internet of Things Applications," *Applied Intelligence*, Vol. 46, No. 3, pp. 569–589, 2017. [Online]. Available: <http://dx.doi.org/10.1007/s10489-016-0849-0>
- [103] J. C. Zhao, J. F. Zhang, Y. Feng, and J.X. Guo, "The Study and Application of the IoT Technology in Agriculture," in *Proceedings of 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, Vol. 2, pp. 462–465, 2010.
- [104] R. Want, "Near Field Communication," *IEEE Pervasive Computing*, Vol. 10, No. 3, pp. 4–7, 2011.
- [105] C. Gomez, J. Oller, and J. Paradells, "Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology," *Sensors*, Vol. 12, No. 9, pp. 11 734–11 753, 2012.
- [106] K. D. M. Vipul, "OASIS: Web Services Dynamic Discovery (WS-Discovery) Version 1.1," 2009.
- [107] C. Bormann, A. P. Castellani, and Z. Shelby, "COAP: An Application Protocol for Billions of Tiny Internet Nodes," *IEEE Internet Computing*, Vol. 16, No. 2, pp. 62–67, 2012.
- [108] Z. Shelby, "Constrained RESTful Environments (CoRE) Link Format," *Internet Engineering Task Force*, pp. 1–22, 2013.

- [109] M. Jeronimo and J. Weast, "UPnP Design by Example: A Software Developer's Guide to Universal Plug and Play," Intel Press, 2003.
- [110] Y. Y. Goland, "Simple Service Discovery Protocol/1.0 Operating without on Arbiter," *IETF INTERNET-DRAFT draft-cai-ssdp-v1-03.txt*, 1999.
- [111] S. Cheshire and M. Krochmal, "Multicast DNS," Technical Report, 2013.
- [112] S. Cheshire and M. Krochmal, "DNS-based Service Discovery," Technical Report, 2013.
- [113] Z. Shelby, C. Bormann, and S. Krco, "IETF Internet-Draft: CoRE Resource Directory," 2013.
- [114] T. Jaffey, J. Davies, and P. Beart, "Hypercat 3.00 Specification," *Hyper-cat Limited*, 2016.
- [115] P. Waher and R. Klauck, "Internet of Things-Discovery," *XMPP Standards Foundation*, 2017.
- [116] R. Cyganiak, D. Wood, and M. Lanthaler, "RDF 1.1 Concepts and Abstract Syntax," *W3C Recommendation*, 2014.
- [117] P. Jayaraman, D. Palmer, A. Zaslavsky, and D. Georgakopoulos, "Do-it-Yourself Digital Agriculture Applications with Semantically Enhanced IoT Platform," in *Proceedings of the IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pp. 311-316, 2015.
- [118] A. Boring, C. Stasch, and J. Echterhoff, "OGC Sensor Observation Service Interface Standard," *Open Geospatial Consortium Interface Standard*, pp. 12-006, 2012.
- [119] A. Broring, S. K. Datta, and C. Bonnet, "A Categorization of Discovery Technologies for the Internet of Things," in *Proceedings of the 6th International Conference on the Internet of Things*, pp. 131-139, 2016.
- [120] S. Z. Selim and M. A. Ismail, "K-means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, No. 1, pp. 81-87, 1984.
- [121] D. Li, L. Wang, Z. Xue, and S. T. C. Wong, "When Discriminative K-means meets Grassmann Manifold: Disease Gene Identification via A General Multi-View Clustering Method," in *Proceedings of IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pp. 364-367, 2016.
- [122] L. Atzori, A. Iera, and G. Morabito, "Understanding the Internet of Things: Definition, Potentials, and Societal Role of a Fast Evolving Paradigm," *Ad Hoc Networks*, Vol. 56, pp. 122 - 140, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1570870516303316>

- [123] G. Salton and M. E. Lesk, "Computer Evaluation of Indexing and Text processing," *Journal of the ACM*, Vol. 15, No. 1, pp. 8–36, 1968.
- [124] S. Krinidis and V. Chatzis, "A Robust Fuzzy Local Information c-means Clustering Algorithm," *IEEE Transactions on Image Processing*, Vol. 19, No. 5, pp. 1328–1337, 2010.
- [125] F. Barrero, J. A. Guevara, E. Vargas, S. Toral, and M. Vargas, "Networked Transducers in Intelligent Transportation Systems based on the IEEE 1451 Standard," *Computer Standards and Interfaces*, Vol. 36, No. 2, pp. 300–311, 2014.
- [126] B. Singh, "National Informatics Centre," 2012. [Online]. Available: <https://data.gov.in/>
- [127] Y. Sharma and N. Goyal, "An Efficient Multi-Component Indexing Embedded Bitmap Compression for Data Reorganization," *Information Technology Journal*, Vol. 7, No. 1, pp. 160–164, 2008.
- [128] S.P. Singh and S.C. Sharma, "A survey on Cluster based Routing Protocols in Wireless Sensor Networks", *Procedia Computer Science, Elsevier*, Vol. 45, pp. 687–695, 2015.
- [129] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, and J. Wang, "A First Look at Cellular Machine-to-Machine Traffic: Large Scale Measurement and Characterization," *ACM SIGMETRICS Performance Evaluation Review*, Vol. 40, No. 1, pp. 65–76, 2012.
- [130] J. Gantz and D. Reinsel, "The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East," *IDC iView: IDC Analyze the Future*, Vol. 2007, No. 2012, pp. 1–16, 2012.
- [131] J. Manyika, M. Chui, J. Bughin, R. Dobbs, P. Bisson, and A. Marrs, "Disruptive Technologies: Advances that will Transform Life, Business, and the Global Economy," *McKinsey Global Institute*, 2013.
- [132] R. R. Madakam, S. and S. Tripathi, "Internet of Things (IoT): A Literature Review," *Journal of Computer and Communications*, vol. 3, 2015.
- [133] G. Rohling, "Internet of Things: Facts and Forecasts Newsletter," 2014. [Online]. Available: <http://www.siemens.com/innovation/en/home/pictures-of-the-future/digitalization-and-software/internet-of-things-facts-and-forecasts.html>
- [134] T. G. Kolda, R. M. Lewis, and V. Torczon, "Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods," *SIAM Review*, Vol. 45, No. 3, pp. 385–482, 2003.
- [135] Y. Tao, J. Zhang, D. Papadias, and N. Mamoulis, "An Efficient Cost Model for Optimization of Nearest Neighbor Search in Low and Medium Dimensional Spaces," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, No. 10, pp. 1169–1184, 2004.

- [136] S. Sergi, F. Pancaldi, and G. M. Vitetta, "A Game Theory Approach to Selection Diversity in Wireless Ad-Hoc Networks," in *Proceedings of IEEE International Conference on Communications*, pp. 1–6, 2009.
- [137] S. Lee, K. Sriram, K. Kim, Y. H. Kim, and N. Golmie, "Vertical Hand-off Decision Algorithms for Providing Optimized Performance in Heterogeneous Wireless Networks," *IEEE Transactions on Vehicular Technology*, Vol. 58, No. 2, pp. 865–881, Feb 2009.
- [138] M. Burger and F. Lucka, "Maximum a Posteriori Estimates in Linear Inverse Problems with Log-Concave Priors are Proper Bayes Estimators," *Inverse Problems*, Vol. 30, No. 11, p. 114004, 2014.
- [139] Z. Li, R. Chen, L. Liu, and G. Min, "Dynamic Resource Discovery based on Preference and Movement Pattern Similarity for Large-Scale Social Internet-of-Things," *IEEE Internet of Things Journal*, Vol. 3, No. 4, pp. 581–589, 2015.
- [140] J. Boubeta-Puig, G. Ortiz, and I. Medina-Bulo, "MEDIT4CEP: A Model-Driven Solution for Real-Time Decision Making in SOA 2.0," *Knowledge-Based Systems*, Vol. 89, pp. 97–112, 2015.
- [141] J. Lanza, L. Sanchez, D. Gomez, T. Elsaleh, R. Steinke, and F. Cirillo, "A Proof-of-Concept for Semantically Interoperable Federation of IoT Experimentation Facilities," *Sensors*, Vol. 16, No. 7, p. 1006, 2016. [Online]. Available: <http://dx.doi.org/10.3390/s16071006>
- [142] D. Hussein, S. N. Han, G. M. Lee, N. Crespi, and E. Bertin, "Towards a Dynamic Discovery of Smart Services in the Social Internet of Things," *Computers and Electrical Engineering*, Vol. 58, pp. 429 – 443, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045790616310369>
- [143] K.H. N. Bui, J. E. Jung, and D. Camacho, "Game Theoretic Approach on Real-Time Decision Making for IoT-based Traffic Light Control," *Concurrency and Computation: Practice and Experience*, Vol. 29, No. 11, pp. e4077–n/a, 2017, e4077 cpe.4077. [Online]. Available: <http://dx.doi.org/10.1002/cpe.4077>
- [144] A. G. D. Prado, G. Ortiz, and J. Boubeta-Puig, "COLLECT: Collaborative Context-Aware Service Oriented Architecture for Intelligent Decision-Making in the Internet of Things," *Expert Systems Applications*, Vol. 85, pp. 231–248, 2017.
- [145] J. Arunadevi, A. John, S. Kumar, and N. Sujatha, "Intelligent Transport Route Planning using Parallel Genetic Algorithms and MPI in High Performance Computing Cluster," in *Proceedings of International Conference on Advanced Computing and Communications (ADCOM)*, pp. 578–583, 2007.
- [146] C. H. Lin, J. L. Yu, J. C. Liu, and C. J. Lee, "Genetic Algorithm for Shortest Driving Time in Intelligent Transportation Systems," in *Proceedings of International Conference on Multimedia and Ubiquitous Engineering (MUE)*, pp. 402–406, 2008.

- [147] L. Mourelle, R. E. Ferreira, and N. Nedjah, "Migration Selection of Strategies for Parallel Genetic Algorithms: Implementation on Networks on Chips," *International Journal of Electronics*, Vol. 97, No. 10, pp. 1227–1240, 2010.
- [148] J. Tang and Y. Hu, "A Memory Gradient Method with a New Non-Monotone Line Search Rule," in *Proceedings of IEEE International Conference on Progress in Informatics and Computing*, Vol. 1, pp. 59–62, Dec. 2010.
- [149] S. M. Asad and A. Zerguine, "Convergence Analysis of a Modified Armijo Rule Step-Size LMF Algorithm," in *Proceedings of 11th International Conference on Information Science, Signal Processing and their Applications*, pp. 343–347, 2012.
- [150] P. Hennig and M. Kiefel, "Quasi-Newton Methods: A New Direction," *Journal of Machine Learning Research*, Vol. 14, No. 1, pp. 843–865, 2013.
- [151] T. Hofmann, A. Lucchi, and B. McWilliams, "Neighborhood Watch: Stochastic gradient descent with neighbors," *CoRR*, abs/1506.03662, 2015.
- [152] K. Ueda and N. Yamashita, "ERRATUM To: A Regularized Newton Method without Line Search for Unconstrained Optimization," *Computational Optimization and Applications*, Vol. 66, No. 1, pp. 219–222, 2017.
- [153] R. M. Lewis, V. Torczon, and M. W. Trosset, "Direct Search Methods: Then and Now," *Journal of Computational and Applied Mathematics*, Vol. 124, pp. 191 – 207, 2000.
- [154] J. Barcelo, E. Codina, J. Casas, J. L. Ferrer, and D. Garcia, "Microscopic Traffic Simulation: A Tool for the Design, Analysis and Evaluation of Intelligent Transport Systems," *Journal of Intelligent and Robotic Systems*, Vol. 41, No. 2, pp. 173–203, 2005.
- [155] T. E. Abrudan, J. Eriksson, and V. Koivunen, "Steepest Descent Algorithms for Optimization Under Unitary Matrix Constraint," *IEEE Transactions on Signal Processing*, Vol. 56, No. 3, pp. 1134–1147, 2008.
- [156] N. Parrado and Y. Donoso, "Congestion based Mechanism for Route Discovery in a V2I-V2V System applying Smart Devices and IoT," *Sensors*, Vol. 15, No. 4, p. 7768, 2015.
- [157] X. Yi, A. Bouguettaya, D. Georgakopoulos, A. Song, and J. Willemsen, "Privacy Protection for Wireless Medical Sensor Data," *IEEE Transactions on Dependable and Secure Computing*, Vol. 13, pp. 369 - 380, 2016.
- [158] F. Yu, H. Zhang, and F. Yan, "A Fuzzy Relation Trust Model in P2P System," in *Proceedings of International Conference on Computational Intelligence and Security*, Vol. 2, pp. 1497–1502, 2006.
- [159] J. Luo, X. Liu, and M. Fan, "A Trust Model based on Fuzzy Recommendation for Mobile Ad-hoc Networks," *Computer Networks*, Vol. 53, No. 14, pp. 2396–2407, 2009.

- [160] M. Momani and S. Challa, “Survey of Trust Models in Different Network Domains,” *CoRR*, abs/1010.0168, 2010.
- [161] S. S. Babu, A. Raha, and M. K. Naskar, “Geometric Mean based Trust Management System for WSNs (GMTMS),” in *Proceedings of World Congress on Information and Communication Technologies*, pp. 444–449, Dec. 2011.
- [162] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, and X. Wang, “TRM-IoT: A Trust Management Model based on Fuzzy Reputation for Internet of Things,” *Computer Science and Information Systems*, Vol. 8, No. 4, pp. 1207–1228, 2011.
- [163] P. N. Mahalle, P. A. Thakre, N. R. Prasad, and R. Prasad, “A Fuzzy Approach to Trust based Access Control in Internet of Things,” in *Wireless VITAE*, pp. 1–5, 2013.
- [164] Y. Ruan and A. Duresi, “A Survey of Trust Management Systems for Online Social Communities, Trust Modeling, Trust Inference and Attacks,” *Knowledge-Based Systems*, Vol. 106, pp. 150 – 163, 2016.
- [165] S. E. A. Rafey, A. A. Hamid, and M. A. ElNasr, “CBSTM-IoT: Context-based Social Trust Model for the Internet of Things,” in *Proceedings of International Conference on Selected Topics in Mobile Wireless Networking (MoWNeT)*, pp. 1–8, 2016.
- [166] R. K. Roul, S. Varshneya, A. Kalra, and S. K. Sahay, “A Novel Modified Apriori Approach for Web Document Clustering,” in *Computational Intelligence in Data Mining*, Vol. 3, pp. 159–171, 2015.
- [167] L. Lukan and E. Spedicato, “Variable Metric Methods for Unconstrained Optimization and Non-Linear Least Squares,” *Journal of Computational and Applied Mathematics*, Vol. 124, No. 1, pp. 61 – 95, 2000.
- [168] T. R. Gruber, “Toward Principles for the Design of Ontologies used for Knowledge Sharing,” *International Journal of Human-Computational Studies*, Vol. 43, No. 5-6, pp. 907–928, 1995.
- [169] S. Castano, V. De Antonellis, M. G. Fugini, and B. Pernici, “Conceptual Schema Analysis: Techniques and Applications,” *ACM Transactions on Database Systems*, Vol. 23, No. 3, pp. 286–333, 1998.
- [170] H. W. Kuhn, *“The Hungarian Method for the Assignment Problem,”* Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 29–47, 2010.
- [171] A. Frank, “On Kuhn’s Hungarian Methods to Tribute from Hungary,” *Naval Research Logistics (NRL)*, Vol. 52, No. 1, pp. 2–5, 2005.
- [172] J. H. Ahrens and U. Dieter, “Sampling from Binomial and Poisson Distributions: A Method with Bounded Computation Times,” *Computing*, Vol. 25, No. 3, pp. 193–208, 1980.

- [173] J. Barcelo, E. Codina, J. Casas, J. L. Ferrer, and D. Garcia, "Microscopic Traffic Simulation: A Tool for the Design, Analysis and Evaluation of Intelligent Transport Systems," *Journal of Intelligent and Robotic Systems*, Vol. 41, No. 2, pp. 173–203, 2005.
- [174] E. Porcu, P. Zastavyi, and M. Bevilacqua, "Buhmann Covariance Functions, their Compact Supports, and their Smoothness," *arXiv preprint arXiv:1606.09527*, 2016.
- [175] S. Xiao and G. Zeng, "Equality-Constrained Minimization of Polynomial Functions," *Science China Mathematics*, Vol. 58, No. 10, pp. 1–24, 2015.
- [176] Z. Wu, "Compactly Supported Positive Definite Radial Functions," *Advances in Computational Mathematics*, Vol. 4, No. 1, pp. 283–292, 1995.
- [177] I. Fister Jr, X.-S. Yang, I. Fister, J. Brest, and D. Fister, "A Brief Review of Nature-Inspired Algorithms for Optimization," *arXiv preprint arXiv:1307.4186*, 2013.
- [178] D. Evans, "The Internet of Things: How the Next Evolution of the Internet is Changing Everything," *In CISCO White Paper*, CISCO: USA, 2011.
- [179] A. Khanna, and S. Bhatia, "Evolution of Internet of Things (IoT) and Its Significant Impact in the Field of Precision Agriculture, " *Computers and Electronics in Agriculture*, Vol. 157, pp. 218-231, 2019.
- [180] M. Burger, and F. Lucka, "Maximum a Posteriori Estimates in Linear Inverse Problems with Log-Concave Priors are Proper Bayes Estimators," *Inverse Problems*, Vol. 30, No. 11, p. 114004, 2014.
- [181] G. Dongbing and H. Huosheng, "Spatial Gaussian Process Regression with Mobile Sensor Networks," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 23, No. 8, pp. 1279-1290, 2012.
- [182] X. ShuiJing and Z. GuangXing, "Equality-constrained Minimization of Polynomial Functions," *Science China Mathematics*, Vol. 58, No. 10, pp. 1-24, 2015.
- [183] W. Zongmin, "Compactly Supported Positive Definite Radial Functions," *Advances in Computational Mathematics*, Vol. 4, No. 1, pp. 283-292, 1995.
- [184] H. Junze and L. X. Yang, "Pickup game: Acquainting Neighbors Quickly and Efficiently in Crowd," *in Proceedings of 11th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), IEEE*, pp. 82-90, 2014.
- [185] K. Murali and N. Thyaga, "Fast and Reliable Estimation Schemes in RFID Systems," *in Proceedings of the 12th Annual International Conference on Mobile Computing and Networking, ACM*, pp. 322-333, 2006.
- [186] L. Junhai, L. Xue and F. Mingyu, "A Trust Model based on Fuzzy Recommendation for Mobile Ad-hoc Networks," *Computer Networks*, Vol. 53, No. 14, pp. 2396–2407, 2009.

- [187] Y. Fajiang, Z. Huanguo and Y. Fei, "A Fuzzy Relation Trust Model in P2P System," in *Proceedings of the International Conference on Computational Intelligence and Security, IEEE*, Vol. 2, pp. 1497-1502, 2006.
- [188] C. Dong, C. Guiran, S. Dawei, L. Jiajia, J. Jie and W. Xingwei, "TRM-IoT: A Trust Management Model based on Fuzzy Reputation for Internet of Things," *Computer Science and Information Systems*, Vol. 8, No. 4, pp. 1207-1228, 2011.
- [189] A. A. Shah and L. J. Dal, "Intelligent Transportation Systems in Transitional and Developing Countries," in *Proceedings of the International Conference on Advances in Space Technologies*, pp. 54-59, 2006.
- [190] U. Bellur, H. Vadodaria, and A. Gupta, "Semantic Matchmaking Algorithms," *Greedy Algorithms*, [IntechOpen], 2008.
- [191] J. F. Iztok, X. S. Yang, I. Fister, J. Brest, and D. Fister, "A Brief Review of Nature-Inspired Algorithms for Optimization," *Computer Science and Information Systems*, 2013.
- [192] W. Liu, Z. Ma, Z. Deng, and L. Xu, "Energy Balance Algorithm based on LEACH," in *Proceedings of the 12th International Conference on Communication Technology, IEEE*, pp. 681-684, 2010.
- [193] J. Singh, B. P. Singh, and S. Shaw, "A New LEACH-based Routing Protocol for Energy Optimization in Wireless Sensor Network," in *Proceedings of the International Conference on Computer and Communication Technology (ICCCT)*, pp. 181-186, 2014.
- [194] W. Twayej, and H. S. A. Raweshidy, "M2M Energy Efficiency Routing Protocol MLCMS by using 6LoWPAN based on IoE," in *Proceedings of the International Black Sea Conference on Communications and Networking (BlackSeaCom), IEEE*, pp. 1-5, 2016.

## List of Publications

### Refereed Journals

- 1) Monika Bharti, Rajesh Kumar and Sharad Saxena, “Clustering-based resource discovery on Internet-of-Things, International Journal of Communication Systems, Vol. 31, no.5, pp. e3501, 2018 (Wiley). SCI Indexed (IF: 1.278)
- 2) Monika Bharti, Rajesh Kumar and Sharad Saxena, “Context-Aware Search Optimization Framework on the Internet of Things, Concurrency and Computation: Practice and Experience, Vol. 30, no. 14, pp. e4426, 2018 (Wiley). SCI Indexed (IF: 1.167)
- 3) Monika Bharti, Sharad Saxena and Rajesh Kumar, “Intelligent Resource Inquisition Framework on Internet-of-Things, in Computers and Electrical Engineering, Vol. 58, pp. 265-281, 2017 (Elsevier). SCI Indexed (IF: 2.189)
- 4) Monika Bharti, Sharad Saxena and Rajesh Kumar, “A Middleware Approach for Reliable Resource Selection on Internet-of-Things (MARRS-IoT)” in International Journal of Communication Systems, 2019 (Wiley). <https://doi.org/10.1002/dac.4278>. (Accepted) SCI Indexed (IF: 1.278)

### Conferences

- 5) Monika Bharti, Sharad Saxena and Rajesh Kumar, “Architectural Survey on Internet-of-Things, in *Proceedings of Fifth International Conference on Image Processing (ICIIP-2019)*, IEEE. (in press) (Scopus Indexed)