UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Engineering and Information Technology

AI-driven Service Broker for Simple and Composite Cloud SaaS Selection

by

Mohammed Abdulaziz Ikram

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Sydney, Australia

2020

Certificate of Authorship/Originality

I, Mohammed Ikram declare that this thesis, is submitted in fulfilment of the requirements for the award of PhD in Computer System, in the school of Computer Science at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note: Signature removed prior to publication.

Date: 30/06/2020

ABSTRACT

AI-driven Service Broker for Simple and Composite Cloud SaaS Selection

by

Mohammed Abdulaziz Ikram

Cloud Software as a Service (SaaS) is one of the three types of services offered in cloud computing. Cloud SaaS is a software application that runs on top of Platform as a Service (PaaS), which in turn works on top of Infrastructure as a Service (IaaS). Due to the numerous advantages offered by cloud SaaS to service consumers, such as reducing the cost of IT expenditures, security capabilities and disaster recovery offered by cloud SaaS service providers, Cloud SaaS is becoming a leading and growing type of cloud service among other cloud services (i.e., IaaS and PaaS). Therefore, Cloud SaaS service consumers may face a difficult task when searching for the most suitable service based on their preferences. Service selection is based on matching the service requirements of functional and non-functional quality attributes. However, selecting a Cloud SaaS service provider with a high number of non-functional quality attributes that fulfils consumer requirements within a large number of similar functional services is a key factor for a Cloud SaaS service selection.

In addition, considering that a cloud SaaS service can involve a long-term contract, Cloud SaaS providers frequently offer a free trial period to test and evaluate services before the consumers make the decision of whether they will use that service.Furthermore, selecting multiple Cloud SaaS service providers in order to create a new business value, known as a service composition in the service-oriented architecture (SOA) model, is very important, since Cloud SaaS services are the first option for deploying IT services for many new enterprises.

Therefore, this research aims to propose intelligent methods for a simple and composite service selection framework based on consumer preferences. By simple, we mean a singular service whereas by composite, we mean an aggregated service. This work seeks to find the services with a high number of non-functional quality attributes that meet the consumer requirements. To achieve the objectives of this research, a design science research methodology will be adopted. Fuzzy logic will be proposed to address the uncertainty of consumer preferences. A ranking service system, evaluation system and composite decision maker system are proposed in this thesis to help a Cloud SaaS service consumer select the optimal service required. Multiple approaches of decision-makers will be developed in order to achieve our research objectives. It is expected that this research work will enhance the selection mechanism of Cloud SaaS, either simple or composite based on service consumer's preferences.

Dissertation directed by Associate Professor Farookh Hussain
School of Computer Science
Centre for Artificial Intelligence
Faculty of Engineering and Information Technology (FEIT)

Dedication

To all of my family...

Acknowledgements

I am thankful to Allah Almighty, who bless me and support me to write the whole thesis. My greatest gratitude goes to my family, I thank my mother Najah Jalal who taught me to be patient and supported me during the PhD journey. Also, I would like to thank my great father Abdulaziz Ikram who taught me to be responsible. Special thanks to my wife Haneen Ekram, and my kids Janna, Ibrahim and Ahmed for supporting me through all my PhD journey. I would like to express my gratitude to all of my family, my sister and brothers who supported me with motivational words and prayers. Also, special thanks to my daughter Janna who help some parts of my thesis. Without all your love I could never have written my thesis.

I would like to thank my supervisor, Prof. Farookh Hussain for his patient guidance, encouragement and advice from the first day through to the end of my PhD journey. Also, I would like to thank my Co-Supervisor Dr Nabin Sharma for his help and support. I would also like to thank all my colleagues during PhD journey Dr Alshehri, Dr Alshaweesh, Dr Shoria, Dr Supanada, Dr Aysha, Dr Thaqeb, Dr Quen and Dr Yahya. I would particularly like to thank Dr Reza for the suggestions he made in Chapter 5 of this work.

I would like to thank the computer science departments staff Level 7 building 11 UTS.

Finally, I would like to express my gratitude and special thanks to the Saudi Arabia Cultural Mission (SACM) in Australia for supporting me all the way from the early days of my masters degree to PhD degree.

Mohammed Ikram

Sydney, Australia, 2019.

List of Publications

The following is a list of my research papers during my PhD study.

Journal Papers

J-1. **Mohammed Abdulaziz Ikram**, Nabin Sharma and Farookh Khadeer Hussain, "Intelligent Service Broker approach for Ranking, Evaluation and Selection of Cloud SaaS".

Conference Papers

C-1. **Mohammed Abdulaziz Ikram**, Farookh Khadeer Hussain: Software as a Service (SaaS) Service Selection Based on Measuring the Shortest Distance to the Consumer's Preferences. EIDWT 2018: pp.403-415

C-2. **Mohammed Abdulaziz Ikram**, Nabin Sharma, Muhammad Raza, Farookh Khadeer Hussain:Dynamic Ranking System of Cloud SaaS Based on Consumer Preferences - Find SaaS M2NFCP. AINA 2019: pp.1000-1010

c-3. **Mohammed Abdulaziz Ikram**, Nabin Sharma, Omar Hussain and Farookh Khadeer Hussain, "Towards Linguistic-based Evaluation System of Cloud Software as a Service (SaaS) Provider".

C-4. **Mohammed Abdulaziz Ikram**, Nabin Sharma, Farookh Khadeer Hussain:The Non-Functional Cloud Software as a Service (SaaS) classification based on Quality of Service (QoS) and Quality of Experience (QoE).

Contents

Certificate ii

Abstract iii

Dedication v

Acknowledgments vi

List of Publications vii

List of Figures xvi

List of Tables xx

Abbreviation xxv

1 Introduction 1

1.1 Significance of the Thesis 5

1.2 Scientific Contributions 6

1.3 Social Contributions 8

1.4 Research Problems and Contribution 8

1.4.1 Research Contributions in Simple Cloud SaaS Selection 9

1.4.2 Research problem and Contribution in Evaluation of Cloud SaaS Providers 10

1.4.3 Research problem and contribution in Composite Cloud SaaS Selection 11

1.5 Thesis Outline 12

2 Literature Review 15

2.1 Introduction 15
2.2 The Concept of Cloud Services 16
2.2.1 Basic concept of Cloud Computing (CC) 16
2.3 Overview of Existing Research on Non-Functional Simple Cloud SaaS Service Selection 22
2.3.1 Non-Functional Approaches for simple Cloud SaaS Service Selection 24
2.3.2 The Important of Non-Functional Quality Attributes 28
2.3.3 Evaluation System of Cloud SaaS Service Provider 32
2.3.4 Shortcomings of existing approaches 37
2.4 Overview of Existing Research on Non-Functional Cloud SaaS Service Composition 38
2.4.1 Non-Functional Approaches of Composite Cloud SaaS Selection 39
2.4.2 Shortcomings of existing approaches 43
2.5 Datasets 44
2.6 Conclusion 45

3 Problem Definition 47

3.1 Introduction 47
3.2 Key Concepts 48
3.2.1 Cloud Computing (CC) 48
3.2.2 Software as a Service (SaaS) 48
3.2.3 Service Oriented Architecture (SOA) 48
3.2.4 Service Level Agreement (SLA) 48
3.2.5 Service Broker 49
3.2.6 Service Consumer 49

3.2.7 Service Provider 49
3.2.8 Simple Service 49
3.2.9 Composite Service 50
3.2.10 Service Selection 50
3.2.11 Ranking System 50
3.2.12 Evaluation System 50
3.2.13 Consumer Preferences 51
3.2.14 Small and Medium Enterprises (SME) 51
3.2.15 Functional Service Selection 51
3.2.16 Non-Functional Service Selection 52
3.2.17 Quality of Service (QoS) 52
3.2.18 Quality of Experience (QoE) 52
3.2.19 Multi-Criteria Decision Making (MCDM) 52
3.2.20 Multi-Objective Optimization (MOO) 53
3.3 The differences between the cloud services and Cloud SaaS 53
3.4 Problem Overview and Problem Definition 54
3.5 Research Issues 57
3.5.1 Research Issue 1: Lack of an intelligent approach for ranking and selecting simple Cloud SaaS service based on consumer's preferences 57
3.5.2 Research Issue 2: Lack of an intelligent approaches for Cloud SaaS simple selection based on the feedback or outcome from the trial evaluation phase 58
3.5.3 Research Issue 3: Lack of an intelligent decision support system for selecting the Cloud SaaS composition based on combining two methods GA and TOPSIS 58

3.5.4 Research Issue 4: Validate the proposed approaches by experiment implementations . . . 59
3.6 Research Questions . . . 60
3.7 Research Aims . . . 60
3.8 Research Objectives . . . 61
3.9 Research methods to Problem Solving . . . 61
3.9.1 Multi-criteria Decision Making (MCDM) . . . 62
3.9.2 Fuzzy Logic . . . 62
3.9.3 Genetic Algorithm . . . 63
3.10 General Research Methodology . . . 63
3.11 Conclusion . . . 65

4 Solution Overview 66

4.1 Introduction . . . 66
4.2 General Framework Architecture of the proposed Service Broker (Find SaaS Framework) . . . 67
4.2.1 Overview of the solution for ranking System of Cloud SaaS Providers (Find SaaS SNFCP, M2NFCP and LNFCP) . . . 69
4.2.2 Overview of the solution for Evaluation System (Find SaaS Evaluation) . . . 71
4.2.3 Overview of the solution for Composite Cloud SaaS service Selection (Find SaaS GA TOPSIS) . . . 73
4.3 Conclusion . . . 76

5 Find SaaS (SNFCP, M2NFCP and LNFCP) Simple Cloud SaaS Service Selection 77

5.1 Introduction . . . 77

5.2 General Architecture of Find SaaS (SNFCP, M2NFCP and LNFCP) Ranking Engine System (RES) for Cloud SaaS Service Providers . . . 79

5.3 Consumer Request Handler (CRH) 81

5.3.1 Weighting Non-Functional Quality Attributes 82

5.3.2 Non-functional Consumer Requirements 85

5.4 Service Registry Repository (SRR) 97

5.5 Ranking System Engine (RSE) 101

5.6 Case Study: Ranking the Services of a Computer Repair Shop (Cloud SaaS) using the Find SaaS SNFCP Approach 105

5.7 Experiments 109

5.7.1 CRM Dataset 112

5.7.2 QWS Dataset 123

5.8 Chapter Discussion 129

5.9 Conclusion 130

6 Find SaaS Evaluation System of Cloud SaaS Providers 132

6.1 Introduction 132

6.2 Find SaaS Evaluation Architecture 134

6.3 Quality of Experience (QoE) Service Repository (QoESR) 135

6.4 A Fuzzy Linguistic-based Evaluation System for Cloud SaaS 140

6.4.1 Consumer Evaluation Handler (CEH) 141

6.4.2 Decision Maker System (DMS) 147

6.5 A Fuzzy Rule-based Evaluation System of Cloud SaaS 150

6.5.1 Consumer Evaluation Handler (CEH) 151

6.5.2 Decision Maker System (DMS) 156

6.6 Case Study 158
6.6.1 Case study: Evaluation of cloud SaaS service provider using a fuzzy linguistic-based system 158
6.6.2 Case Study: Evaluation of cloud SaaS using a fuzzy rule-based system 164
6.7 Chapter Discussion 168
6.8 Conclusion 169

7 Find SaaS GA TOPSIS - Composite Cloud SaaS Service Selection 170

7.1 Introduction 170
7.2 Find SaaS Composition GA TOPSIS Framework Architecture 172
7.2.1 System Notation 174
7.2.2 System Process 174
7.3 Consumer Request Handler (CRH) 175
7.3.1 Workflow Handler 176
7.3.2 Selected Criteria 177
7.3.3 Consumer Constraints 177
7.3.4 Weighting Non-Functional Quality Attributes 178
7.4 Service Registry Repository (SRR) 179
7.4.1 Service Provider 180
7.4.2 Previous Consumer Feedback 181
7.4.3 Monitoring Tool 181
7.5 Optimizer Unit (OU) 181
7.5.1 Total Composite cost 183

7.5.2 Reputation-based Service Reviewers (RbSR) 183
7.5.3 Aggregation Function 186
7.5.4 Genetic Algorithm (GA) 188
7.6 Decision Maker System (DMS) 192
7.7 Experiment 196
7.7.1 Case study: Emar Ltd Company 196
7.7.2 Experiment Setup 197
7.7.3 Experiment Result 199
7.7.4 Experiment Evaluation and Discussion 203
7.8 Chapter Discussion 205
7.9 Conclusion 206

8 Find SaaS System Prototype 208

8.1 Introduction 208
8.2 Simple Cloud SaaS Service Selection Ranking Service Providers Prototype 209
8.3 System Prototype of Cloud SaaS Evaluation System 213
8.4 System Prototype of Cloud SaaS Service Composition Decision Making 218
8.5 Conclusion 220

9 Recapitulation and Future Research Direction 221

9.1 Introduction 221
9.2 Problems addressed in this thesis 223
9.3 Contributions of this thesis to the existing literature 223
9.3.1 **Contribution 1:** State-of-the-art survey of present literature 224

9.3.2 **Contribution 2:** Simple Cloud SaaS Service Selection Model based on consumer's preferences (Find SaaS SNFCP, M2NFCP and LNFCP) 225

9.3.3 **Contribution 3:** Evaluation of Cloud SaaS Service Provider 226

9.3.4 **Contribution 4:** Composite Cloud SaaS Service Selection (Find SaaS GA TOPSIS) 226

9.3.5 **Contribution 5:** System prototype to demonstrate the Find SaaS service porker framework 227

9.4 Limitations 228

9.5 Future work 230

9.5.1 Service provider instead of Service consumer 230

9.5.2 Cloud SaaS selection technique based on economical and environmental value 231

9.5.3 The interoperability of on-premises software and cloud Software as a service (SaaS) 232

9.5.4 The risk assessment module for selecting Cloud SaaS 233

9.5.5 The IoT and Cloud SaaS 233

References **235**

Appendix **245**

A Chapter 5 data results **245**

B Chapter 7 data results **264**

List of Figures

2.1 Cloud Computing Services Architecture 17

2.2 Licensed software versus Cloud SaaS 19

2.3 Service-Oriented Architecture (SOA) Model 20

2.4 Service Broker Architecture 22

2.5 Overview of the literature review on Cloud SaaS Service Selection . . 24

3.1 Design Science Research Methodology Iterations Process 64

4.1 Find SaaS Service Broker Framework Structure 68

4.2 Thesis Structure 69

4.3 Ranking System Engine of Cloud SaaS Providers Solution Structure . 70

4.4 Evaluation System of Cloud SaaS Solution Structure 72

4.5 Composite Cloud SaaS Service Selection Solution Structure 74

5.1 Ranking Framework Architecture 80

5.2 Non-functional preferences classification 83

5.3 Degree of membership function of triangular fuzzy logic or linguistic variables 84

5.4 Box plot explanation 94

5.5 Membership function of linguistic values for the non-functional attributes proposed in the study 95

5.6 Affordable price membership function of the CRM dataset 95

5.7 Services Registry Repository 99

5.8 Site24x7 analysis for (Apptivo) service 113

5.9 The response time analysis by the list of servers 113

5.10 nDCG result for Find SaaS SNFCP, FSSP and CASCP approaches . 119

5.11 nDCG result for Find SaaS M2NFCP, FSSP and CASCP approaches 119

5.12 nDCG result for Find SaaS LNFCP, FSSP and CASCP approaches . 120

5.13 The average nDCG for our proposed approaches and compared with the existing research approaches 122

5.14 The mean average precision (mAP) for Find SaaS SNFCP in the top-(5,10,20 and 50) and compared with the existing research approaches 127

5.15 The mean average precision mAP for Find SaaS M2NFCP in top-(5,10,20 and 50) and compared with the existing research approaches 128

5.16 The precision of Find SaaS SNFCP compared with other ranking approaches 128

5.17 The precision of Find SaaS M2NFCP compared with other ranking approaches 129

6.1 The Find SaaS Evaluation Methods 133

6.2 The components of the Find SaaS Evaluation architecture framework 135

6.3 Evaluation of the QoE 136

6.4 Triangular fuzzy number 142

6.5 Membership function for linguistic values of QPriority 143

6.6 Membership function for linguistic values of QEvaluation 144

6.7 A fuzzy rule-based evaluation system architecture 151
6.8 Trapezoidal fuzzy number 154
6.9 Membership functions of linguistic values for QPriority 154
6.10 Membership functions of linguistic values for QEvaluation 155
6.11 Membership functions of linguistic values for QEvaluationScore . . . 155
6.12 The square rules (FAM) representation 156
6.13 Three-dimensional plot for evaluation system rule 158

7.1 The Find SaaS Composition Framework Architecture 173
7.2 Workflow control process 176
7.3 Services Registry Repository 180
7.4 Find SaaS GA TOPSIS main algorithms 182
7.5 Membership functions of linguistic values for Service rating (RA) . . 184
7.6 Membership functions of linguistic values for Service Reviewers (RV) 185
7.7 Reputation-based service reviewers 185
7.8 The basic genetic algorithm flowchart 189
7.9 The Service Composition Model 190
7.10 Genetic crossover operator for Cloud SaaS Composition 193
7.11 Genetic mutation operator for Cloud SaaS Composition 194
7.12 The first main page of ranking services of Find SaaS 197
7.13 The execution paths of RbSR 198
7.14 The execution paths of composite costs 198
7.15 The Fitness function 200

8.1 The first main page of ranking services of Find SaaS 210

8.2 Selected the non-functional preferences for ranking the services 211

8.3 The main page of ranking the cloud services using the SNFCP ranking approach 211

8.4 The main page of ranking the cloud services using the M2NFCP ranking approach 212

8.5 The main page of ranking the cloud services using the LNFCP ranking approach 213

8.6 The main page of evaluation system of Find SaaS 215

8.7 The main page of cloud SaaS services using the the linguistic for evaluation the criteria 216

8.8 The decision maker for selection the best cloud software service after the testing the service 217

8.9 The main page for rule based evaluation system of cloud software services 218

8.10 The main page of selection after the decision using the percentage of selection or using the rule based evaluation system 219

8.11 The Find SaaS GA TOPSIS Composite solution 220

List of Tables

2.1 Comparison methods of Cloud SaaS Simple selection approaches . . . 27

2.2 Comparative Analysis of Existing Research Studies on Cloud SaaS Selection Attributes . . . 33

2.3 Existing Research Studies on Evaluation System of Cloud SaaS Provider . . . 36

2.4 Existing Research Studies on Cloud SaaS Service Composition . . . 42

5.1 Linguistic terms with their fuzzy numbers . . . 84

5.2 The examples of non-functional requirements for the price attribute . 87

5.3 The linguistic values used to measure consumer preferences . . . 96

5.4 Computer repair shop services with non-functional attributes . . . 105

5.5 Consumer request parameters . . . 106

5.6 BIS and WIS service vectors generated by CRH . . . 106

5.7 Weighting quality attributes using fuzzy logic . . . 106

5.8 Normalization matrix for the services . . . 107

5.9 The weighting normalization matrix for the services . . . 107

5.10 The values of DB and DW . . . 108

5.11 The values of CSB for each service . . . 108

5.12 Ranking result of consumer request . . . 109

5.13 Top 20 services of CRM by Capterra web site . . . 114

5.14 Find SaaS SNFCP queries input 115
5.15 Find SaaS M2NFCP queries input 116
5.16 Find SaaS LNFCP queries input 116
5.17 Summary of CRM dataset 117
5.18 The fuzzy number of ranking request to Find SaaS LNFCP 117
5.19 CRM result analysis 121
5.20 Definitions of QWS non-functional attributes 124
5.21 Summary of the QWS dataset 124
5.22 The service queries for the Find SaaS SNFCP approach 125
5.23 The service queries for the Find SaaS M2NFCP approach 126

6.1 Evaluation QoE 137
6.2 Linguistic variables of weighting qualities with their fuzzy numbers . 143
6.3 Linguistic variables of evaluating the qualities and their fuzzy number 144
6.4 Linguistic variables and their ranges 153
6.5 The rule 157
6.6 Linguistic values to evaluate the services 159
6.7 Linguistic weight criteria 160
6.8 Fuzzy numbers for weighting criteria 160
6.9 Fuzzy numbers for an evaluation of the services 161
6.10 Fuzzy numbers for evaluation matrix 161
6.11 The Final crisp number of all services 162
6.12 The positive ideal solution and negative ideal solution 163
6.13 The distance to the positive and negative ideal solutions 163
6.14 Similarity to the positive ideal solution 164

6.15 Ranking of the services for selection 164
6.16 The Consumer's evaluation criteria 166
6.17 The priority of qualities 166
6.18 The final evaluation score for the service providers 168

7.1 System notations 174
7.2 The Find SaaS GA TOPSIS interacting stages 175
7.3 Linguistic terms with their fuzzy numbers 179
7.4 Linguistic values of Service Rating (RA) and their fuzzy numer ranges 184
7.5 Linguistic values of service reviewers (RV) and their fuzzy numbers range 185
7.6 Linguistic values of reputation-based service reviewers (RbSR) and their fuzzy numbers range 186
7.7 The square rules (FAM) representation 187
7.8 Non-functional aggregation functions for sequence patterns 187
7.9 Non-functional preferences input 199
7.10 Consumer Constraints 200
7.11 The Top-20 Composite services generated by GA 202
7.12 The Top-20 Composite services generated by GA TOPSIS 203
7.13 The Top-20 Composite services generated by SAW 204
7.14 The Top-20 Composite services generated by TOPSIS 204
7.15 The Evaluation of Find SaaS GA TOPSIS Compared with other approaches using nDCG metric 205

A.1 Ranking Cloud SaaS top-20 CRM providers using Find SaaS SNFCP considering two non-functional preferences 245

A.2 Ranking Cloud SaaS top-20 CRM providers using Find SaaS SNFCP considering four non-functional preferences 246

A.3 Ranking Cloud SaaS top-20 CRM providers using Find SaaS SNFCP considering six non-functional preferences 247

A.4 Ranking Cloud SaaS top-20 CRM providers using Find SaaS M2NFCP considering two non-functional preferences 248

A.5 Ranking Cloud SaaS top-20 CRM providers using Find SaaS M2NFCP considering four non-functional preferences 249

A.6 Ranking Cloud SaaS top-20 CRM providers using Find SaaS M2NFCP considering six non-functional preferences 250

A.7 Ranking Cloud SaaS top-20 CRM providers using Find SaaS LNFCP considering two non-functional preferences 251

A.8 Ranking Cloud SaaS top-20 CRM providers using Find SaaS LNFCP considering four non-functional preferences 252

A.9 Ranking Cloud SaaS top-20 CRM providers using Find SaaS LNFCP considering six non-functional preferences 253

A.10 Ranking top 20 services for QWS Dataset using Find SaaS SNFCP considering two non-functional preferences 254

A.11 Ranking top 20 services for QWS Dataset using Find SaaS SNFCP considering four non-functional preferences 255

A.12 Ranking top 20 services for QWS Dataset using Find SaaS SNFCP considering six non-functional preferences 256

A.13 Ranking top 20 services for QWS Dataset using Find SaaS SNFCP considering eight non-functional preferences 257

A.14 Ranking top 20 services for QWS Dataset using Find SaaS SNFCP considering ten non-functional preferences 258

A.15 Ranking top 20 services for QWS Dataset using Find SaaS M2NFCP considering two non-functional preferences 259

A.16 Ranking top 20 services for QWS Dataset using Find SaaS M2NFCP considering four non-functional preferences 260

A.17 Ranking top 20 services for QWS Dataset using Find SaaS M2NFCP considering six non-functional preferences 261

A.18 Ranking top 20 services for QWS Dataset using Find SaaS M2NFCP considering eight non-functional preferences 262

A.19 Ranking top 20 services for QWS Dataset using Find SaaS M2NFCP considering ten non-functional preferences 263

B.1 Top 20 Cloud SaaS providers of Customer Relationship Management (CRM) based on Capttera 265

B.2 Top 20 Cloud SaaS providers of Energy Software based on Capttera . 266

B.3 Top 20 Cloud SaaS providers of Architecture Software based on Capttera 267

B.4 Top 20 Cloud SaaS providers of Project Software based on Capttera . 268

Abbreviation

AHP - Analytic Hierarchy Process

AV - Availability

CC - Cloud Computing

CRM - Customer Relationship Management

DMS - Decision Maker System

ELECTRE - ELimination Et Choix Traduisant la REalit

ERP - Enterprise Resource Planning

FAM - Fuzzy Associative Matrix

GA - Genetic Algorithm

HR - Human Resources

IaaS - Infrastructure as a Service

mAP - Mean Average Precision

MCDM - Multi-Criteria Decision Making

MIX - Mixed Integer Programming

MODM - Multi Objective Decision Making

MOO - Multi Objective Optimization

nDCG - Normalized Discounted Cumulative Gain

NF - Non-Functional

OU - Optimizer Unit

PaaS - Platform as a Service

PR - Price

PSO - Particle Swarm Optimization

RbSR -Reputation-based Service Reviewers

RS - Ranking System

RT - Response Time

RV - Service Reviewers

SaaS - Software as a Service

SAW - Simple Additive Weighting

SOA - Service-Oriented Architecture

SR - Service Rating

SRR - Service Registry Repository

TH - Throughput

TOPSIS - Technique for Order of Preference by Similarity to Ideal Solution

Chapter 1

Introduction

Cloud computing offers hardware resources and software applications over the Internet platform (Armbrust et al., 2010). Cloud computing is a new paradigm that offers on-demand resources such as infrastructure, platforms and software to customers in the same way that utilities such as water, electricity and gas do (Vijaya & Neelanarayanan, 2015). There are three main services in cloud computing: (1) Infrastructure as a Service (IaaS), (2) Platform as a Service (PaaS) and (3) Software as a Service (SaaS) (Mell, Grance, et al., 2011). Cloud computing has many advantages for service consumers, including providing them with access to their applications anywhere and at any time. It also offers advantages such as high scalability, reduced IT costs and a pay-as-you-go price model (Marian & Hamburg, 2012)(Vijaya & Neelanarayanan, 2015).

Cloud Software as a Service (SaaS) is a software application that can be accessed over computer networks. There are a growing number of Cloud SaaS providers that can be found on the internet. Different service providers of Cloud SaaS offer services to their consumers such as typing software, drawing software, marketing software, human resources (HR) software, IT software and monitoring software and more. The Cloud SaaS run through the service provider's data center, thereby eliminating the need to install the applications on consumers' machines so that the user only needs a web browser or application programming interface (API) to access them. (Alkalbani, Ghamry, Hussain, & Hussain, 2015a) found that in 2015, there were around 5,000 Cloud SaaS applications offering different kinds of services. In fact,

the last study conducted by IDC (International Data Corporation) shows that the Cloud SaaS market made 48.8 billion US dollars in revenue in 2014 and it is expected to surpass $112.8 billion by 2019 (Benjamin McGrath, 2015). Cloud SaaS is growing five times faster in the global software market than traditional licensed software.

The variety of Cloud SaaS providers makes the selection of the optimum service difficult. Service selection is based on meeting the requirements of functional and non-functional quality attributes (Wagle, Guzek, & Bouvry, 2015). In fact, the service consumer may face difficulty when searching for a service among the huge number of Cloud SaaS providers that have similar functional categories such as HR software, but having different non-functional quality attributes such as service price and service availability (Badidi, 2013) (ur Rehman, Hussain, & Hussain, 2011) (Wagle et al., 2015). Thus, the service selection of Cloud SaaS that has met with the service consumer's requirements of non-functional quality attributes is a key factor in Cloud SaaS selection. In addition, Cloud SaaS providers often involve a long-term contract, so Cloud SaaS providers frequently offer a trial version to test and evaluate services before requiring a contract for with selected service. Therefore, this research proposes intelligent methods to select the optimum Cloud SaaS based on the consumer's preferences. To achieve that purpose, the service broker is developed to understand the service consumer's requirements and meeting with the service provider that satisfy their preferences. This work classifies the service consumer's preferences of non-functional attributes into two shapes: (1) Weighting the importance of attributes and (2) Quantifying the real value of attributes. The Fuzzy logic concept has been used to address the weighting of attributes using the linguistic terms for describing the consumer's preferences such as service price weights as an Important factor among other attributes. Regarding the second shape of preferences, we have addressed multiple options of service requirements of attributes, as follows:

- Single-based value of attributes (e.g. service price $20).
- Minimum and maximum-based values of attributes (e.g. service price $20 - $40)
- Linguistic-based value of attributes (e.g. affordable price)

This thesis has developed three algorithms to address these multiple options of preferences. This is done by ranking and sorting Cloud SaaS providers by considering the non-functional quality preferences, and by using the measuring of the shortest distance to the service provider that has matched with the service consumer's requirements.

In addition, this thesis has developed an evaluation system to assess the service providers in order to select the optimum service based on the service consumer's experiences. The Quality of Experiences (QoE) are collected first that may be considered by a service consumer when evaluating the service provider. Reviewing the literature and asking some expert people in Cloud SaaS selection are two approaches to obtain all these attributes. The evaluation system is based on two intelligent methods: (1) Linguistic values to evaluate the individual experience, then using the decision- maker approach for final service selection. (2) Fuzzy rules system by evaluating the individual experience as a percentage, then using the aggregation function to make a decision for selecting the service.

Finally, a key issues related to service composition of Cloud SaaS has addressed in this thesis. The service composition is the second flavor of service based on the service-oriented architecture (SOA) (Papazoglou, 2003). The service composition involves selecting aggregating of multiple services in order to fulfil consumer's requirements of functional service categories. This is very important as Cloud SaaS services are often the first option for new enterprises to deploy IT services for their organization, and are usually preferred to select multiple services to satisfy their

requirements of functional services. Service composition is based on declaring a workflow by using a language such as BPEL (Jordan et al., 2007) that includes some abstract services (i.e. tasks) to obtain a specific job. After that is done, a list of candidate concrete services that match each task is compiled. Therefore, the goal of service composition approaches addressed selecting the best candidate service for each task in order to satisfy user requirements (Alrifai & Risse, 2009). Different studies have been conducted recently that examine the service composition in web services and the cloud service levels (i.e. IaaS, PaaS and SaaS). However, very few publications can be found in the literature that discusses the issue of the selection of the Cloud SaaS composition taking into account consumer's preferences. Additionally, the majority of previous works focused on developing their approaches based on an optimization problem, and mainly they focused on reducing the search space to select optimum services. This means no decision action takes into account selecting the best composite service after aggregating the services taking into account the consumer's preferences. Therefore, this research aims to introduce a service composition model for Cloud SaaS services based on consumer's preferences. Moreover, our research examines an alternative technique for multi-criteria decision-making (MCDM) that uses the technique for order of preference by similarity to ideal solution (TOPSIS) for final composite selection. The reputation factor has been addressed in the optimization problem for selecting the composite services that have maximum reputation factor.

In order to evaluate our proposed approaches, there are multiple methods conducted in this thesis for measuring the performance of proposed approaches. In terms of simple Cloud SaaS selection, the ranking system engine is proposed; therefore, we measured the performance by comparing the result with two existing approaches by using the mAP and $nDCG$ metric. The QWS (Al-Masri & Mahmoud, 2007) dataset is used as a benchmark for measuring the performance of our proposed method for

simple Cloud SaaS selection. In addition to this, a new dataset was collected that has the top 20 Cloud SaaS providers that have CRM functionality. Furthermore, in terms of demonstrating the evaluation system approach proposed, a case study is introduced to achieve this purpose. Finally, the service composition approach is evaluated using the case study to explain the service workflow scenario, then the experiment is conducted to find and select the optimum Cloud SaaS selection. Also, the $nDCG$ is a performance metric used to evaluate the performance of the service composition approach proposed. In this thesis, also we provide the system prototype to evaluate the effectiveness of the approaches proposed.

The remainder of this chapter elaborates the significance of the thesis, scientific and social contributions, research problem addressed in this thesis and the organization of the thesis.

1.1 Significance of the Thesis

This thesis is the first and only attempt in the literature to simple and composite Cloud SaaS selection, by considering consumer's preferences. Specifically, the significance of this thesis arises from the following:

1. This thesis introduces an intelligent methodology for simple Cloud SaaS selection, considering the consumer's preferences. The proposed approach developed ranking service providers using multiple options of non-functional consumer's requirements. The fuzzy logic approach is used to weight the most important quality attributes for the service consumer. In addition, multiple algorithms have been proposed for addressing the consumer's requirements of particular quality attributes. Finally, the ranking approach is developed by measuring the shortest distance to the service consumer's requirements.

2. This thesis developed an intelligent approach for evaluation system of Cloud

SaaS provider using the linguistic for assessing the quality of experiences. The evaluation system uses the fuzzy logic based approach to address the uncertainty of quality of experiences and to develop the multi-criteria decision making (MCDM) approach for final service selection.

3. This thesis develops an intelligent approach for evaluation system using the fuzzy rule system for assessing the service provider to select an optimum service provider after testing the services by the service consumer. The fuzzy rule system is proposed to achieve this objective using the fuzzy associative matrix (FAM) for generating the fuzzy rule to make a decision and select the service provider.

4. This thesis has proposed and used the reputation factor for service composition. The optimization problem has been addressed to select the best composite service that has the maximum reputation value. The reputation factor is compared by using the service rating and service reviewers using fuzzy logic. The reputation-based service reviewers (RbSR) is called to describe the reputation factor.

5. This thesis has developed smart approach for Cloud SaaS service composition using the genetic algorithm (GA) for aggregating the service providers that have maximum reputation factor and minimum service cost. In addition to this, the decision maker system has been proposed to select the best service composition generated by the GA in order to select the service composition that has been matched with the consumer's preferences.

1.2 Scientific Contributions

1. This thesis focuses on developing Cloud SaaS service selection by developing a ranking system engine. In order to address the uncertainty of consumer's re-

quirements, the fuzzy logic has been developed to achieve this purpose. Also, the ranking system is used for measuring the similarity to non-functional consumer's requirements. Multiple metrics such as $nDCG$ and mAP have been used to evaluate the proposed approaches

2. This thesis also focuses on developing the evaluation system approach by combining two scientific approaches which are fuzzy logic and multi-criteria decision making. Fuzzy logic is used to translate the linguistic terms to evaluate the individual attribute for each service. After collecting all evaluation mechanisms for each service provider, the multi-criteria decision process using the TOPSIS method is used to decide and select the optimum Cloud SaaS provider.

3. This thesis also focuses on developing the evaluation system using the fuzzy rule- based approach. The fuzzy associative matrix (FAM) is the technique used for generating the fuzzy rule for evaluation of the Cloud SaaS providers. In the end, the aggregation function is used to decide and select the optimum service provider after the evaluation process.

4. This thesis also focuses on developing the service composition of Cloud SaaS by using a combination of two scientific methods: Genetic Algorithm (GA) and Multi-criteria decision making (MCDM) approach. The genetic algorithm has been developed to address the optimization problem when aggregating the service providers and has been developed to select the service providers by maximizing the reputation factor and minimizing the cost of aggregating services. After that, the MCDM approach by the TOPSIS method has been used to decide and select the optimum composite service from those generated using the genetic algorithm approach.

5. This thesis compares the performance of proposed approaches using different benchmark techniques such as the dataset used, multiple traditional ap-

proaches, performance metrics. In the end, the system prototype is proposed to demonstrate the proposed approaches.

1.3 Social Contributions

1. The proposed approach helps a Cloud SaaS consumer to find and select the optimum service based on their preferences. It takes the quality attributes on consideration of either the quality of service (QoS) for ranking the services or the quality of experience (QoE) for evaluating the service.

2. The proposed approach helps the service consumer, especially the new business enterprise to focus on deploying the Cloud SaaS solution most appropriate to their business. It is known that selecting the appropriate Cloud SaaS provides an opportunity to reduce the cost of IT expenditures by minimizing the cost of operation, maintenance and modification. The Cloud SaaS composition is usually used by a Cloud SaaS consumer who is looking to select aggregating service providers. Hence, by using the proposed approach, the service consumer can select an optimum service composition to deploy the Cloud SaaS solution.

1.4 Research Problems and Contribution

In this research, the primary objective is to help a Cloud SaaS consumer to select simple or composite service providers based on its preferences. To achieve this purpose, the service broker is called the Find SaaS to be as an broker between the service consumer and the service providers. Our research addresses the following research problems makes the following research contributions:

1.4.1 Research Contributions in Simple Cloud SaaS Selection

The simple Cloud SaaS selection is the process to select a single service provider that matches with the Cloud SaaS consumer's requirements in terms of a functional and non-functional attribute such as service price. The following are the major shortcomings which were identified as needing to be addressed:

Cloud SaaS consumer's preferences are ignored in most existing approaches. The weighting of non-functional attributes is only considered in the previous research to address the consumer's preferences. In addition, the previous researchers used either compare pairs using the criteria selected or by crisp number to weight the importance of attributes.

To address the aforesaid problem, a new ranking system is proposed to rank and sort the Cloud SaaS providers based on service consumer's preferences. The system proposed classify the consumer's preferences into two shapes: (1) weighting the quality attributes to identify the importance of attributes comparing with others. The linguistic terms are used to weight the attributes such as the service price which is weighted as Important. (2) The service consumer's requirements for individual attributes as the attribute value such as the service required with $20. Also, three approaches are proposed to identify these values to give more options for a Cloud SaaS consumer to present their preferences. They are as follows:

- Single-based value, such as the service price required is $20. This proposed approach is called Find SaaS SNFCP.
- Minimum and maximum based value such as the range of service price required is between 15$ to 20$. This proposed approach is called Find SaaS M2NFCP.
- Linguistic-based value such as the service price required is an affordable price. This proposed approach is called Find SaaS LNFCP.

The system proposed ranks the Cloud SaaS providers based on measuring the shortest distance to the service consumer's requirements. The proposed approaches have been tested and evaluated with different real services and comparing the result with some traditional ranking approaches.

1.4.2 Research problem and Contribution in Evaluation of Cloud SaaS Providers

The evaluation system is the process for testing and evaluating the Cloud SaaS provider by a service consumer taking the advantages that Cloud SaaS providers offer for testing the service for a one-month trial version. The evaluation service is an essential process to select the best service provider that matches with the service consumer's experiences. The following are the major shortcomings which were identified as needing to be addressed:

1. The majority of existing approaches focus on developing their approach using the comparison features for each service provider to make a service selection. The linguistic terms to evaluate the quality of experiences (QoE) are missing in the existing approaches. Also, the rule-based system to evaluate the Cloud SaaS provider is missing in the existing approaches.

2. In terms of the Quality of Experience (QoE), the previous researchers proposed the list of quality attributes without studying which of the attributes are suitable for evaluation and which are not. That means they provide the quality attributes that a service consumer should consider when selecting the service provider, and they provide the definitions of these quality attributes.

A new list of quality attributes that describe the Cloud SaaS consumer's experiences are presented. For evaluation of Cloud SaaS providers, the new evaluation system has been proposed to help a Cloud SaaS consumer to evaluate the Cloud SaaS

provider before making a contract with the selected service. The system proposed has two options for evaluating the Cloud SaaS provider, as follows:

1. The evaluation system proposed is based on linguistic terms to evaluate Cloud SaaS provider. The MCDM approach as TOPSIS method is applied for final decision to find and select the optimum service provider. This approach is called the Find SaaS Linguistic Evaluation System.

2. The evaluation system proposed is based on using the fuzzy rule that takes evaluation attributes with the importance of that attribute as the system input, then generates the evaluation score based on these two inputs as the output. After that, the system aggregates all the outputs in order to generate the final score obtained for a selected Cloud SaaS provider. Ranking all the service providers beginning from the service that has the biggest number for the evaluation score is the last step the system process.

1.4.3 Research problem and contribution in Composite Cloud SaaS Selection

The service composition of Cloud SaaS is the process to select multiple Cloud SaaS providers in order to satisfy the functional workflow for the Cloud SaaS consumer. This is very important for some specific Cloud SaaS consumers who are looking to select multiple services for deploying the Cloud SaaS solution for their work environment. The following are the major shortcomings which were identified as needing to be addressed:

1. The majority of existing approaches are focused on reducing the search space for aggregating the service provider from each task by developing the multi-objective optimization approach. That means they have only developed their approaches to aggregate the optimum service provider for each task. Therefore,

the decision maker has no input in selecting the optimum composite services that are generated using this method.

2. The reputation factor for the service composition is ignored in the existing approach. The majority of previous researches have aimed to find and select the optimum composite services based only on minimizing the negative quality attributes such as the service price and maximizing the positive quality attributes such as the service availability.

To address the above problems, we propose intelligent Cloud SaaS composite service selection algorithms. We term as Find SaaS GA TOPSIS. Find SaaS GA TOPSIS addresses the selection of composite Cloud SaaS using the genetic algorithm and TOPSIS method. In the first step, the genetic algorithm is applied to address the multi-objective optimization problem for selecting the aggregate/composite services that have maximum reputation factor and minimum cost. The result of this step is to list the composite services that matched with the service consumer's constraints of reputation factor, and the composite cost. The result of the multi-objective optimization problem is called the Pareto set. Therefore, the TOPSIS method is applied to select the optimum composite service that has matched with the consumer's preferences for other quality attributes such as composite availability and composite throughput. In addition to this, the new reputation factor is proposed using service reviewers and service rating. This reputation factor is called RbSR, which is short for Reputation-based Service Reviewers.

1.5 Thesis Outline

In this thesis, the intelligent approaches have been proposed to help a Cloud SaaS consumer to select simple and Composite Cloud SaaS providers. The proposed system methodologies are divided across different chapters. Each of the chapters

presents the proposed techniques to achieve the objectives. This thesis is divided into nine chapters and is organized as follows:

- Chapter 1 provides a brief introduction to the thesis topic, defines the significance of this work and outlines the research problem with the major contributions.
- Chapter 2 presents the existing state-of-the-art approaches that have been proposed for simple and composite service selection of Cloud SaaS.
- Chapter 3 defines the main keywords that have been used in this work. The main research problems addressed in this thesis are defined in this chapter. Furthermore, the main research methodologies that have been used to achieve our research objectives are introduced in this chapter.
- Chapter 4 introduces an overview of the solutions to issues identified in this thesis. Related sections and subsections of proposed solutions are presented in this chapter.
- Chapter 5 presents the ranking system methodology to address the simple Cloud SaaS selection. The main experiments, results and performance of this work are disused in this chapter.
- Chapter 6 presents the evaluation system methodology of a Cloud SaaS provider. Two solutions are introduced in this chapter. The case study to demonstrate our work is also presented in this chapter.
- Chapter 7 presents the methodology of composite Cloud SaaS selection. The experiments, results and evaluation are discussed in this chapter.
- Chapter 8 provides the system prototype to demonstrate the solution methodologies of this work.

- Chapter 9 concludes the thesis by giving a summary of the main contributions, limitations and potential of future work.

Chapter 2

Literature Review

This chapter presents a brief overview of the relevant research and the state-of-the-art approaches in Cloud SaaS simple and composite selection. The issues and limitations of existing approaches are discussed to identify research gaps and research questions of this thesis.

2.1 Introduction

In this chapter, we present an overview of the existing literature on Cloud SaaS Simple and Composite selection. An overview of Cloud Computing with their advantages and classifications are discussed in the chapter. This chapter has been classified into four main sections, as follows:

In Section 2.2, we present an overview of the concept of Cloud Computing (CC), including the service classifications. Also, we cover the concept of Service-Oriented Architecture (SOA) with main definitions of some relevant research keywords such as service broker, service selection and service composition.

In Section 2.3, we present an overview of existing approaches in Cloud SaaS service selection methods, including the main attributes authors concerns when proposed their approaches. Also, we cover the existing literature review on evaluating Cloud SaaS mechanisms. Finally, we present the main shortcomings of existing approaches.

In Section 2.4, we present an overview of existing approaches in Cloud SaaS service composition methods, we also discuss the main shortcomings of existing

literature review in this section.

In Section 2.5, we present an overview of the main dataset that the authors used for tested and evaluated their proposed approaches, we will discuss the advantages and disadvantages of using these datasets.

2.2 The Concept of Cloud Services

2.2.1 Basic concept of Cloud Computing (CC)

Cloud Computing is a new paradigm that provides the hardware resources as well as the software application over the Internet platform (Armbrust et al., 2010). There are a number of different definitions of cloud computing. The National Institute of Standards and Technology (NIST) (Mell et al., 2011) defines cloud computing as follows:

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models".

Furthermore, Cloud Computing becomes the first option for the organization to provide the IT services. It also clear that Cloud Computing has utility-based model similar to electricity and water for providing computing resources. For instance, the organization does not need to build power plants to obtain the electricity required, while it needs to contract with the electricity service provider to obtain electricity they need. It is a similar case with water. Therefore, the new businesses can obtain the IT services from the companies that offer IT services over Cloud Computing in different service levels (i.e. IaaS, PaaS and SaaS). As a result, we can summarize

that Cloud Computing has great benefits to the customers such as offering access to their application, anywhere and at any time. It also offers such advantages such as high scalability, reduced IT costs and the pay-as-you-go price model (Vijaya & Neelanarayanan, 2015).

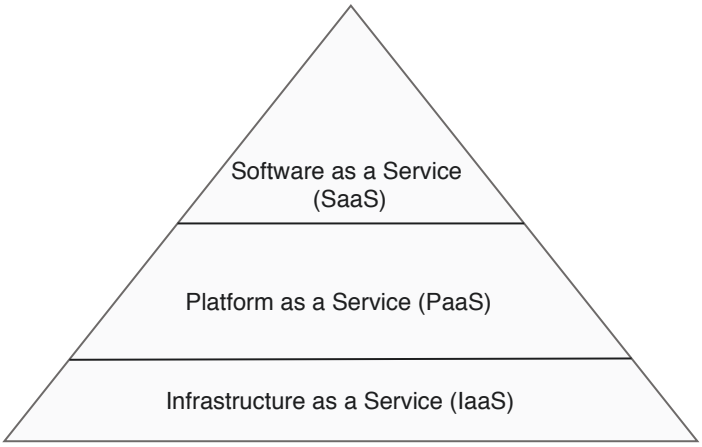


Figure 2.1 : Cloud Computing Services Architecture

The services in the Cloud Computing can be classified into three main services as shown in Figure (2.1) which are as follows (Mell et al., 2011):

1. **Infrastructure as a Service (IaaS)** provides computing resources through the Internet and the user can scale up or scale down based on the user's requirements. This service level includes virtual computers, virtual storage and virtual central processing units (CPUs). The leading vendors in this category are Amazon EC2, Amazon S3, GoGrids cloud server and Joyent.

2. **Platform as a Service (PaaS)** provides tools to manage the user's infrastructure resources such as operating systems, databases and web servers. The

most notable vendors under this classification are Google App Engine, Microsoft Azure and Salesforce [*].

3. **Software as a Service (SaaS)** is a software application that enables users to reach a service over a connection such as the Internet. This type of service has grown dramatically in usage since it provides significant advantages to the user, such as reducing costs and removing the need to build IT in-house. Cloudera [†], Dropbox [‡] and MindMeister [§] are such example of SaaS providers.

Cloud computing can be classified into four deployment models as follows (Jula, Sundararajan, & Othman, 2014) (Mell et al., 2011):

1. **Public cloud**: This is the main model of cloud computing. It provides virtual resources that can be accessed over an Internet platform. Cloud owners such as AWS and Microsoft Azure provide public services to customers based on policies, service features, service level agreements (SLA) and pricing models.

2. **Private cloud**: This kind of cloud computing is designed to offer all the same cloud features similar to the public cloud but only available to specific organization/person. VMware and OpenStack are two examples where an organization can transfer their datacenter to a cloud computing environment.

3. **Community cloud**: This kind of cloud computing offers shared cloud computing to be used by some specific community members only.

---

[*]https://www.salesforce.com

[†]https://www.cloudera.com

[‡]https://www.dropbox.com

[§]https://www.mindmeister.com

4. **Hybrid cloud**: This kind of cloud computing refers to the combination of all previous cloud types (private, public and hybrid).

In this thesis, we focus on Cloud Software as a Service (SaaS) because this is the area where the most growth is currently occurring and it has the highest demand in comparison with other cloud services (i.e. PaaS and IaaS) (Safari, Safari, & Hasanzadeh, 2015) (Tsai, Bai, & Huang, 2014).

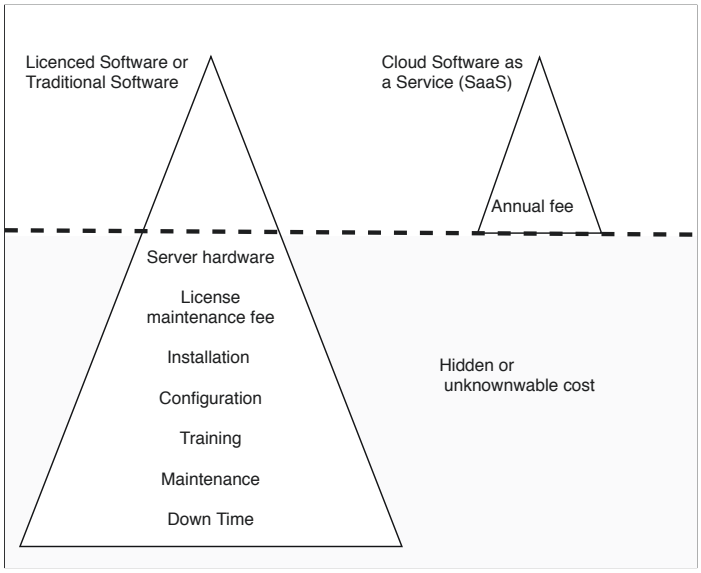


Figure 2.2 : Licensed software versus Cloud SaaS

(Waters, 2005)

Cloud Software as a Service (SaaS) is a software application that can be accessed over computer networks. It runs on vendors datacenter cloud without installing it on the end users' machine (Tyrväinen & Selin, 2011). There are several cloud software applications offering services to consumers such as drawing software, marketing software, human resources (HR) software, IT software and monitoring software and more. Alkalbani et al. found that in 2015, there were around 5,000 Cloud SaaS Service providers offering different kinds of services (Alkalbani, Ghamry, Hussain,

& Hussain, 2015b). In fact, the last study conducted by IDC (International Data Corporation) shows that the Cloud SaaS market made 48.8 billion US dollars in revenue in 2014 and it is expected to surpass $112.8 billion by 2019 (McGrath, 2015). Finally, Cloud SaaS is growing five times faster in the global software market than traditional licensed software. Figure (2.2) shows a comparison between traditional software and Cloud SaaS (Waters, 2005).

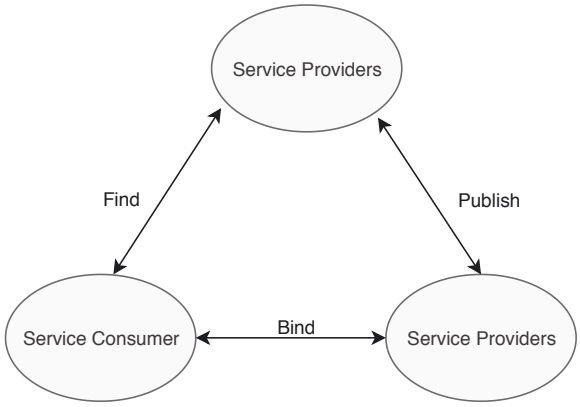


Figure 2.3 : Service-Oriented Architecture (SOA) Model

Cloud SaaS is a web-based service which can be demonstrated in the Service Oriented Architecture (SOA). SOA aids in managing large problems such as mission-critical applications by analyzing the problems sincerely by decomposition it into smaller manageable pieces which are interrelated to each other. In the SOA model, three basic entities are interacting and exchange the message between them as shown in Figure (2.3) (Papazoglou, 2003):

1. **Service Provider**: The service provider describes a service description of a service and publishes it to a service consumer or service registry through a

service description. The service providers make their services available publicly or privately based on the service consumer situation. Also, the service provider publishes important attributes that describe the characteristics of their services, such as price. The term service provider or Cloud SaaS service provider are synonymous in this thesis.

2. **Service Consumer**: The service consumer can define as the service requester uses a find operation to retrieve the service description typically from service registry or repository like UDDI and uses the service description to bind with the service provider and entreat the service or interact with service implementation. The term service consumer or Cloud SaaS service consumer are synonymous in this thesis.

3. **Service registry**: The service registry contains all the service descriptions launched by service providers. The service registry may include some important attributes that describe the service characteristics. The term service registry repository or pool of service or list of services are synonymous in this thesis.

The concept of service broker is key in this thesis. The service broker is the middleware between the service consumer and service provider. It acts as a bridge between the service consumer and service provider in order to help a service consumer to select an optimum service. The service can be shaped into two types: simple or composite service (Papazoglou, 2003). Figure (2.4) describes the service broker architecture.

In the next section, we discuss the state-of-the-art of Cloud SaaS selection including the evaluation system of Cloud SaaS provider.

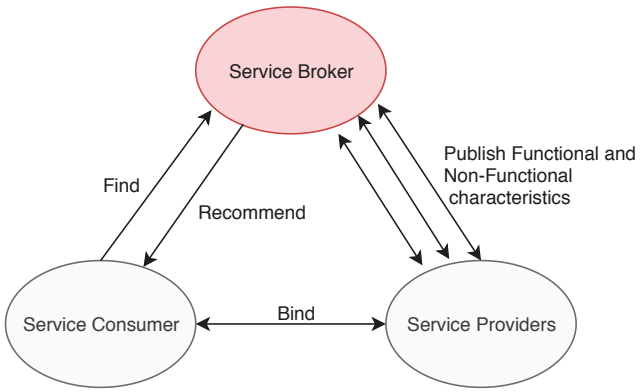


Figure 2.4 : Service Broker Architecture

2.3 Overview of Existing Research on Non-Functional Simple Cloud SaaS Service Selection

This section provides a brief overview of the recent advances and the various techniques proposed by the researchers for Cloud SaaS simple service selection.

In recent years, there are growing in the number of Cloud SaaS service providers. This has increased the difficulty of selecting an appropriate service. For example, according to Cloud SaaS engines such as Capttera, there are more than 500 services matching CRM functionality.

Cloud SaaS service selection is based on matching the functional of service characteristics and non-functional attributes such as service price. Therefore, several scientific types of research have focused on service selection based on developing semantic approaches for matching the functional elements of the service (Kang & Sim, 2011) (Rezaei, Chiew, Lee, & Aliee, 2014). Other researchers have focused on

developing selection mechanisms by studying non-functional attributes. The proposed multiple techniques by using multi-criteria decision making (MCDM), multi-objective decision making (MODM) or recommender system to select a Cloud SaaS service provider. In fact, Cloud SaaS service selection based on non-functional quality attributes has increased in popularity due to the increasing number of services with similar functional service characteristics.

The existing literature on Cloud SaaS selection are classified on the following:

1. **Decision Maker Approach**: This approach proposes service selection as a decision maker using different decision maker methods such as multi-criteria decision making (MCDM) or multi-objective decision making (MODM). A ranking system approach is proposed in order to rank and sort all the Cloud SaaS service providers for a service consumer matching their preferences.

2. **Studying Non-Functional Quality Attributes**:Selecting important non-functional parameters during Cloud SaaS service selection is important. This kind of work is very necessary to help a Cloud SaaS service consumer in identifying the important factors for selecting the service.

3. **Proposing Evaluation System for Cloud SaaS Selection**: This kind of contribution is to deal with a service after selecting the service or considering the advantage of one-month trial version that a Cloud SaaS service provider offers for a service consumer.

The rest of the section is composed of three main subsections as shown in Figure (2.5): (a) Decision maker approach for selecting a simple service selection based on non-functional attribute, (b) The main non-functional attributes or Quality of Service (QoS) for selection Cloud SaaS service provider, and (c) Evaluation system of Cloud SaaS service provider.

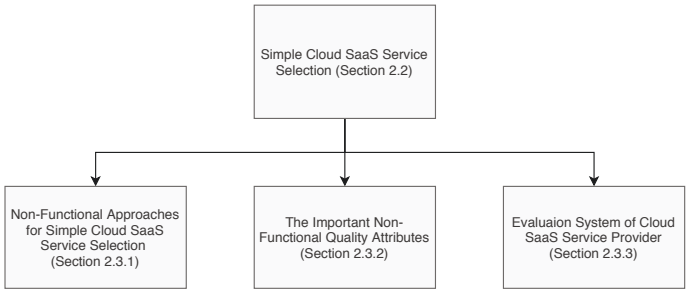


Figure 2.5 : Overview of the literature review on Cloud SaaS Service Selection

2.3.1 Non-Functional Approaches for simple Cloud SaaS Service Selection

An Analytic Hierarchy Process (AHP) for Cloud SaaS selection Godse and Mulik (2009)introduced a framework to select a Cloud SaaS service provider based on multi-criteria decision making (MCDM) using the AHP approach. The authors first investigate the importance of various quality attributes in selecting the Cloud SaaS service provider. The result was the following criteria: functionality, architecture, usability, vendor reputation and cost. The following stage proposed AHP MCDM methods for weighting the most critical quality attributes to address the consumers' preferences and selecting the best service using the same AHP technique. Finally, the authors present a case study to verify their approach (Godse & Mulik, 2009) .

In addition, Mital et al. (2014) proposed a framework to identify the main criteria that affect a Cloud SaaS consumer when adopting Cloud SaaS solutions for new businesses. The authors discuss the motivation for going to deploy Cloud SaaS to the business rather than an on-premises software solution. The authors proposed weighting the importance of criteria using the AHP technique. The authors conclude that five main principles affect a Cloud SaaS service consumer to adopt the Cloud

SaaS as follows: technology, process, cost, quality and the network. However, the major drawback of this approach is the lack of a selection method for final selection (Mital, Pani, & Ramesh, 2014).

Selection a Cloud SaaS-based on user preferences (Fuzzy AHP) approach Boussoualim and Aklouf (2015) proposed an AHP with the fuzzy measure to quantify software attributes and extract the best Cloud SaaS providers among different services. The authors proposed a matching algorithm to obtain all similar services required by a service consumer. The authors also investigated the leading quality attributes concerns by a Cloud SaaS consumer, which can be summarized as follows reputation, cost, usability, structure, configurability and personalization. The problem with this approach is in that it is only suitable with a small number of services or alternative options since the AHP mechanism used pairwise comparison. As a result, more and more processing is required to analyze the user's preferences as well as for making a final decision about which service to adopt. (Boussoualim & Aklouf, 2015a).

A selecting framework for Cloud SaaS-based ERP software A few methods were considered selecting the specific type of Cloud SaaS functional types, such as Venkataraghavan and Sundarraj (Sundarraj et al., 2011). In their approach, a selection mechanism for a Cloud SaaS based on enterprise resource planning (ERP) service by studying the differences between the Cloud SaaS-based on ERP and the regular software type. The authors proposed a selection method by using the two-stage process, (1) analyzing the considerations for choosing the Cloud SaaS-based solution, (2) applying the fundamental objective hierarchy.

Selection Cloud SaaS using aggregated the Utility function To address a large number of Cloud SaaS service providers, Bididi (2013) proposed a framework for Cloud SaaS service selection using the concept of a cloud service broker (CSB) as

a mediator service that helps to understand the consumer's requirements and then select the most appropriate service to the consumer. The framework is based on matching the user's needs with Cloud SaaS providers and then ranking the services using an aggregate utility function. However, the authors test their approach using a simulated dataset which contains only four services (Badidi, 2013).

Selection cloud service provider using the TOPSIS MCDM approach Arman et al. (2016) introduced a ranking system of cloud services to help a service consumer to select the best cloud service based on consumers requests of cloud service. The ranking system is based on MCDM TOPSIS approach. Ching-Lai Hwang and Yoon in 1981 developed the TOPSIS standard for Technique for Order of Preference by Similarity to Ideal Solution. In addition, in their approach, the authors considered the final cloud service selection based on a consensus-based voting technique by using the Borda score. A service consumer selects the service with the highest Borda score. However, the authors ignore the main consumer requirements as numerical values to give more options to a service consumer to present their preferences (Arman, Foresti, Livraga, & Samarati, 2016).

Ranking cloud services by considering the weighting of important criteria using the rough set theory Liu et al. (2016) proposed an assessment method of the importance of Cloud SaaS criteria/attributes using rough set theory. They start by explaining the main quality attributes that may considers by a service consumer when selecting the cloud service. The authors' summarize their criteria as follows: availability, scalability, elasticity, security, innovation, cost, capability, solution and brand influence (Liu, Esseghir, & Boulahia, 2016).

Recommender system for Cloud SaaS service provider selection Afify et al. (2017) proposed Software-as-a-Service (SaaS) Recommender (SaaSRec), a personalized reputation-based system based on QoS- ranking cloud software services.

The basic concept of the recommender system relies on measuring the matching preferences between the active user who requested the service and the previous users. Therefore, the system will display all the services that are of interest to the active user for service selection based on similar characteristics of the previous user. The recommender system has widely been used in the web services. The authors used semantic matching of functional cloud services, then matched the active user with the previous user in terms of cost, profile and reputation. The authors test and evaluate their approach by using WS-Dream dataset (Afify, Moawad, Badr, & Tolba, 2017).

Table 2.1 : Comparison methods of Cloud SaaS Simple selection approaches

Author (s)	Ranking System for service selection measuring the the shortest distance to consumer's preferences	Consumer Non-Functional Preferences				Dataset	Method (Mechanism/Type)
		Weighting non-functional attribute by linguistic/subjective method	Consumer Non-Functional Requirements				
			Single Requirements	Minimum and Maximum Requirements	Linguistic Requirements		
Godse and Mulik (2009)	No	No	No	No	No	Simulated	MCDM /AHP
Mital, Pani, and Ramesh (2014)	No	No	No	No	No	-	-
Boussoualim and Aklouf (2015)	No	No	No	No	No	Simulated	MCDM/AHP
Sundarraj et al. (2011)	No	No	No	No	No	-	-
Badidi (2013)	No	No	Yes	No	No	Simulated	Aggregated Utility function
Arman, Foresti, Livraga and Samarati (2016)	No	No	No	No	No	Simulated	TOPSIS and Borda score for final selection
Liu, Esseghir and Boulahia (2016)	No	No	No	No	No	Real services	-
Afify, Moawad, Badr and Tolba (2017)	No	-	No	No	No	Web services (WS-DREAM)	Matching Algorithm

Based on the comparative analysis in the area of selection simple Cloud SaaS service provider presented in table 2.1, we can summarize the shortcomings of the existing methods as follows: (1) The existing research works are focused on selection mechanisms, which only consider the weighting of non-functional attributes as the consumer's preferences. (2) They proposed their approach using selection mechanism such as pair-wise comparison, or by matching with the previous service consumer, however, no one proposed a ranking system in order to find the service that aligns most closely to a service consumer request. (3) Finally, they only test their approaches using a few services. Hence, (Ikram, Sharma, Raza, & Hus-

sain, 2019) proposed a cloud software service selection considering the linguistic for weighting the most important criteria as well as taking in to account the minimum and maximum of consumer's requirements. We also test our approach with using more than 2500 services. We evaluated our approach against existing approaches and found a superior result in terms of performance of selection match with the consumer's preferences. In addition, (Ikram & Hussain, 2018) introduced ranking services considering the consumer's requirements as a single of service consumer requirement describing consumer preferences. We provide a case study to select an optimum service among multiple services that have computer repair shop functionality. Moreover, both proposed techniques are used to match with service consumer preferences to find an optimum service provider.

2.3.2 The Important of Non-Functional Quality Attributes

The non-function of Cloud SaaS can be classified into two major types: (a) Quality of Service (QoS): This includes all major attributes that can be measured either by client-side or server-side. All the criteria in this type are measured correctly without considering the service consumer perspective. For an example of this type, service price, service availability and service rating. (b) Quality of Experience (QoE): This includes all factors describing the consumer's experience for using a particular service. It consists of many factors, including the quality of service. However, this is usually described by a service consumer. Therefore, this information is typically ambiguous and "vague" not specific. For example, the usability is "Excellent" or the service price is "Very Expensive".

Therefore, the non-functional attributes of Cloud SaaS can be classified either qualitative type or quantitative type. The qualitative types usually described by a consumer experience or by an expert of cloud services. It includes many factors such as security, performance and support. On the other hand, the quantitative type can

be measured numerically using information collected either consumer-side or server-side. This section looks at existing research work that have been conducted recently in order to identify the most important non-functional attributes that should be considered when selecting a Cloud SaaS provider.

Design since research methodology for analysing 651 services Repschleager et al. (2012) investigated the main criteria/attributes of Cloud SaaS to help service consumer to select an optimum service. The authors picked all criteria based on studying the characteristics of the Cloud SaaS service provider. In this study, they followed a Design since research approach and conducted a systematic literature review. They conducted an extensive market analysis of 651 Cloud SaaS service providers and found that there are six criteria for selecting a Cloud SaaS provider: maintenance and service cycle, functional coverage, service category, user scaling, probability of date and browser compatibility (Repschlaeger, Wind, Zarnekow, & Turowski, 2012).

Quantify the Quality of Experience (QoE) based on other quality attributes Al-Shammari and Al-Yasiri (2014) provided valuable information on how to quantify the quality of experience (QoE) in order to help a service consumer and service provider obtain a service level agreement (SLA). They proposed a method to quantify the quality of experience (QoE) attributes based on numerical value assigned by a service provider or by other available resources (Al-Shammari & Al-Yasiri, 2014).

Main definitions for 33 Quality of Services (QoS)/ non-functional attributes Khanjani et al. (2014) conducted a review study on quality attributes for Cloud SaaS. The study includes reviewed on 33 quality attributes. The quality attributes explained in the study as follows: Adaptability, Extensibility, flexibility, customizability, scalability, changeability, composability, availability, maintainabil-

ity, reliability, fault tolerance, recoverability, stability, serviceability, robustness, accuracy, efficiency, functionality, response time, suitability, data integrity, privacy, security, accessibility, learnability, commonality, multi-tenancy, operability, agility, assurance, performance, security management and usability (Ghani & Sultan, 2014).

Comparing the characteristics of Cloud SaaS and on-premises software service to determine the essential QoS of Cloud SaaS Burkon (2013) establishes a set of quality attributes to manage Cloud SaaS services. The author also considers the main differences between Cloud SaaS and IT outsourcing software (ITO) to identify the leading quality of service of Cloud SaaS. The author concludes that the main differences as follows: (1) on-demand self-service, which allows Cloud SaaS service consumers to interact with service capabilities automatically without considering human interaction using programmed user interface, while the ITO requires human interaction between the service provider and the consumer; (2) ubiquitous network access, which allows Cloud SaaS service consumers to obtain services through computer networks such as the Internet; (3) location-independent resource pooling, which describes how Cloud SaaS providers serve thousands of customers at the same time based on monthly or annually contracts between the service provider and the consumer; (4) multi-tenant architecture, which allows Cloud SaaS service providers to be automatically scalable based on customers needs since it depends on cloud infrastructure (IaaS and PaaS); and (5) service contract, since it uses a pay-as-you-go cost model or pay as-consumption payment model. Therefore, the author concludes the main non-functional attributes for managing and selecting the Cloud SaaS provider as follows: availability, performance, reliability, scalability, security, support, interoperability, modifiability, usability and testability (Burkon, 2013).

The main quality attributes/non-functional factors consider with the new business in order to adopt Cloud SaaS Safari et al. (2015) ranked the fac-

tors that should be taken into account to adopt Cloud SaaS into small and medium enterprises (SMEs). The authors mentioned that investigating the most important factors for the adoption of Cloud SaaS is highly important in order to help businesses to understand the advantages and disadvantages of adopting the Cloud SaaS solution. The methodology used is based on Technology, Organization and Environment (TOE) characteristics. These characteristics are as follows: Relative advantages, Compatibility, Complexity, Trialability, Observability, Security and Privacy, IT resources, Sharing and Collaboration, Competitive pressure and Social Influence. Fuzzy AHP is used by the authors in order to rank and find the most critical attributes among all of them (Safari et al., 2015).

Non-functional quality attributes of Cloud SaaS involved for adopting cloud computing for Small and Medium Enterprises (SMEs) in the European Union Marian and Hamburg (2012) investigated the criteria for adopting cloud computing for SMEs in the European Union. The authors found that cost is the determining factor for choosing to deploy services over the Cloud. On the other hand, they were concerned about some issues such as security, privacy and availability (Marian & Hamburg, 2012).

Security concerns for adopting Cloud SaaS Sultan (2011) has mentioned that around 75% of business decision-makers were worried about the security (Sultan, 2011).

The impact factors for adopting Cloud SaaS on the Small and Medium Enterprises (SMEs) Rodrigues et al. (2014) discuss the impact factors of using Cloud SaaS on the Small and Medium Enterprises (SMEs) performance. The authors conducted a study of 32 articles to identify the most important reasons for choosing Cloud SaaS and deploying it to the organization. The authors found that Small and Medium Enterprises are interested in deploying Cloud SaaS solutions since this type

of service offers many advantages to the organization in terms of reducing the cost, flexibility to scale, ubiquitous access and security (Rodrigues, Ruivo, & Oliveira, 2014).

According to the existing approaches summaries in Table 2.2, we can summarize the main shortcomings in this section as follows: (1) No one considers the classifications of consumer's experience attributes (QoE), and the Quality of Service (QoS). Also, the technique that can be used to measure these attributes in terms of subjective analysis and objective analysis. (2) No one helps a service consumer by measuring the non-functional factors that should be involved when selecting a Cloud SaaS provider, specifically QoS and QoE. Hence, our new work proposed a classification technique for the quality of service (QoS) and the quality of Experience (QoE). Furthermore, we also introduce the main factors that should be treated when the selection of the Cloud SaaS provider in terms of QoS and QoE.

2.3.3 Evaluation System of Cloud SaaS Service Provider

Evaluation of Cloud SaaS service provider aims to help a Cloud SaaS service consumer to assess selected Cloud SaaS service for evaluation purposes. As a result, there is an improved selection process with greater satisfaction in Cloud SaaS provider choice. The majority of Cloud SaaS service providers offers a one-month trial version with limited capabilities to help a service consumer to test and evaluate their services, and then a service consumer can make an educated decision on whether or not to continue using the service and enter a contract with the service provider. Therefore, the evaluation system is based on the consumer perspective, which is known as quantifying a service consumer's experiences. A few techniques (Godse & Mulik, 2009; Reixa, Costa, & Aparicio, 2012; Jagli, Purohit, & Chandra, 2016; Boussoualim & Aklouf, 2015a; Naseer & Nazar, 2016) proposed on evaluation methods for the Cloud SaaS service provider.

Table 2.2 : Comparative Analysis of Existing Research Studies on Cloud SaaS Selection Attributes

Author (s)	QoS and QoE Classification	Quality attributes
Repschleager et al. (2012)	No	Maintenance and service cycle, functional coverage, service category, user scaling, probability of date and browser compatibility
Al-Shammari and Al-Yasiri (2014)	No	Not Mention
Khanjani et al. (2014)	No	Adaptability, Extensibility, flexibility, customizability, scalability, changeability, composability, availability, maintainability, reliability, fault tolerance, recoverability, stability, serviceability, robustness, accuracy, efficiency, functionality, response time, suitability, data integrity, privacy,security, accessibility, learnability, commonality, multi-tenancy, operability, agility, assurance, performance, security management and usability
Burkon (2013)	No	Availability, Performance, Reliability, Scalability, Security, Support, Interoperability,Modifiability, Usability and Testability
Safari et al. (2015)	No	Relative advantages, Compatibility, Complexity, Trialability, Observability, Security and Privacy, IT resources, sharing and Collaboration, Competitive pressure and Social Influence.
Marian and Hamburg (2012)	No	Cost, security, privacy and availability
Sultan (2011)	No	Security
Rodrigues et al. (2014)	No	Cost, flexibility to scale, ubiquitous access and security

A pair-wise comparison analytic hierarchy process (AHP) technique for evaluation Cloud SaaS service provider Godse and Mulik (2009) proposed evaluation method for Cloud SaaS service provider using the analytic hierarchy process (AHP) approach in order to select an optimum service. They also consider some parameters for evaluation purposes such as service functionality, architecture, usability, vendor, reputation and cost (Godse & Mulik, 2009). In addition, and based the same evaluation technique, Boussoualim and Aklouf (2015) proposed pairwise comparison for weighting the criteria as well as an evaluation service provider, and prior to decision making regarding a service. The authors were also considering the functional attributes of Cloud SaaS providers and non-functional quality attributes for the evaluation process. The main non-functional quality attributes considered in their work as follows: reputation, cost, usability, structure, configurability and personalization. The authors finally demonstrated their approach by a case study (Boussoualim & Aklouf, 2015a). However, from existing approaches developed the evaluation system by an AHP method is only suitable with a small number of alternatives. In the case where many services need to be evaluated, this does not appear to be a suitable solution.

Aggregation function for evaluation Cloud SaaS service provider Reixa et al. (2012) proposed an evaluation approach based on aggregation function based on a service consumer perspective with also considering the opinion of an expert person. The highest score indicates that the service has many of the appropriate features. The authors provide factors that a service consumer should consider when evaluating a Cloud SaaS provider. The factors generated by extensive studies using the expert group report "European commission" that focus and analyze the main characteristics of cloud services. These factors as follows: suitability, economic value, control, usability, reliability and security. The author also provides a case study for selecting only three office cloud services: Zoho Docs, google docs and Microsoft

office365 (Reixa et al., 2012).

Tree-data mining model for evaluation Cloud SaaS Service Provider Jagli et al. (2016) proposed an evaluation mechanism using a decision tree-data mining model. The authors investigate the main challenges for representing the qualities that a Cloud SaaS consumer are considering when evaluating a Cloud SaaS service provider. They concluded with the following criteria: pay-per-use, availability, reusability, scalability, data managed by provider and service customizability (Jagli et al., 2016). However, the main shortcoming of this work is the lack of demonstration of its use in a case study or experiment.

Analytic Network Process (ANP) for evaluating the service provider Naseer and Nazar (2016) proposed an evaluation of Cloud SaaS provider based on the analytic network process (ANP) to make a decision and then select the best service provider. The authors also investigated the main criteria for evaluating Cloud SaaS service providers including usability, security, reliability, tangibility, responsiveness and empathy (Naseer & Nazar, 2016).

According to the comparative analysis provides in the table 2.3, we can summarize the main shortcomings of existing methods as follows: (1) none of the existing methods considers an evaluation process using the consumer's experience perspective. (2) the service consumer opinions may be described as vague for example, usability is evaluated as "excellent". Therefore, it may be worthwhile to design a framework using linguistic-terms evaluation system. (3) evaluating the consumer perspective as a percentage level seems to be a suitable solution to describe consumer experiences. Hence, the rule-based system can help to evaluate the criteria to match with a service consumer perspective.

To address the above shortcomings, chapter (6) two approaches are proposed for evaluating the Cloud SaaS provider using a linguistic-based evaluation criterion and

Table 2.3 : Existing Research Studies on Evaluation System of Cloud SaaS Provider

Author (s)	Service Consumer's Perspective		Method	Evaluation Attributes
	Linguistic-based Evaluation	Percentage-based Evaluation		
Godse and Mulik (2009)	No	No	AHP	Functionality, Architecture, Usability, Vendor Reputation and Cost
Boussoualim and Aklouf (2015)	No	No	AHP	Reputation, Cost, Usability, Structure, Configurability and Personalization
Reixa et al. (2012)	No	No	aggregation function	Suitability, Economic Value, Control, Usability, Reliability and Security
Jagli et al. (2016)	No	No	Tree-data mining	Pay-Per-Use, Availability, Reusability, Scalability, Data Managed and Service Customizability
Naseer and Nazar (2016)	No	No	ANP	Usability, Security, Reliability, Tangibility, Responsiveness and Empathy

a fuzzy rule-based evaluation mechanism. These two proposed approaches help a service consumer to analyze and evaluate a service provider taking into account the advantages of testing the service for a month. Also, we introduce new non-functional experience attributes for evaluation of the Cloud SaaS provider.

2.3.4 Shortcomings of existing approaches

According to a comparative study of existing literature review on simple Cloud SaaS selection, we can summarize the main shortcomings into the following points:

1. The majority of previous research focuses on weighting non-functional quality attributes as service consumer's preferences. There is little or no attention given to quantifying the consumer requirements for particular criteria.

2. While a few studies consider weighting non-functional quality attributes using linguistic values, there has been little work done to identify the consumer's preferences among selected criteria.

3. There are no existing approaches to develop a service selection mechanism based on quantifying the value of selected criteria describing the consumer's requirements.

4. According to the quality of service (QoS) and Quality of Experience (QoE), none of the existing approaches develops Cloud SaaS service selection based on these two types describing the consumer's perspective.

5. A significant majority of existing work demonstrates their approaches use a simulated dataset or as a case study.

6. None of the existing approaches evaluates the service provider using the linguistic values that describe a consumer experience perspective. In addition

to this, none of the existing approaches develops a rule-based fuzzy system to evaluate the Cloud SaaS service providers.

To address the above gaps in the research literature, this research aims to develop a service broker called (Find SaaS) (Details in chapter 5 and 6) to help a Cloud SaaS service consumer to find the best service matching with the non-functional consumer's perspective in terms of (QoS and QoE).

2.4 Overview of Existing Research on Non-Functional Cloud SaaS Service Composition

Service composition is the other type of service that a consumer prefers. Compared with simple Cloud SaaS selection, Cloud SaaS service composition refers to the selection of multiple sequences of Cloud SaaS service providers in order to fulfil a service consumer's functionality (Jian, Zhu, & Xia, 2016). In order to provide clear explanation, Wu & Khoury (2012) did a case study to develop a service composition for a healthcare supply chain management (SCM) (Wu & Khoury, 2012). The healthcare SCM runs four main services: ordering, manufacturing, transporting and distribution. Those services should be composed of different cloud service providers. It means that each service, such as ordering (called an abstract service), has a variety of services (called candidate concrete services) that satisfies each service. The service composition should select an optimum candidate service from each abstract service, and then provide it to SCM which a service consumer as a composite service. The authors consider that the service composition should be minimized for price and time and maximized for reliability and availability.

Therefore, service composition is a decision problem that considers multiple aggregating service providers in terms of functional and non-functional quality attributes. According to a literature review, service composition is an NP-Hard prob-

lem (Alrifai & Risse, 2009) (Wu & Khoury, 2012).Service composition consists of three main steps. Firstly, identifying the Cloud SaaS abstract services. This step is known as a control service structure or data flow. The WS-BPEL is frequently used for ideal abstract composition. Secondly, the preprocessing step. In this step, it will generate Cloud SaaS service providers that match with abstract services functionality by using (syntactically or semantically) (Jordan et al., 2007). As a result of the previous step, list of services, namely candidate concrete services having similar of each abstract service with different non-functional quality attributes. Finally, build a service composition by selecting an optimum service provider in each abstract service matches with non-functional service consumer's preferences (Jian et al., 2016).

2.4.1 Non-Functional Approaches of Composite Cloud SaaS Selection

Tree-based SaaS Composition Wu and Khoury in 2012 proposed a Cloud SaaS service composition framework that uses a tree-based search algorithm in order to optimize the QoS. The proposed approach consists of three steps. Firstly, the tree creation model creates all possible web Cloud SaaS-based services to find composite services. Secondly, the tree filtration algorithm aims to reduce the tree paths by reducing the response time, reducing complexity and improving the performance. The algorithm consists of two main steps: pruning and combining the tree based on the quality attributes. Thirdly, each path is evaluated and ranked in order to select the optimal composition of services. The evaluation algorithm is based on the multi-objective evolutionary function called DFBMOEA (distance function-based multi-objective evolutionary algorithm). The authors used a case study of healthcare supply chain management (SCM) to demonstrate their approach. In this case study, the SCM needs to apply four services for composition: order, manufacture, transport and distribution. The composition function is used to minimize total cost and time

as well as maximize the reliability and availability (Wu & Khoury, 2012).

Multi-tenant SaaS Composition using Genetic Algorithm (GA) Yu Toh and Tang (2018) proposed a framework for the selection of Cloud SaaS service composition based on Genetic Algorithm (GA). The authors considered the properties of cloud computing characteristics such as multi-tenant software model. In addition, the authors performed optimization for selection of the Cloud SaaS to minimize the total running cost and maintain the overall performance. The heuristic algorithm is used as a benchmark to compare the performance of the algorithm proposed (Toh & Tang, 2018).

Evolutionary Optimization for multi-tenant middleware Cloud SaaS Composition Kunar et al. (2018) proposed multi-tenant middleware broker to select the Cloud SaaS service composition. They used an evolutionary optimization for encoding Cloud SaaS in order to obtain the best Cloud SaaS composition. The authors also considered cost optimization for composite selection. The consumers' constraints are considered in the selection of composite service. In their approach, the experiment was conducted with WS-Dream dataset, and the authors tested their approaches with different evolutionary approaches such as MOEA and NSGA-II (Kumar, Bahsoon, Chen, Li, & Buyya, 2018).

Clustering approach for multi-tenant SaaS selection Wang et al. (2015) proposed a service recommendation approach for QoS-aware service selection for multi-tenant Cloud SaaS named Clustering-Global. The clustering approach using data mining was applied in order to cluster tenants (i.e. candidate services) by matching the similarity of multi-dimensional QoS requirements. They used Mixed Integer Programming (MIX) to solve the optimization problem in order to maximize the service composition utility and minimize the overall cost of resource usage. The authors tested their approach by QWS dataset and compared the result with

multiple existing approaches by measuring the ratio of computation time (Y. Wang, He, & Yang, 2015).

MSSOptimiser for QoS-driven service Selection for Multi-tenant Cloud SaaS He et al. (2012) proposed MSSOptimiser for selection a multi-tenant Cloud SaaS service provider taking into account the functional process specifications of Cloud SaaS. The optimizer proposed generates the sequence of services as the output for the execution plans in order to fulfilment the end-user's requirements. Moreover, the optimizer is based on maximizing the utility of composing services as well as minimize the total cost. The authors tested and evaluated their approach using the QWS dataset by comparing with six existing optimizer approaches (He, Han, Yang, Grundy, & Jin, 2012).

Genetic Algorithm for Cloud SaaS optimization towards service composition Bentaleb and Ettalbi (2016) proposed Cloud SaaS service composition using the genetic algorithm for optimization. The proposed approach based on Skylines operator selection problem (Bentaleb & Ettalbi, 2016).

Multi-tenant Service selection Schroeter et al. (2012) proposed the requirements for multi-tenant applications (MTA) for Cloud SaaS in order to provide the reference architecture for Cloud SaaS stakeholders. They introduced the multi-quality auto-tuning (MQuAT) as an example framework to support the requirements for MTAs (Schroeter, Cech, Götz, Wilke, & Aßmann, 2012).

According to extensive studies in literature review of selection a composite service of Cloud SaaS, (see table 2.4), we can summarise the main shortcomings as follows: (1) None of the existing works investigate the reputation-level based on previous consumer reviewers when selecting a composite service. (2) The majority of existing works only developed their approaches based on reducing the search space by developing the optimization processor such as GA or MIX when selecting

Table 2.4 : Existing Research Studies on Cloud SaaS Service Composition

Author (s)	Reputation service level	Decision maker for selection composite service	Method for optimization	Optimization units
Wu and Khoury (2012)	No	No	Tree-based optimization	Minimize the cost, and maximize reliability and availability
Yu Toh and Tang (2018)	No	No	Genetic Algorithm (GA)	Minimize the total running cost and maintain the overall performance
Kumar et al. (2018)	No	No	Genetic Algorithm (GA)	Minimize the cost
Wang et al. (2015)	No	No	Mixed Integer Programming (MIX)	Minimize the cost, and maximize the utility of composite service
He et al. (2012)	No	No	None	Minimize the cost, and maximize the utility of composite service
Bentaleb and Ettalbi (2016)	No	No	Genetic Algorithm (GA)	None
Schroeter et al.(2012)	No	No	None	None

the composite service.

2.4.2 Shortcomings of existing approaches

To summarize the main shortcomings of existing literature review of composite Cloud SaaS service selection, and after comprehensive studies on the service composition sector, we can summarize these points as follows:

1. The selection of a composite service is a decision-making problem. To assist the decision-maker in reducing the search space, and after using the multi-objective optimization approaches such as GA, the decision-maker approaches such as TOPSIS can be used to select optimal composite service.

2. None of the existing approaches is taking into account the reputation of previous consumer's reviewers when selecting the composite service.

3. The existing works focus only to reduce the search space without taking into account the consumer's preferences. In other word, there is no approach for weighting quality attributes and considering the consumer's constraints for composite cost as well as composite reputation based when selecting the composite Cloud SaaS providers.

To address the above shortcomings, this research aims to address all these shortcomings by adding a feature of existing previous service broker to select the composite service of Cloud SaaS called Find SaaS GA TOPSIS. We have used Genetic Algorithm (GA) for multi-objective optimization to maximize the reputation-based service reviewers (RbSR) and minimize the total cost of composite service. We also develop a decision-maker using the TOPSIS technique in order to select the optimum composite service based on consumer's preferences of multiple criteria such as availability, response time and throughput (Details in chapter 7).

2.5 Datasets

There are a few datasets available to test and evaluate simple and composite service selection proposed approach. The following datasets can be used for evaluation and as a benchmark:

1. **QWS**: Is real web services dataset introduced by Al-Masri in (2007) (Al-Masri & Mahmoud, 2007). The great thing to use this dataset is to measure the performance for large services number, and multiple criteria can be used as well for testing and evaluating. The dataset contains around 2500 services, and each service has nine criteria describes the non-functional attributes.

2. **WS-Dream**: Is a Distributed Reliability Assessment Mechanism for Web services (Delac, Silic, & Srbljic, 2015). It contains three sets of data: (1) QoS (Quality-of-Service) datasets; (2) log datasets; and (3) review datasets. However, this kind of data is suitable only for the testing matching algorithm as a recommender system.

3. Liu et al. (2016) introduced services dataset contains around 50 services with multiple criteria for evaluating the approach of multi-criteria decision making (Liu et al., 2016).

4. **Blue SaaS dataset**: Is proposed by Alkalbani et al. in 2015 (Alkalbani, Ghamry, Hussain, & Hussain, 2015). This dataset contains more than 5000 software services. However, it includes only a few criteria describing the non-functional attributes. Also, many services have missing values in terms of the service cost and service rating.

Therefore, none of the existing datasets contains real Cloud SaaS providers, with multiple criteria describing client-side and server-side.

2.6 Conclusion

In this chapter, a brief review of the existing literature on Cloud SaaS simple and composite service selection was presented. The literature review shows that few works consider consumer's preferences when selecting a simple and composite of Cloud SaaS provider. Three main directions have been classified in this chapter.

Firstly, the first part of this chapter, an overview of existing simple Cloud SaaS selection techniques has been presented.In the area of simple Cloud SaaS selection we identified the following major gaps that we intend to address in this research: (1)The majority of the existing work considers only weighting quality attributes as consumer's preferences when selecting a service. However, these works do not consider the consumer's preferences as quantifying the actual value describing the quality attributes. For example, the Cloud SaaS provider required as 20$ service price. (2) The existing literature does not propose any mechanism to take into account the feedback from a Cloud SaaS service provider based on the trial phase during the process of selecting a simple Cloud SaaS service.

In the second part of this chapter, the existing works on Cloud SaaS service composition have been reviewed. The existing literature has focused on developing an optimization process to minimize the search space and then select an optimal composite Cloud SaaS providers. However, none of the existing literature has designed the decision-maker after the optimization process when it takes place. Also, none of the existing works develops the reputation-based service reviewers using the service rating and service reviewers for Cloud SaaS provider.

The datasets for evaluation process have finally reviewed. The literature review shows that researchers have used web services for testing the Cloud SaaS service selection. Also, some researchers have tested their approaches by simulated datasets. Therefore, in our research, the new datasets have created contains multiple Cloud

SaaS providers using the Capptera web site. Also, the set of huge services has used for the evaluation of our proposed approaches.

The next chapter discusses the problem definitions that have been addressed in this thesis.

Chapter 3

Problem Definition

The first chapter covered the importance of Cloud SaaS simple and composite selection. In the previous chapter, a review of the existing literature was presented. It was noted that researches proposed various methods to address the simple and composite Cloud SaaS selection. However, the major drawback of these approaches is that they fail to offer a comprehensive intelligent methodology as a service broker to select an optimum simple and composite service based on consumer's preferences. In addition to this, we have identified various shortcomings within the existing literature that needs to be addressed in order to propose a complete methodology for our new service broker of Cloud SaaS simple and composite selection. This chapter provides an overview of the research problem to be addressed in this thesis. In addition, the research issues, aims, objectives and questions are defined in this chapter. Finally, the main research methodology that has been used to achieve our objectives is discussed finally.

3.1 Introduction

The main objective of this chapter is to give an overview of the research problem addressed by this thesis. To achieve that, this chapter is organized into ten sections, as follows: Section 3.2 provides the research keywords of this thesis with their definitions. Section 3.3 discusses the main differences between Cloud SaaS and other cloud services (i.e. PaaS and IaaS). Overview of the research problem of this thesis is presented in Section 3.4. Section 3.5 discusses the research issues. Section 3.6 will introduce the research questions. Section 3.7 and 3.8 provide the research aims and

objectives respectively. The overview of the main research methods is discussed in Section 3.9. introduces the research methodology that has been followed to achieve the objective of this research. Finally, Section 3.11 concludes this chapter.

3.2 Key Concepts

This section displays a collection of research keywords and their formal definitions that have been used in this thesis.

3.2.1 Cloud Computing (CC)

Cloud Computing is defined as a new paradigm that provides the hardware resources and software applications over the global network system (e.g. the Internet) (Armbrust et al., 2010).

3.2.2 Software as a Service (SaaS)

Software as a Service (SaaS) is defined as a software application that runs and operates over cloud computing infrastructure. It runs on the vendor's (i.e. Service Provider) datacenter without installing it on the end-user machine (Tyrväinen & Selin, 2011).

3.2.3 Service Oriented Architecture (SOA)

SOA is defined as a way to build integration applications including an interconnected series of services, each accessible through standard interfaces and messaging protocols. It is the standard architecture content of three main entities: service provider, service requestor and service discovery agency (Papazoglou, 2003).

3.2.4 Service Level Agreement (SLA)

SLA is defined as an agreement between the customer and the service provider (Badidi, 2013). SLA describes agreed service functionality and qualities provided in

the contract between them, such as cost and availability.

3.2.5 Service Broker

Service broker is defined as an agent service in between the service consumer and service provider. A service broker addresses the service requested by a service consumer in order to match service description requirements and non-functional attributes that are registered in its service registry repository (Cai & Tong, 2009).

3.2.6 Service Consumer

Service consumer is defined as any entity who uses a particular service. The service consumer could be a human user, business user, or another service. The service consumer is one kind of entity in the SOA model (Papazoglou, 2003).

3.2.7 Service Provider

The service provider is defined as the service that provides particular services to the service consumer or the service registry (i.e. Service broker in this thesis). The service provider makes the service accessible and can identify it through the service description and characterize it by non-functional attributes such as service price (Papazoglou, 2003).

3.2.8 Simple Service

Simple service selection may be defined as a process that can be taken by a service consumer in order to select a single service to fulfil its requirements (Papazoglou, 2003). Recently, multiple approaches are proposed to help a service consumer to find an optimal service. These approaches are based on studying the factors that a service consumer considers when selecting a service provider, or by a proposed smart agent or service broker in between a service provider and service consumer.

3.2.9 Composite Service

Service composition is defined as an aggregation of multiple services in order to fulfil a user's requirements; the result of these aggregation will be a single composite service (Papazoglou, 2003). Selecting a composite service is an NP-Hard problem, which means there is no optimum solution for selection of a composite service (Alrifai & Risse, 2009).

3.2.10 Service Selection

Service selection is defined as selecting an optimum service provider that provides a specific service for a service consumer from among multiple services based on consumer's preferences. Service selection is based on matching the requirements of service functional and non-functional consumer's preferences (Sun, Dong, Hussain, Hussain, & Chang, 2014).

3.2.11 Ranking System

Ranking system (RS) is defined as a way to display all the items (could be services, movies or things) beginning with those most related to client's favourites and ending with those least aligned with the client's preferences (Joshi, 2020). The traditional way to rank services usually works based on the most critical metrics selected by the client such as service price, service rating and service reviews.

3.2.12 Evaluation System

Evaluation system is defined as a way to assess services in order to help a consumer select an optimum service (Boussoualim & Aklouf, 2015a). The evaluation system is usually based on a decision matrix which involves multiple alternatives and various parameters.

3.2.13 Consumer Preferences

Consumer preferences is defined as the way to understand the consumer's requirements when requesting a particular service. The consumer's preferences are classified as functional and non-functional preferences (Sun, Dong, Hussain, Hussain, Ma, & Zhang, 2014). The consumer's preferences in terms of non-functional requirements could be subjective requirements such as need low service price, or objective such as need a service starting with 20 dollars. In addition, in terms of functional requirements, the consumer requests the functional service type by declaring the service description requirements. Then, any proposed approach can be used to match their service description requirements to the pool of services, which could be a approach or Semantics approach.

3.2.14 Small and Medium Enterprises (SME)

SME may be defined as business sector focusing on improving their products to gain more revenue. Therefore, one of the most important strategies for SME is reducing the cost of building its facilities, such as IT services. In the literature review, Cloud SaaS offers an attractive service for SME to reduce expenditure and increase the flexibility of the IT environment (Haselmann & Vossen, 2011).

3.2.15 Functional Service Selection

Functional service selection may be defined as the way to discover and invoke the service based on the service description published by a service provider to the service registry repository (Sun, Dong, Hussain, Hussain, & Chang, 2014). Functional service selection could use a syntactic or semantic approach.

3.2.16 Non-Functional Service Selection

Non-functional service selection is defined as the selection process of the service-based match with the non-functional requirements such as a service price or service availability (Sun, Dong, Hussain, Hussain, & Chang, 2014). The researchers developed a multi-criteria decision-making approach or multi-objective decision-making approach to solving this kind of selection problem.

3.2.17 Quality of Service (QoS)

The QoS is defined as the description or measurement attributes of the overall performance of a particular service. Collecting the QoS could be carried out by client-side or server-side (Zheng, Zhang, et al., 2010). Price, availability, rating are some parameters of QoS.

3.2.18 Quality of Experience (QoE)

The QoE is defined as a measure of the delight or description of a consumer's experiences with a service (Al-Shammari & Al-Yasiri, 2014). QoE and QoS are the same measurement concept, except that the QoS is not based on the consumer's experiences.

3.2.19 Multi-Criteria Decision Making (MCDM)

MCDM is defined as a decision-making problem that involves multiple alternatives or options described by multiple criteria or attributes (Godse & Mulik, 2009). MCDM is known as Multi-attribute decision making (MADM) in some references as a word synonym. The authors proposed many approaches to solve this kind of problems such as SAW, AHP, TOPSIS and ELECTRE.

3.2.20 Multi-Objective Optimization (MOO)

The MOO is defined as the problem having multiple conflicting objectives when deciding. Minimizing cost while maximizing comfort while purchasing a new car and minimizing fuel consumption and emission of pollutants whilst maximizing the performance of a vehicle are examples of MOO problems involving more than two objectives (Zhang, Cao, Liu, & Huang, 2016).

3.3 The differences between the cloud services and Cloud SaaS

Cloud Software as a Service (SaaS) is a cloud service operating and running under the cloud infrastructure (i.e. PaaS and IaaS). However, compared with other cloud services such as Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) in terms of popularity, Cloud SaaS is usually the first choice to deploy IT services especially for small and medium enterprise (SME). Cloud SaaS provides many advantages to service consumer to reduce the cost of IT services and also offers benefits in terms of scalability and availability. Recently, the authors proposed many approaches to select a cloud service without considering the service type. Therefore, we introduce some points in order to compare the main differences between Cloud SaaS and other cloud services such as PaaS and IaaS.

1. Cloud infrastructure as a service (IaaS) provides hardware resources as a service, and also in terms of user expertise, the end- user who used this type of service has a capability to program and is usually classified as the software developer. The end-user of Cloud software service (SaaS) usually has less skill to program and is usually classified as a regular person or enterprise. In addition, cloud PaaS often provides services to the software developer in order to help them to enhance the user experience when using cloud infrastructures

(i.e. IaaS).

2. Regarding the payment method, all cloud services provide multiple options to pay for services used. However, with Cloud IaaS and Cloud PaaS, the payment method is usually based on a pay-as-you-go or consumption-based pricing model (Armbrust et al., 2010). However, Cloud SaaS provides a monthly/annual subscribe per user or an enterprise version depending on the number of users who would like to use the software. Moreover, some service providers charge users based on additional features such as the size of data and supporting features.

3. Cloud SaaS is based on the consumers experience for using and adopting a particular service. Therefore, many service providers offer a one-month trial version to help Cloud SaaS service consumers' test and evaluate the service before committing to a contract with the service provider.

Therefore, the authors in cloud service selection should consider these differences between all cloud services when proposing a new approach in order to understand consumer's preferences properly.

3.4 Problem Overview and Problem Definition

As discussed in the previous chapter, selecting an optimal cloud service is a critical topic since the number of service providers is growing rapidly (ur Rehman, Hussain, & Hussain, 2014). Cloud computing can be classified into three types of services which are: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). In addition, Cloud Computing has three types of deployment models (Mell et al., 2011): Private Cloud, Public Cloud and Hybrid Cloud. The number of cloud services is growing, especially in the Cloud SaaS service

level. This means that it is important to provide a service selection model based on the consumer's perspective.

Cloud services come in two basic forms (Papazoglou, 2003): simple service and composite service. This classification is based on the service-oriented architecture (SOA) model. The Cloud SaaS uses the SOA model to deploy and publish their services to the public (Service consumer). Therefore, developing an approach to select a simple or composite service is important in order to ensure comprehensive selection solution.

Previous research has made significant advances and impacts on the issue of simple selection of Cloud SaaS. One example of simple Cloud SaaS selection is in (Badidi, 2013). The author has proposed a framework to help a Cloud SaaS consumer to select a simple Cloud SaaS provider using the aggregated utility function. However, the author ignores the consumer's preferences when selecting a cloud service. It means that identifying the value of specific quality attributes is missing in the author's work (e.g. service price 20$). As we discussed previously, the Cloud SaaS simple selection goes through three major steps: ranking, evaluation and selection. We will focus on each of these three steps in turn.

When ranking Cloud SaaS providers for a Cloud SaaS consumer, very few approaches tried to rank many of the service providers available for selection. Most previous approaches are suitable only for choosing between three to five services. Moreover, they used either a case study or simulated dataset to demonstrate their methods.

As stated in the literature review, the selection of a Cloud SaaS provider involves considering the consumers preferences to identify a preferred service provider. Previous research proposes service selection methods which only consider functional attributes in service selection. When choosing a Cloud SaaS service, consumers will

also consider non-functional features of the selected service such as the monthly price. Furthermore and importantly, the extant literature has failed to consider the weights or rank assigned by the Cloud SaaS consumer during the selection process.

Evaluation of Cloud SaaS service providers has generally focused on the comparison of two providers. There is a shortage of research using linguistically based evaluation systems to measure non-functional characteristics and no data regarding QoS and QoE.In general, previous research has not considered the evaluation of non-functional qualities using linguistic data. Most Cloud SaaS service providers offer a one-month free trial; therefore, it is likely that consumers take advantage of this option to evaluate multiple service providers prior to making a final decision. By doing so, it is likely they make their decision based on our proposed approach. However, the existing literature has not developed any approaches to take into account the Cloud SaaS consumer's feedback on the trial phase into account during the service selection. Incorporation of the SaaS consumer's feedback based on actual experience with the SaaS provider (at the trial phase) is critical is ensuring personalized selection based on real experience with the Cloud SaaS provider.

Previous researchers evaluating the systems of Cloud SaaS service provider ignore evaluation service providers based on a fuzzy rule system. It means that the service consumers can assess individual attributes as a percentage level. Then the evaluation system then selects the best service based on the fuzzy rule system.

As discussed in the previous chapter regarding the service composition of Cloud SaaS, it classified under the NP-Hard problem. This means that no optimum solution can be found for this problem. However, the researchers have attempted to address this problem by using multi-objective optimization methods. This method is only useful in reducing the search space. No decision-making design has been developed to select the best composite service. We propose to develop a selection

method by combining two methods multi-objective optimization and a decision-making approach such as TOPSIS to address the issue of composite Cloud SaaS selection.

The reputation of a service provider is a critical factor for the consumers choosing the best composite Cloud SaaS. Previous researchers have ignored this factor in service composition. They have focused their attention either on cost minimization or maximized positive attributes such as availability while disregarding cost. No previous approach has considered selection based on maximum reputation coupled with minimization of cost. The reputation score used in this study has been developed based on service rating and reviews of the service by consumers.

The focus of this research is to address the above mentioned shortcomings in previous methods used to select Cloud SaaS providers and to develop and intelligent method for simple and composite service selection based on the consumer's preferences.

3.5 Research Issues

Based on the literature review raises some issues and limitations of previous researches that need to be addressed in order to build an intelligent service broker to select a simple and composite service.

3.5.1 Research Issue 1: Lack of an intelligent approach for ranking and selecting simple Cloud SaaS service based on consumer's preferences

In this thesis, we address the consumer's preferences when ranking Cloud SaaS simple service providers. Furthermore, in addition, there are shortcomings in the existing methods that only they demonstrate their approaches by the small number of service providers. Therefore, we address that by developing a new approach that

acknowledges the consumer's preferences.

Regarding the consumer's preferences in our system, we have classified the consumer's preferences into two types: (1) weighting the non-functional quality attributes. This preference type is based on all the previous researches approaches developed recently. However, we are taking one more step and developing our weighting preferences using linguistic-based weighting preferences. (2) Quantifying the real value describing the requirements of non-functional quality attributes. We term that preferences as the consumers requirements.

3.5.2 Research Issue 2: Lack of an intelligent approaches for Cloud SaaS simple selection based on the feedback or outcome from the trial evaluation phase

The majority of Cloud SaaS service providers offers a one-month trial version to test and evaluate the Cloud SaaS before making a contract and payment schedule with the selected service provider. To take into account the SaaS consumer's experience or feedback from the trial phase, during the selection a Cloud SaaS simple service, we proposed an evaluation system to assess each service provider using linguistic terms for evaluating individual quality attributes. Furthermore, in order to take into account the scenario, where in the service consumer evaluates the service providers by the level of percentage, we developed a fuzzy rule-based system for decision making.

3.5.3 Research Issue 3: Lack of an intelligent decision support system for selecting the Cloud SaaS composition based on combining two methods GA and TOPSIS

In this research issue, we address the differences between the service rating and service reviewers. It means that some service providers have many service reviewers,

but with middle service rating such as 3 out of 5. On the other hand, some services have high service rating such as 5 out of 5, but with few service reviewers. Therefore, it worth noting that it is essential to find a new factor that is based on these two inputs which is classified under the previous user experiences. We have termed this factor as reputation-based service reviewers (RbSR). The fuzzy rule-based system is proposed to assist the user in selecting the optimal SaaS combination.

Furthermore, during the decision making process of selecting the optimal Cloud SaaS combination, the decision-maker is required to select an optimum composite of Cloud SaaS service provider. The previous researchers found that service composition is an NP-Hard problem. Therefore, they designed a method to address the service composition by applying objective optimization methods such as GA and PSO. However, the authors are only attempted to solve the aggregation problem or how the optimum combinations will be produced. Hence, they only tried to reduce the search space for service composition. Therefore, in this research, we combine two methods in order to help a service consumer to select the optimum composite of Cloud SaaS service providers. These two methods are a genetic algorithm (GA) which is used to address the optimization problem. Then, apply TOPSIS MCDM in order to select an optimum composite service among all composite services that are generated by the previous method, which is the GA.

3.5.4 Research Issue 4: Validate the proposed approaches by experiment implementations

Finally, we need to validate the results of the research proposed for research issues 1 and 3. Also, we demonstrate the other research issues by case studies for testing and validating purpose. We compared the results with other existing previous approaches as well as traditional service selection and compositions approaches. $nDCG$ and mAP are two metrics that have used for validation purpose of our

researches approach. In addition, we present the system prototype in Chapter 8 in order to make clear validation and evaluation purpose.

3.6 Research Questions

This research aims to solve the following main research question:

What is the optimal way to select simple and composite Cloud SaaS provider taking into account consumer's preferences?

In order to solve our main question, we break it down to sub-questions as follows:

Sub-question 1) How we develop intelligent methods to automatically rank Cloud SaaS service providers based on multiple consumer's requirements?

Sub-question 2) How can intelligent methods be used to systematically and automatically select a Cloud SaaS service provider based its trial evaluation?

Sub-question 3) How can intelligent methods be used to select composite Cloud SaaS automatically?

Sub-question 4) How can we evaluate and benchmark the solutions developed for research question 1-3?

3.7 Research Aims

This research includes the following research aims:

Aim 1) To improve the selection mechanism of simple Cloud SaaS service provider taking into account the consumer's preferences.

Aim 2) To improve the evaluation method in order to help service consumer to assess the Cloud SaaS service provider before making a contract.

Aim 3) To improve the selection mechanism for Cloud SaaS service composition considering the reputation of composite services as well as making a decision to select an optimum composite service.

Aim 4) To extend the application of MCDM, Fuzzy Logic and Genetic Algorithm (GA) to the selection approach for simple and composite Cloud SaaS.

3.8 Research Objectives

This research includes the following research objectives:

Objective 1) To develop a dynamic ranking system of Cloud SaaS-based on consumer's preferences by using different approaches such as MCDM and Fuzzy logic.

Objective 2) To develop an evaluation system of Cloud SaaS service provider taking into account the consumer's experiences of individual service provider using the linguistic-based technique as well as rule-based technique.

Objective 3) To develop a new methodology for using the Genetic Algorithm (GA) and TOPSIS MCDM in the selection of Cloud SaaS service composition and using the fuzzy-rule system to generate the reputation factor of Cloud SaaS service provider.

Objective 4) To validate and evaluate the proposed approaches by using the extensive comparisons between the traditional or existing approaches, in addition to this, using the terminology of proof of concept to develop the system prototype to demonstrate how our system work with presenting different cases.

3.9 Research methods to Problem Solving

This research applies multiple methods to achieve each research objectives. The artificial intelligence (AI) sector plus multi-criteria decision making (MCDM) approaches have been used widely in this research. The AI research methods are used

to determine and quantify the output of multiple inputs such as using fuzzy logic for identifying the weighting quality attributes, also for generating the reputation factor. In addition, the genetic algorithm (GA) is also a kind of AI method used in this research to address the optimization problem in Cloud SaaS service composition. On the other hand, the TOPSIS method is used as a MCDM method to select the optimum options for a Cloud SaaS service consumer.

3.9.1 Multi-criteria Decision Making (MCDM)

The multi-criteria decision making is the process to help people to choose the best alternative from several potential candidates based on their preferences. The authors in this field contribute different methods for this reason to find the best service or best option. The decision-maker usually deals with the problem involved in a different way. For example, the AHP is standard for the Analytic Hierarchy Process. In this method, the decision-maker makes pairwise comparisons between all alternatives to make a decision, then select the best alternative. The other example is used recently is TOPSIS. The TOPSIS standard for Technique for Order of Preference by Similarity to Ideal Solution (Chamodrakas, Leftheriotis, & Martakos, 2011). In this method, the decision-maker performs the decision matrix, then makes a decision based on the shortest distance to the best-ideal vector, which is generated from its matrix. Simple Additive Weighting (SAW) and ELECTRE are other examples of MCDM.

3.9.2 Fuzzy Logic

Fuzzy logic is the mathematical principle which deals with the degree of membership rather than on the crisp number of classical binary logic 0 or 1. It was introduced by Lotfi Zadeh in 1965. The fuzzy logic goes through several steps in order to find the final result. For example, the fuzzy system needs to identify first the fuzzy input variables and output. Then, it needs to define the fuzzy membership

for each variable. The last step is to define the rules that will apply for determining the fuzzy output. The fuzzy reasoner is usually described as the system that used the fuzzy logic concept. The fuzzy logic has been applied in a variety of applications such as management, finance and technology.

3.9.3 Genetic Algorithm

John Holland introduces the genetic algorithm (GA) in 1970 (Holland et al., 1992). Holland aims to make a computer works as nature does. For example, the computer can run an algorithm to manipulate a string of binary digits 0 or 1 (called chromosome), to simulate the natural evaluation. This algorithm generates from each chromosome (parent) to move forward to next-generation called child chromosome by using the crossover and mutation process. The result of this algorithm is to find an optimum generation. The idea of GA led to other equations to simulate nature behaviours such as particle swarm optimization (PSO) and ant colony optimization (ACO). This kind of algorithms classified under evolutionary algorithms. They are used mainly for addressing the optimization problem and have been applied in a variety of study areas such as engineering, medical, finance and computer science.

3.10 General Research Methodology

This thesis follows the design science research methodology (DSRM) to achieve our research objectives. The DSRM process consists of six main activities in a nominal sequence (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007). Figure (3.1) shows the main activities that are followed by us to achieve thesis objectives.

Phase 1) *Problem identification and motivation.*

In this phase, we study the relevant literature review on topics related to

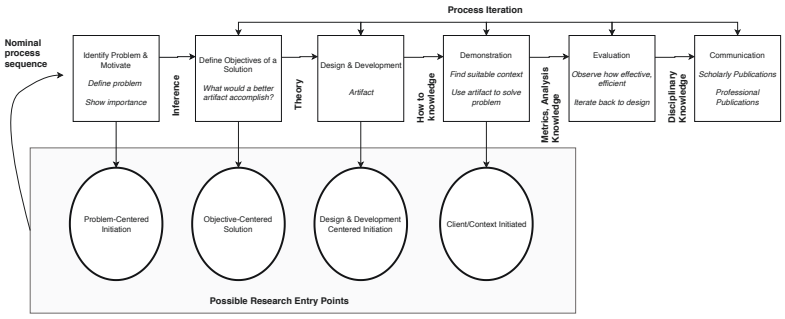


Figure 3.1 : Design Science Research Methodology Iterations Process (Peffers et al., 2007)

simple and composite Cloud SaaS selection. Based on the extensive review of the existing literature, the problem which needs be addressed was identified.

Phase 2) *Define the objectives for a solution.*

In the next step, the objectives for this research are recognized by developing an intelligent service broker to address the problem of simple and composite service selection based on consumer's preferences.

Phase 3) *Design and Development.*

After that, layout algorithms (e.g. Find SaaS GA TOPSIS) using different intelligent methods such as fuzzy logic, genetic algorithm, and by using the decision maker such as TOPSIS approaches were conceptualized, designed and developed.

Phase 4) *Demonstration.*

Apply and demonstrate the algorithms proposed on some dataset such as QWS and CRM Cloud SaaS providers in order to obtain the result is the next phase required.

Phase 5) *Evaluation.*

Next, evaluate the result by comparing the result with the existing approaches

and by using the performance metrics such as $nDCG$ and mAP.

Phase 6) *Communication.*

Share the result obtained by publishing the works with the researchers' community interested in the Cloud SaaS selection is the final phase in DSRM (e.g. AINA and EIDWT).

3.11 Conclusion

This chapter presented formal definitions of the main research keywords that we intend to address in this thesis. Based on this, the primary research question was identified. We also introduced the research problem overview, as decomposed into four research issues which need to be addressed them in our research project. The research questions, aims and objectives are given in this chapter, which were taken and analyzed from the shortcomings found in Cloud SaaS simple and composite service selection. Finally, we produced the general methods that will be used to address the research problems in the field of Cloud SaaS simple and composite service selection.

In the next chapter, we perform an overview of the solution for the questions being addressed in this thesis. In addition, the detailed framework of service broker to address the Cloud SaaS simple and composite service selection is presented in the next chapter.

Chapter 4

Solution Overview

The previous chapter described the main research keywords definitions, the research issues, research aims and objectives. Four issues have been found that intend to be addressed in our thesis outcome. In this chapter, we determine the main research methodology that we follow in order to obtain our objectives. Also, we explain the overview of solutions for our proposed system.

4.1 Introduction

The Cloud SaaS has become the first option to deploy an IT service for the user who looks to reduce the cost of using the IT service, and the availability (Waters, 2005). Recently, many services are available to the end-user, and in some studies show the growth of the services to around more than 5000 services available. Therefore, the service consumer may face a problem when selecting the optimum service. Selection of Cloud SaaS is based on matching the functional requirements and meeting the non-functional requirements such as the service price. However, the Cloud SaaS consumer may pay more attention for selecting the Cloud SaaS based on meeting the non-functional preferences because growing in the number of Cloud SaaS service providers that have similar functional service. In addition, the services based on the SOA model have two flavors: a simple service and composite service. Therefore, this research aims to develop an intelligent service broker system to help a Cloud SaaS consumer to select the simple or composite service based on their preferences.

As discussed in Chapter 2, several existing researches have attempted to solve the problem of simple and composite Cloud SaaS selection. However, based on the discussion in Chapter 2 and Chapter 3, it is apparent that there are still significant research gaps concerning simple and composite Cloud SaaS selection need to be addressed especially selecting the service based on the consumer's preferences.

This chapter presents an overview of the proposed approach of Find SaaS service broker to work as an agent in between the Cloud SaaS consumer and Cloud SaaS provider.

4.2 General Framework Architecture of the proposed Service Broker (Find SaaS Framework)

The main outcome of this thesis is to design and develop a service broker called Find SaaS, which is able to find an optimum simple and composite cloud SaaS service provider for selection. The Find SaaS module has three main functionalities, in the beginning, ranking simple service providers, evaluation system based on consumer's experiences and decision-maker for composite Cloud SaaS for selection. Figure (4.2) represents the main thesis structure for the proposed service broker (Find SaaS).

Moreover, each functional service of service broker has multiple methods inside in order to obtain the best service result for a service consumer who requests the service. As we mentioned before, our service broker (Find SaaS) provides three main services as follows:

1. Ranking Cloud SaaS providers for a simple Cloud SaaS service selection.
2. Evaluation System based on the Cloud SaaS consumer's experiences.
3. Decision-maker system for selection the Cloud SaaS service composition.

In this chapter, we introduce the main methodology that have been implemented

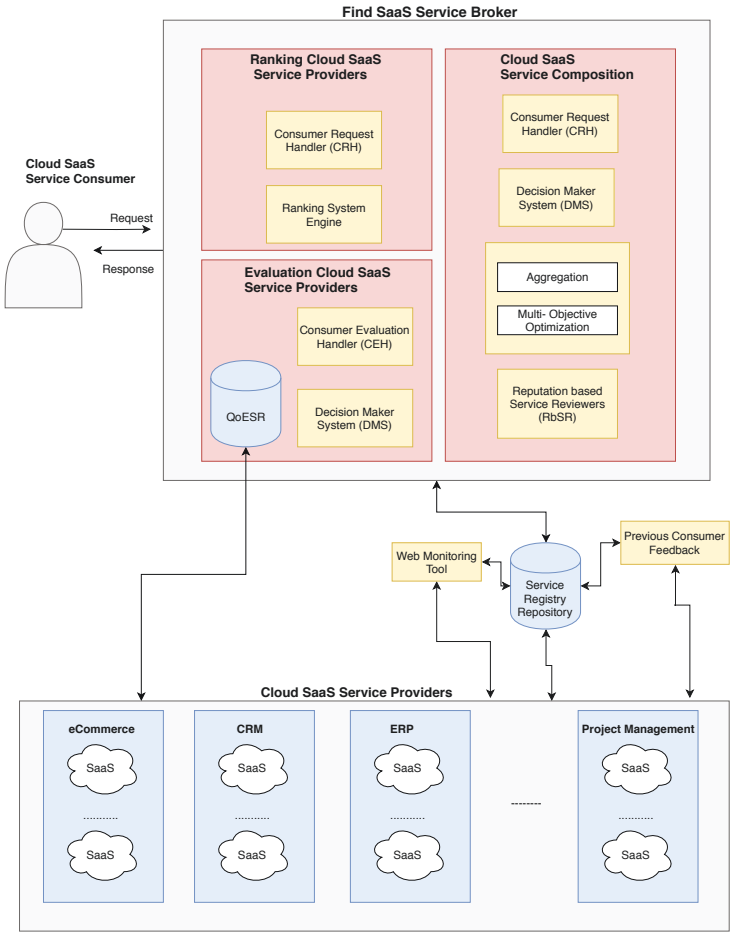


Figure 4.1 : Find SaaS Service Broker Framework Structure

in order to give to the thesis readers an overview of the methodology used in the thesis.

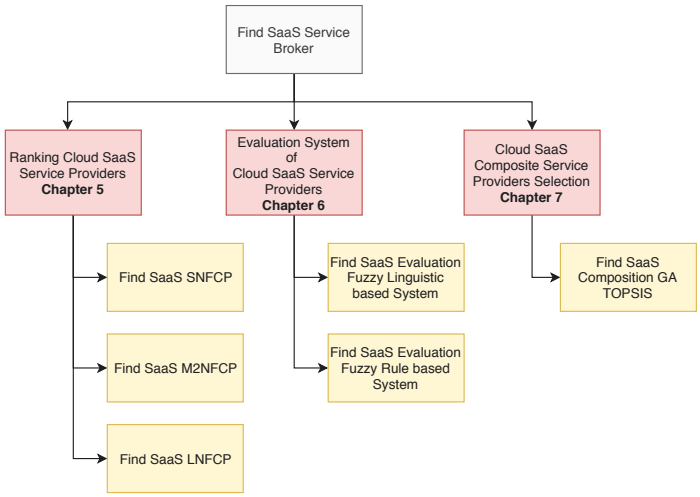


Figure 4.2 : Thesis Structure

4.2.1 Overview of the solution for ranking System of Cloud SaaS Providers (Find SaaS SNFCP, M2NFCP and LNFCP)

As was mentioned in the last chapter, the previous researches do not propose a complete methodology for Cloud SaaS service selection, taking into account the service consumer's preferences. The ranking system proposed is to address issues number 1 and part of issue number 2.

Our proposed ranking system goes through multiple stages in order to find and rank all service to the service consumer as shown in figure 4.3. The stages are as follows:

Stage 1: Consumer request handler - (CRH) is used to collect all service requirements from the service consumer (See section 5.3 for details).

This component has two major parts to address the service consumer's prefer-

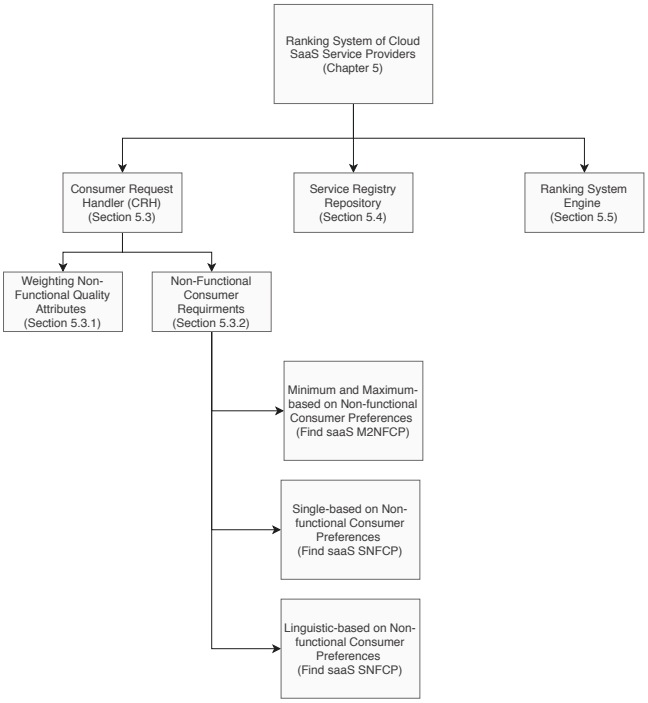


Figure 4.3 : Ranking System Engine of Cloud SaaS Providers Solution Structure

ences as follows:

1. Weighting Non-Functional Quality Attributes (See section 5.3.1 for details) This is used for weighting the quality attributes from the service consumer. The linguistic term based is used to weight the quality attributes such as the service price weighting as "Important" in selecting the service. The fuzzy logic is used to address the linguistic and to translate the linguistic values to a crisp number for further analysis.

2. Non-Functional Consumer Requirements (See section 5.3.2 for details) This section describes the consumer's requirements. In this thesis, we term the quantifying real value of consumer's non-functional requirements, such as the service price required is 20 dollar as consumer's non-functional requirements.

This thesis proposed three solutions to address the consumer's requirements. Firstly, addressing the single value of consumer's non-functional requirements, such as the service price is "20" dollar. Secondly, we address the minimum and maximum of consumer's non-functional requirements, such as the range of service price is between "20 to 30" dollar. Thirdly, we address the consumer's non- functional requirements as linguistic values such as the service price required as "Affordable price"

Stage 2: Service Registry Repository - (SRR) (See section 5.4 for details) In this section, we provide overview of registry repository that is used to interact with the consumer's requirements and the main ranking system engine. The service registry repository (SRR) consider the functional services of Cloud SaaS providers which is represented by service description as well as all the numerical non-functional quality attributes that appears with all services. Moreover, the quality attributes can be collected in three primary resources as follows: (1) Service provider (2) Monitoring tool (3) Previous consumer feedback.

Stage 3: Ranking System Engine - (RSE) (See section 5.5 for details) In this section, we introduce the main ranking system engine algorithm that is used to interact with the service registry repository and all service requirements concerns by a service consumer. The result will be all the list of services equivalents to consumer's preferences. The shortest distance method has been used as a core of the ranking engine.

4.2.2 Overview of the solution for Evaluation System (Find SaaS Evaluation)

Other studies have considered the evaluation of Cloud SaaS provider, taking into account the service consumer's experiences. As was mentioned in the literature review on evaluation services, the majority of authors developed assessment

methodologies based on a pairwise comparison of each service provider. However, there is no method to develop an evaluation system based on linguistic term describing the user experience of particular quality experience such as service usability. In addition, there is no method to provide the solution to evaluate the service consumer as percentage level and based on fuzzy rule method. Hence, this chapter 6 presents approaches to address the linguistic-based as well as fuzzy rule-based when evaluating Cloud SaaS provider.

In order to develop a comprehensive solution of the evaluation system, chapter 6 provides the methodology that has been used to assess the Cloud SaaS provider, and it composed into three main parts as shown in figure (4.4).

This chapter 6 covers solution to address part of issue 2, as well as issues 3 and 4.

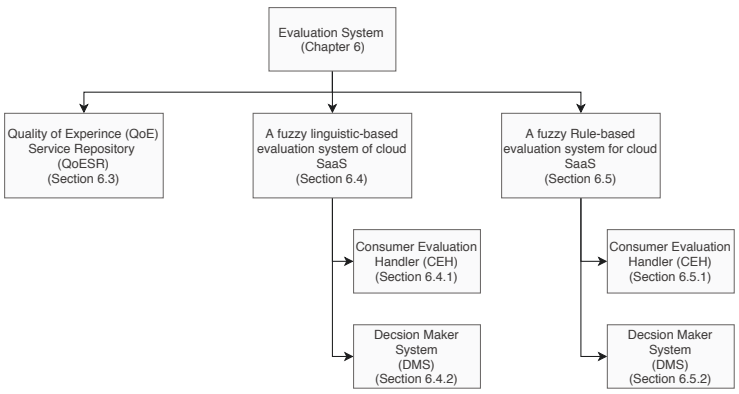


Figure 4.4 : Evaluation System of Cloud SaaS Solution Structure

Part 1) Quality of Experience (QoE) Service Repository (QoESR) (See section 6.3 for details) In this section, we provide the main quality experience attributes concerns by a service consumer when evaluating the Cloud SaaS provider. These qualities have been collected by relevant literature review published

recently in view of this issue and by several experts in the field of Cloud SaaS service selection. We have classified the (QoESR) into three main groups: Vendor, Data and Service.

Part 2) A fuzzy linguistic-based evaluation system of Cloud SaaS (See section 6.4 for details) In this section, we provide the solution to address the evaluation system using the linguistic terms for evaluation of the particular quality experience. The fuzzy logic concept has been used to translate the linguistic values to a crisp number for final assessment mechanism. The TOPSIS MCDM is used for final decision making for evaluation of the Cloud SaaS provider in order to find which Cloud SaaS provider has a match with the consumer's preferences.

Part 3) A fuzzy rule-based evaluation system for Cloud SaaS (See section 6.5 for details) In this section, we provide a solution to address the consumer's experiences using the percentage level of the quality of experiences. The Fuzzy rule-based is used to make a decision for the evaluation process. It also used the FAM fuzzy concept for generating the fuzzy rule for evaluation purpose using two linguistic variables as a system inputs and one output result. In the end, the aggregation function to aggregate all the results from the fuzzy logic and to make final evaluation score for each Cloud SaaS provider.

4.2.3 Overview of the solution for Composite Cloud SaaS service Selection (Find SaaS GA TOPSIS)

The final research contribution of this thesis is to design and develop the service selection mechanism of Cloud SaaS service composition. As was mentioned in the literature review of Cloud SaaS service composition, the previous works concerned only to develop service composition by only taking into account to reduce search space for selecting best combination of Cloud SaaS providers. It means that none of

the existing approaches develops decision-making approach after aggregating Cloud SaaS providers. Also, the reputation factor in finding the optimum composite services is missing in the existing research approaches. Hence, our research work in this thesis develops two combining methods which are the genetic algorithm (GA) and multi-criteria decision-making (MCDM) approach to find and select the optimum composite of Cloud SaaS providers. We also develop the reputation factor using the service reviewers with service rating called the reputation-based service reviewers (RbSR).

Chapter 7 covers solution to address the issues number 3.

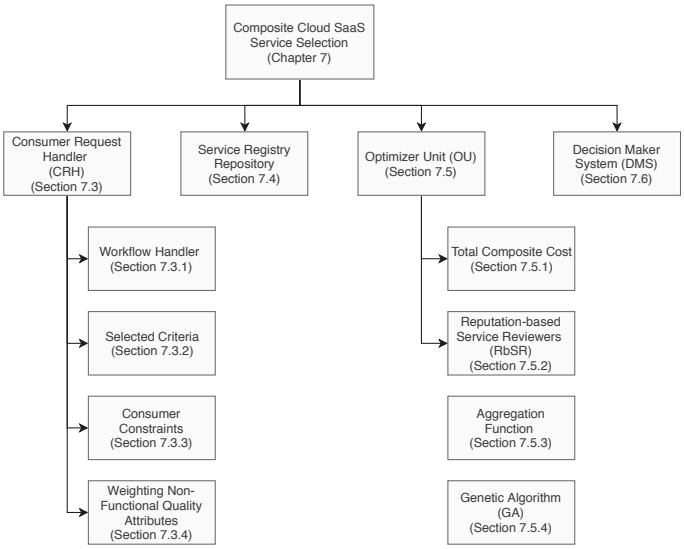


Figure 4.5 : Composite Cloud SaaS Service Selection Solution Structure

In order to build our research methodology for selecting the composite Cloud SaaS, multiple phases have been utilized for final service selection of composite Cloud SaaS providers.

Phase 1) Consumer Request Handler (See section 7.3 for details) In this framework component, the consumer's preferences are obtained to find the best Cloud

SaaS service composition meeting the consumer's preferences. This component has multiple inputs to address the consumer's requirements, as follows:

(a) Workflow Handler (See section 7.3.1 for details) This unit addresses the functional service demands or as known as the abstract services required by a Cloud SaaS consumer.

(b) Selected Criteria (See section 7.3.2 for details) This unit is used to gather the non-functional quality attributes concerns by a service consumer when selecting the service composition. That is used for the decision-maker process in order to select the composite services that meet the consumer's preferences.

(c) Consumer Constraints (See section 7.3.3 for details) This unit is responsible for gathering the service consumer's constraints of the reputation factor and the total cost of composite service. The service consumer requires to declare the value of the constraints of minimum and maximum for both factors. The genetic algorithm is used to address the optimization problem handled the consumer's constraints during the optimization process in order to find the composite services satisfy service consumer's constraints.

(d) Weighting Non-Functional Quality Attributes (See section 7.3.4 for details) In this section, after gathering the non-functional interests by the service consumer, the weighting process is required in order to weight the significance of attributes comparing with the other. The linguistic terms based is used to weight the importance of attributes similar to the ranking system approach proposed in the chapter 5.

Phase 2) Service Registry Repository (See section 7.4 for details) In this component, we provide the service registry repository which is used to interact with the con-

sumer request handler and with the optimizer unit. The proposed repository has the same functionality similar to the service registry repository architecture in chapter 5.

Phase 3) Optimizer Unit (See section 7.5 for details) This component is responsible for addressing the optimization problem in order to find the composite services with maximum reputation factor and minimum total cost of service composition. The Genetic algorithm (GA) is used for this purpose.

Phase 4) Decision Maker System (DMS) (See section 7.6 for details) DMS is the last component of our proposed approach to select the composite Cloud SaaS provider. This component is responsible for making the decision in order to select the optimum composite Cloud SaaS providers that meet with the service consumer's preferences. The TOPSIS method is used for decision making purpose.

4.3 Conclusion

In this chapter, we presented the solution overview of Find SaaS service broker framework architecture to address the research objectives in this thesis. The overall architecture was presented, and the functional layers were discussed in detail.

In the next chapter, we describe the first research objective in order to select the optimum simple Cloud SaaS provider. The ranking system is introduced for this purpose called Find SaaS (SNFCP, M2NFCP and LNFCP).

Chapter 5

Find SaaS (SNFCP, M2NFCP and LNFCP) Simple Cloud SaaS Service Selection

In this chapter, we propose a methodology for carrying out the ranking system of Cloud SaaS providers based on consumer's preferences for selecting an optimal service. The proposed approach is called Find SaaS (SNFCP, M2NFCP and LNFCP). An examination of the previously published state-of-the-art techniques for Cloud SaaS selection, as described in Chapter (2), reveals that few studies have been conducted on the service selection of Cloud SaaS provider taking into account the Cloud SaaS consumer's preferences. Hence, this research proposes multiple options of Cloud SaaS consumer preference options to rank and sort the Cloud SaaS providers in order to select an optimum service.

The proposed methods were tested on diverse datasets and the results obtained from various experiments are summarized. Major parts of the chapter have been published in two conference articles, titled: Software as a Service (SaaS) Service Selection based on Measuring the Shortest Distance to the Consumers Preferences; and Dynamic Ranking System of Cloud SaaS based on Consumer Preferences - Find SaaS M2NFCP (Ikram & Hussain, 2018) (Ikram et al., 2019).

5.1 Introduction

Cloud Software as a Service (SaaS) has become the first option used to deploy an IT service with a view to reducing the cost of building an IT house as it provides low-cost management, and a more secure environment. Several large IT vendors, such

as IBM, SAP, and Oracle, provision their services through Cloud SaaS. Recently, there has been an increase in the number of Cloud SaaS service providers, which makes it challenging for a Cloud SaaS service consumer to select the best service.

The literature proposes several theories to explain how to select a Cloud SaaS service provider for a service consumer. The majority of existing approaches analyze the selection of Cloud SaaS as a multi-criteria decision-making (MCDM) research problem. Therefore, they only consider weighting the most important quality attributes consumer's use when selecting the best Cloud SaaS service provider for their needs. The objective of this study is to propose a ranking system engine (RSE) of Cloud SaaS service providers based on consumer preferences including not only the weighting quality attributes but also by quantifying the value of selected quality attributes. We term the requesting of the particular value of quality attributes in this research as non-functional consumers' requirements. Therefore, we focus on creating a ranking system engine of Cloud SaaS providers which considers and focuses on a consumer's preferences of multiple quality attributes.

In this chapter, we propose a service broker (Find SaaS) to interact with a service consumer to rank and sort Cloud SaaS service provider for optimum service selection purposes. The proposed approach ranks the services based on measuring the shortest distance to the consumers non-functional preferences. A case study is used to explain the proposed approach. In addition, to evaluate our proposed method, a CRM dataset is created for this purpose. The mean average precision mAP and normalized discounted cumulative gain $nDCG$ metrics are used to measure the performance and to validate the results. Finally, the limitations of the proposed approach are discussed, and suggestions are made for future research directions to improve our research work.

The remainder of the chapter is organized into nine sections: Section (5.2) de-

scribes the general framework of the ranking system, Section (5.3) discusses/analyzes the consumer request handler (CRH) part of ranking framework, Section (5.4) discusses the Service Registry Repository (SRR) part, Section (5.5) describes the ranking system mechanisms, Section (5.6) provides a case study, Section (5.7) presents the experimental aspects, Section (5.8) discusses the chapter outcomes, and finally, Section (5.9) concludes the chapter.

5.2 General Architecture of Find SaaS (SNFCP, M2NFCP and LNFCP) Ranking Engine System (RES) for Cloud SaaS Service Providers

We introduce a ranking system approach called Find SaaS (SNFCP, M2NFCP and LNFCP) for our service broker to rank and sort Cloud SaaS service providers based on consumer preferences. We hypothesize that a service broker, which is named Find SaaS (SNFCP, M2NFCP and LNFCP), can mediate the communication process between service consumers and service providers. The proposed system architecture of our approach is shown in Figure (5.1). It is composed of three main components that can interact with each other, described as follows:

(Section 5.3) **Consumer Request Handler (CRH):** This part of the service broker addresses the consumer's functional and non-functional preferences in order to select a service based on the consumer's preferences. Our proposed approach classifies the consumers non-functional preferences into two types: (a) Weighting the most important quality attributes (see section 5.3.1) and (b) Quantifying value of non-functional quality attributes (Note: Consumer non-functional requirements describing the quantifying value of consumer non-functional requirements) (See section 5.3.2). In addition to this, the non-functional consumer requirements are divided into three categories. Each category makes

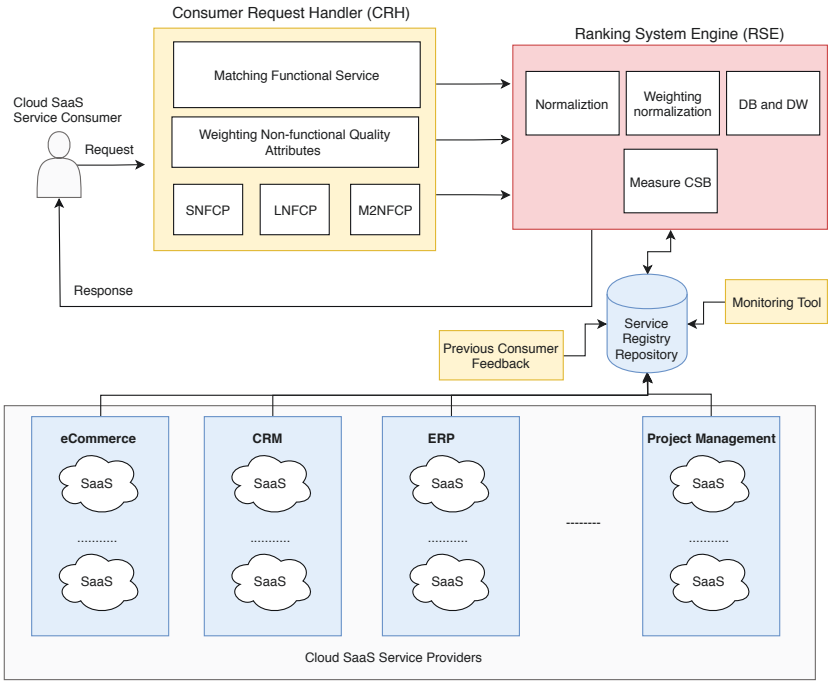


Figure 5.1 : Ranking Framework Architecture

a new contribution of ranking and ordering Cloud SaaS service providers for Cloud SaaS service consumers, described in section (5.3.2, 5.3.2 and 5.3.2).

(Section 5.4) **Service Registry Repository (SRR):** This part describes the service broker repository that will be interacting between the service provider and service consumer. We identify in this section the main important parameters in terms of service description and non-functional quality attributes. This part metadata in between consumer request handler (CRH) and ranking system engine (RSE). Section (5.4) presents the details of our proposed approach.

(Section 5.5) **Ranking System Engine (RSE):** This part describes the technique of ranking approach of our service broker. The ranking approach is based on measuring the shortest distance to consumer non-functional preferences. It considers the parameters of the requirements generated by CRH, then ranks all ser-

vice providers based on these requirements. Section (5.5) provides detailed information about our ranking approach.

5.3 Consumer Request Handler (CRH)

The consumer request handler (CRH) is responsible for gathering the information required from the service consumer to rank and sort Cloud SaaS service providers based on their preferences. The ranking approach begins when a service consumer submits a new service request that includes service description preference and non-functional quality preferences. Matching service description of consumer preferences is done mainly via semantic matching (Tahamtan, Beheshti, Anjomshoaa, & Tjoa, 2012) (Cortzar, Zapater, & Snchez, 2012) or syntactic matching (Sajjanhar, Hou, & Zhang, 2004). The service description matching component uses to generate a list of service providers satisfying consumers functional requirements. For this purpose, the algorithm by Sajjanhar et al. (Sajjanhar et al., 2004) was applied. However, due to the variety of similar service providers offered the same functional type; therefore, a service consumer pays increasing attention to non-functional quality attributes for selecting the service. Therefore, this research focuses on developing multiple approaches to rank services based on non-functional preferences.

After matching with a service Cloud SaaS functional description, such as CRM or ERP, the Cloud SaaS service consumer must present their preferences for non-functional quality attributes. Firstly, Cloud SaaS consumer selects their preferences of quality attributes for selection purpose. For example, the consumer (x) selects quality attributes such as service availability and service price, which are used to rank and sort the services available. After this, the service consumer provides their preferences for these quality attributes in two types, as shown in Figure (5.2). Type (1) The importance of quality attributes where a Cloud SaaS consumer

needs to weight their preferences. There are different mechanisms to weight quality attributes, such as by ranking-based objective, or by the analytic hierarchy process (AHP) method. However, most Cloud SaaS consumers prefer the simplest way to indicate their preferences, which is by using linguistic values, such as the price of service is Not Important, but the service availability is Very Important. Therefore, this research uses linguistic terms to weight the importance of quality attributes for a service consumer. Type (2) quantify value describing selected non-functional service requirements. For example, a consumer may consider a service with a price of $20 or a service availability that is 100 per cent. Moreover, in order to offer more options to present their non-functional regarding to quantify value for service consumer's preferences, we classify the second types into three categories: (1) Single-based on Non-Functional Consumer Preferences - Find SaaS SNFCP approach, (2) Minimum and Maximum-based on Non-Functional Consumer Preferences - Find SaaS M2NFCP approach, and (3) Linguistic-based on Non-Functional Consumer Preferences - Find SaaS LNFCP approach.

The consumer request handler CRH after processing service requirements generates the best-ideal service BIS_i, worst-ideal service (WIS_i), and weighting quality attributes vector w_i, and sent it to the ranking system engine RSE which interacts with the service registry repository SER for ranking the services required to service consumer.

5.3.1 Weighting Non-Functional Quality Attributes

Several approaches proposed recently to measure the importance of quality attributes, such as ranking-based objective or using on the min and max in the weighting approach. Since service consumers express their preferences in ambiguous or vague terms, such as "Extremely Important" or "Not Important", the weighting quality attributes that have been used for this study are based on linguistic terms.

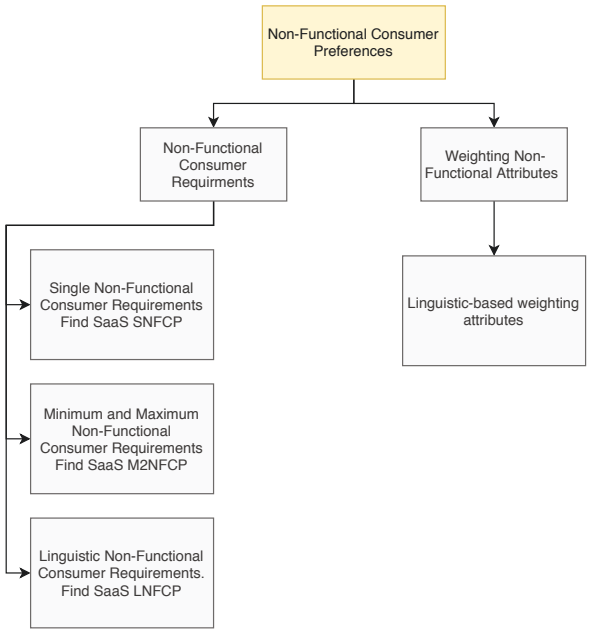


Figure 5.2 : Non-functional preferences classification

This study classifies the linguistic values for weighting quality attributes into seven linguistic values, as follows: Extremely Important (EI), Very Important (VI), Important (I), Somewhat Important (SI), Not Important (NI), Not Very Important (NVI), and Not Important at All (NIA). Note that, if a service consumer does not select certain quality attribute, the CRH will weight these quality attribute as Not Important at All - NIA. That means the system will consider these quality attributes to have very low preferences.

Fuzzy set

In order to translate these linguistic values into numerical quantity for further analysis, the fuzzy logic concept is used. Moreover, the triangular fuzzy number $\hat{A} = (a, b, c)$ has been launched to compute the membership function, as shown in Figure (5.3). Table 7.4 describes the linguistic terms with a set of fuzzy numbers.

Defuzzification

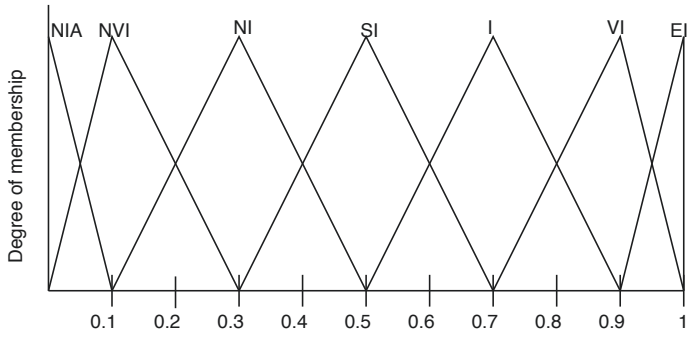


Figure 5.3 : Degree of membership function of triangular fuzzy logic or linguistic variables

Table 5.1 : Linguistic terms with their fuzzy numbers

Linguistic terms	Triangular fuzzy number
Extremely Important (EI)	(0.9,1.0,1.0)
Very Important (VI)	(0.7,0.9,1.0)
Important (I)	(0.5,0.7,0.9)
Somewhat Important (SI)	(0.3,0.5,0.7)
Not Important (NI)	(0.1,0.3,0.5)
Not Very Important (NVI)	(0.0,0.1,0.3)
Not Important at All (NIA)	(0.0,0.0,0.1)

Defuzzification is a process to convert fuzzy numbers defined over an output universe of discourse into a non-fuzzy (crisp) value. It generates a crisp value that best represents the possibility distribution of an inferred fuzzy number. There are many different defuzzification methods such as RCOM (Random Choice Of Maximum), FOM (First of Maximum), LOM (Last Of Maximum), COG (Center Of Gravity), MeOM (Mean Of Maxima), WFM (Weighted Fuzzy Mean), QM (Quality Method), EQM (Extended Quality Method), and COA (Center Of Area) (Van Leekwijck & Kerre, 1999).

To obtain the crisp value of a triangular fuzzy number $\hat{A} = (a, b, c)$,we compute the COG of $\hat{A}$ using the following formula (5.1):

$$d(\hat{A}) = a + \frac{(a-b) + (c-b)}{3} \tag{5.1}$$

Weighting Quality Attributes

The crisp weight of non-functional (w_i) is calculated using the following formula (7.2):

$$w_j = \frac{d(\hat{A}_j)}{\sum_{j=1}^{n} d(\hat{A}_j)} \tag{5.2}$$

As a result, the list of weighting quality attributes $W = w_1, w_2, ..., w_n$ is generated to the ranking system describing the set of the importance of non-functional attributes.

The following algorithm (1) is used to weight the importance of quality attributes.

5.3.2 Non-functional Consumer Requirements

The consumer's requirements of non-functional attributes used to quantify the particular values that match a consumer's preferences of non-functional attributes.

vlined 1 Algorithm for weighting quality attributes using linguistic values

Input: Linguistic values with their fuzzy number and importance of quality attributes Output: List of Weighting quality attributes (W)

foreach *non-functional attribute* **do**

match the linguistic values Fuzzy number $(a, b, c), (a^{'}, b^{'}, c^{'})and(a^{''}, b^{''}, c^{''})$ compute the membership function $(membership(NFQ_i))$

end

return list of membership function for each quality attributes Calculate the summation of membership function $sum(membership(NFQ_i))$; **foreach** *non-functional attribute* **do**

calculate the weighting quality attributes $w_i = membership(NFQ_i)/sum(membership(NFQ_i))$

end

return the list of (W)

The specification is done using multiple approaches, which can be classified as follows:

1. Single-based on Non-Functional Consumer Preference: This approach is called - Find SaaS SNCP.

2. Minimum and Maximum-based on Non-Functional Consumer Preferences: This approach is called - Find SaaS M2NFCP.

3. Linguistic-based on Non-Functional Consumer Preferences: This approach is called - Find SaaS LNFCP.

Table (5.2) shows examples of non-functional requirements for the criterion of price.

Definition Non-Functional Classification

There are two types of non-functional attributes:

Table 5.2 : The examples of non-functional requirements for the price attribute

Single-based on Non-Functional Consumer Preferences (SNFCP)	$90
Minimum and Maximum-based Non-Functional Consumer Preferences (M2NFCP)	$80-$90
Linguistic-based on Non-Functional Consumer Preferences (LNFCP)	Affordable Price

1. Positive non-functional quality attributes are those that have the best amount when they increase, such as availability, number of service reviewers, and service ratings.

2. Negative non-functional quality attributes are those that have the best amount when they decrease, such as price and service response time.

$$Non-FunctionalAttributes = \begin{cases} \alpha, & Positive(Non-Functionalattribute) \\ \beta, & Negative(Non-Functionalattribute) \end{cases}$$

Single-based on Non-Functional Consumer Preferences (Find SaaS SNFCP)

In this section, we propose a Find SaaS SNFCP to understand the consumers non-functional requirement as a single value for individual criterion. For example, the service price that a service consumer prefers is $90 monthly for a subscription. In the literature review, many researchers proposed a service selection approach based on matching consumer requirements with the service list. However, this approach, in some cases, there are no services that match consumer requirements. Therefore, zero services will be retrieved. In addition, some methods proposed a recommender system to generate services based on matching the similarity of the active consumer who requests a service with a previous consumer (Afify et al., 2017). This approach

is frequently used to address the service selection in the web service. However, the cloud services are different from the web services in terms of the number of services, service level agreement (SLA) and the contract period.

Assume that cp_1, cp_2,....,cp_n is the set of selected quality attributes by a service consumer, and cr_1, cr_2,...,cr_n is the set of single quantifying requirements of qualities assign by a service consumer. Note that the quantifying values of consumer's requirements must be in between the range of the minimum and maximum of these selected attributes. Moreover, in case the service consumer has not selected some quality attributes as preferences, the CRH generates the minimum and the maximum value of these none selected attributes, but with a very low weighting (Not Important at All) linguistic weight. Using these inputs, the CRH will generate the set of BIS, and the worst-ideal service WIS to rank service providers to Cloud SaaS service consumer.

If the selected non-functional preferences are positive, the set of BIS is calculated using equation (5.3), and the set of WIS is calculated using equation (5.4).

$$BIS = Cr_{cp}^{\alpha_1}, Cr_{cp}^{\alpha_2}, ..., Cr_{cp}^{\alpha_i}, ..., Cr_{cp}^{\alpha_n} \tag{5.3}$$

where $cr_{cp}^{\alpha_i}$ the consumer non-functional requirements requested by a service consumer.

$$WIS = min_{cp}^{\alpha_1}, min_{cp}^{\alpha_2}, ..., min_{cp}^{\alpha_i}, ..., min_{cp}^{\alpha_n} \tag{5.4}$$

where $min_{cp}^{\alpha_i}$ is the minimum value of quality attribute i.

On the other hand, if selected non-functional attributes are negative, the set of BIS is calculated by equation (5.5), and the set of WIS is calculated by equation (5.6).

$$BIS = cr_{cp}^{\beta_1}, cr_{cp}^{\beta_2}, ..., cr_{cp}^{\beta_i}, ..., cr_{cp}^{\beta_n} \tag{5.5}$$

Where $cr_{cp}^{\beta_i}$ is the consumer non-functional requirements requested by a service consumer.

$$WIS = max_{cp}^{\beta_1}, max_{cp}^{\beta_2}, ..., max_{cp}^{\beta_i}, ..., max_{cp}^{\beta_n} \tag{5.6}$$

where $max_{cp}^{\beta_i}$ is the maximum value of quality attribute i.

The algorithm for the Find SaaS SNFCP is illustrated in algorithm (2).

A case study is introduced in section (5.6) to explain the working functionality of Find SaaS SNFCP proposed approach.

Minimum and Maximum-based on Non-Functional Consumer Preferences (Find SaaS M2NFCP)

In this section, the Find SaaS M2NFCP is proposed. This approach proposes to modify the previous approach to accept a range of consumer non-functional requirements instead of a single-based requirement for each criterion. Therefore, we consider the minimum and maximum consumer non-functional requested requirements. For instance, our proposed method can represent the price of a service by providing a price range such as \$70 to \$90. Therefore, the CRH generates the BIS vector and WIS vector considering the ranges of non-functional requirements.

If positive quality attributes are selected as consumer's preferences, equation (5.7) is used to calculate the set of BIS and (5.8) is used to calculate the set of WIS.

$$BIS = max_{cr}^{\alpha_1}, max_{cr}^{\alpha_2}, ..., max_{cr}^{\alpha_i}, ..., max_{cr}^{\alpha_n} \tag{5.7}$$

vlined 2 Algorithm for Find SaaS SNFCP

initialization

Input: dataset including the list of similar services functionality and non-functional descriptions Set of the minimum of non-functional attributes $min_i^{NF} = (min_1^{NF}, min_2^{NF}, ..., min_n^{NF})$ Set of the maximum of non-functional attributes $max_i^{NF} = (max_1^{NF}, max_2^{NF}, ..., max_n^{NF})$ Set of the non-functional quality types $NFt_i = (Nft_1, nft_2, ..., nft_n)$ $nft_i = positive/negative$ Select the non-functional preferences $CP_i = (cp_1, cp_2, ..., cp_n)$

Result: The set of BIS_i and set of WIS_i

for $\forall$ *attribute* i *in* $1 : n$ **do**

  while $NF_i == cp_i$ **do**

    if $NF_i ==$ *positive* **then**

      Ask for the consumer's request (cr_i)

      while $cr_i > min_i^{NF}$ *&* $cr_i < max_i^{NF}$ **do**

        $BIS_i = cr_i$ $WIS_i = min_i^{NF}$

      end

      return(BIS_i) *return*(WIS_i)

    end

    if $NF_i ==$ *negative* **then**

      Ask for the consumer's request (cr_i) **while** $cr_i > min_i^{NF}$ *&* $cr_i < max_i^{NF}$ **do**

        $BIS_i = cr_i$ $WIS_i = max_i^{NF}$

      end

      return(BIS_i) *return*(WIS_i)

    end

  end

  while $NF_i \neq cp_i$ **do**

    if $NF_i ==$ *positive* **then**

      $Lw_i =$ *Not Important at All (NIA)* $BIS_i = max_i^{NF}$ $WIS_i = min_i^{NF}$ *return*(BIS_i) *return*(WIS_i)

    end

    if $NF_i ==$ *negative* **then**

      $Lw_i =$ *Not Important at All (NIA)* $BIS_i = min_i^{NF}$ *and* $WIS_i = max_i^{NF}$ *return*(BIS_i)

      return(WIS_i)

    end

  end

  return(NF)

end

$$WIS = min_{cr}^{\alpha_1}, min_{cr}^{\alpha_2}, ..., min_{cr}^{\alpha_i}, ..., min_{cr}^{\alpha_n} \tag{5.8}$$

where, $max_{cr}^{\alpha_i}$ is the maximum non-functional requirement demanded by a service consumer, and $min_{cr}^{\alpha_i}$ is the minimum non-functional requirements requested by a service consumer.

On the other hand, if negative quality attributes are selected, the BIS and WIS generated by CRH are calculated using equations (5.9) and (5.10) respectively.

$$BIS = min_{cr}^{\beta_1}, min_{cr}^{\beta_2}, ..., min_{cr}^{\beta_i}, ..., min_{cr}^{\beta_n} \tag{5.9}$$

$$WIS = max_{cr}^{\beta_1}, max_{cr}^{\beta_2}, ..., max_{cr}^{\beta_i}, ..., max_{cr}^{\beta_n} \tag{5.10}$$

Where $min_{cr}^{\beta_i}$ is the minimum non-functional requirement of attribute i, and $max_{cr}^{\beta_i}$ is the maximum non-functional requirement of attribute i. All these requirements requested and addressed by a service consumer.

The algorithm for Find SaaS M2NFCP is shown in algorithm (3).

Linguistic-based on Non-Functional Consumer Preferences - Find SaaS LNFCP

In this section, we address the service consumer requirements based on linguistic requested to quantify the value of the quality attribute. As we know that the other previous approaches deal with the numerical requesting for quantifying the service requirements, the LNFCP proposed approach will be novel, as it is for requesting ranking services based on linguistic-based requirements. The proposed approach helps a service consumer to identify their preferences as a linguistic method. An example of this kind of request is that a price criterion can be described as an affordable price or low price.

vlined 3 Algorithm for Find SaaS M2NFCP

initialization

Input: dataset including the list of similar services functionality and non-functional descriptions Set of the minimum of non-functional attributes $min^i_{NF} = (min^1_{NF}, min^2_{NF}, ..., min^n_{NF})$ Set of the maximum of non-functional attributes $max^i_{NF} = (max^1_{NF}, max^2_{NF}, ..., max^n_{NF})$ Set of the quality type $nft = (nft_1, nft_2, ..., nft_n)$ $nft_i = positive/negative$ Select the non-functional preferences $CP_i = (cp_1, cp_2, ..., cp_n)$ **Result:** The set of BIS_i and set of WIS_i

for *i in 1:n* **do**

  while $NF_i == cp_i$ **do**

    if $NF_i == positive$ **then**

      Ask for the minimum consumer's request (min^i_{cr}) Ask for the maximum consumer's request (min^i_{cr})

      while $min^i_{cr} \geq min^i_{cr}$ $min^i_{cr} < max^i_{NF}$ $max^i_{cr} > min^i_{NF}$ $max^i_{cr} \leq max^i_{NF}$ **do**

        $BIS_i = max^i_{cr}$ $WIS_i = min^i_{cr}$

      end

      return(BIS_i) return(WIS_i)

    end

    if $NF_i == negative$ **then**

      Ask for the minimum consumer's request (min^i_{cr}) Ask for the maximum consumer's request (max^i_{cr})

      while $min^i_{cr} \geq min^i_{NF}$ $min^i_{cr} < max^i_{NF}$ $max^i_{cr} > min^i_{NF}$ $max^i_{cr} \leq max^i_{NF}$ **do**

        $BIS_i = min^i_{cr}$ $WIS_i = max^i_{cr}$

      end

      return(BIS_i) return(WIS_i)

    end

  end

  while $NF_i \neq cp_i$ **do**

    if $NF_i == positive$ **then**

      Lw_i= Not Important at All (NIA) BIS_i= max^i_{NF} $WIS_i = min^i_{NF}$

      return(BIS_i) return(WIS_i)

    end

    if $NF_i == negative$ **then**

      Lw_i = Not Important at All (NIA) $BIS_i = min^i_{NF}$ and $WIS_i = max^i_{NF}$

      return(BIS_i) return(WIS_i)

    end

  end

  return(NF)

end

In order to address this problem, a fuzzy logic concept is proposed. Lotfi Zadeh introduced Fuzzy logic in 1965 (Zadeh, 1999). The basic concept of a fuzzy set is to address the uncertainty and vagueness of human responses and subjective judgements to design an expert system. The fuzzy set has different shapes for presenting the fuzzy number, such as a trapezoidal or triangular fuzzy number (Chauhan & Singh, 2016). Because of the simplicity of triangular fuzzy numbers, Find SaaS LNFCP uses this type of fuzzy number to describe the linguistic terms for the quality attributes.

Fuzzy set

To build a fuzzy system, the fuzzy set must first be obtained. To do this, the following different techniques have been proposed: (1) obtaining knowledge from a single expert to form fuzzy set information; (2) obtaining knowledge from multiple experts to obtain all the information to produce the fuzzy sets; (3) using artificial neural networks that learn the operation data and then develop the fuzzy sets; and (4) predicting the fuzzy sets using the box plot diagram (Asanka & Perera, 2017) and the interquartile range (IQR), as shown in Figure (5.4). Find SaaS LNFCP uses the box plot of each criterion to build the fuzzy membership function.

Figure (5.5) shows the membership functions of the fuzzy system for each criterion. Figure (5.5a) shows the membership of service price and includes the following linguistic terms: low price, affordable price, and high price. Figure (5.5b) shows the membership of service rating, Figure (5.5c) shows the membership of service reviewers, Figure (5.5d) shows the membership of service founded, Figure(5.5e) shows the membership of service availability, Figure (5.5f) shows the membership of response time, and Figure (5.5g) shows the membership of throughput. Table 5.3 presents the main quality attributes considered in the system with the proposed fuzzy number. All quality attributes except for the service reviewers are designed to have three linguistics values, such as low price, affordable price, and high price. The reason why

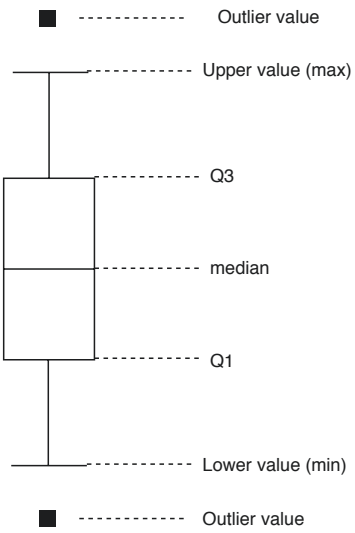


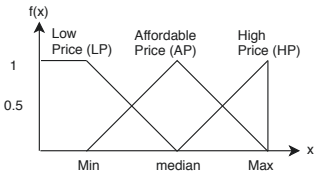
Figure 5.4 : Box plot explanation

the service re- viewers have five notations is due to the differences in the number of reviewers for similar services. For example, in the CRM dataset (Section 5.7.1), Salesforce has 5869 reviewers, but the other CRM services have fewer than one thousand. Salesforce is the most popular service, as it offers a customer relationship management component.

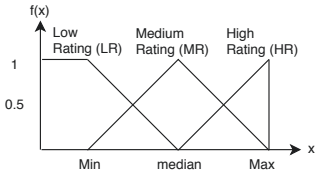
Defuzzification

We provide the following example to explain how the system works. If a service consumer requests a service with an affordable price then, based on table 5.3, the fuzzy number is (a,b,c) = (min, median, max). Based on the CRM dataset in the experiment section, the minimum price for CRM services is 7.94, the maximum price is 82.18, and the median is 19.99. Therefore, in this case, the fuzzy number will be (7.94,19.99,82.18), so the interpreter generates the BIS and WIS on the edge of the fuzzy number, as shown in Figure (5.6). Therefore, the BIS for the previous example is (7.94) and the WIS is 82.18.

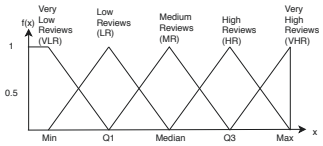
Consider $NF_1, NF_2, ..., NF_n$ are the set of quality attributes, $min_1, min_2, ..., min_n$



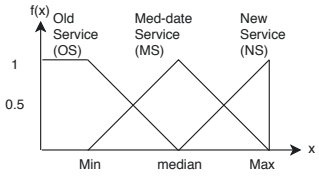
(a) Membership of Price



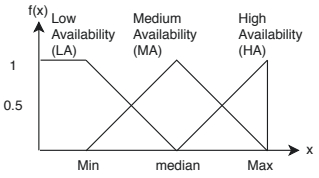
(b) Membership of service rating



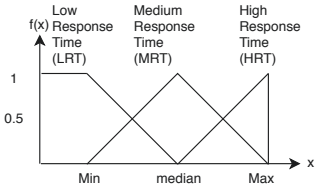
(c) Membership of service reviewers



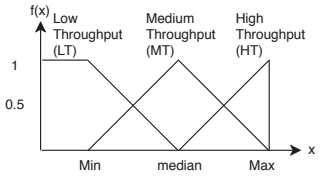
(d) Membership of service Refounded



(e) Membership of availability



(f) Membership of Response Time



(g) Membership of Throughput

Figure 5.5 : Membership function of linguistic values for the non-functional attributes proposed in the study

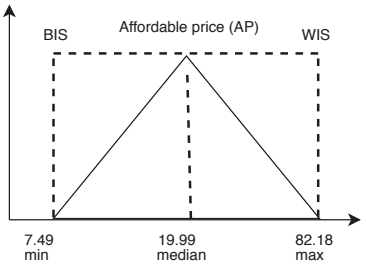


Figure 5.6 : Affordable price membership function of the CRM dataset

Table 5.3 : The linguistic values used to measure consumer preferences

Linguistic variable: Price		
Linguistic value	Notation	Numerical range
Low Price	LP	(0,min,median)
Affordable Price	AP	(min,median,max)
High Price	HP	(median,max,max)
Linguistic variable: Rating Review		
Linguistic value	Notation	Numerical range
Low Rating	LR	(0,min,median)
Medium Rating	MR	(min,median,max)
High Rating	HR	(median,max,max)
Linguistic variable: Number of reviewers		
Linguistic value	Notation	Numerical range
Very Low reviewers	VLR	(0,min,Q1)
Low reviewers	LR	(min,Q1,median)
Medium reviewers	MR	(Q1,median,Q3)
High reviewers	HR	(median,Q3,max)
Very High reviewers	VHR	(Q3,max,max)
Linguistic variable: Year Founds		
Linguistic value	Notation	Numerical range
Old Service	OS	(0,min,median)
Med-date Service	MD	(min,median,max)
New Service	NS	(median,max,max)
Linguistic variable : Availability		
Linguistic value	Notation	Numerical range
Low Availability	LA	(0,min,median)
Medium Availability	MA	(min,median,max)
High Availability	HA	(median,max,max)
Linguistic variable : Response Time		
Linguistic value	Notation	Numerical range
Low Response Time	LRT	(0,min,median)
Medium Response Time	MRT	(min,median,max)
High Response Time	HRT	(median,max,max)
Linguistic variable : Throughput		
Linguistic value	Notation	Numerical range
Low Throughput	LT	(0,min,median)
Medium Throughput	MT	(min,median,max)
High Throughput	HT	(median,max,max)

are the set of minimum quantity for each quality attributes, $max_1, max_2, ..., max_n$ are the set of maximum quantity for each quality attribute, and $median_1, median_2, ..., median_n$ are the set of median for each quality attributes. Also, as consumers make requests linguistically, therefore suppose $cp_1, cp_2, ..., cp_n$ is the set of selected quality attributes as consumer preferences and $Lin_1^{NF}, Lin_2^{NF}, ..., Lin_n^{NF}$ is the set of linguistic non-functional requirements requested by a service consumer. Therefore, CRH generates BIS and WIS for ranking purposes considering the selected quality attributes as preferences for the non-functional parameters as shown in algorithm (4).

5.4 Service Registry Repository (SRR)

Cloud SaaS is a software application that can be accessed via the Internet. The Cloud SaaS service provider advertises their services to the public with two primary identifications, (1) service description or functional software type such as Salesforce which provides customer relationship management (CRM) as functional type, and (2) non-functional quality attributes, such as the service price.

This section describes the service registry repository (SRR) for our service broker to register all service providers information. SRR interacts with the RSE and the CRH to retrieve all services requested by a service consumer. The SRR as shown in Figure (7.3), contains the following multiple functional Cloud SaaS types $CSF = \{CSF_1, CSF_2, CSF_3, ..., CSF_m\}$. Each functional type CSF_i includes multiple service providers sharing similar functionality.

Let us assume CSF_1 match with CRM service functionality. In this case, the pool of services in the repository looks like the following matrix:

vlined 4 Algorithm for Find SaaS LNFCP

initialization Input: dataset including the list of similar services functionality and non-functional descriptions

Set of the minimum of non-functional attributes $min^{i}_{NF} = (min^{1}_{NF}, min^{2}_{NF}, ..., min^{n}_{NF})$ Set of the median of non-functional attributes $median^{i}_{NF} = (median^{1}_{NF}, median^{2}_{NF}, ..., median^{n}_{NF})$ Set of the maximum of non-functional attributes $max^{i}_{NF} = (max^{1}_{NF}, max^{2}_{NF}, ..., max^{n}_{NF})$ Set of the attributes types $nft_i = (nft_1, nft_2, ..., nft_n)$ $nft_i = positive/negative$ Select the non-functional preferences $CP_i = (cp_1, cp_2, ..., cp_n)$

Result: The set of BIS_i and set of WIS_i

for *i in 1:n* **do**

  while $NF_i == cp_i$ **do**

    if $NF_i ==$ *positive* **then**

      Ask for the requirement requested (Lin^{NF}_{i}) **while** Lin^{NF}_{i} *is* $(a, b, c), (a^{'}, b^{'}, c^{'}) and (a^{''}, b^{''}, c^{''})$ **do**

        match BIS = c$^{''}$ $WIS = a^{''}$ **end**

        return(BIS_i) return(WIS_i)

      end

      if $NF_i ==$ *negative* **then**

        Ask for the requirement requested (Lin^{NF}_{i}) **while** Lin^{NF}_{i} *is* $(a, b, c), (a^{'}, b^{'}, c^{'}) and (a^{''}, b^{''}, c^{''})$ **do**

          match BIS = a$^{''}$ $WIS = c^{''}$ **end**

          return(BIS_i) return(WIS_i)

        end

      end

      while $NF_i \neq cp_i$ **do**

        if $NF_i ==$ *positive* **then**

          Lw_i= Not Important at All (NIA) $BIS_i = max^{i}_{NF}$ $WIS_i = min^{i}_{NF}$ return(BIS_i) return(WIS_i)

        end

        if $NF_i ==$ *negative* **then**

          Lw_i = Not Important at All (NIA) $BIS_i = min^{i}_{NF}$ and $WIS_i = max^{i}_{NF}$ return(BIS_i) return(WIS_i)

        end

      end

      return(NF)

    end

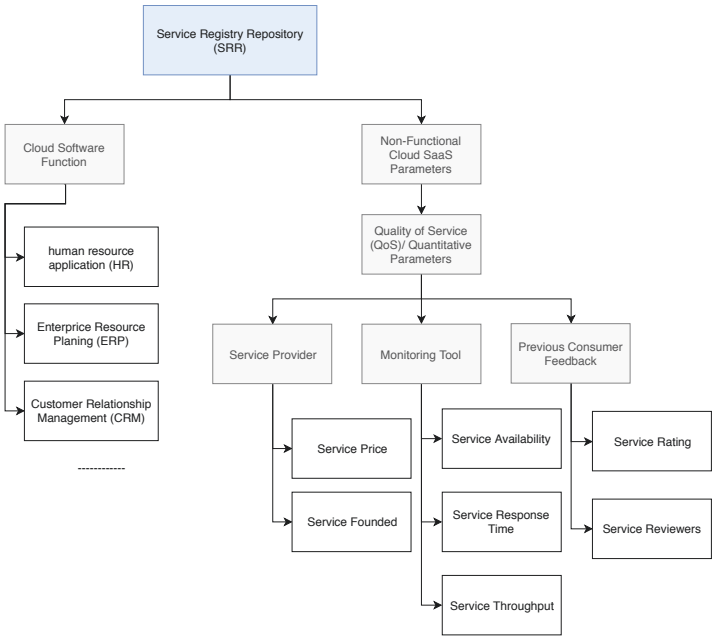


Figure 5.7 : Services Registry Repository

$$\begin{bmatrix} & c_1 & c_2 & c_3 & ... & c_n \\ SaaS_1 & NF_1 1 & NF_1 2 & NF_1 3 & ... & NF_1 n \\ SaaS_2 & NF_2 1 & NF_2 2 & NF_2 3 & ... & NF_2 n \\ ... & ... & ... & ... & ... & ... \\ SaaS_m & NF_m 1 & NF_m 2 & NF_m 3 & ... & NF_m n \end{bmatrix}$$

Each service has multiple non-functional quality attributes represented as follows: $SaaS_1 = \{NF_1, NF_2, ..., NF_n\}$.

The service registry repository classified the non-functional registry into three main resources. Furthermore, we focus on keeping the service registry repository in this section within quantitative parameters. Therefore, the non-functional classification of the service registry is as follows:

(A) **Service Provider**

A service provider publishes its services, along with multiple related non-functional parameters. One of the most important quality attribute is the cost of using the service. Most of the pricing models for Cloud SaaS include monthly/annually subscription per user. Cloud SaaS service providers also offer other options for enterprises version. In addition, most Cloud SaaS service providers offer a one-month free trial version for service consumers before the consumer makes a decision and enters a contract with the service provider. Mostly, the services offered during the month trial come with limited features. A second criterion is the history of a service and when the service was founded. This provides a useful indicator of the quality of the service. A cloud software service with a long history implies a good reputation and instils consumers with more confidence when selecting relevant services. Usually, services with longer lifespans have more user feedback and more training resources available to the public. This does not necessarily mean that relatively new services are less credible: rather, it provides more information in the form of user feedback and ratings. Also, there are other useful quality attributes considered frequently, such as service features, training support, and customer support (Chapter 6). As all of these quality attributes are considered qualitative, it is much easier to define them in terms of good value or otherwise.

(B) **Monitoring Tool**

This includes features that can be collected using publicly available monitoring tools. Several web monitoring tools are available on the Internet, such as up-down, statusOK, and site24x7. All these services are available through subscription and can be used to monitor different resources on the Internet. Most monitoring tools measure the performance of cloud software service, web services, or a specific network by calculating the availability, response time,

and throughput. This research considers three attributes collected by a third party monitoring tool: (1) availability, which refers to the number of successful invocations/total in- vocations and is calculated as a percentage; (2) response time, which refers to the time taken to send a request and receive a response (in milliseconds); and (3) throughput, which refers to the number of invocations for a given period of time.

(C) **Previous Consumer Feedback**

There are many parameters that can be obtained from the feedback of previous consumers, such as performance and support. Due to consumer privacy, many useful parameters cannot be obtained for assessment. However, two important attributes that can be obtained are very useful and can help to gather important information: service ratings and service reviewers. Typically, service ratings are in the range of 1 to 5, where 5 indicates a high rating. Service reviewers refer to the number of previous consumers who assisted in the rating of the service. Usually, services with high popularity have a higher number of service reviewers.

5.5 Ranking System Engine (RSE)

The ranking system engine (RSE) of our service broker is the core component for ranking and sorting the services for consumers. It is responsible for filtering and classifying each service to obtain the most matches with consumer's preferences. The RSE proposed is based on measuring the shortest distance to non-functional consumer preferences.

Lets assume that $BIS = \{bis_1, bis_2, ..., bis_n\}$ is the vector for the best-ideal service and $WIS = \{wis_1, wis_2, ..., wis_n\}$ is the vector for the worst-ideal service. Also, $W = \{w_1, w_2, ..., w_n\}$ is the vector of weighting the importance of quality attributes. These parameters are transmitted by the CRH to the RSE for further

process in order to rank service providers for a service consumer. We introduce the RSE processes in detail as follows:

Step 1: The BIS vector and WIS vector are added to the pool of services. This step is necessary to consider consumer requirements when ranking the services.

Step 2: The normalization step takes place to normalize the dataset and to keep the pool of services in the range [0,1]. It uses equation (5.11) to normalize positive non-functional attributes such as the service availability. Moreover, equation (5.12) is used to normalize the negative non-functional attributes such as the service price. Then, all these parameters are merged as a new normalized dataset.

$$r_ij = \frac{NF_ij}{\sqrt{\sum_{i=1}^{m} NF_ij^2}} \tag{5.11}$$

$$r_ij = \frac{1/NF_ij}{\sqrt{\sum_{i=1}^{m} 1/NF_ij^2}} \tag{5.12}$$

Step 3: To consider the weighting of consumer preferences for each attribute, this step calculates the weighted normalized ratings. Equation (5.13) is used for this purpose. The result of this step is a new dataset that includes all the service parameters with consideration of the weighting of the preferences.

$$v_ij = w_j.r_ij \tag{5.13}$$

where w_j is the weighting value of attribute j, and r_ij refers to the normalized value for service i and non-functional attribute j.

Step 4: To measure the distance to the best-ideal service BIS and to worst-ideal service WIS, Euclidean distance is used in this step. This step is implemented in each service after the previous step. Therefore, to measure the distance to the best-ideal service, equation (5.14) is used. The result of this equation will be one matrix

size $(1 \times n)$, which includes the distance of each service to the best-ideal service. It is called the Distance to Best DB.

$$DB_j = \sqrt{\sum_{i=1}^{n} (v_i j - bis_i^*)^2} \tag{5.14}$$

where $v_i j$ is the weighted normalized non-functional parameters of service i and attribute j. Also, bis_i^* denotes the best-ideal service vector for attribute i.

Equation (5.15) is used to measure the distance to the worst-ideal service WIS. The result of this equation is also one matrix size $(1 \times n)$, called the Distance to Worst DW.

$$DW_j = \sqrt{\sum_{i=1}^{n} (v_i j - wis_i^*)^2} \tag{5.15}$$

where $v_i j$ is the weighted normalized non-functional parameters of service i and attribute j. Also, wis_i^* denotes the worst-ideal service vector for attribute i.

Step 5: Before the last step, in order to rank the service providers, the similarity to the best-ideal service is measured. The DW vector is divided by the summation of the DB vector with DW using equation (5.16). The result is the vector called CSB, size $(1 \times n)$.

$$CSB_j = \frac{DW_j}{DB_j + DW_j} \tag{5.16}$$

Step 6: The last step is ranking and sorting all the services using the CSB vector. It ranks in descending order, which means the service that has the highest similarity will be ranked first.

To describe the ranking system engine approach, algorithm (5) provides pseudocode for the RSE.

vlined 5 Ranking system of cloud services (SaaS) based on measuring the similarity to ideal-service

initialization Input: The pool of similar cloud software services, Set of the best-ideal service vector $BIS_i = (bis_1, bis_2, ..., bis_n)$, Set of the worse-ideal service vector $WIS_i = (wis_1, wis_2, ..., wis_n)$ and Set of weighting the most important of quality attributes $W_i = (w_1, w_2, ..., w_n)$ Output: Ranking cloud services

saas_services(update) = Combine the BIS vector and WIS vector with the pool of services;

for $\forall$ *saas_services(update)* j *in* $1 : m$ **do**

  foreach $NF_i ==$ *positive* **do**

    $normalized_positive_data = \frac{x_i j}{\sqrt{\sum_{i=1}^{m} x_i j^2}}$

  end

  return(normalized_positive_data)

  foreach $NF_i ==$ *negative* **do**

    $normalized_negative_data = \sqrt{\sum_{i=1}^{m} 1/x_i j^2}$

  end

  return(normalized_negative_data)

end

saas_normalized = Combinenormalized_positive_datawithnormalized_negative_data

for $\forall$ *saas_normalized* j *in* $1 : m$ **do**

  $weighted_saas = saas_normalized_{ij} \times w_i$

end

return (weighted_saas)

for $\forall$ *weighted_saas* j *in* $1 : m$ **do**

  $Distance(best)_j = \sqrt{\sum_{i=1}^{n} (weighted_saas_{ij} - bis(weighted_saas)_i)^2}$

end

return(Distance(best))

for $\forall$ *weighted_saas* j *in* $1 : m$ **do**

  $Distance(worse)_j = \sqrt{\sum_{i=1}^{n} (weighted_saas_{ij} - wis(weighted_saas)_i)^2}$

end

return (Distance(worse))

calculate $Similarity_Best_Service_j = \frac{Distance(worse)_j}{Distance(best)_j + Distance(worse)_j}$

Ranking services according to Similarity_Best_Service in descending order;

5.6 Case Study: Ranking the Services of a Computer Repair Shop (Cloud SaaS) using the Find SaaS SNFCP Approach

The purpose of the case study is to illustrate the process of using the Find SaaS SNFCP to select a service provider. The case study includes six Cloud SaaS service providers in the computer repair shop category that were collected from the Capterra website (www.capterra.com). The Capterra website provides a variety of software applications to help individuals and businesses find software solutions. Moreover, site24x7 (www.site24x7.com) is used to measure the response time (RT), availability (AV), and throughput (TH) for these six services. The services were monitored from 21 September 2017 to 4 October 2017. Table 5.4 shows the six software services in the computer repair shop category. It contains service names, web service link, the number of reviewers, service rating, service location, service founded, service price (monthly subscription per user), availability (AV), response time (RT) and throughput (TH). However, the throughput factor does not appear for some services due to the short period over which the services were monitored.

Table 5.4 : Computer repair shop services with non-functional attributes

Service number	Service name	SaaS website link	Number of reviewers	Service Rating	Service location	Founded	Service price	Availability(AV)	Response time (RT)	Throughput (TH)
1	Repair Shopr	http://www.repairshopr.com/	47	5	USA	2010	9.999	100	370	0
2	Repair Pilot	http://www.repairtrackingsoftware.com	5	3.5	UK	N/A	7.94271	99.91	1307	0
3	Open RMA	https://www.openrma.com/	3	4.5	Greece	2016	82.1788	100	2,709	0
4	RSRS	https://www.synolonsoft.com	3	4.5	Cyprus	2010	59.3047	100	7123	0
5	Busy Bench	https://busybench.com/	1	5	USA	2015	9.999	100	969	74
6	Cell Store	https://cellstore.co/	1	5	Canada	2012	29.99	100	660	121

Table 5.5 shows an example of the input data presenting the consumer's preferences for all non-functional attributes.

Table 5.5 : Consumer request parameters

Non-functional (QoS)	Number of reviewers	Service Rating	Service Founded	Service Price	Availability	Response Time	Throughput
Consumer non-functional requirements	20	5	2010	20	100%	20 ms	20 ms
Consumer weighting (w)	Not Important (NI)	Important (I)	Not Important (NI)	Extremely important (EI)	Important (I)	Important (I)	Not Important (NI)

Firstly, the CRH generates the BIS and WIS vectors to the ranking system, as shown in the table 5.6.

Table 5.6 : BIS and WIS service vectors generated by CRH

Quality Attributes	Number of reviewers	Service Rating	Service Founded	Service Price	Availability	Response Time	Throughput
Best-ideal service (BIS)	20	5	2010	20	100	500	20
Worst-ideal service (WIS)	1	3.5	2016	82.179	99.91	7123	0

Also, the CRH interprets the linguistic terms and measures the weighting quality attributes. Table 5.7 shows the fuzzy logic values and the weighting of the importance of non-functional attributes.

Table 5.7 : Weighting quality attributes using fuzzy logic

Quality Attributes	Linguistic terms	Fuzzy Number	Weighting quality attributes
Number of reviewers	(NI)	(0.1,0.3,0.5)	0.07563
Service Rating	(I)	(0.5,0.7,0.9)	0.17647
Service Founded	(NI)	(0.1,0.3,0.5)	0.07563
Service Price	(EI)	(0.9,1.0,1.0)	0.24370
Availability	(I)	(0.5,0.7,0.9)	0.17647
Response Time	(I)	(0.5,0.7,0.9)	0.17647
Throughput	(NI)	(0.1,0.3,0.5)	0.07563

As shown in table (5.7), the total of the weighting quality attributes is equal 1 ($\sum W = 1$).

After this, the RSE ranks all the services based on the consumer's request. Therefore, the ranking system process as described in algorithm (5), is applied to rank all the services.

Step 1: Add BIS and WIS service vectors to the pool of services.

Step 2: Generate the normalization matrix using equations (5.11) and (5.12). Table 5.8 shows the normalization matrix for the services.

Table 5.8 : Normalization matrix for the services

Service Name	Number of reviewers	Service Rating	Service Founded	Service Price	Availability	Response Time	Throughput
Repair Shopr	0.912148599	0.389249472	0.353154264	0.499753073	0.353632931	0.688123504	0
Repair Pilot	0.097037085	0.27247463	0.352803216	0.62913426	0.353314662	0.194801604	0
Open RMA	0.058222251	0.350324525	0.35210321	0.060806814	0.353632931	0.093985122	0
RSRS	0.058222251	0.350324525	0.353154264	0.084260286	0.353632931	0.035744166	0
Busy Bench	0.019407417	0.389249472	0.352277951	0.499753073	0.353632931	0.262750977	0.516624253
Cell Store	0.019407417	0.389249472	0.352803216	0.16662324	0.353632931	0.385766207	0.844750468

Step 3: Calculate the weighted normalized matrix, taking into account the weighting non-functional quality attributes. Using equation (7.2), the weighted normalization matrix is shown in table 5.9.

Table 5.9 : The weighting normalization matrix for the services

Service Name	Number of reviewers	Service Rating	Service Founded	Service Price	Availability	Response Time	Throughput
Repair Shopr	0.068986028	0.068691083	0.026709146	0.121788564	0.062405811	0.121433559	0
Repair Pilot	0.007338939	0.048083758	0.026682596	0.153318433	0.062349646	0.034376754	0
Open RMA	0.004403364	0.061821975	0.026629655	0.014818467	0.062405811	0.01658561	0
RSRS	0.004403364	0.061821975	0.026709146	0.020534019	0.062405811	0.006307794	0
Busy Bench	0.001467788	0.068691083	0.02664287	0.121788564	0.062405811	0.046367819	0.039072423
Cell Store	0.001467788	0.068691083	0.026682596	0.040605664	0.062405811	0.068076389	0.063888691

Step 4: Measure the distance to the best-ideal service DB using equation (5.14), as well as the distance to the worst-ideal service DW using equation (5.15). Table 5.10 shows the values of DB and DW for each service.

Table 5.10 : The values of DB and DW

Service Name	DB	DW
Repair Shopr	0.079925162	0.172278656
Repair Pilot	0.112441303	0.141437584
Open RMA	0.090957978	0.01740669
RSRS	0.096907984	0.015166859
Busy Bench	0.084802863	0.122469151
Cell Store	0.067140985	0.094798238

Step 5: Measure the similarity to the best-ideal service CSB for each service to obtain the most similar service to the consumer's request. Using equation (5.16), calculate the values of CSB for each service. Table 5.11 shows the values of CSB for each service.

Table 5.11 : The values of CSB for each service

Service Name	CSB
Repair Shopr	0.683092972823959
Repair Pilot	0.55710652244561
Open RMA	0.160630676869119
RSRS	0.1353279555534
Busy Bench	0.590861972758796
Cell Store	0.585393929101185

Step 6: Rank all the services using the CSB. The highest value of CSB for the service ranks first, which means it has the most similarity to the consumer's request. Table 5.12 shows the ascending order for all services beginning with the service that has the highest value of CSB.

Table 5.12 : Ranking result of consumer request

Service Name	CSB
Repair Shopr	0.683092972823959
Busy Bench	0.590861972758796
Cell Store	0.585393929101185
Repair Pilot	0.55710652244561
Open RMA	0.160630676869119
RSRS	0.1353279555534

As shown in table 5.12, the service Repair shopr ranks first with a similarity value of (0.683092972823959), which means it is very close to the consumer's preferences by 68%. Moreover, the service RSRS ranks last, which means it is the service that least matches the consumers preferences by 13%.

5.7 Experiments

We use the R language version 1.1.383 to test and evaluate the Find SaaS (SNFCP, M2NFCP and LNFCP) proposed approaches. The experiment was run on a MAC operating system. We develop our algorithm to understand the consumer's request, and then rank and sort all the service providers using our proposed approaches and compared the efficiency with two existing ranking approaches as a benchmark.

The existing ranking approaches to which we compare our results are as follows:

1. A consensus-based approach for selecting a cloud plan (CASCP) was proposed by (Arman et al., 2016). This approach uses the TOPSIS method to rank services based on a positive-ideal solution and negative-ideal solution. Ching-

Lai Hwang and Yoon introduced the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) in 1981 (Yoon & Hwang, 1995). This approach is similar to our ranking mechanism, except that only the consumer's preferences are taken into account in relation to the weighting the importance of non-functional attributes.

2. Badidi (2013) proposed the framework for software as a service selection and provisioning (FSSP) (Badidi, 2013). This approach aggregates the utility function for each service provider. This method has been applied in many existing systems approaches such as (Garg, Versteeg, & Buyya, 2013). Like the TOPSIS method, the utility function only considers weighting non-functional attributes when computing the utility value.

To measure the performance of Find SaaS (SNFCP, M2NFCP and LNFCP) against the other ranking approaches, the mean average precision mAP, as well as the normalized discounted cumulative gain $nDCG$, are examined. The $nDCG$ metric is widely used to evaluate the ranking system. $nDCG$ works by providing the ideal descending services ranking and predicting descending services ranking. The $nDCG$ is then calculated for the top-k using equation (5.17). However, the $nDCG$ metric has only been used to determine the performance of the ranking service approach with the CRM dataset. CRM is a small dataset collected by Capterra web site, and it includes top 20 service providers that offer CRM functionality. In addition, the mAP metric is used to measure the performance of our system and compared with other existing approaches by using the QWS dataset (See section 5.7.2) dataset.

$$nDCG_k = \frac{DCG_k}{iDCG_k} \tag{5.17}$$

$$DCG_k = \sum_{j=1}^{k} \frac{rel_j}{log(1 + R_j)} \tag{5.18}$$

where DCG_k is the discounted cumulative values of the top-k for the predicted service ranking system, while $iDCG$ is the ideal value of the top-k for ideal service ranking. The value of DCG_k can be calculated using equation (5.18).

On the other hand, the mAP is used to measure the accuracy of the object detectors, which is the average of the maximum precision at different recall values. The mAP metrics are used to evaluate Find SaaS against the other ranking approaches for the second dataset which is the QWS. To measure the mAP in the top-K services, we use the following equation:

$$mAP_k = \frac{1}{k} \sum_{j=1}^{k} P_j \tag{5.19}$$

Where P_j is the precision value for the service j.

$$mAP_k = \frac{1}{k} \sum_{i=1}^{k} \frac{1}{i} \sum_{r=1}^{i} rel_r \tag{5.20}$$

To calculate the relevant ranking services, Euclidean distance is used to calculate the distance between the consumer non-functional requirements with each service in the dataset. To calculate this, equation (5.21) is used.

$$d(cr, NF) = \sqrt{\sum_{i=1}^{n} (cr_i - NF_i)} \tag{5.21}$$

5.7.1 CRM Dataset

Dataset description

One of the most common software service available to the public are CRM services. There are a number of CRM service providers: we have collected the top-20 services based on the Capterra website. In the dataset (5.13), 13 columns describe each service characteristic. These columns are as follows: Service name, Supporting mobile (answers Yes/No), Free trial version (answers Yes/No), Service Rating, Service Reviewers, Service Price (based on monthly rate per person), language, Service Location, Year business was founded, web link, Service Availability, Service Response Time, and Service Throughput. We have gathered this information from the CRM providers website, and by Capterra web site. In addition, the site24x7 monitoring tool is used to obtain service availability, service response time and service throughput for each service. Figure (5.8) shows the information collected by the site24x7 monitoring tool for the (Apptivo) CRM service. During September, the Apptivo had no accident down-times with 100 availability, 255 ms response time, and 555 KB/sec Throughput. In addition to this, site24x7 has the capability to choose the location of monitoring servers to ping the service; six servers have been used, as shown in Figure (5.9).

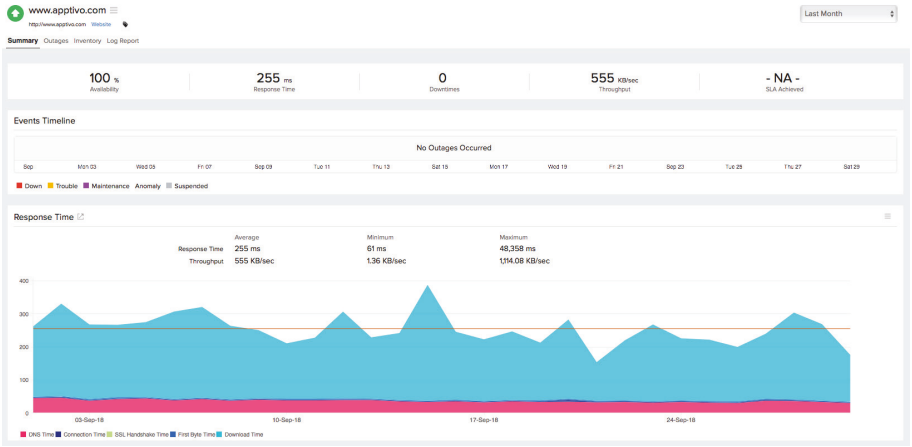


Figure 5.8 : Site24x7 analysis for (Apptivo) service

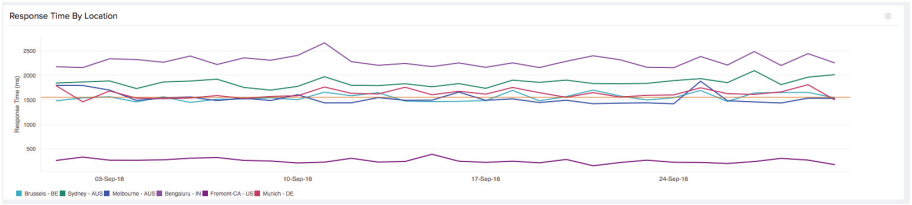


Figure 5.9 : The response time analysis by the list of servers

Table 5.13 : Top 20 services of CRM by Capterra web site

Number	Service Name	Supporting mobile	Free trial version	Review	Number of reviewers	Price	language	Location	Founded	Web link	Availability	Response Time	Throughput
1	Salesforce	Yes	Yes	4	5869	25	English	USA	1999	www.salesforce.com	100	583	681
2	Zoho	Yes	Yes	4	881	12	English	USA	1996	www.zoho.com	100	420	104
3	Odoo	Yes	Yes	4	110	20	English	USA	2002	www.odoo.com	99.86	1499	35
4	Microsoft Dynamics	Yes	Yes	4	1545	65	English	USA	1975	www.microsoft.com	100	335	596
5	OroCRM	Yes	Yes	5	7	295	English	USA	2012	www.oroinc.com/	99.92	508	118
6	HubSpot CRM	Yes	Yes	4.5	865	50	English	USA	2006	www.hubspot.com/	100	360	1194
7	Act!	Yes	Yes	4	182	10	English	USA	1987	www.swiftpage.com	100	183	199
8	Pipedrive	Yes	Yes	4.5	1317	15	English	USA	2010	www.pipedrive.com	99.81	673	126
9	Maximizer	Yes	Yes	4	49	60	English	Canada	1987	www.maximizer.com	100	1452	52
10	SAP	NA	NA	4	58	60	English	Germany	1972	www.sap.com	100	362	1345
11	Insightly	Yes	Yes	4	245	29	English	USA	2010	www.insightly.com	100	1597	37
12	Oracle Sales Cloud	NA	NA	4	43	600	English	USA	1977	www.oracle.com	100	174	345
13	Apptivo CRM	Yes	Yes	4.5	167	10	English	USA	2010	www.apptivo.com	100	255	555
14	Infusionsoft	Yes	Yes	4	533	99	English	USA	2001	www.infusionsoft.com	100	815	91
15	SugarCRM	Yes	Yes	4	183	40	English	USA	2004	www.sugarcrm.com	100	654	43
16	SuiteCRM	Yes	Yes	4	10	0	English	UK	2006	www.suitecrm.com	100	3230	30
17	Sage CRM	Yes	Yes	3	25	45	Enlish	UK	1981	www.sagesoftware.com	99.39	1839	79
18	Get Base	Yes	Yes	4.5	93	45	English	USA	2008	www.getbase.com	100	229	517
19	Salesboom	Yes	Yes	4.5	22	14	English	Canada	2003	www.salesboom.com	100	352	262
20	bpm'online	Yes	Yes	5	67	25	English	UK	2002	www.bpmonline.com/	100	1009	214

As shown in figure (5.9), the server in Fremont-CA in the US has the best measurement of response time, since the data center of the Apptivo service is located in the USA. The server Bengaluru has the worst response time, since the speed link of the connection between these two countries may be affected by distance, time of ping, and other factors.

Experiment setup

In order to test and evaluate Find SaaS (SNFCP, M2NFCP and LNFCP) ranking approaches using the CRM dataset, three queries for each proposed approach were conducted. The first query considers two non-functional qualities preferences, and the second query considers four non-functional qualities preferences. The third and final query considers six non-functional qualities preferences. In addition, we tested our ranking methods with seven non-functional quality attributes as follows: service rating, service reviewers, service price (monthly subscription per user), service founded, availability, response time and throughput.

Table 5.14 : Find SaaS SNFCP queries input

Consumer request (weight)	Service Reviewers	Service Rating	Service Price	Service Founded	Availability	Response time	Throughput
1	N/A	5 (VI)	30 (EI)	N/A	N/A	N/A	N/A
2	N/A	N/A	30 (EI)	N/A	100 (SI)	200 (SI)	N/A
3	500 (I)	5(EI)	30 (I)	2000 (SI)	100 (EI)	200 (I)	N/A

Table 5.14 shows the parameters of the queries for service selection approach using the Find SaaS SNFCP approach. For example, in the first query, the ranking request considers two preferences for ranking the services: the service price and the service rating. Therefore, the CRH will obtain two parameters ($5_{VI}^{Rating} \quad AND \quad 30_{I}^{price}$). This means the consumer considers a service rating of 5 out of 5 as a Very Important priority and a service price of $30 as an Important priority.

Table 5.15 shows the system queries requested to Find SaaS M2NFCP. It is ev-

Table 5.15 : Find SaaS M2NFCP queries input

Consumer request (weight)	Service Reviewers	Service Rating	Service Price	Service Founded	Availability	Response time	Throughput
1	8-50 (I)	3-5 (EI)	N/A	N/A	N/A	N/A	N/A
2	50-800 (SI)	4-5 (EI)	10-50 (EI)	2000-2010 (I)	N/A	N/A	N/A
3	50-4000 (SI)	4-5 (EI)	10-100 (EI)	2000-2010 (I)	N/A	200-1000 (EI)	1000-1345 (SI)

ident that the requesting ranking services are based on the range. Therefore, the ranking system will consider the minimum and maximum requirements for each non-functional attributes to obtain the best result. For example, in the first query, the requesting ranking services considers the number of reviewers attributes with a range of (8 to 50) reviewers in (Important) weighting attribute, and the service rating concerns with a range of (3 to 5) in (Extremely Important) weighting attribute. Therefore, the consumer request handler will get two parameters $(8^{reviewers} - 50^{reviewers})_I \quad AND \quad (3^{rating} - 5^{rating})_{EI}$.

Table 5.16 : Find SaaS LNFCP queries input

Consumer request (weight)	Service Reviewers	Service Rating	Service Price	Service Founded	Availability	Response time	Throughput
1	HR (I)	N/A	LP (EI)	N/A	N/A	N/A	N/A
2	VHR (I)	HR (SI)	AP (EI)	MD (SI)	N/A	N/A	N/A
3	MR (I)	MR (I)	LP (EI)	OS (EI)	HA (SI)	LRT (SI)	N/A

Table 5.16 shows the queries requested to Find SaaS LNFCP approach. It is evident that each ranking request includes the linguistic terms to describe the requirement for the non-functional attributes, as well as the linguistic terms for weighting the importance of qualities. Regarding table 5.17, the CRH, based on the fuzzy interpreter, generates the fuzzy number for each requirement in order to obtain the crisp values representing the consumer request requirements.

As shown in the table 5.18, the fuzzy number describes each ranking query of the table 5.16. For example, in the first query, the requested ranking service providers are based on two preferences ($HR_I^{Reviewers} \quad AND \quad LP_{EI}^{Price}$). Therefore, in this

Table 5.17 : Summary of CRM dataset

Service reviewers	Service rating	Service founded	Service Price	Availability	Response Time	Throughput
Min. : 1.0	Min. :3.50	Min. :2010	Min. : 7.94	Min. : 99.9	Min. : 370	Min. : 0.0
1st Qu.: 1.5	1st Qu.:4.50	1st Qu.:2010	1st Qu.:10.00	1st Qu.:100.0	1st Qu.: 737	1st Qu.: 0.0
Median : 3.0	Median :4.75	Median :2012	Median :19.99	Median :100.0	Median :1138	Median : 0.0
Mean :10.0	Mean :4.58	Mean :2012	Mean :33.24	Mean :100.0	Mean :2190	Mean : 32.5
3rd Qu.: 4.5	3rd Qu.:5.00	3rd Qu.:2014	3rd Qu.:51.98	3rd Qu.:100.0	3rd Qu.:2358	3rd Qu.: 55.5
Max. :47.0	Max. :5.00	Max. :2016	Max. :82.18	Max. :100.0	Max. :7123	Max. :121.0

Table 5.18 : The fuzzy number of ranking request to Find SaaS LNFCP

Service reviewers	Service rating	Service Price	Service Founded	Availability	Response time	Throughput
(3,4.5,47)	(N/A)	(0,7.94,19.99)	(N/A)	(N/A)	(N/A)	(N/A)
(4.5,47,47)	(4.75,5,5)	(7.94,19.99,82.18)	(2010,2012,2015)	(N/A)	(N/A)	(N/A)
(1,3,4.5)	(3.5,4.75,5)	(0,7.94,19.99)	(0,2010,2012)	(100,100,100)	(0,370,1138)	(N/A)

case, the service reviewers include the fuzzy number (3,4.5,47) and service price (0,7.94,19.99). Thus, the CRH interprets these fuzzy numbers and translates them to a crisp number for further analysis. For the non-functional attributes describes in (N/A), the CRH generate the BIS and WIS based on the minimum and the maximum of these attributes with (Not Important at All -NIA).

Experiment Results

We studied the proposed approaches based on the top (5,10,15 and 20) service providers generated by our system. The evaluation performance of our proposed methods is compared with other existing approaches, FSSP and CASCP, using the $nDCG$ metric. The experimental results are presented in Table 5.19. For further details, the Appendix A part of this thesis shows the service ranking tables for each request. The key findings can be summarized as follows:

1. Within the two of non-functional preferences, we observe that the accuracy in

terms of $nDCG$ metric decrease as N is increased from 5 to 20 services.

2. However, and more critical for Find SNFCP and Find M2NFCP. We observe the following:

   (a) The accuracy level of Find SaaS SNFCP is much higher than both FSSP and CASP approaches in terms of nDCG metric.

   (b) For example, for the two non-functional preferences, the accuracy of Find SaaS SNFCP is approximately 50% higher than CASCP and nearly 60% higher than FSSP. We see a similar trend of higher Find SaaS SNFCP across $N = 5, 10, 15 and 20$ top service providers.

   (c) Similarly to (a), the accuracy level of Find SaaS M2NFCP is much higher than both CASCP, and FSSP approaches.

   (d) For example, for the two non-functional preferences with N= 5, the accuracy of Find SaaS M2NFCP is higher than FSSP and CASCP by approximately 30%. We see a similar trend across four and six non-functional preferences.

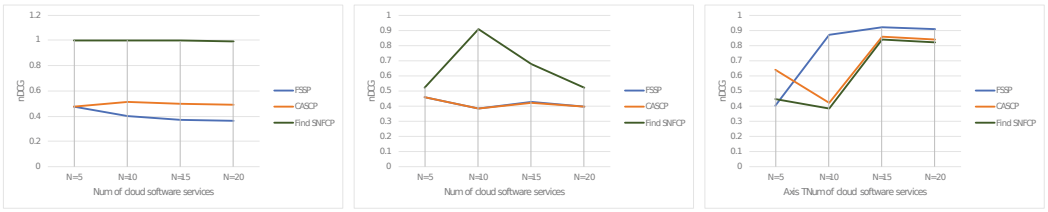
3. In contrast to (3) above, generally speaking, the accuracy of Find SaaS LNFCP is comparable to FSSP and CASCP with conflicting findings. For example, for N= 20; the accuracy of Find SaaS LNFCP is 46.9% while that of CASCP is 51%. Find SaaS LNFCP is 5% lower than CASCP. On the contrary, compared to FSSP 41%, Find SaaS LNFCP is 5% higher. However, for the four non-functional preferences, FSSP accuracy is the best followed by Find SaaS LNFCP and least FSSP. Conversely, with six attributes (extremely case), Find LNFCP is the best followed by CASCP then FSSP.

4. It is worth noting that ranking service providers in terms of matching the non-functional parameters that consider more than four preferences is still

challenging. However, if a consumer requests ranking services by quantifying the consumer non-functional requirements, they may get some services that match their requirements, or they may not. It depends on the pool of services list.

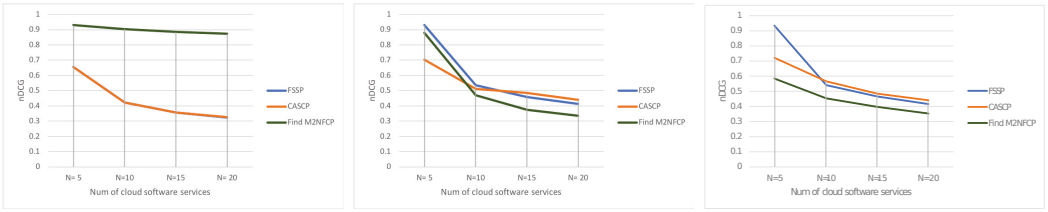
5. In general, the *nDCG* metric of all proposed approaches decreases with the increasing number of preference concerns when requesting a service provider for selection.

6. In general, the *nDCG* metric decrease with the increasing number of services.



(a) nDCG within two preferences (b) nDCG within four preferences (c) nDCG within six preferences

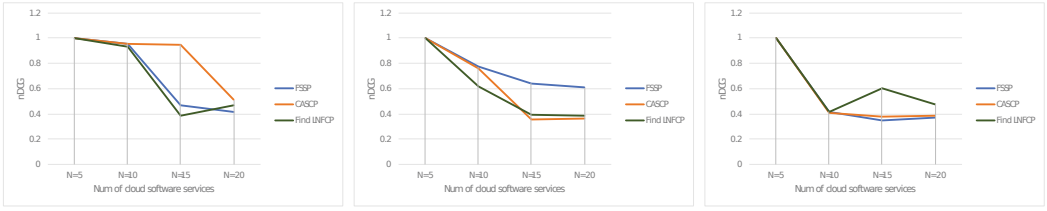
Figure 5.10 : nDCG result for Find SaaS SNFCP, FSSP and CASCP approaches



(a) nDCG within two preferences (b) nDCG within four preferences (c) nDCG within six preferences

Figure 5.11 : nDCG result for Find SaaS M2NFCP, FSSP and CASCP approaches

The results of the experiment for *nDCG* matrices for Find SaaS SNFCP compared with the FSSP and *CASCP* approach are shown in Figure (5.10). The result



(a) nDCG within two preferences (b) nDCG within four preferences (c) nDCG within six preferences

Figure 5.12 : nDCG result for Find SaaS LNFCP, FSSP and CASCP approaches

proves that Find SaaS SNFCP, in terms of considering two and four preferences, outperforms the other approaches using different given number of services with respect to $nDCG$. Figure (5.10a,5.10b) demonstrates that Find SaaS SNFCP is higher than other approaches by approximately 50% when two and four preferences are used for ranking the services. On the other hand, Figure (5.10c) shows that Find SaaS SNFCP underperforms in comparison to other approaches where there are six conflicting preferences.

Figure (5.11) shows that the performance of Find SaaS M2NFCP compared with FSSP and $CASCP$ using the $nDCG$ metric. The result proves that ranking the services by considering two preferences outperforms other existing approaches, as shown in Figure (5.11a). Regarding four preferences, as shown in Figure (5.11b, the Find SaaS M2NFCP outperforms the other approaches with respect to the top-5 services. Moreover, when six conflicting preferences are used to rank the services, Find SaaS M2NFCP it outperformed by the FSSP and $CASCP$ approaches.

The results in Figures (5.12a, 5.12b and 5.12) show the performance of Find SaaS LNFCP compared with other approaches using the $nDCG$ metric. In general, the Find SaaS LNFCP ranking of the top-5 services produces similar results as the other approaches.

Table 5.19 : CRM result analysis

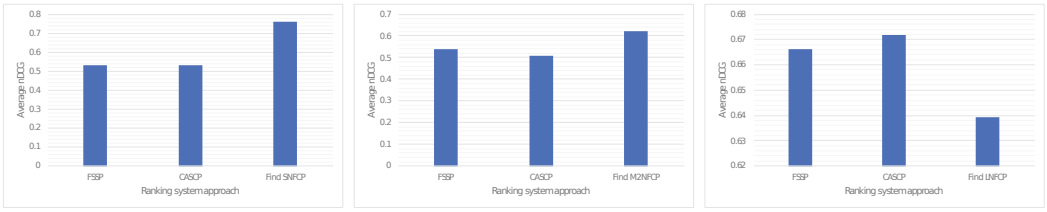
Approach	2- Attributes				4- Attributes				6- Attributes			
	N= 5	N=10	N=15	N= 20	N= 5	N=10	N=15	N= 20	N= 5	N=10	N=15	N= 20
FSSP	0.4717532	0.3971027	0.3687536	0.3624668	0.4556533	0.3825136	0.4255493	0.3968845	0.4037312	0.8699604	0.9160177	0.9040686
CASCP	0.4717532	0.5154495	0.4989939	0.493587	0.4556533	0.3825129	0.4237599	0.3956011	0.6360104	0.4223828	0.8579019	0.8373529
Find SNFCP	1	0.9997387	0.9974817	0.9871743	0.5211027	0.9083838	0.6749247	0.5222999	0.4443679	0.381437	0.8376763	0.8194429

Approach	2- Attributes				4- Attributes				6- Attributes			
	N= 5	N=10	N=15	N= 20	N= 5	N=10	N=15	N= 20	N= 5	N=10	N=15	N= 20
FSSP	0.6537042	0.422419	0.3568053	0.3247566	0.9293594	0.5357732	0.458215	0.4137461	0.9293594	0.536337	0.4628558	0.417706
CASCP	0.6537042	0.4222528	0.3569805	0.3249625	0.7018146	0.5104939	0.4843049	0.4394452	0.720443	0.5648681	0.4852642	0.4386697
Find M2NFCP	0.929677	0.9038872	0.8848709	0.8719815	0.8784734	0.4710346	0.3738221	0.3357397	0.5834047	0.4532713	0.3974053	0.3507552

Approach	2- Attributes				4- Attributes				6- Attributes			
	N= 5	N=10	N=15	N= 20	N= 5	N=10	N=15	N= 20	N= 5	N=10	N=15	N= 20
FSSP	0.9999873	0.9551955	0.4643594	0.4150233	0.9999873	0.7708342	0.6411425	0.6088441	1	0.4181812	0.3516224	0.3699979
CASCP	0.9999873	0.9551955	0.9478496	0.5109682	0.9999873	0.7617656	0.3554693	0.3608608	0.9994713	0.4106557	0.3753058	0.3820629
Find LNFCP	0.9999873	0.9283363	0.3892694	0.4698207	0.9999873	0.6154092	0.3913145	0.3866236	0.9992689	0.4123749	0.6033231	0.4768312

Experiment Discussion

Cloud SaaS service selection with respect to consumer non-functional preferences obtains different results compared to those that only consider weighting non-functional attributes. Figure (5.13) shows the mean value of nDCG value for all ranking requests regarding Find SaaS (SNFCP, M2NFCP and LNFCP) compared with FSSP and CASCP. The figure proves that FSSP and CASCP obtain a similar level of performance since FSSP uses the SAW for the ranking engine and CASCP uses the TOPSIS mathematical approach to rank the services.



(a) Average nDCG for Find SaaS SNFCP and compared with the existing approach

(b) Average nDCG for Find SaaS M2NFCP and compared with the existing approach

(c) Average nDCG for Find SaaS LNFCP and compared with the existing approach

Figure 5.13 : The average nDCG for our proposed approaches and compared with the existing research approaches

Figure (5.13a) shows the bar chart of average value for Find SaaS SNFCP, FSSP, and CASCP for all queries requested. It demonstrates that the average of Find SaaS SNFCP is 77%; FSSP and CASCP are similar, with approximately 53%.

Figure (5.13b) shows the bar chart of the average of Find SaaS M2NFCP with FSSP and CASCP. It demonstrates that the average of Find SaaS M2NFCP is 62% compared with FSSP and CASCP, which have approximately 52%.

Finally, Figure (5.13c) shows the average of Find SaaS LNFCP compared with

FSSP and CASCP. It demonstrates that the mean value of our proposed approach is less than the other approaches. The average of Find SaaS LNFCP is 64% compared with 67% in FSSP and 68% in CASCP.

In conclusion, CRM is a useful dataset on which to test and evaluate any service selection module of Cloud SaaS service provider considering multiple conflicting non-functional preferences. A future direction could be to conduct more experiments to consider smaller ranges of consumer preferences, such as a price range between $25 and $26. In addition, more tests need to be conducted to investigate non-functional dependency. For example, if a consumer needs a service with superior advantages without being concerned about the cost. Therefore, the future research direction of Cloud SaaS service selection involves the dependency of multiple preferences.

5.7.2 QWS Dataset

Dataset Description

This section details the experiment using the QWS dataset to evaluate the proposed ranking approaches. Al-Masri and Mahmoud published the QWS dataset (Al-Masri & Mahmoud, 2007). This dataset contains multiple non-functional attributes with approximately 2,506 web services. We generated a random price range from 20 to 40 in order to get the service price to involve with the experiment result. Therefore, there is a total of 10 Attributes, defined in the table 5.20.

Experiment Setup

In order to test and evaluate our ranking services approaches, a list of queries is generated to obtain a ranking result. Find SaaS LNFCP has not been tested on the QWS dataset, since this kind of request requires linguistic values to quantify the consumer's non-functional requirements. Therefore, Find SaaS SNFCP and Find SaaS M2NFCP have been only tested on this dataset. The mean average preci-

Table 5.20 : Definitions of QWS non-functional attributes

Non-functional attributes	Measurement	quality type	Definition
Price	Dollar	Negative	Generated randomly to be simulated as the price of each service.
Response time	Ms	Negative	The duration time required to send a request and receive a response.
Availability	%	Positive	The average of all total invocations.
Throughput	Ips	Positive	The number of invocations for the period of time.
Successability	%	Positive	The average of the responses for the number of requests.
Reliability	%	Positive	The ratio of error messages to the total number of messages.
Compliance	%	Positive	The percentage of the criterion that matches the WSDL description.
Best practice	%	Positive	The percentage of the criterion that matches the WS-I basic profile..
Latency	Ms	Negative	The processing time required for a given request.
Documentation	%	Positive	The amount of documentation of the WSDL file.

sion (mAP) is the evaluation metric used to evaluate the proposed approach, and the other ranking approaches. The study compares the performance of Find SaaS (SNFCP and M2NFCP) against the framework for software as a service selection and provisioning (FSSP), the consensus-based approach for selecting a cloud plan (CASCP). In addition, we compared the result with other ranking approaches based on selected non-functional as preferences such as according to price, so the services will be ranked in ascending order of price from lowest to highest. Table 5.21 summarizes the QWS dataset that required the ranking request should be within the range of each non-functional attributes.

Table 5.21 : Summary of the QWS dataset

Price	Response time	Availability	Throughput	Successability	Reliability	Compliance	Best practice	Latency	Documentation
Min. :20.00	Min. : 37.0	Min. : 7.00	Min. : 0.100	Min. : 8.00	Min. :33.00	Min. : 33.00	Min. :50.00	Min. : 0.25	Min. : 1.00
1st Qu.:27.00	1st Qu.: 142.3	1st Qu.: 75.00	1st Qu.: 2.800	1st Qu.: 76.00	1st Qu.:67.00	1st Qu.: 78.00	1st Qu.:75.00	1st Qu.: 4.60	1st Qu.: 6.00
Median :35.00	Median : 226.6	Median : 87.00	Median : 6.900	Median : 95.00	Median :73.00	Median : 89.00	Median :82.00	Median : 11.85	Median :11.00
Mean :34.91	Mean : 383.9	Mean : 81.14	Mean : 9.036	Mean : 83.88	Mean :69.78	Mean : 88.44	Mean :79.31	Mean : 54.61	Mean :31.32
3rd Qu.:43.00	3rd Qu.: 348.8	3rd Qu.: 93.00	3rd Qu.:13.300	3rd Qu.: 98.00	3rd Qu.:73.00	3rd Qu.:100.00	3rd Qu.:84.00	3rd Qu.: 44.61	3rd Qu.:42.00
Max. :50.00	Max. :4989.7	Max. :100.00	Max. :43.100	Max. :100.00	Max. :89.00	Max. :100.00	Max. :95.00	Max. :4140.35	Max. :97.00
Negative Type	Negative Type	Positive Type	Positive Type	Positive Type	Positive Type	Positive Type	Positive Type	Negative Type	Positive Type

Table 5.22 displays the service queries that were used to test Find SaaS SNFCP.

The table shows five ranking requests based on selected quality attributes. For example, in the first ranking request, a consumer prefers a service price of $25 a month, with the weighting of Extremely Important, and a service response time of 50 ms, with the weighting of Important. The other non-functional attributes are Not Important at All, so the ranking system weights these attributes as Not Important at All. The other ranking queries take into account more non-functional preferences. For example, the second query considers four preferences; the third service query considers six preferences, and so on.

Table 5.22 : The service queries for the Find SaaS SNFCP approach

Experiment	Price	Response time	Availability	Throughput	Successability	Reliability	Compliance	Best practice	Latency	Documentation
1	25 (EI)	50 (I)	NIA	NIA	NIA	NIA	NIA	NIA	NIA	NIA
2	NIA	NIA	100 (EI)	42 (I)	100 (VI)	NIA	NIA	NIA	5 (I)	NIA
3	25 (EI)	50 (I)	100 (I)	30 (I)	100 (I)	NIA	NIA	NIA	NIA	90 (I)
4	22 (EI)	100 (EI)	100 (EI)	20 (I)	100 (I)	80 (I)	NIA	80 (EI)	NIA	90(VI)
5	20 (SI)	100(EI)	100 (EI)	42 (I)	100 (I)	80 (SI)	100 (SI)	80 (I)	50 (SI)	90 (SI)

Table 5.23 shows the service queries to test and evaluate the ranking approach of Find SaaS M2NFCP against other ranking systems. The table shows five service queries which take into consideration a different number of preferences ranging from two to eight. For example, in the first service request, a consumer considers a service price in the range of $25 to $30 with the weighting of "Extremely Important", and response time in the range of 50 ms to 100 ms with the weighting "Important". The other preferences are Not Important at All. Therefore, the system is weighted Not Important at All. The next service requests consider more and more non-functional preferences.

Experiment Result

This section explained the results of our proposed approaches Find SaaS (SNFCP and M2NFCP) and compared the results with other existing approaches, FSSP and

Table 5.23 : The service queries for the Find SaaS M2NFCP approach

Experiment	Price	Response Time	Availability	Throughput	Successability	Reliability	Compliance	Best practice	Latency	Documentation
1	25 - 30 (EI)	50- 100 (I)	NIA	NIA	NIA	NIA	NIA	NIA	NIA	NIA
2	NIA	NIA	90-100 (EI)	30-42 (I)	90-100 (VI)	NIA	NIA	NIA	5-100 (I)	NIA
3	25-30(EI)	50 -100(I)	88-100 (I)	30-40 (I)	80-100 (I)	NIA	NIA	NIS	NIA	80-90 (I)
4	22 -40(EI)	100-200 (EI)	90-100 (EI)	20-30 (I)	80-100 (I)	70-80 (I)	NIA	80 -90(EI)	NIA	80-95(VI)
5	30-40(I)	100-500(EI)	90-100 (EI)	20-42 (I)	70-100 (I)	60-80 (SI)	80-100 (SI)	80-90 (I)	50-500 (SI)	70-90 (SI)

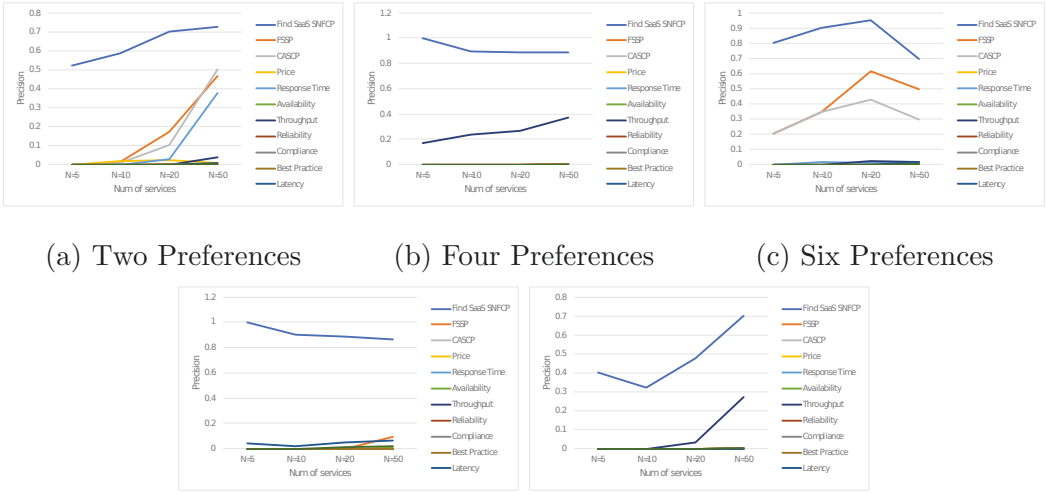
CASCP. We also compared our proposed method by ranking the services based on consumer's preferences, such as ranking the services beginning with the lowest price to the highest price. We studied the mean average precision mAP for ranking services on top (5,10, 20, and 50) in order to measure the performance of the ranking approaches. The results of Find SaaS SNFCPs performance compared with FSSP and CASCP are shown in Figure(5.14). The main findings can be summarized as follows:

1. In general, Find SaaS SNFCP outperforms the ranking system approaches. The average precision of Find SaaS SNFCP is approximately 75%, whereas the FSSP has 12% and CASCP has 9%. Therefore, ranking services by considering non-functional attribute preferences by quantifying service requirements of particular attributes is crucial to determine the service that matches with the consumer non-functional preferences.

2. Ranking service providers based on attributes such as service price is not the best way to match a service with a consumer's preferences.

3. The results in Figures (5.14a,5.14b,5.14c and 5.14d) show that the increase the number of services considered generally has a positive impact on the accuracy of the proposed Find SaaS SNFCP approach in terms of precision.

Moreover, Figure (5.15) illustrates the precision metric for Find SaaS M2NFCP



(a) Two Preferences (b) Four Preferences (c) Six Preferences

(d) Eight Preferences (e) Ten Preferences

Figure 5.14 : The mean average precision (mAP) for Find SaaS SNFCP in the top-(5,10,20 and 50) and compared with the existing research approaches

compared with other ranking system approaches. We can summarize the main findings into the following points:

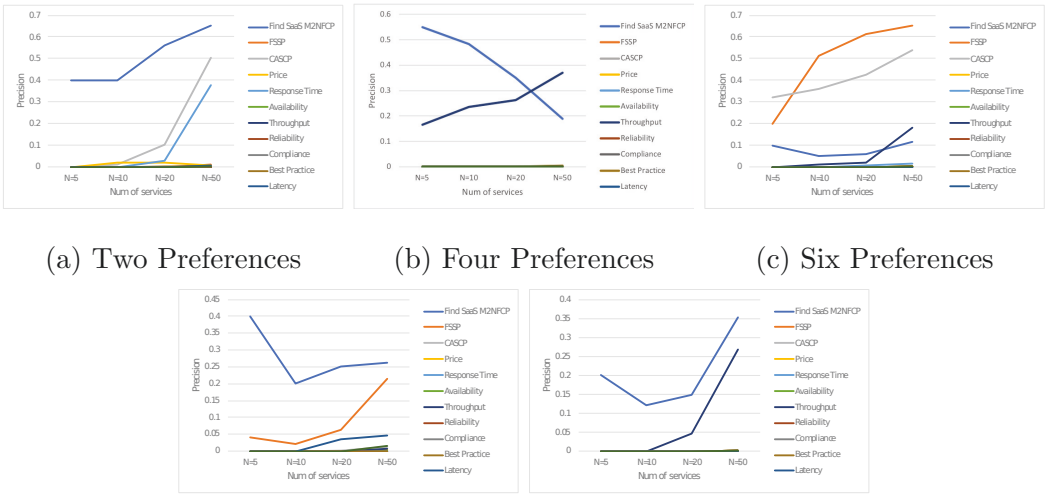
1. In general, ranking services by Find SaaS M2NFCP outperforms other ranking approaches.

2. The precision of Find SaaS M2NFCP decreases in cases where the consumer considers more non-functional preferences. It is evident that the mAP in the top-5 is around 20%, as shown in Figure (5.15e).

Experiment Discussion

This section compares the performance of Find SaaS (SNFCP and M2NFCP) ranking approaches compared with the existing ranking approaches.

Figure (5.16) shows the precision curves for each selection methods including Find SaaS SNFCP for each experiment. The figure shows the high performance



(a) Two Preferences (b) Four Preferences (c) Six Preferences

(d) Eight Preferences (e) Ten Preferences

Figure 5.15 : The mean average precision mAP for Find SaaS M2NFCP in top-(5,10,20 and 50) and compared with the existing research approaches

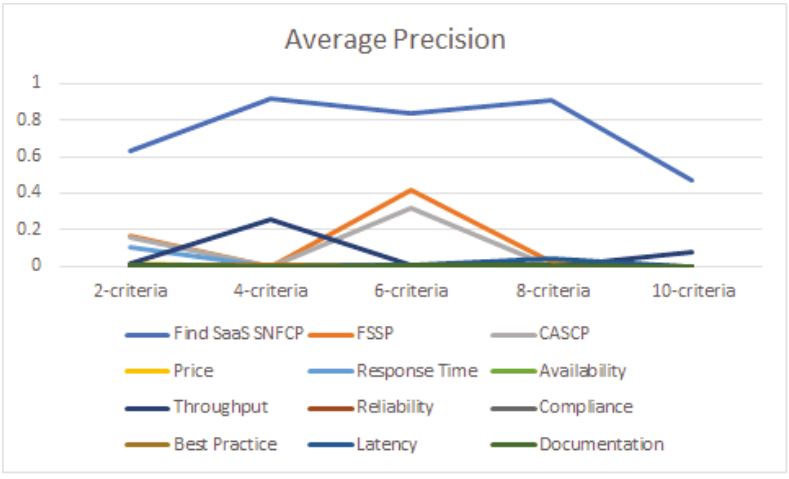


Figure 5.16 : The precision of Find SaaS SNFCP compared with other ranking approaches

of Find SaaS SNFCP in matching consumer's preferences compared with the other approaches. The other ranking approaches do not demonstrate stable performance results. FSSP and CASCP have the second performance results compared with other ranking services approaches that depend on selected quality attributes by a service consumer.

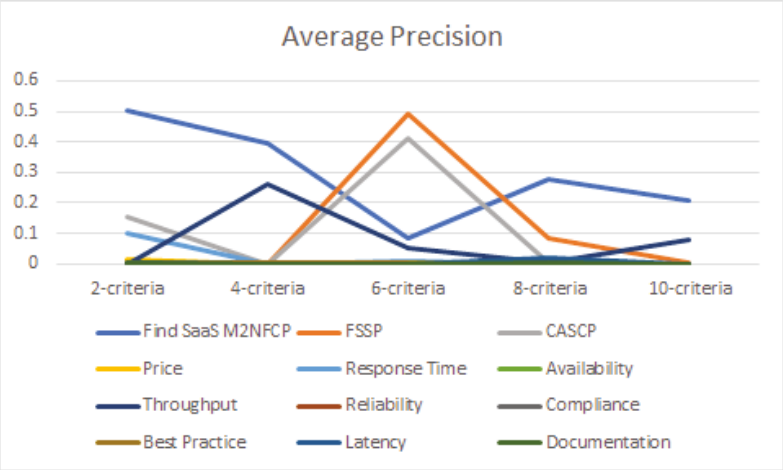


Figure 5.17 : The precision of Find SaaS M2NFCP compared with other ranking approaches

Figure (5.17) presents the performance of Find SaaS M2NFCP compared with other existing approaches. As shown in this figure, the Find SaaS M2NFCP has the highest precision values in the first experiment. For the third experiment, which involves ranking the services taking into account six preferences, the Find SaaS M2NFCP approach records the lowest precision compared with the FSSP approach and CASCP approach.

5.8 Chapter Discussion

In general, our proposed service broker approach obtains a significantly more accurate result compared with other existing approaches.

In future work, it is recommended that the following improvements are made to obtain an accurate service selection result:

1. Proposed ranking services approaches need to be modified by adding an acceptable range of consumer non-functional requirements. This means the ranking approaches should consider these ranges of the services because our proposed methods generate some services that are out of the scope of consumer preferences, as exactly matching the consumer's requirements is impossible. Therefore, this improvement process is essential to obtaining an accurate ranking service that matches with the consumer's requirements.

2. The Find SaaS LNFCP needs to improve to obtain numerous, accurate results for ranking the services. In our proposed method, after mapping the linguistic requirements to the fuzzy numbers, we consider the range value of the fuzzy number. However, this technique causes some issues when ranking the services. Therefore, we can improve our method by calculating the membership number of the linguistic variables and then sending all of these membership numbers to the ranking system to rank and sort the services.

5.9 Conclusion

In this chapter, we proposed a service broker Find SaaS (SNFCP, M2NFCP and LNFCP) to rank and order cloud software services based on consumer preferences. The proposed service broker used to quantify the consumer non-functional requirements and weighted the most important attributes to rank and sort cloud services. Additionally, we classified consumer non-functional requirements for quantifying value of attributes into three categories, as follows: (1) single-based of non-functional of consumer preferences (Find SaaS SNFCP), (2) minimum and maximum-based of non-functional consumer preferences (Find SaaS M2NFCP), and (3) linguistic-based of non-functional consumer preferences (Find SaaS LNFCP).

In this chapter, we also proposed a ranking system engine (RSE) based on measuring the shortest distance to the consumer non-functional preferences. The RSE gathered the consumer non-functional preferences that generated by CRH. This information includes multiple parameters such as Cloud SaaS service description, the best-ideal service, the worst-ideal service, and the weighting non-functional attributes. Moreover, the service registry repository (SRR) is used as metadata between CRH and RSE is uniquely proposed. In this chapter, we considered the numerical qualities for service registry and handled by three main resources, as follows: service provider, monitoring tool, and previous consumer feedback.

The case study explained the ranking of the cloud services, whereby the single-based of consumer non-functional requirement was used to rank the services of computer repair shops. Moreover, extensive experiments were conducted to test and compare the efficiency of the proposed approach. The $nDCG$ and mAP metrics were used to measure the performance of the ranking approaches. The results show that our service broker approach provides the best ranking results when compared with existing ranking approaches that do not consider consumer non-functional preferences when ranking and sorting services.

In the next chapter, we propose an evaluation system to assist Cloud SaaS service consumers to evaluate a service by taking advantage of the free testing period offered by Cloud SaaS service providers. We assume that a Cloud SaaS service consumer will test and evaluate several services suggested by the ranking system and evaluate them based on their experience. Therefore, we consider the subjective attributes of experience (i.e. QoE) to evaluate the services for final service selection.

Chapter 6

Find SaaS Evaluation System of Cloud SaaS Providers

To ensure a consumer chooses a Cloud SaaS service provider which best suits their needs, it is important to undertake an evaluation of the Cloud SaaS service provider. The existing state-of-the-art techniques on Cloud SaaS service selection find that the final selection of service provider is based on the consumers experiences (Naseer & Nazar, 2016; Boussoualim & Aklouf, 2015b, 2015a). However, the existing research does not consider linguistic values to evaluate the consumers experience. In addition, there is scant research which investigates Cloud SaaS Service providers based on an expert fuzzy rule-based system for evaluation purposes. Hence, to address these issues, two new methods are proposed to assess cloud software services for service selection called the Find SaaS Evaluation (linguistic-based and rule-based) methods. The proposed methods are detailed and demonstrated using two case studies.

6.1 Introduction

In Chapter 5, we proposed a service broker to rank cloud software services based on consumer's preferences. The list of services built on the quantitative non-functional attributes were taken from three main sources: the service provider, the monitoring tool and the previous consumer feedback. Most Cloud SaaS service providers offer a one-month trial so that potential consumers can examine and evaluate a service before deciding to enter into a contract for a particular service. The state-of-the-art methods reviewed in Chapter 2 mostly focus on the evaluation of services based on a pairwise comparison of quality attributes known as an analytic

hierarchy process (AHP) method to evaluate a service for service selection whereas, the consumer experience is easier to convey using linguistic terms, such as the *usability is excellent*. Furthermore, there is currently no cloud quality evaluation system which uses an expert rule-based system. Therefore, this chapter proposes a tool, called Find SaaS Evaluation (Fuzzy linguistic-based system and Fuzzy rule-based system) to help a Cloud SaaS service consumer evaluate a service based on their experience. We propose two mathematical models to evaluate the Cloud SaaS service providers, as shown in Figure (6.1). The first is a fuzzy linguistic-based evaluation system and the second is a fuzzy rule-based evaluation system.

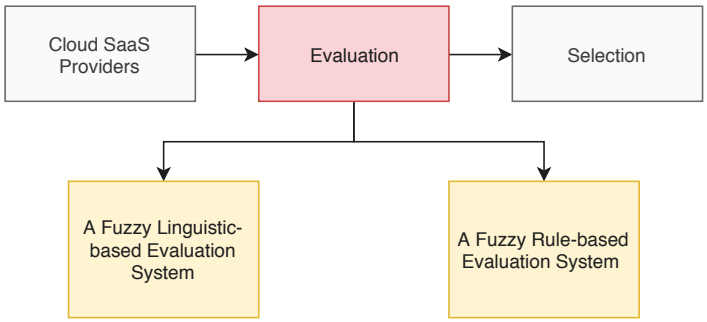


Figure 6.1 : The Find SaaS Evaluation Methods

In this chapter, we address the following:

- The Find SaaS Evaluation architecture.
- The quality of experience (QoE) parameters that should be considered when evaluating a service.
- A Fuzzy Linguistic-based evaluation system for Cloud SaaS.
- A Fuzzy Rule-based evaluation system for Cloud SaaS.

- Case study to demonstrate the idea of Find SaaS Evaluation for the two methods.

- Discussion of the shortcomings of Find SaaS Evaluation.

The remainder of the chapter is organized as follows: Section 6.2 describes the Find SaaS Evaluation framework architecture, Section 6.3 outlines the quality of experience registry repository; Section 6.4 describes the methods to evaluate the Cloud SaaS service providers based on linguistic terms. The fuzzy rule-based system is introduced in section 6.5. A case study is detailed in section 6.6. Section 6.7 discusses the main findings and the limitations of our proposed approach. Finally, Section 6.8 concludes the chapter.

6.2 Find SaaS Evaluation Architecture

The system architecture of Find SaaS Evaluation for the two proposed methods comprises of three components as shown in Figure (6.2).

- The quality of experience (**QoE**) service repository (**QoESR**) is the data store which includes all the services selected by a consumer for cloud quality evaluation purposes. There are three parameters connected with a service which need to be taken into consideration: the vendor, data and service. (details in section 6.3)

- The consumer evaluation handler (**CEH**), which collects the consumers' evaluation for each service. We propose two methods to address the evaluation services mechanism: (1) based on linguistic values to evaluate the criteria 6.4.1); and (2) based on percentage values to evaluate the criteria (details in section 6.5.1).

- The decision maker system (**DMS**) which is responsible for making a decision

to select the best service after the cloud quality evaluation process. Two methods are proposed for this purpose. The **MCDM TOPSIS** method is proposed to handle the linguistic evaluation of criteria (details in section 6.4.2); and the fuzzy rule-based method is proposed to handle the percentage variables to evaluate the criteria (details in section 6.5.2).

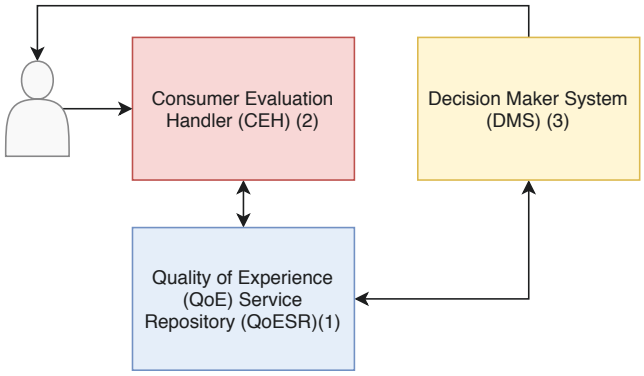


Figure 6.2 : The components of the Find SaaS Evaluation architecture framework

6.3 Quality of Experience (QoE) Service Repository (QoESR)

This section details the consumer's evaluation of the services of the Cloud SaaS service provider. We consider the quality attributes based on a consumer experience to evaluate the Cloud SaaS service provider. The criteria are the factors associated with the QoE, for example, usability and security factors are two quality concerns for a service consumer. However, due to the lack of a standard repository for the QoE for the Cloud SaaS, we propose a new standard of QoE based on the views of an expert team and with consideration of the literature review in this section.

The QoE responses were collected by an expert team and with reference to the literature review that identified the main quality attributes that should be considered

when evaluating cloud software services. The QoE responses were divided into three groups as shown in Figure (6.3): (1) **vendor**; (2) **data**; and (3) **service**. Table 6.1 details the parameters which Cloud SaaS service consumers should consider when evaluating the quality of a service. There are many factors to measure QoE. However, we assume that the QoE is measured by a service consumer during the trial evaluation period offered by a service provider. This trial period allows the consumer to test and evaluate the cloud software service prior to entering a contract.

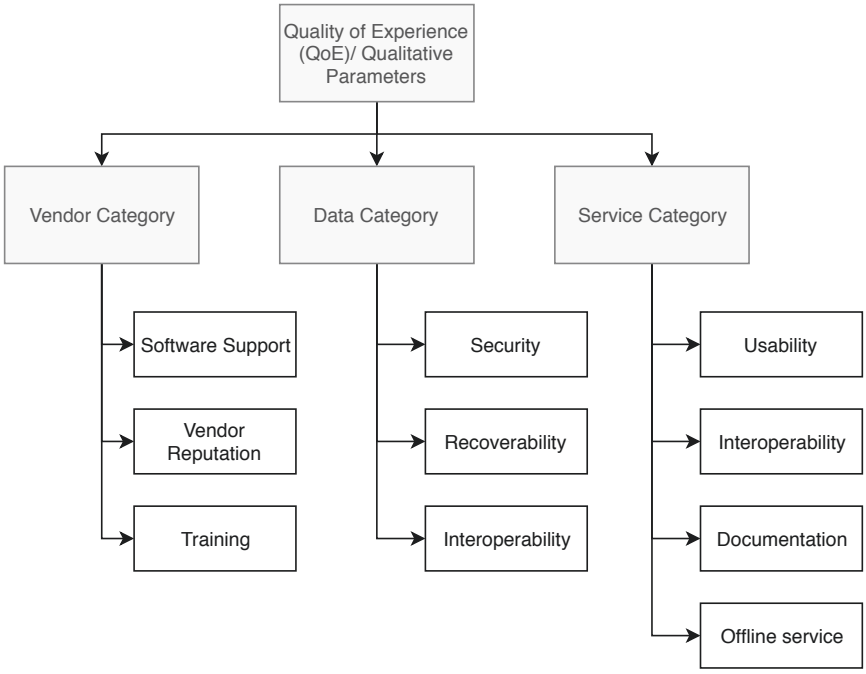


Figure 6.3 : Evaluation of the QoE

1. **Vendor category** includes factors affecting a Cloud SaaS consumer when selecting a service related to a service vendor. For example, is there a cloud software service support team to help consumers solve any issues they may have relating to the service. The vendor category comprises software support, vendor reputation and training.

Table 6.1 : Evaluation QoE

QoE Factor	Evaluation Concerns
Software Support	Helpdesk support, software update and problem solving response.
Vendor Reputation	Service history, number of clients, service rating and social media rating.
Training	Training materials, videoed explanations and community support.
Security	Security certificates, HTTPS and trust certificate.
Recoverability	Supporting service backup.
Interoperability	Supporting the export of different kinds of files.
Usability	Easy-to-use and supporting Mobile App.
Integration	Supporting integration with other services.
Documentation	E-book, White papers and service report.
Offline support	Supporting offline service and synchronizes service automatically after reconnecting service.

(a) **Software support**: Cloud software service providers offer several support strategies to their service consumers, such as communication via chatbots, email and hotline telephone numbers in case of emergency. Cloud SaaS consumers can check how quickly and efficiently software vendors solve software conflicts and usability problems and provide updates to fix bugs and improve security.

(b) **Vendor reputation**: It is important that the customer considers the reputation of the service provider, taking into account the number of

clients who have used the software and their satisfaction level. Another consideration is the brand value of the software vendor. Some consumers prefer to select an established software vendor which has a good reputation, however, a new software vendor may have a good reputation if their services have expanded and if they provide sufficient documentation on the Internet (Godse & Mulik, 2009; Boussoualim & Aklouf, 2015a).

(c) **Training**: A service consumer should also consider whether a vendor provides training, such as workshops, tutorials, or webinars covering both local and international events and information on video channel platforms such as YouTube to help consumers understand the full range of features offered by the service.

2. **Data category** includes all the data-related factors of Cloud SaaS. The data category comprises security, recoverability and interoperability.

   (a) **Security**: This tends to be the main concern for most Cloud SaaS service consumers since Cloud SaaS is a multi-tenant service concept. Security is a process or set of actions to ensure data is protected from unauthorized persons or systems (Reixa et al., 2012). Security involves confidentiality, authenticity, integrity, and availability. A Cloud SaaS service consumer should check if a service is certified with security certifications such as SAS 70, SSAE 16, ISAE 3402, ISO/IEC 9126, ISO/IEC 27001, and ISO/IEC 27002 to ensure the service has an appropriate level of security (Burkon, 2013).

   (b) **Recoverability**: Backup periodicity specifies how often backups are made, either in a continuous manner or at regular intervals. Recovery velocity specifies how quickly data can be recovered from the backup in case of application or infrastructure issues, including the backup service

in terms of using on-premise devices or cloud storage options.

(c) **Interoperability**: Data interoperability refers to the ability of data to be represented in different forms so that it can be accessible over different software platforms. A consumer needs to verify if a system can export their data in commonly used data forms. For example, if a service is selected by a hospital to manage the input of patient information, then the selected service must be able to export the data in various forms, such as PDF, XLS and CSV. This is an important feature in case a consumer wants to adopt another cloud service, or in the backup process, or even for further research or study purposes.

3. **Service Category** covers all the quality parameters involved with services such as the design of the windows screen and the integrity of the system. The service category comprises usability, integration, documentation and offline support.

   (a) **Usability** is a measure of the consumers satisfaction with a service and the quality of their experience in interacting with the Cloud SaaS (Burkon, 2013). A Cloud SaaS provider should ensure there is good user interface (UI) to enhance the users experience of service. Usability is evaluated in terms of the following parameters: (1) UI is intuitive and easy-to-use for frequently implemented tasks and has an attractive graphical windows design; (2) the availability of user manuals to enhance consumer understanding of the windows design and e-Learning modules; (3) support for mobile applications such as mobile phones and tablets (Godse & Mulik, 2009).

   (b) **Integration** refers to the ability of software to communicate easily with other service platforms to share information and is generally connected to

the application programming interface (API) which enables one service to be integrated with other services and systems (Burkon, 2013). A consumer should verify if the service API has the features to enable it to connect with other software platforms. For instance, if a consumer selects the ERP service and needs additional functional software such as a storage service, then the consumer should investigate whether the ERP service for the selected software is able to be integrated with other services, which in this case, is a storage service.

(c) **Documentation** It is important that the service consumer understands the functions of the service, and how the software operates in various stages. Documentation also includes different kinds of materials provided either by a service provider or service community or service consumers, such as E-Books, reports and white papers.

(d) **Offline support** is an important factor due to the continuous connectivity with the software server and measures whether the service supports the connectivity with the system in offline mode and synchronizes data once it reconnects to the Internet.

6.4 A Fuzzy Linguistic-based Evaluation System for Cloud SaaS

The main objective of this approach is to propose a systematic quality evaluation model to help Cloud SaaS service consumers select an optimal service from the available alternatives. We propose methods based on fuzzy logic to evaluate the cloud service criteria subjectively and linguistically. In addition, the multi-criteria decision making method based on the TOPSIS technique is used for the final service selection.

A service consumer firstly selects the main quality attributes which are of con-

cern when selecting a service. These attributes are based on consumer experience, as explained in section 6.3. Then, a consumer weights each quality attribute to determine the importance of each quality attribute. Then, during a cloud software service trial period, a consumer evaluates multiple selected attribute for each service. For example, the usability of a service can be assessed to determine whether it is a medium subjective performance which indicates an average user interface (UI) and an ordinary windows design. Finally, our system decides to select the optimum service from the multiple alternatives after the evaluation process by a service consumer.

6.4.1 Consumer Evaluation Handler (CEH)

The main objective of CEH is to register the consumers feedback on the quality factors for each service and to assign weights to indicate the priority of the quality factor. The CEH deals with two important linguistic variables from a service consumer: (1) Quality priority (QPriority), ρ and (2) Quality evaluation (QEvaluation), ε. The result after fuzzified is the quality evaluation score (QEvaluationScore), δ.

Six linguistic terms or values are used to evaluate the criteria as follows: Excellent (EX), Very High (VH), High (H), Medium (M), Low (L) and Very Low (VL). In order to assign weights to the quality priority, seven linguistic terms used as follows: Extremely Important (EI), Very Important (VI), Important (I), Somewhat Important (SI), Not Important (NI), Not Very Important (NVI) and Not Important at All (NIA).

There are different shapes of presenting the fuzzy membership function such as the triangular fuzzy set and the trapezoidal fuzzy set. In this work, the triangular fuzzy set will be used to measure the quality priority and evaluate the service. The reason for using the triangular to represent our method is that this type of membership function is easy to use and calculate (Karsak & Tolga, 2001). Let us

assume the $(\tilde{a})$ is triangular fuzzy number has three values (a, b, c) as shown in figure (6.4). The membership function $\mu_{\tilde{a}(x)}$ is defined as follows:

$$\mu_{\tilde{a}}(x) = \begin{cases} 0, & \text{x} < a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{x-c}{b-c}, & b \leq x \leq c \\ 0, & \text{x} > c \end{cases}$$

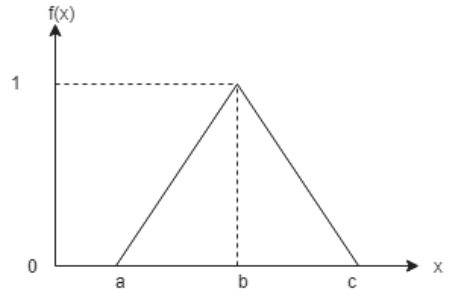


Figure 6.4 : Triangular fuzzy number

Quality Priority (QPriority) (ρ_j)

This is the first linguistic variable of our proposed system. It is used to assign weights to determine the importance of quality for the evaluation process. We propose seven linguistic values which are decomposed into triangular fuzzy numbers using the triangular fuzzy set shown in Figure (6.5). Table 6.2 shows the seven linguistic terms which are used to measure the quality priority.

Quality Evaluation (QEvaluation) ($\varepsilon_i j$)

The second linguistic variable in our proposed approach is the quality evaluation, $\varepsilon_i j$. A Cloud SaaS service consumer evaluates individual quality variable as linguistic values after trialling the service during the free test period. Six linguistic values are

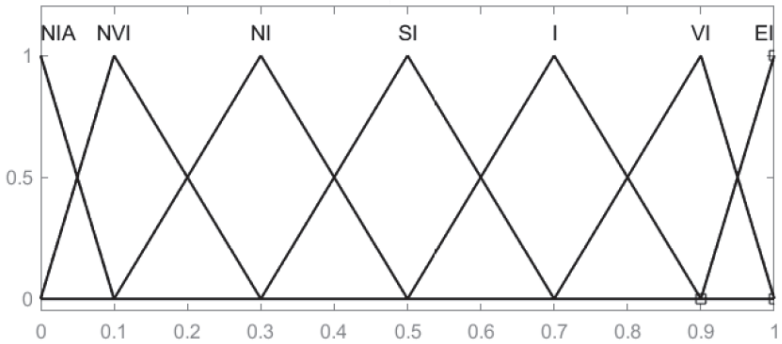


Figure 6.5 : Membership function for linguistic values of QPriority

Table 6.2 : Linguistic variables of weighting qualities with their fuzzy numbers

Linguistics term	Fuzzy number
Extremely Important (EI)	(0.9,1.0,1.0)
Very Important (VI)	(0.7,0.9,1.0)
Important (I)	(0.5,0.7,0.9)
Somewhat Important (SI)	(0.3,0.5,0.7)
Not Important (NI)	(0.1,0.3,0.5)
Not Very Important (NVI)	(0.0,0.1,0.3)
Not Important at All (NIA)	(0.0,0.0,0.1)

proposed for evaluation quality as shown in Table 6.3. The membership functions of the linguistic values are shown in Figure (6.6).

Table 6.3 : Linguistic variables of evaluating the qualities and their fuzzy number

Linguistic Values	Fuzzy numbers
Very low (VL)	(0.0.0.2)
Low (L)	(0,0.2,0.4)
Medium (M)	(0.2,0.4,0.6)
High (H)	(0.4,0.6,0.8)
Very high (VH)	(0.6,0.8,1)
Excellent (E)	(0.8,1,1)

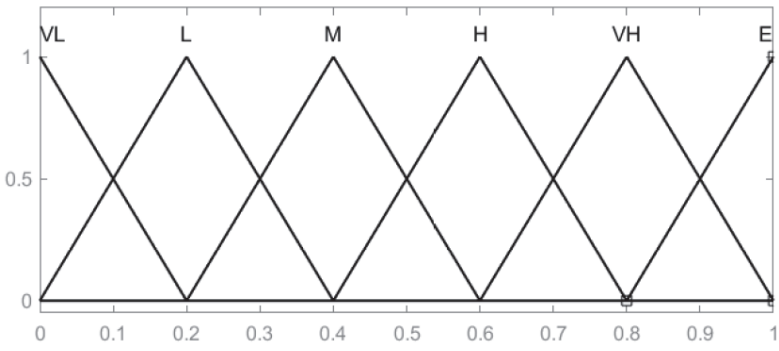


Figure 6.6 : Membership function for linguistic values of QEvaluation

As a result, the list of services evaluated linguistically by a service consumer appears to be similar to the following matrix:

$$\begin{bmatrix} & qoe_1 & qoe_2 & qoe_3 & ... & qoe_n \\ eSaaS_1 & \tilde{\varepsilon_1}1 & \tilde{\varepsilon_1}2 & \tilde{\varepsilon_1}3 & ... & \tilde{\varepsilon_1}n \\ eSaaS_2 & \tilde{\varepsilon_2}1 & \tilde{\varepsilon_2}2 & \tilde{\varepsilon_2}3 & ... & \tilde{\varepsilon_2}n \\ ... & ... & ... & ... & ... & ... \\ eSaaS_m & \tilde{\varepsilon_m}1 & \tilde{\varepsilon_m}2 & \tilde{\varepsilon_m}3 & ... & \tilde{\varepsilon_m}n \end{bmatrix}$$

where, $qoe_1, qoe_2, qoe_3, ..., qoe_n$ are the criteria selected by a consumer for the evaluation process; $eSaaS_1, eSaaS_2, ..., eSaaS_m$ are the m services selected by a service consumer in order to select the best service. Furthermore, $\tilde{\varepsilon_i}j$ is the evaluation for service i and criteria j.

Quality Evaluation Score *(*$QEvaluationScore$*)*, δ

The final process before deciding and selecting a service is to aggregate the fuzzy weighting of the quality priority with the evaluation criteria to obtain the quality evaluation score. The quality evaluation score, δ, is the overall quality score of the individual quality of each service. This process is a metadata step to transmit all this information to the decision maker system to select the best service that matches the consumers requirements.

Let us assume the weighting quality attributes vector assigned by a consumer to weight each criterion are as follows: $\rho = \{\tilde{\rho_1}, \tilde{\rho_2}, ..., \tilde{\rho_n}\}$. Here, $\tilde{\rho_j}$ is a fuzzy number mapped from linguistic terms as detailed in Table 6.2. Moreover, let us assume the evaluation criteria of cloud services by a Cloud SaaS service consumer is as follows:

$$\begin{bmatrix} & qoe_1 & qoe_2 & qoe_3 & ... & qoe_n \\ eSaaS_1 & \tilde{\varepsilon_1}1 & \tilde{\varepsilon_1}2 & \tilde{\varepsilon_1}3 & ... & \tilde{\varepsilon_1}n \\ eSaaS_2 & \tilde{\varepsilon_2}1 & \tilde{\varepsilon_2}2 & \tilde{\varepsilon_2}3 & ... & \tilde{\varepsilon_2}n \\ ... & ... & ... & ... & ... & ... \\ eSaaS_m & \tilde{\varepsilon_m}1 & \tilde{\varepsilon_m}2 & \tilde{\varepsilon_m}3 & ... & \tilde{\varepsilon_m}n \end{bmatrix}$$

where the $\tilde{\varepsilon_i}j$ is the fuzzy numbers which represent the evaluation of criteria j for service i. Additionally, we need to aggregate these two fuzzy numbers together. To do this, the fuzzy number can be aggregated with different arithmetic equations (Klir, 1997) such as:

$$\begin{aligned} \tilde{a} \otimes \tilde{b} = \quad & (a_1, a_2, a_3) \otimes (b_1, b_2, b_3) = \\ & (a_1 \times b_1, a_2 \times b_2, a_3 \times b_3) \end{aligned} \tag{6.1}$$

$$\begin{aligned} \tilde{a} \oplus \tilde{b} = \quad & (a_1, a_2, a_3) \oplus (b_1, b_2, b_3) = \\ & (a_1 + b_1, a_2 + b_2, a_3 + b_3) \end{aligned} \tag{6.2}$$

$$\begin{aligned} \tilde{a} \ominus \tilde{b} = \quad & (a_1, a_2, a_3) \ominus (b_1, b_2, b_3) = \\ & (a_1 - b_1, a_2 - b_2, a_3 - b_3) \end{aligned} \tag{6.3}$$

Therefore, equation (6.1) is applied to aggregate these fuzzy numbers from the evaluation of services and the weighting criteria.

$$\begin{bmatrix} & qoe_1 & qoe_2 & qoe_3 & ... & qoe_n \\ eSaaS_1 & \tilde{\rho_1} \otimes \tilde{\varepsilon_1}1 & \tilde{\rho_2} \otimes \tilde{\varepsilon_1}2 & \tilde{\rho_3} \otimes \tilde{\varepsilon_1}3 & ... & \tilde{\rho_n} \otimes \tilde{\varepsilon_1}n \\ eSaaS_2 & \tilde{\rho_1} \otimes \tilde{\varepsilon_2}1 & \tilde{\rho_2} \otimes \tilde{\varepsilon_2}2 & \tilde{\rho_3} \otimes \tilde{\varepsilon_2}3 & ... & \tilde{\rho_n} \otimes \tilde{\varepsilon_2}n \\ ... & ... & ... & ... & ... & ... \\ eSaaS_m & \tilde{\rho_1} \otimes \tilde{\varepsilon_m}1 & \tilde{\rho_2} \otimes \tilde{\varepsilon_m}2 & \tilde{\rho_3} \otimes \tilde{\varepsilon_m}3 & ... & \tilde{\rho_n} \otimes \tilde{\varepsilon_m}n \end{bmatrix}$$

The result of these fuzzy aggregations is as follows:

$$\begin{bmatrix} & qoe_1 & qoe_2 & qoe_3 & ... & qoe_n \\ eSaaS_1 & \tilde{\delta_1}1 & \tilde{\delta_1}2 & \tilde{\delta_1}3 & ... & \tilde{\delta_1}n \\ eSaaS_2 & \tilde{\delta_2}1 & \tilde{\delta_2}2 & \tilde{\delta_2}3 & ... & \tilde{\delta_2}n \\ ... & ... & ... & ... & ... & ... \\ eSaaS_m & \tilde{\delta_m}1 & \tilde{\delta_m}2 & \tilde{\delta_m}3 & ... & \tilde{\delta_m}n \end{bmatrix}$$

For the final step, the fuzzy interpreter will defuzzify the fuzzy number and transform all these values to a crisp number. To do this, there are different methods for the defuzzification of fuzzy numbers, such as the centre of gravity (CoG), First of Maximum (FOM) , Last Of Maximum (LOM), COG (Center Of Gravity), Mean Of Maxima (MeOM), Weighted Fuzzy Mean (WFM), Quality Method (QM), Extended Quality Method (EQM) and Center Of Area (CoA)(Van Leekwijck & Kerre, 1999). CoG has been used recently in different research to deal with linguistic terms (Herrera, Lozano, & Verdegay, 1995). Equation (6.4) is applied to transfer these values to a crisp number.

$$d(\hat{A}) = \frac{1}{3}(a + b + c) \tag{6.4}$$

6.4.2 Decision Maker System (DMS)

The TOPSIS method is proposed to select the best Cloud SaaS after the evaluation under a fuzzy environment. The TOPSIS standard of a technique for order performance by similarity to the ideal solution was developed by Hwang Yoon (Yoon

& Hwang, 1995). The idea of TOPSIS is to select the best alternative based on the shortest distance to the positive ideal solution and the furthest distance from the negative ideal solution. The advantages of the TOPSIS method over the other MCDM approaches is that it is easy to develop with different programming language platforms. Moreover, TOPSIS can be used with many alternatives, which can be easily applied with $m \times n$ matrix. where m denotes the number of alternatives and n denotes the number of criteria. Other multi-criteria decision-making approaches such as AHP, ANP and ELECTRE are only suitable for use with a small number of alternatives due to the expense in processing and time where there are numerous alternatives. The TOPSIS MCDM process comprises six steps to make a decision and selects the best alternative (Yoon & Hwang, 1995). However, if we apply the TOPSIS method in Find SaaS Evaluation to select the best service, we leave out some steps, such as the normalization process and calculating the weighting normalized matrix because these two steps have already been calculated by the fuzzy logic calculation proposed previously.

Accordingly, to apply the TOPSIS MCDM method into our proposed approach, four steps are needed to select the best Cloud SaaS service provider after the evaluation of the services by a cloud SaaS service consumer as follows:

Let us assume the QEvaluationScore matrix that will interact with the decision maker system is as follows:

$$\begin{bmatrix} & qoe_1 & qoe_2 & qoe_3 & ... & qoe_n \\ eSaaS_1 & \delta_1 1 & \delta_1 2 & \delta_1 3 & ... & \delta_1 n \\ eSaaS_2 & \delta_2 1 & \delta_2 2 & \delta_2 3 & ... & \delta_2 n \\ ... & ... & ... & ... & ... & ... \\ eSaaS_m & \delta_m 1 & \delta_m 2 & \delta_m 3 & ... & \delta_m n \end{bmatrix}$$

where $\delta_i j$ is the quality evaluation score after the defuzzification process for

service i and quality j.

The steps involved in our method are as follows:

1. Calculate the positive ideal solutions and the negative ideal solutions. This step is used to select the service based on measuring the shortest distance to the positive ideal solution and the farthest distance to the negative ideal solution. The result of each calculation will be a vector size of $1 \times n$. where n denotes the number of qualities that are of concern for a service consumer. Equation (6.5) is used to calculate the positive ideal solution, and equation (6.6) is used to calculate the negative ideal solution.

$$\begin{aligned} &A^* = v_1^*, v_2^*, ..., v_j^*, ..., v_n^* = \\ &\{(\max_i v_{ij} | j \in J_1), (\min_i v_{ij} | j \in J_2) |\ i = 1, ..., m\} \end{aligned} \tag{6.5}$$

$$\begin{aligned} &A^- = v_1^-, v_2^*, ..., v_j^-, ..., v_n^- = \\ &\{(\min_i v_{ij} | j \in J_1), (\max_i v_{ij} | j \in J_2) |\ i = 1, ..., m\} \end{aligned} \tag{6.6}$$

2. Calculate the separation measures for each service from the positive and negative ideal solution. To do this, the TOPSIS method applies Euclidean distance to measure the distance to the positive and negative ideal solution. The result of the separation measures will be between [0,1]. Equation (6.7) is used to calculate the separation of each service from the positive ideal solution and equation (6.8) is is used to calculate the separation of each service from the negative ideal solution.

$$S_i^* = \sqrt{\sum_{j=1}^{n} (v_{ij} - v_j^*)}, i = 1, ..., m \tag{6.7}$$

$$S_i^- = \sqrt{\sum_{j=1}^{n} (v_{ij} - v_j^-)}, i = 1, ..., m \tag{6.8}$$

3. Calculate the relative closeness to the ideal solution for each cloud service using equation (6.9). The result of this step is between [0,1]. The service that has a higher number indicates better performance and that it is a good match with the consumer's requirements.

$$C_i^* = \frac{S_i^-}{S_i^* + S_i^-} \tag{6.9}$$

4. The last step is ranking and sorting the cloud services based on the index value C_i^j. The service that has a higher index value will be selected first which means it is very close to the consumer's evaluation of services. The result represents the evaluation score for each service which shows the position of each service.

6.5 A Fuzzy Rule-based Evaluation System of Cloud SaaS

We introduce a fuzzy rule-based system to evaluate the cloud software services for a service consumer. The evaluation process goes through several phases in order to find the best cloud SaaS service provider. The first phase is presenting the list of services which match a consumer's non-functional quality requirements $SaaS_ij$. Selecting the QoE is a vital phase to present the consumer's preferences in the second phase. Next, the evaluation system requires the priority of the quality list (QPriority (ρ_j)) to be weighted. Also, the consumer must evaluate each quality for each service (QEvaluation ($\varepsilon_i j$)). Both QPriority and QEvaluation are measured as a percentage by a service consumer. Then, the evaluation system investigates the score of the evaluation quality (QEvaluationScore) using a fuzzy rule-based is proposed for this purpose. Finally, after collecting the score evaluation of each quality for each service, the evaluation system calculates the average evaluation score for each service for the final service selection (ServiceEvaluationScore - SES). The architecture of system is shown in figure (6.7)

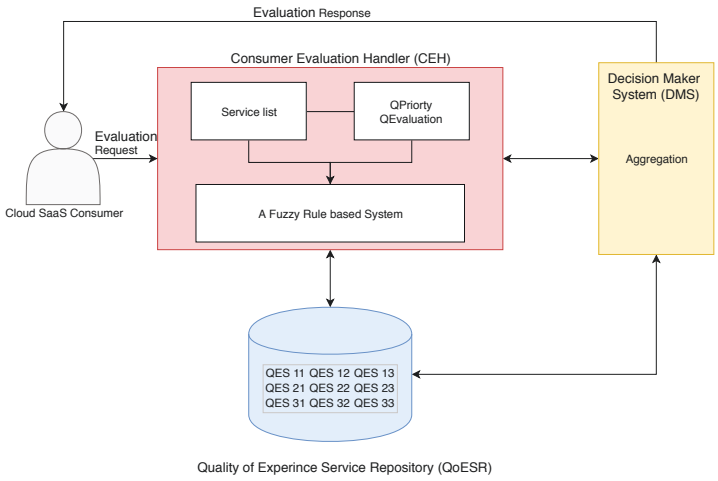


Figure 6.7 : A fuzzy rule-based evaluation system architecture

6.5.1 Consumer Evaluation Handler (CEH)

This part of the system is responsible for gathering the primary information from a service consumer to evaluate the cloud software services. This part interacts with the quality of experiences service repository (QoESR) through bidirectional communication. There are four main inputs collected from a service consumer for the evaluation of the services. Firstly, a service consumer needs to list the cloud software services which satisfy the non-functional requirements, such as the service price and service rating. We assume that a service provider offers the customer the opportunity to trial and evaluate a cloud service for approximately one month. Therefore, we assume that the evaluation process is conducted during the period in which the service is trialed.

Three linguistic variables are proposed using the fuzzy set, two as the inputs and one as a result of the aggregation of the inputs, namely Quality priority (QPriority) ρ, Quality evaluation (QEvaluation) ε and Quality evaluation score (QEvaluation-Score) δ. The quality priority (QPriority), ρ, is used to determine the importance

of the criteria measured as a percentage. Moreover, the quality evaluation (QEvaluation), ε, is the evaluation mechanism for individual criteria obtained from a service consumer when the service is being trialed, also measured as a percentage. Finally, the quality evaluation score (QEvaluationScore), δ, is the result of the aggregation of (QPriority) ρ and (QEvaluation) ε using the fuzzy rule, also measured as a percentage.

A Fuzzy Rule-based System

There are different ways to present fuzzy numbers, such as triangular fuzzy numbers and trapezoidal fuzzy numbers. In this work, trapezoidal fuzzy numbers are proposed to present the QEvaluation, QPriority and QEvaluationScore. A trapezoidal fuzzy number ($\tilde{a}$) can be represented as (a, b, c, d)), as shown in Figure (6.8). The membership function $\mu_{\tilde{a}(x)}$ can be given as follows:

$$\mu_{\tilde{a}}(x) = \begin{cases} 0, & \text{x} < a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{x-c}{d-c}, & c \leq x \leq d \\ 0, & \text{x} > d \end{cases}$$

The steps in the working of fuzzy-rule based system:

Step 1: Fuzzy Sets

The first step identifies the linguistic variables input and output for our evaluation system. Table 6.4 shows that the intervals for ρ ε δ are generated to be within the range of [0,100] as a percentage.

Figures 6.9 to 6.11 show the fuzzy sets for all the linguistic variables used in our evaluation system. For (QPriority) ρ, we consider only three linguistic values - low,

Table 6.4 : Linguistic variables and their ranges

Linguistic variable: quality evaluation		
Linguistic value	Notation	Numerical range (percentage)
Bad	B	[0, 0, 20, 40]
Fair	F	[20, 40, 60, 80]
Excellent	E	[60, 80, 100, 100]
Linguistic variable: quality priority		
Linguistic value	Notation	Numerical range (percentage)
Low	L	[0, 0, 20, 50]
Medium	M	[20, 40, 60, 80]
High	H	[60, 80, 100, 100]
Linguistic variable: Evaluation score		
Linguistic value	Notation	Numerical range (percentage)
Very Low	VL	[0, 0, 10, 30]
Low	L	[20, 30, 40, 50]
Medium	M	[40, 50, 60, 70]
High	H	[60, 70, 80, 90]
Very High	VH	[80, 90, 100, 100]

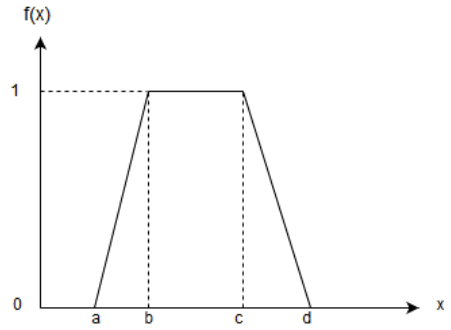


Figure 6.8 : Trapezoidal fuzzy number

medium and high. For (QEvaluation) ε, we consider three linguistic values - bad, fair and excellent. For (QEvaluationScore) δ, we consider five linguistic values - very low, low, medium, high and very high.

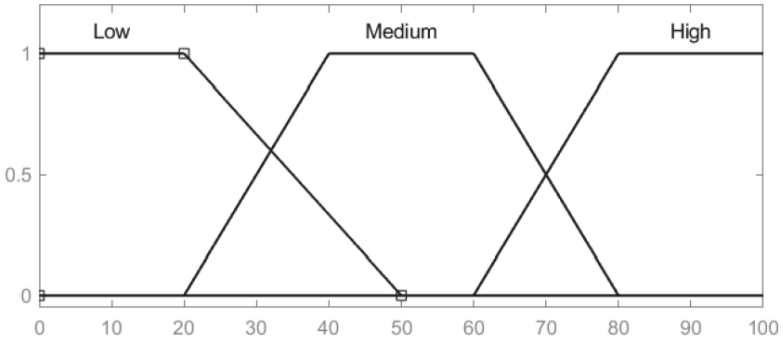


Figure 6.9 : Membership functions of linguistic values for QPriority

Step 2: Construct fuzzy rules

The second step is to build the fuzzy rules for our system. There are different techniques to construct fuzzy rules, such as asking an expert to describe how the problem can be solved using fuzzy linguistic variables. Another method is to use a flow diagram, computer database and human behaviour (Negnevitsky, 2005).

In this work, we build the fuzzy set rules using the research method proposed by (Turksen, Tian, & Berg, 1992). There are two variable inputs, QPriority and

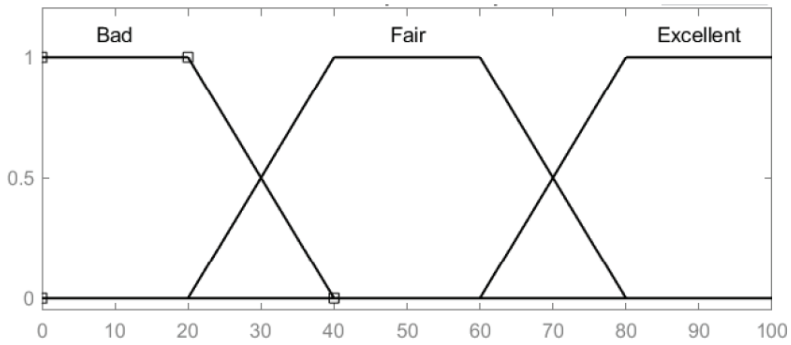


Figure 6.10 : Membership functions of linguistic values for QEvaluation

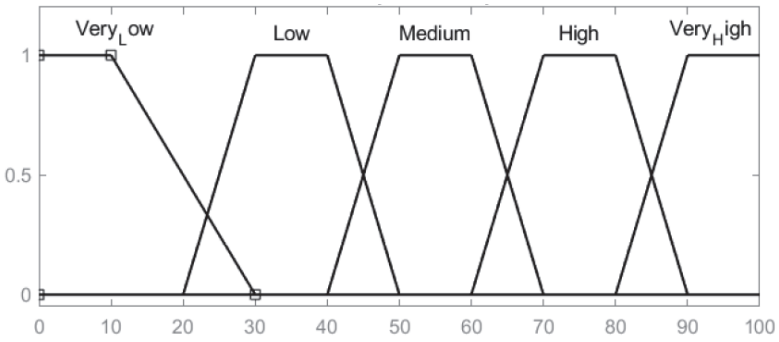


Figure 6.11 : Membership functions of linguistic values for QEvaluationScore

QEvaluation, and one output QEvaluationScore. Commonly, fuzzy rules are represented in the form of a matrix called a fuzzy associative matrix (FAM). Therefore, in our case, there is 3×3 FAM that represents the rules of our evaluation system as shown in Figure 6.12. Table 6.5 contains all the rules used in our evaluation system.

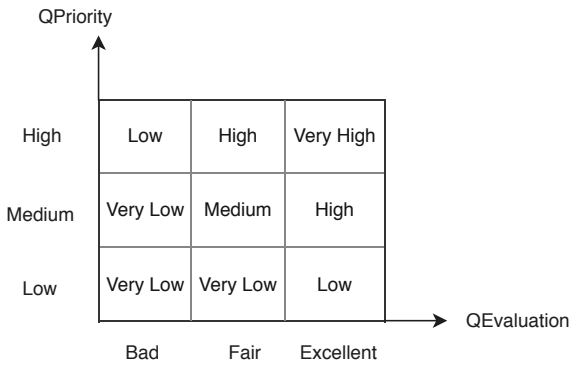


Figure 6.12 : The square rules (FAM) representation

Figure 6.13 represents the three-dimensional plots for QPriority and QEvaluation as the input variables and QEvaluationScore as the output variable. The figure was generated using the Matlab Fuzzy Logic Toolbox.

6.5.2 Decision Maker System (DMS)

This component calculates the service evaluation score - SES for each service in order to find an optimum service for service selection. Equation (6.10) is used to calculate the final evaluation service result for service i.

$$SES_i = \frac{\sum_{j=1}^{n} \delta_{ij}}{N} \tag{6.10}$$

where δ_{ij} represents the quality evaluation score - $QEvaluationScore$ of criteria i and service j. N refers to the number of attributes concerns by a service consumer.

Table 6.5 : The rule

Rule	ρ	ε	δ
1	Low	Bad	Very Low
2	Low	Fair	Very Low
3	Low	Excellent	Low
4	Medium	Bad	Very Low
5	Medium	Fair	Medium
6	Medium	Excellent	High
7	High	Bad	Low
8	High	Fair	High
9	High	Excellent	Very High

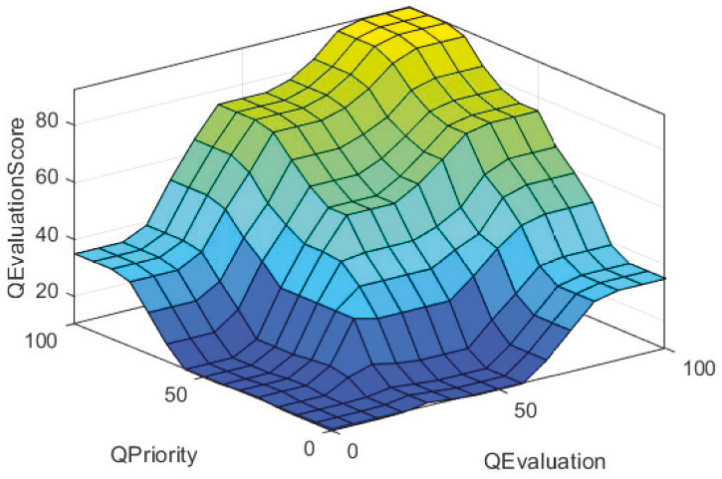


Figure 6.13 : Three-dimensional plot for evaluation system rule

6.6 Case Study

This section details our proposed approach with the assistance of two case studies. The first case study in section 6.6.1 uses a linguistic-based approach to evaluate the cloud software services (Find SaaS Linguistic-Based Evaluation System) and the second case study in section 6.6.2 uses a rule-based approach to evaluate the cloud software services (Find SaaS Rule-based Evaluation System).

6.6.1 Case study: Evaluation of cloud SaaS service provider using a fuzzy linguistic-based system

To better illustrate our proposed evaluation approach, a case study is used to explain the process of evaluating services to make a final service selection using the linguistic-based evaluation technique. This case study was chosen to illustrate how the proposed evaluation services can help a new business select a cloud SaaS.

Let us assume that a new business called Misbar is interested in adopting a Cloud SaaS solution using the CRM SaaS application. The expert team at Misbar understand the benefits of using cloud services instead of building their own software

application, such as availability, elasticity and support for a mobile app. Therefore, the Misbar team used the ranking cloud SaaS system to find the best CRM service which matches their preferences. They also selected six services which match their requirements regarding service functionality and other QoS features such as service cost, availability rate etc.

For the final service selection, the Misbar team needs to evaluate their six services using the evaluation criteria described in Table 6.1. We assume all these selected services can be trialed and evaluated during the free test period offered by the service provider. As shown in Table 6.6, the Misbar team evaluates each service based on multiple QoE parameters. For instance, the first service provider evaluates software support as medium. This means this service has been evaluated as being medium in terms of supporting problem solving, fixing bugs, updating software and providing help desk support. Similarly, they evaluate the following criteria: vendor reputation, training, security, recoverability, interoperability, usability, documentation and offline support for all the selected services.

Table 6.6 : Linguistic values to evaluate the services

Service Provider	Software Support	Vendor reputation	Training	Security	Recoverability	Interoperability	Usability	Documentation	Offline Support
SaaS provider 1	M	H	VH	M	VH	M	VH	M	L
SaaS provider 2	VL	L	M	M	VH	M	E	L	L
SaaS provider 3	VH	E	VH	VH	H	H	M	VH	H
SaaS provider 4	VL	H	H	H	M	M	M	L	L
SaaS provider 5	M	VH	L	M	H	H	M	L	M
SaaS provider 6	E	H	H	H	VH	M	VH	M	VH

After this, the Misbar team weights the priority of the criteria among all conflict-

ing criteria to make a service selection. Table 6.7 shows the priority of the criteria provided by the Misbar team. As shown in this table, the Misbar team used linguistic values to weight the QoE parameters. For example, the Misbar team considers software support as very important, however, they consider the usability criteria as important.

Table 6.7 : Linguistic weight criteria

Weight criteria	Software Support	Vendor reputation	Training	Security	Recoverability	Interoperability	Usability	Documentation	Offline Support
Linguistic values	VI	I	EI	EI	VI	SI	I	NI	VI

Then, the Find SaaS Evaluation system transfers their linguistic values that describe the evaluation criteria into fuzzy numbers. For example, the software support quality is weighted very important and the fuzzy number of this linguistic value is (0.7,0.9,1.0). Table 6.8 illustrates the fuzzy numbers of the linguistic values for weighting the quality attributes.

Table 6.8 : Fuzzy numbers for weighting criteria

X	Software Support	Vendor reputation	Training	Security	Recoverability	Interoperability	Usability	Documentation	Offline Support
weight	(0.7,0.9,1.0)	(0.5,0.7,0.9)	(0.9,1.0,1.0)	(0.9,1.0,1.0)	(0.7,0.9,1.0)	(0.3,0.5,0.7)	(0.5,0.7,0.9)	(0.1,0.3,0.5)	(0.7,0.9,1.0)

The Find SaaS Evaluation system then transfers the linguistic values of the evaluation of the quality attributes into fuzzy numbers. For example, the evaluation of the first service in relation to vendor reputation is high and the fuzzy number of high is (0.4,0.6,0.8). Table 6.9 shows the fuzzy numbers of the evaluation criteria for all services.

After collecting all the inputs from the Misbar team, the Find SaaS (Linguistic-based Evaluation system) is used to select the service which best matches the con-

Table 6.9 : Fuzzy numbers for an evaluation of the services

X	Software Support	Vendor reputation	Training	Security	Recoverability	Interoperability	Usability	Documentation	Offline Support
SaaS provider 1	(0.2,0.4,0.6)	(0.4,0.6,0.8)	(0.6,0.8,1.0)	(0.2,0.4,0.6)	(0.6,0.8,1.0)	(0.2,0.4,0.6)	(0.6,0.8,1.0)	(0.2,0.4,0.6)	(0.0,0.2,0.4)
SaaS provider 2	(0.0,0.0,0.2)	(0.0,0.2,0.4)	(0.2,0.4,0.6)	(0.2,0.4,0.6)	(0.6,0.8,1.0)	(0.2,0.4,0.6)	(0.8,1.0,1.0)	(0.0,0.2,0.4)	(0.0,0.2,0.4)
SaaS provider 3	(0.6,0.8,1.0)	(0.8,1.0,1.0)	(0.6,0.8,1.0)	(0.6,0.8,1.0)	(0.4,0.6,0.8)	(0.4,0.6,0.8)	(0.2,0.4,0.6)	(0.6,0.8,1.0)	(0.4,0.6,0.8)
SaaS provider 4	(0.0,0.0,0.2)	(0.4,0.6,0.8)	(0.4,0.6,0.8)	(0.4,0.6,0.8)	(0.2,0.4,0.6)	(0.2,0.4,0.6)	(0.2,0.4,0.6)	(0.0,0.2,0.4)	(0.0,0.2,0.4)
SaaS provider 5	(0.2,0.4,0.6)	(0.6,0.8,1.0)	(0.0,0.2,0.4)	(0.2,0.4,0.6)	(0.4,0.6,0.8)	(0.4,0.6,0.8)	(0.2,0.4,0.6)	(0.0,0.2,0.4)	(0.2,0.4,0.6)
SaaS provider 6	(0.8,1.0,1.0)	(0.4,0.6,0.8)	(0.4,0.6,0.8)	(0.4,0.6,0.8)	(0.6,0.8,1.0)	(0.2,0.4,0.6)	(0.6,0.8,1.0)	(0.2,0.4,0.6)	(0.6,0.8,1.0)

sumer evaluation for each service. Firstly, our system aggregates the fuzzy evaluation of the attributes (QEvaluation) with the fuzzy quality priority (QPriority) to obtain the quality evaluation score (QEvaluationScore). For example, regarding the first service, for the software support criteria, the fuzzy (QEvaluation) is (0.2,0.4,0.6) and the fuzzy (QPriority) is (0.7,0.9,1.0). Therefore, the aggregation of these two fuzzy numbers is $(0.2, 0.4, 0.6) \otimes (0.7, 0.9, 1.0) = (0.2 \times 0.7, 0.4 \times 0.9, 0.6 \times 1.0)$. The final result is (0.14,0.36,0.6). Table 6.10 shows the evaluation score matrix for all cloud services with respect to the quality evaluation score.

Table 6.10 : Fuzzy numbers for evaluation matrix

X	Software Support	Vendor reputation	Training	Security	Recoverability	Interoperability	Usability	Documentation	Offline Support
Escore 1	(0.14, 0.36, 0.60)	(0.20, 0.42, 0.72)	(0.54, 0.80, 1.00)	(0.18, 0.40, 0.60)	(0.42, 0.72, 1.00)	(0.06, 0.20, 0.42)	(0.30, 0.56, 0.90)	(0.02, 0.12, 0.30)	(0.00, 0.18, 0.40)
Escore 2	(0.0, 0.0, 0.2)	(0.00, 0.14, 0.36)	(0.18, 0.40, 0.60)	(0.18, 0.40, 0.60)	(0.42, 0.72, 1.00)	(0.06, 0.20, 0.42)	(0.4, 0.7, 0.9)	(0.00, 0.06, 0.20)	(0.00, 0.18, 0.40)
Escore 3	(0.42, 0.72, 1.00)	(0.4, 0.7, 0.9)	(0.54, 0.80, 1.00)	(0.54, 0.80, 1.00)	(0.28, 0.54, 0.80)	(0.12, 0.30, 0.56)	(0.10, 0.28, 0.54)	(0.06, 0.24, 0.50)	(0.28 0.54 0.80)
Escore 4	(0.0, 0.0, 0.2)	(0.20, 0.42, 0.72)	(0.36, 0.60, 0.80)	(0.36, 0.60, 0.80)	(0.14, 0.36, 0.60)	(0.06, 0.20, 0.42)	(0.10, 0.28, 0.54)	(0.00, 0.06, 0.20)	(0.00, 0.18, 0.40)
Escore 5	(0.14, 0.36, 0.60)	(0.30, 0.56, 0.90)	(0.0, 0.2, 0.4)	(0.18, 0.40, 0.60)	(0.28, 0.54, 0.80)	(0.12, 0.30, 0.56)	(0.10, 0.28, 0.54)	(0.00, 0.06, 0.20)	(0.14 0.36 0.60)
Escore 6	(0.56, 0.90, 1.00)	(0.20, 0.42, 0.72)	(0.36, 0.60, 0.80)	(0.36, 0.60, 0.80)	(0.42, 0.72, 1.00)	(0.06, 0.20, 0.42)	(0.30, 0.56, 0.90)	(0.02, 0.12, 0.30)	(0.42 0.72 1.00)

Secondly, the fuzzy interpreter defuzzifies the fuzzy numbers for the (QEvaluationScore) to obtain the crisp number. To do this, the Center of Gravity (CoG) as proposed in Equation (6.4 is used to calculate the crisp number. For example, the fuzzy evaluation score for the first service for software support is (0.14,0.36,0.6). Therefore, to calculate the CoG is $\frac{1}{3}(0.14+0.36+0.6)$ with a result of 0.366666667. Table 6.11 shows the crisp numbers of all the services.

Table 6.11 : The Final crisp number of all services

X	Software Support	Vendor reputation	Training	Security	Recoverability	Interoperability	Usability	Documentation	Offline Support
Escore 1	0.366666667	0.446666667	0.78	0.393333333	0.713333333	0.226666667	0.586666667	0.146666667	0.193333333
Escore 2	0.066666667	0.166666667	0.393333333	0.393333333	0.713333333	0.226666667	0.666666667	0.086666667	0.193333333
Escore 3	0.713333333	0.666666667	0.78	0.78	0.54	0.326666667	0.306666667	0.266666667	0.54
Escore 4	0.066666667	0.446666667	0.586666667	0.586666667	0.366666667	0.226666667	0.306666667	0.086666667	0.193333333
Escore 5	0.366666667	0.586666667	0.2	0.393333333	0.54	0.326666667	0.306666667	0.086666667	0.366666667
Escore 6	0.82	0.446666667	0.586666667	0.586666667	0.713333333	0.226666667	0.586666667	0.146666667	0.713333333

TOPSIS MCDM is a method which is used to assist decision making based on measuring the distance for each alternative based on the shortest distance to the positive ideal solution and furthest distance to the negative ideal solution. The positive ideal solution is the highest value for each criterion, whereas the negative ideal solution is the lowest value for each criterion. In the next step, our proposed system obtains the positive ideal solution and negative ideal solution to make a decision based on measuring the distance to these two vectors. Table 6.12 shows the positive ideal solution vector and negative ideal solution vector.

After this, the TOPSIS method makes a decision by measuring the distance to the positive ideal solution (S^+) and the negative ideal solution (S^-) for each service

Table 6.12 : The positive ideal solution and negative ideal solution

X	Software Support	Vendor reputation	Training	Security	Recoverability	Interoperability	Usability	Documentation	Offline Support
A	0.82	0.666666667	0.78	0.78	0.713333333	0.326666667	0.666666667	0.266666667	0.713333333
A-	0.066666667	0.166666667	0.2	0.393333333	0.366666667	0.226666667	0.306666667	0.086666667	0.193333333

using Euclidean distance. Table 6.13 shows the distance to the positive ideal solution and the negative ideal solution for all services.

Table 6.13 : The distance to the positive and negative ideal solutions

SaaS Provider	S^+	S^-
SaaS provider 1	0.839417788	0.840819706
SaaS provider 2	1.195547294	0.535868972
SaaS provider 3	0.448404579	1.160478828
SaaS provider 4	1.119484008	0.515062024
SaaS provider 5	1.005009674	0.580076623
SaaS provider 6	0.392371706	1.142531303

Before the last step, the TOPSIS method calculates the similarity to the positive ideal solution in order to obtain the evaluation score for each service. Table 6.14 shows the similarity of the positive ideal solution for all services. The values are between [0,1] where a value close to one indicates better service.

The final step is to rank the services based on the value of the similarity to the positive ideal solution. Table 6.15 shows the ranking of the services for the Misbar company. It can be seen that service provider 6 has the most similarity to Misbars preferences at approximately 74.4 per cent. The service provider which is ranked last is service provider 2 with 30.9 per cent. Therefore, service provider 6 is the

Table 6.14 : Similarity to the positive ideal solution

SaaS Provider	C^+
SaaS provider 1	0.500417179
SaaS provider 2	0.309497481
SaaS provider 3	0.721294547
SaaS provider 4	0.315110137
SaaS provider 5	0.365959017
SaaS provider 6	0.744367101

most suitable for selection based on Misbars evaluation mechanisms.

Table 6.15 : Ranking of the services for selection

SaaS Provider	C^+
SaaS 6	0.744367101
SaaS 3	0.721294547
SaaS 1	0.500417179
SaaS 5	0.365959017
SaaS 4	0.315110137
SaaS 2	0.309497481

6.6.2 Case Study: Evaluation of cloud SaaS using a fuzzy rule-based system

The purpose of this case study is to demonstrate how our proposed system evaluates cloud software services using the fuzzy rule-based technique. We used the MATLAB Fuzzy Logic Toolbox to obtain the quality evaluation score (QEvalua-

tionScore). First, we defined two fuzzy variables as the inputs (QPriority and QEvaluation). The linguistic variable QPriority has a range (the universe of discourse) between 0% to 100% and the range comprises three fuzzy sets low, medium and high. The universe of discourse of linguistic variable QEvaluation can be between 0% to 100% and includes three fuzzy sets namely bad, fair and excellent. We define one linguistic variable as an output QEvaluationScore. The universe of discourse of the QEvaluationScore is between 0% to 100% and is represented by five linguistic values: very low, low, medium, high and very high.

In the first step, a service consumer identifies the quality attributes that are of concern to evaluate the cloud software service. A service consumer selects the following criteria to evaluate the services: software support, security, usability, documentation, training, vendor reputation and recoverability. The consumer must also identify the services that match their QoS requirements such as service price described in Chapter 5.

Step 1: Fuzzification

The fuzzification process occurs when a consumer weights the quality priority as a percentage ρ_j. This value in the fuzzy set is called the crisp number. Also, after testing each service, the service consumer evaluates each quality as a percentage $\varepsilon_i j$ as a crisp value. The value of ρ_j and $\varepsilon_i j$ are limited to the universe of discourse (QPriority and QEValuation), which is between [0,100]. Once the crisp inputs ρ_j and $\varepsilon_i j$ are obtained, both are fuzzified against the appropriate linguistic fuzzy sets to generate the QEvaluationScore, δ. Table 6.16 shows the crisp values of evaluation criteria j for each services i. Table 6.17 shows the weighting priority of the criteria assigned by a service consumer.

Step 2: Rule evaluation

The second step is to use the fuzzified inputs of QPriority and QEvaluation to

Table 6.16 : The Consumer's evaluation criteria

service	Software Support	Security	Usability	Documentation	Training	Vendor reputation	Recoverability
1	50%	50%	90%	70%	80%	90%	90%
2	20%	30%	90%	10%	60%	20%	80%
3	80%	90%	40%	70%	30%	60%	70%
4	25%	80%	80%	20%	80%	80%	50%
5	60%	60%	50%	20%	20%	60%	70%
6	80%	75%	80%	60%	70%	70%	85%

Table 6.17 : The priority of qualities

x	Software Support	Security	Usability	Documentation	Training	Vendor reputation	Recoverability
Priority	90%	90%	40%	10%	80%	70%	90%

obtain the QEvaluationScore. For example, regarding the first service with regard to software support, the service consumer evaluates $\varepsilon_1 1 = \%50$, and weights the QPriority by $\rho_1 = \%90$. The membership of QPriority $\mu(90) = 1(High)$, and The membership of QEvlaution $\mu(50) = 1(Fair)$. Therefore, we apply the antecedents of the fuzzy rule. In this case, there is only one antecedent fuzzy rule, therefore, the AND operation is used to obtain QEvaluationScore. The result is $\mu(90 \cap 50) = min[\mu(90), \mu(50)] = 1(High)$. However, in case there are multiple antecedents, it will be multiple operations are used to obtain the QEvaluationScore. For instance, if a consumer evaluates the criteria as 30%, then $\mu(30) = 0.5(Bad)$ and $0.5(Medium)$. Therefore, the aggregation process as shown in the next step is necessary to obtain the final result of QEvaluationScore.

Step 3: Aggregation of rule output

The aggregation process is the combination of the output of all the rules. As previously mentioned, there may be multiple antecedents for some cases. As a result, multiple rules will be applied to obtain the output result. Lastly, the output of all the rules output is aggregated into a single fuzzy set for the overall fuzzy output.

Step 4: Defuzzification

The last step in the fuzzy inference process is to obtain the final output of the fuzzy system which is the last crisp number for the QEvaluationScore. To do this, the Center of Gravity (CoG) method will be applied to obtain the final QEvaluationScore as shown in equation (6.11).

$$QEvaluationScore = \frac{1}{4}(a * b * c * d) \tag{6.11}$$

Step 5: Evaluation Score

To find the overall performance of each service, we use the evaluation score as calculated by equation (6.11). Table 6.18 shows the QEvaluationScore for all the

service providers as well as the evaluation score. The results show that service provider 1 obtained the highest score of 71.8142%, followed by service provider 6 with 71.057%. The service consumer can choose the service by checking how much each criterion matches their preferences. For example, in relation to software support, service provider 1 has around 75% of consumer preferences, however service provider 5 maps to 92.5% of consumer preferences. Service provider 2 has the lowest performance as it maps to only 50.61% of consumer preferences.

Table 6.18 : The final evaluation score for the service providers

Service	Software Support	Security	Usability	Documentation	Training	Vendor reputation	Recoverability	Service Evaluation Score (SES)
1	75%	75%	62.7%	23.5%	92.5%	81.5%	92.5%	71.81428571%
2	35%	55%	62.7%	10.6%	75%	23.5%	92.5%	50.61428571%
3	92.5%	92.5%	42%	23.5%	55%	65%	81.5%	64.57142857%
4	46.6%	92.5%	62.7%	10.6%	92.5%	81.5%	75%	65.91428571%
5	75%	75%	42%	10.6%	35%	65%	81.5%	54.87142857%
6	92.5%	86.1%	62.7%	10.6%	81.5%	71.5%	92.5%	71.05714286%

6.7 Chapter Discussion

This chapter proposes an evaluation mechanism based on the evaluation of the quality of consumer experience attributes. In this chapter, we discuss a consumer quality of experience when evaluating the services provided by a cloud software service provider during the introductory trial period. Two evaluation systems are proposed based on the fuzzy linguistic evaluation system and the fuzzy rule evaluation system. The primary shortcoming of our work is that it is based on subjective feedback form Cloud SaaS consumers. In our future work, we aim to investigate the use of AI for objective QoE assessment.

6.8 Conclusion

A new method based on a fuzzy linguistic evaluation system and a fuzzy rule evaluation system is presented in this chapter to evaluate cloud software services to assist the process of selecting the most optimum and suitable service. The proposed method identifies the main QoE concerns of consumers when evaluating a cloud software service. The evaluation mechanism used in our system measures the evaluation criteria in two ways: by linguistic values or as a percentage. The decision maker for the linguistic system is based on multi-criteria decision making via the TOPSIS approach, whereas the evaluation based on the rule system uses fuzzy rule to make the decision to select a service. Two case studies illustrate the proposed approaches. The next stage in improving our system is to design a recommender system to help a service consumer evaluate the criteria to obtain the best evaluation mechanism.

Next chapter introduce the new service composition approach called Find SaaS GA TOPSIS to select the optimum composite Cloud SaaS providers.

Chapter 7

Find SaaS GA TOPSIS - Composite Cloud SaaS Service Selection

In this chapter, we have investigated the scenario where multiple Cloud SaaS providers are selected for combined service delivery as also known as service composition. Service composition is necessary for a consumer to provide a satisfactory full Cloud SaaS solution for a new business sector or regular SaaS consumer. In this chapter, we propose a service composition decision maker system called Find SaaS GA TOPSIS framework for selecting a composite service. Our proposed approach used the Genetic Algorithm (GA) for the optimization process to maximize the reputation-based service reviewers and minimize the total cost of service composition. Further, the TOPSIS MCDM is used for ranking and selecting the optimal composite service composition in order to fulfil a consumer's preferences. A case study is used to evaluate the proposed approach. A major part of the chapter has been published in one journal paper, titled: Cloud Software as a Service (SaaS) Service Composition approach using Genetic Algorithm (GA) and TOPSIS MCDM (Find SaaS Composition Framework).

7.1 Introduction

Cloud SaaS is a software application which runs and operates over the cloud computing infrastructures. One of the advantages of Cloud SaaS is that multiple functional service providers can be worked as one composite service. That means a new business can adopt Cloud SaaS solution and can select aggregated multiple Cloud SaaS providers in order to fulfil their requirements. Service composition is a

suite of services which are combined to work as a functional unit. The advantage of service composition of Cloud SaaS is to adopt a full Cloud SaaS solution to meet a consumer's requirements. Previous studies indicate that the selection of service composition is a complex and challenging task classified under the NP-Hard problem (Toh & Tang, 2018; Wu & Khoury, 2012; D. Wang, Yang, & Mi, 2015).

The literature review on the selection of a Cloud SaaS service composition shows a variety of approaches. He et al. (2012) and Wu & Khoury (2012) proposed service composition of Cloud SaaS by optimizing the selection criteria such as selection based on maximizing the positive factor such as the utility or availability factors, while minimizing the negative criteria such as total cost of service composition (He et al., 2012; Wu & Khoury, 2012). However, most previous works only concern is to reduce the search space when selecting a composite service. That means they have developed service composition based exclusively on addressing the optimization problem when selecting a composite service of Cloud SaaS. In addition, the reputation factor when selecting a composite service plays a crucial role in selecting a composite service for the consumer. Nevertheless, the reputation factor is largely overlooked in the selection process in the extant literature.

In this chapter, the Find SaaS GA TOPSIS is proposed to address the selection mechanism of Cloud SaaS service composition. We have addressed three issues related to the service composition of Cloud SaaS. Firstly, the reputation factor of service composition, we address this factor using fuzzy logic. To do that, we used the service rating and service reviewers as the system inputs and generated the reputation-based service reviewers (RbSR) using the fuzzy rule system by creating the fuzzy associative matrix (FAM) method.

Secondly, we addressed the optimization problem using the genetic algorithm (GA) for minimizing the total cost of composite service and maximizing the repu-

tation factor (RbSR).

Thirdly, after obtaining the list of service composition that generates from the optimization method, known as the (Pareto solution), the decision-maker system is used to select the optimum composite service that meets the service consumer's preferences. In order to do that, the TOPSIS method is applied to rank the composite services in order to select the best composite service among all the composite services generated by the optimization method.

The case study and experiment are introduced to demonstrate and evaluate our research approach for service composition. We have compared and evaluated our proposed approaches with multiple other composition approach, and nDCG metric is used for measuring the performance.

The remainder of the chapter is organized as follows: Section 7.2 describes the main framework architecture of Find SaaS GA TOPSIS, Section 7.3 describes the consumer request handler (CRH) which is the first part of Find SaaS GA TOPSIS. The service registry repository (SRR) is discussed in Section 7.4. Section 7.5 introduces the optimizer unit (OU) which is used to address the optimization problem of Find SaaS GA TOPSIS, Section 7.6 describes the decision-maker methods in Find SaaS GA TOPSIS. The experiment part is presented in Section 7.7. Section 7.8 provides the discussion part of our system framework. Finally, Section 7.9 concludes with a summary.

7.2 Find SaaS Composition GA TOPSIS Framework Architecture

We introduce a Find SaaS GA TOPSIS framework architecture to help a Cloud SaaS service consumer to select an optimum service composition. The proposed system architecture of Find SaaS GA TOPSIS is shown in Figure (7.1). The proposed

approach consists of four main components, as follows:

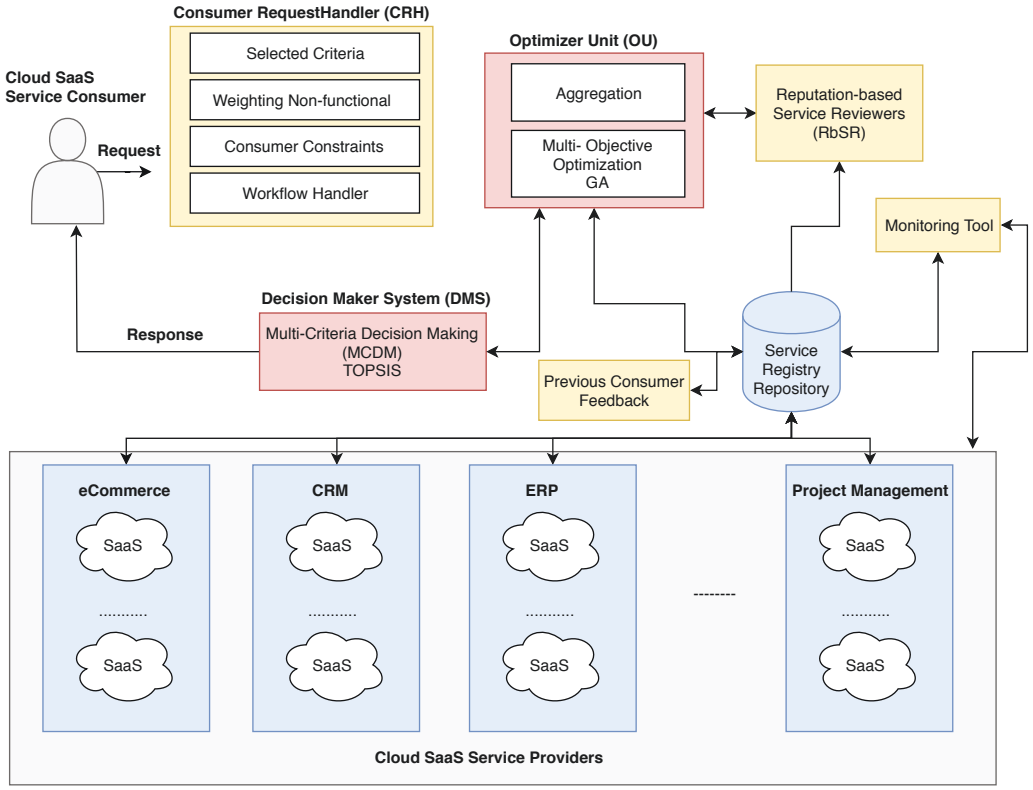


Figure 7.1 : The Find SaaS Composition Framework Architecture

(Section 7.3) **Consumer Request Handler (CRH)**: This component is responsible for gathering all the information required from service consumer to apply ranking the composition of cloud software services.

(Section 7.4) **Service Registry Repository (SRR)**: This part is responsible for retrieving all the services to the optimization unit (OU) based on the consumer's requests for functional software service. All the classifications are organized based on section (5.3) in the ranking system part.

(Section 7.5) **Optimizer Unit (OU)**: This component is responsible for the optimization process. The optimization is based on minimization of the total cost of cloud service composition and maximization of reputation-based service reviewers. The optimization process is carried out using the Genetic Algorithm (GA) optimization method.

(Section 7.6) **Decision Maker System (DMS)**: This component is responsible for ranking all service compositions using the TOPSIS MCDM approach. The decision maker uses the optimum services of composition generated by the optimization process or as known as the global Pareto optimal set (or the Pareto set).

7.2.1 System Notation

This System contains the following parameters as system input and output for the Find SaaS Compositions GA TOPSIS model.

Table 7.1 : System notations

Notation	Meaning
$Cost_i, j$	Service cost for abstract service (i), and concrete service (j)
$RbSR_i, j$	The reputation-based service reviewers for abstract service (i), and concrete service (j)
AB_i	A set of Abstract service requested by a service consumer matching the functional workflow
CS_i, j	A set of concrete service for matching an abstract service (i), and service candidate (j)
$NF^k_{i,j}$	A set of non functional parameters,for abstract service (i), concrete service (j) and criteria number (k)
CP_j	A set of selected criteria defined by a service consumer
epl_k	Execution path
W_{CP_j}	A set of weighting selected criteria by a service consumer
$Cons^{Cost/RbSR}_{min/max}$	Minimum and maximum consumer's constraint of (Cost - Total Cost of service composition) and (RbR - Reputation-based service reviewers)
LSC_i	List of service composition after optimization process
RSC_i	List of service composition after decision maker system (DMS) process
$Fitness_{Cost}$	The fitness function for the cost attribute
$Fitness_{RbSR}$	The fitness function for the reputation-based service reviewers

7.2.2 System Process

Find SaaS GA TOPSIS works in three stages to rank composite services for selection based on the consumer's preferences. Table 7.2 shows the proposed stages

to select the optimum Cloud SaaS service composition.

Table 7.2 : The Find SaaS GA TOPSIS interacting stages

Stage	Input	Process	Output
Weighting consumer' preferences	$Ling(CP_j)$	Fuzzy	W_{CP_j}
Multi-Objective Optimization	$Cons_{min/max}^{Cost/RbSR}$, AB_i, $NF_{i,j}$	GA	LSC_i
	$RV_{i,j}$, $RA_{i,j}$	Fuzzy	$RbSR_{i,j}$
Decision Making	W_{CP_j}, LSC_i	TOPSIS	RSC_i

7.3 Consumer Request Handler (CRH)

In order to find the best Cloud SaaS service composition for the service consumer, there are four main steps required by a service consumer who needs to select and obtains the best service as follows:

(Step 1) Select the primary functional services as known as the objective operational workflow for the Cloud SaaS service providers. The workflow handler is responsible for addressing this step.

(Step 2) Select the main non-functional quality attributes concerns by the service consumer.

(Step 3) Weight the importance of selected non-functional quality attributes to satisfy the preferences.

(Step 4) Provide the service consumer's constraints for the total cost of service composition as well as the average reputation-based service reviewers required to select the cloud software service composition.

7.3.1 Workflow Handler

Workflow describes the process of aggregating multiple Cloud SaaS functional services in order to make a composite of multiple Cloud SaaS providers. The workflow has multiple abstract services with different invocation relations between them. Once the process is complete these are known as the service composition. Each abstract service has multiple concrete candidate services having similar functional characteristics. According to the literature review, four types of composition structures are found: *sequence, branch, loop* and *parallel* (D. Wang et al., 2015; Ye, Zhou, & Bouguettaya, 2011). As shown in figure (7.2),there are four control flows provided by a service composition, also several abstract Cloud SaaS services described in each node. For example, AB_1 and AB_2 run in a sequence flows pattern, also, AB_3 runs in parallel with AB_4. There is a conditional flows pattern between AB_5 or AB_6. Finally, there is a loop flows pattern in abstract service number seven. The BPEL platform is widely used to implement a business workflow for some environment (Alrifai & Risse, 2009).

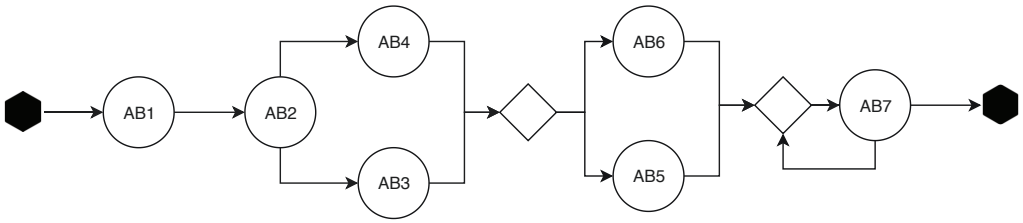


Figure 7.2 : Workflow control process

This research acknowledges the sequential composition model. This model generates abstract services to generate a service composition for confirming the service composition model. This step provides the number of Cloud SaaS services that are required to build a service composition. Let us assume the AB is a set of abstract services for a particular business environment $AB = \{AB_1, AB_2, ..., AB_n\}$, where

AB_1 denotes the abstract service as a task considered by a service consumer in a composite service. For example, when a service consumer needs to select four functional clouds. They require SaaS tasks including marketing SaaS, human resources SaaS, IT SaaS and email SaaS. Then, the service composition model needs to find the optimal composition of Cloud SaaS service providers among thousands of similar services.

Therefore, let us assume $CS_{i,j}$ is the concrete service for task i, and j concrete service number. For this reason, in the first step, our proposed approach (Find SaaS GA TOPSIS) retrieves all the services matching a similar task category. Therefore, the matching algorithm used by (Sajjanhar et al., 2004) has been applied to retrieve all the services required by a service consumer. This method is only used to retrieve all services corresponding with the consumer's requirements in terms of the Cloud SaaS tasks.

7.3.2 Selected Criteria

After gathering all the information being considered by a service consumer in terms of abstract services, a service consumer needs to present their preferences for non-functional quality attributes. For example, a consumer may want to consider the total cost and availability when selecting a composite service. This step is necessary for selecting the service composition considering the consumer's preferences. Let assume, the selected non-functional attribute as follows $sp_1, sp_2, ..., sp_n$. Where sp_1 is, for example, a composite cost of service composition.

7.3.3 Consumer Constraints

Thirdly, consumer's constraints are required by a service consumer for solving the multi-objective optimization (MOO). Find SaaS GA TOPSIS method uses the Genetic Algorithm (GA) to obtain all service compositions or in other words to

get all non-dominated set (or the Pareto set). Therefore, our research concerns the consumer's constraints to determine the consumer's boundary in terms of the total cost of composite services and the reputation-based service reviewers (RbSR) factor. Let us assume that $cons_{min}^{Cost}$ and $cons_{max}^{Cost}$ is the minimum and maximum of composite service formed by a service consumer respectively. Moreover, let us assume that the $cons_{min}^{RbSR}$ and $cons_{min}^{RbSR}$ is the minimum and maximum of reputation-based service reviewers (RbSR) respectively provided by a service consumer.

7.3.4 Weighting Non-Functional Quality Attributes

Finally, the Cloud SaaS service consumer needs to weight their preferences among all selected criteria. Our proposed approach used the TOPSIS method for selecting the composite service matching the consumer's non-functional preferences. We have used the same technique introduced in chapter (5) to weight the consumer's preferences using the linguistic terms. Therefore, seven linguistic values are used for this purpose as follows: Extremely Important (EI), Very Important (VI), Important (I), Somewhat Important (SI), Not Important (NI), Not Very Important (NVI), and Not Important at All (NIA). In addition, if a consumer does not recognize specific criteria, our system will weigh the non-functional as Not Important at All - NIA. Therefore, our system will acknowledge these criteria to have very low weighting. Table 7.4 shows the linguistic terms with a set of fuzzy numbers.

In order to obtain the crisp value of the fuzzy number $\hat{A} = (a, b, c)$,we compute the center of gravity (COG) of $\hat{A}$ using the following formula (7.1):

$$d(\hat{A}) = a + \frac{(a-b) + (c-b)}{3} \tag{7.1}$$

Finally, the weighting the most important criteria for a service consumer (w_i) is calculated using the following formula (7.2):

Table 7.3 : Linguistic terms with their fuzzy numbers

Linguistic terms	Triangular fuzzy number
Extremely Important (EI)	(0.9,1.0,1.0)
Very Important (VI)	(0.7,0.9,1.0)
Important (I)	(0.5,0.7,0.9)
Somewhat Important (SI)	(0.3,0.5,0.7)
Not Important (NI)	(0.1,0.3,0.5)
Not Very Important (NVI)	(0.0,0.1,0.3)
Not Important at All (NIA)	(0.0,0.0,0.1)

$$w_j = \frac{d(\hat{A}_j)}{\sum_{j=1}^{n} d(\hat{A}_j)} \tag{7.2}$$

As a result, the set of weighting attributes w_i $W = w_1, w_2, ..., w_n$ is generated to decision maker system (DMS) in order to find the optimal composite services for a service consumer.

7.4 Service Registry Repository (SRR)

Service registry repository (SRR) contains information about all the Cloud SaaS service providers. Cloud SaaS Service providers publish its services wit the functional and non-functional characteristics. Each service has information in terms of service description, service category and multiple non-functional quality attributes such as a service price. We have classified the non-functional quality attributes into three main types as described in section (5.2) as shown in Figure (7.3): (1) Service providers, (2) monitoring tool and (3) previous consumer feedback.

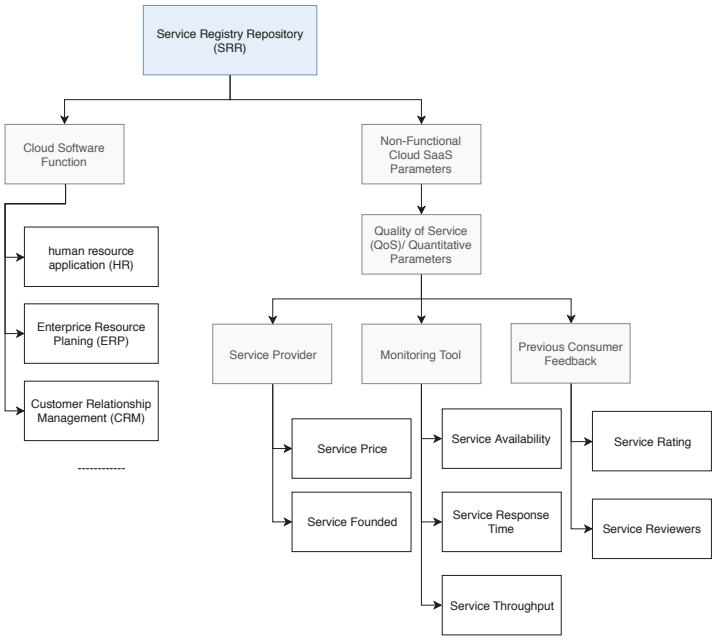


Figure 7.3 : Services Registry Repository

7.4.1 Service Provider

A "service price" has the major crucial factor that a service consumer considers when selecting a service. There are different categories of payment methods of Cloud SaaS providers, such as monthly/annual payment methods. However, it may have some services provide their service based on pay as-consumption payment concept. The majority of Cloud SaaS service provider provide their services based on the size of service consumer such as individual user/ business user. Also, "service founded" is the second factor may consider by a service consumer when selecting a service provider. A service consumer may prefer to choose an old service rather than a new service, since, the old service having a big community and multiple documentation can be found in public.

7.4.2 Previous Consumer Feedback

Two main factors or quality attributes will be collected to present the previous consumer's feedback: (1) "Service Rating" (2) "Service Reviewers". Service rating is in the range of 1 to 5, where 5 indicates a high rating and 1 shows the low rating of service. Furthermore, service reviewers indicate the number of the consumer has evaluated or rate the service. Usually, services with high popularity have a higher number of service reviewers.

7.4.3 Monitoring Tool

The monitoring tool is used for gathering three crucial quality attributes for evaluating the service provider. (1) "Service Availability (AV)" refers to the number of successful invocations divided by total invocation (measured as a percentage). (2) "Service Response Time (RS)" refers to the time taken to send a request and receive a response (measured as millisecond). (3) "Service Throughput" refers to the number of invocations for a given time (measured by a bit per second "bps"). In addition, there are a number of third-party monitoring tools available that can scale these attributes ideally, such as site24×7.

7.5 Optimizer Unit (OU)

Find SaaS GA TOPSIS is a technique to rank and sort composite services of Cloud SaaS using the Genetic algorithm (GA) and TOPSIS method as described in figure (7.4). The genetic algorithm is used to solve the optimization problem. Previous research has shown that the selection of service composition is an NP-hard problem, which means that no optimum solution can be obtained. Therefore, the GA is an algorithm that has been used for this research to solve this issue. The main purpose of the optimizer unit (OU) is to reduce the search space to select the optimum service composition.

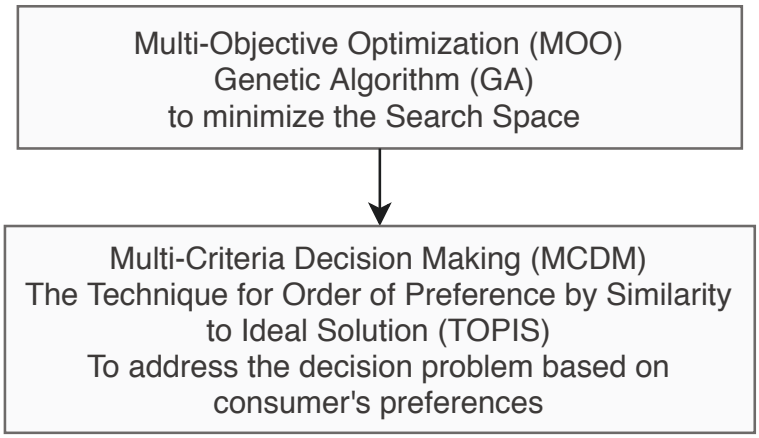


Figure 7.4 : Find SaaS GA TOPSIS main algorithms

Our proposed approach has two objectives which can be described as a multi-objective optimization (MOO) problem. The multi-objective optimization refers to the problem having multiple conflicts objectives when making a selection decision. For example, solution x obtains the best solution or called dominates solution y in terms of all objectives. However, if x does not dominate y, also, in the same way, y does not dominate x, in this case, it will be referred to as non-dominated solutions or Pareto set. Many mathematical algorithms can be applied to solve this issue, such as using evolutionary programming such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) or Genetic Algorithm (GA). Furthermore, mixed- integer programming (MIP) can solve this problem. However, the Genetic Algorithm (GA) is very widely used for solving the multi-objective optimization problem, and have been applied for a different domain such as health sector (Zhang et al., 2016) and fingerprint recognition (Oliveira, Sabourin, Bortolozzi, & Suen, 2002).

Find SaaS GA TOPSIS is used for ranking and sorting the composite service of Cloud SaaS in order to satisfy consumer's constraints of minimizing the total cost of

composite service and maximizing the reputation-based service reviewers (RbSR).

7.5.1 Total Composite cost

The *Total cost* of service composition is the total amount of service composition for using the Cloud SaaS service. In fact, selecting the composite service that has low cost may play a high priority in order to adopt the Cloud SaaS instead of building its software service (Safari et al., 2015). Therefore, we proposed our approach to select composite services by minimizing the total cost of service composition.

7.5.2 Reputation-based Service Reviewers (RbSR)

Reputation-based service reviewers (RbSR) is the second factor involved in our optimization problem. The reason that we create this factor is to measure the reputation based on service reviewers. We find that some services have low number of reviewers and having high service rating. Moreover, some services have high reviewers and have average rating (such as 3 out of 5). Therefore, we will address this issue by developing a fuzzy logic system. The fuzzy logic is a rule-based system taken into account service rating and service reviewers as the input and generating the reputation-based service reviewers (RbSR) as the system output. The service rating is within the range of [0,5] and has shaped as a triangular fuzzy membership as shown in figure (7.5). Table 7.4 shows the three linguistic values (Few, Average and High) with their fuzzy numbers described the service rating attribute as the linguistic variable in our fuzzy system. On the other hand, the service reviewers are within the range of [min, max] for the functional service category. For example, if we consider selection service in CRM functional category, and the range of service reviewers from 25 to 1330 (it means that the minimum service reviewers for example in service x have got 25, and the maximum service reviewers for instead service y have got 1330). We have classified the service reviewers into four linguistic values

as the triangular fuzzy memberships, as shown in Figure (7.6). Table 7.5 shows the service reviewers fuzzy number with their linguistic values. Finally, the reputation-based service reviewers (RbSR) as the output of our fuzzy system, we have classified as four linguistic terms (Low, Medium, High and Very High) as shown in table 7.6. Also, the Trapezoidal fuzzy shape is proposed for representing the RbSR as shown in Figure (7.7).

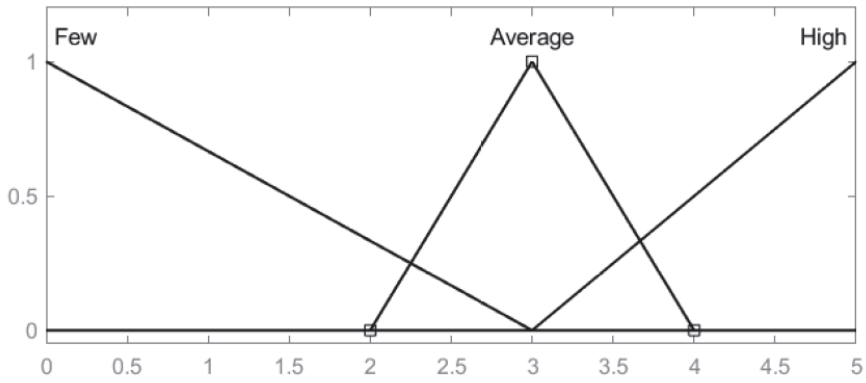


Figure 7.5 : Membership functions of linguistic values for Service rating (RA)

Table 7.4 : Linguistic values of Service Rating (RA) and their fuzzy numer ranges

Linguistic Values	Fuzzy numbers
Few (F)	(0,0,3)
Average (A)	(2,3,4)
High (H)	(3,5,5)

In order to construct fuzzy rules for our system, the fuzzy associative matrix (FAM) is used for this purpose. FAM is introduced by Turksen ea al. (1992), and it has been widely used for different fuzzy scenario in order to build fuzzy rules (Azzam, Leung, & Horwood, 2005). Therefore, in our proposed system, there are two inputs and one output. The first input, which is the service rating (RA) has

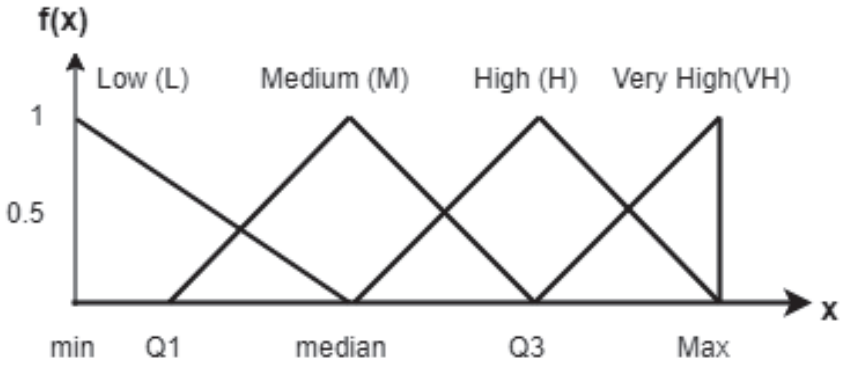


Figure 7.6 : Membership functions of linguistic values for Service Reviewers (RV)

Table 7.5 : Linguistic values of service reviewers (RV) and their fuzzy numbers range

Linguistic Values	Fuzzy numbers
Low (L)	(min,min,median)
Medium (M)	(Q1,median,Q3)
High (H)	(median,Q3,max)
Very High (VH)	(Q3,max,max)

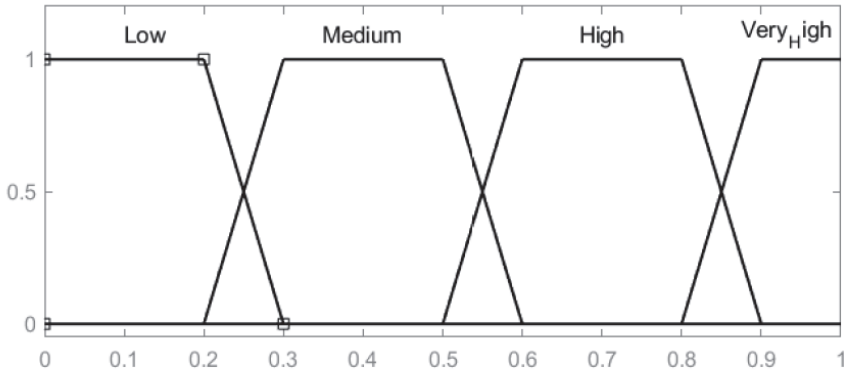


Figure 7.7 : Reputation-based service reviewers

Table 7.6 : Linguistic values of reputation-based service reviewers (RbSR) and their fuzzy numbers range

Linguistic Values	Fuzzy numbers
Low (L)	(0,0,0.2,0.3)
Medium (M)	(0.2,0.3,0.5,0.6)
High (H)	(0.5,0.6,0.8,0.9)
Very High (VH)	(0.8,0.9,1,1)

three linguistic values, and the second input, which is the service reviewers (RV) has four linguistic values. As a result, twelve rules have been constructed by used very sample method 3×4. Table 7.7 contains all the rules used for measuring the reputation-based service reviewers (RbSR).

7.5.3 Aggregation Function

The optimizer unit (OU) manages all candidate services generated by matching the abstract services. Therefore, after gathering all candidate services, the optimizer unit delivers the set of optimal composite service as an execution plan epl^k according to non-functional quality attributes. The execution plan created for Cloud SaaS consumer must meet the consumer's constraints of total cost and RbSR as well as consumer's non-functional preferences. An execution plan is a sequence of services in service composition, from the initial service to the last service. As mentioned in the service registry repository, we have considered the following non-functional attributes: service cost, service rating, service reviewers, response time, availability and reputation-based service reviewers (RbSR). We have used the aggregation functions as (Y. Wang et al., 2015; He et al., 2012). Table 7.8 shows the aggregation of non-functional qualities for all composite services.

Table 7.7 : The square rules (FAM) representation

Rule	RV	RA	RbSR
1	Low	Few	Low
2	Low	Average	Low
3	Low	High	Average
4	Medium	Few	Low
5	Medium	Average	Average
6	Medium	High	High
7	High	Few	Low
8	High	Average	Average
9	High	High	High
10	Very High	Few	Low
11	Very High	Average	High
12	Very High	High	Very High

Table 7.8 : Non-functional aggregation functions for sequence patterns

Quality Parameter	Aggregation Function
Cost	$epl_k^{Cost} = \sum NF_{i,j}^{Cost}$, $i \in \{1, 2, ..., AB_n\}$ and $j \in \{1, 2, ..., CS_m\}$
Rating	$epl_k^{RA} = \frac{1}{m} \sum NF_{i,j}^{RA}$, $i \in \{1, 2, ..., AB_n\}$ and $j \in \{1, 2, ..., CS_m\}$
Reviewers	$epl_k^{RV} = \frac{1}{m} \sum NF_{i,j}^{RV}$, $i \in \{1, 2, ..., AB_n\}$ and $j \in \{1, 2, ..., CS_m\}$
Availability	$epl_k^{AV} = \prod NF_{i,j}^{AV}$, $i \in \{1, 2, ..., AB_n\}$ and $j \in \{1, 2, ..., CS_m\}$
Response Time	$epl_k^{RT} = \sum NF_{i,j}^{RT}$, $i \in \{1, 2, ..., AB_n\}$ and $j \in \{1, 2, ..., CS_m\}$
Throughput	$epl_k^{TH} = minNF_{i,j}^{TH}$, $i \in \{1, 2, ..., AB_n\}$ and $j \in \{1, 2, ..., CS_m\}$
RbSR	$epl_k^{RbSR} = \frac{1}{m} \sum NF_{i,j}^{RbSR}$, $i \in \{1, 2, ..., AB_n\}$ and $j \in \{1, 2, ..., CS_m\}$

7.5.4 Genetic Algorithm (GA)

The Genetic Algorithm (GA) is used to address the multi-objective optimization (MOO) in order to reduce the search space for selecting the optimum Cloud SaaS Composition. We will describe the overview of Genetic algorithm, then explain and demonstrate how we will apply the algorithm in our scenario problem.

Genetic Algorithm (GA): Overview

The Genetic Algorithm (GA) is introduced in the early 1970s by John Holland (Holland et al., 1992). GA works based on biological evolution and can be used to find the optimal solution in an ample search space. The basic concept of GA can be represented as in Figure (7.8). GA go through several steps as follows:

Step (1) Define the chromosome structure and the size of population N, the crossover probability P_c and the mutation probability P_m.

Step (2) Define the fitness function to measure and evaluate the performance of individual chromosome in problem domain.

Step (3) Start randomly generating a population. In this step, each chromosome is eliminated or duplicated for one or more times based on its relative quality. The population size is typically kept fixed $x_1, x_2, ..., x_N$.

Step (4) Evaluate the performance by calculating the fitness of each individual chromosome $f(x_1), f(x_2), ..., f(x_N)$.

Step (5) Generate a pair of offspring chromosomes by applying crossover and mutation operators.

Step (6) Place the created offspring chromosomes in new population.

Step (7) Evaluate the new population.

Step (8) Repeat step (5) till the end of iterations or generations.

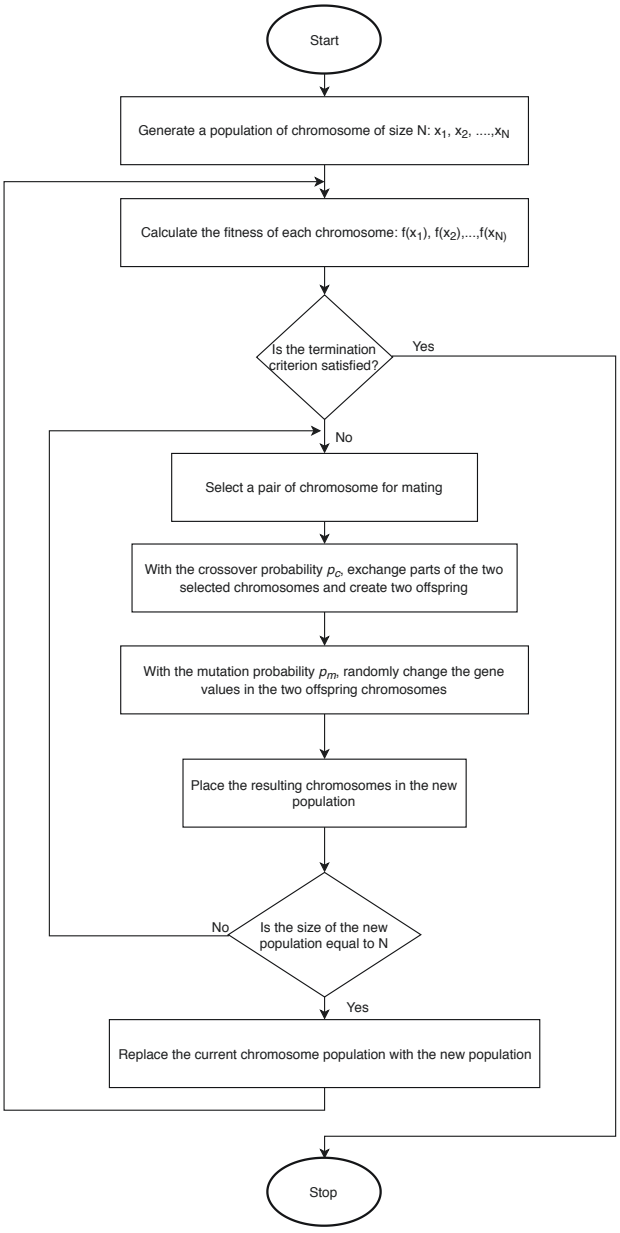


Figure 7.8 : The basic genetic algorithm flowchart

Genetic Algorithm for Cloud SaaS Composition

The Genetic Algorithm (GA) will be used to address the multi-objective optimization (MOO) problem to find the composite services that fulfilment service

consumer's constraints. The output result for this step is the list of all composite services match with the consumers' constraints requirements LSC_i (Feasible solution/Pareto set/on-dominated solution).

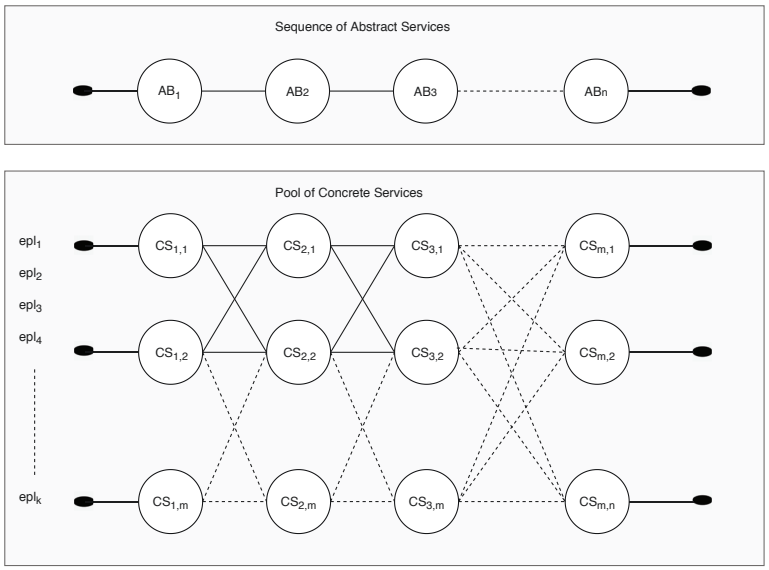


Figure 7.9 : The Service Composition Model

Suppose that a composite service $AB = \{Ab_1, AB_2, ..., AB_n\}$ has n abstract component services. In each abstract service AB_i can be bound to one of the m concrete services $CS_{i,1}, CS_{i,2}, ..., CS_{i,m}$ as shown in Figure (7.9), which are having similar functionalities but with different non-functional attributes (such as service cost, service rating, service reviewers, service availability and service response time).

We assume that a service consumer requests a service composition with minimum and maximum composite cost constraint $Cons^{Cost}_{min,max}$, as well as minimum and maximum reputation-based service reviewers (RbSR) composite constraints $Cons^{RbSR}_{min,max}$. In fact, our GA algorithm focuses on selection the optimum concrete service for each abstract service from the existing concrete candidate in order to fulfilment the

consumer's constraints by minimizing the total cost of composite services and maximizing the RbSR. Hence, two main objectives optimization developed for selecting the best composite baths having the minimum cost $Cost^k_{epl}$ and the maximum reputation based service reviewers $RbSR^k_{epl}$. Equation (7.3) represents the objective functions by maximizing the $RbSR$ and minimizing the total cost of composite service. Furthermore, equation (7.4) express consumer's constraints of $RbSR$ with the total cost.

$$\begin{aligned} max(\sum_{i=1}^{n}\sum_{j=1}^{m} epl^k(RbSR_{(i,j)})), epl^k \in L \\ min(\sum_{i=1}^{n}\sum_{j=1}^{m} epl^k(Cost_{(i,j)})), epl^k \in L \end{aligned} \tag{7.3}$$

$$\begin{aligned} Cons^{RbSR}_{min} \leqslant RbSR_{i,j} \leqslant Cons^{RbSR}_{max} \\ Cons^{Cost}_{min} \leqslant Costi,j \leqslant Cons^{Cost}_{max} \end{aligned} \tag{7.4}$$

where $epl_k \in \{1, 2, ..., L\}$ are execution path of composite services of abstract service i, and candidate concrete service j.

Fitness Function

As a part of GA is define the fitness function. The fitness function is used to evaluate selected chromosome performance, either good or bad. Therefore, for our problem, we apply simple function concerned with constraint violation for maximum and minimum of total cost and RbSR. Since we have two main objectives which are the cost of service composition and the RbSR, therefore, two equations build to measure the performance of the chromosome. Firstly, Equation 7.5 is used for measuring the performance in terms of cost of service composition and how the chromosome positions are for a service consumer's constraints. The result of this

equation is the range in the range up to the value one, where (1) describes optimum composite services fullfillment consumer's constraint requirments of the service cost of scomposition. On the other hand, equation (7.6) is proposed for measuring the performance of composite service in terms of RbSR. In addition, the result of the equation is in the range of [0,1], where (0) shows optimum composite selection, and (1) is the opposite.

$$F_{Cost} = \frac{Cons_{max}^{Cost} - \sum_{i=1}^{n}\sum_{j=1}^{m} epl_k(Cost_{i,j})}{Cons_{max}^{Cost} - Cons_{min}^{Cost}} \tag{7.5}$$

$$F_{RbSR} = \frac{Cons_{max}^{RbSR} - \sum_{i=1}^{n}\sum_{j=1}^{m} epl_k(RbSR_{i,j})}{Cons_{max}^{RbSR} - Cons_{min}^{RbSR}} \tag{7.6}$$

Finally, we calculate the fitness function by measuring the differences between F_{Cost} and F_{RbSR}, as shown in equation (7.7). Where $Fitness \in [-1, 1]$ is a constant, denotes the rate of chromosome for a composite service. The chromosome with (1) value shows an optimum solution, and (-1) shows bad fitness value. As mentioned before, the result will be Petro set or LSC_i.

$$Fitness = F_{Cost} - F_{RbSR} \tag{7.7}$$

Figure (7.10) and (7.11) show the proposed chromosome structure with crossover operation and mutation operation, respectively. The size of the chromosome of our problem in the case study in section 7.7.1, consists of 5-bit string describes 20 services.

7.6 Decision Maker System (DMS)

The TOPSIS method will be used to rank composite services generated by GA. As mentioned before, the list of composite services is generated as matched with

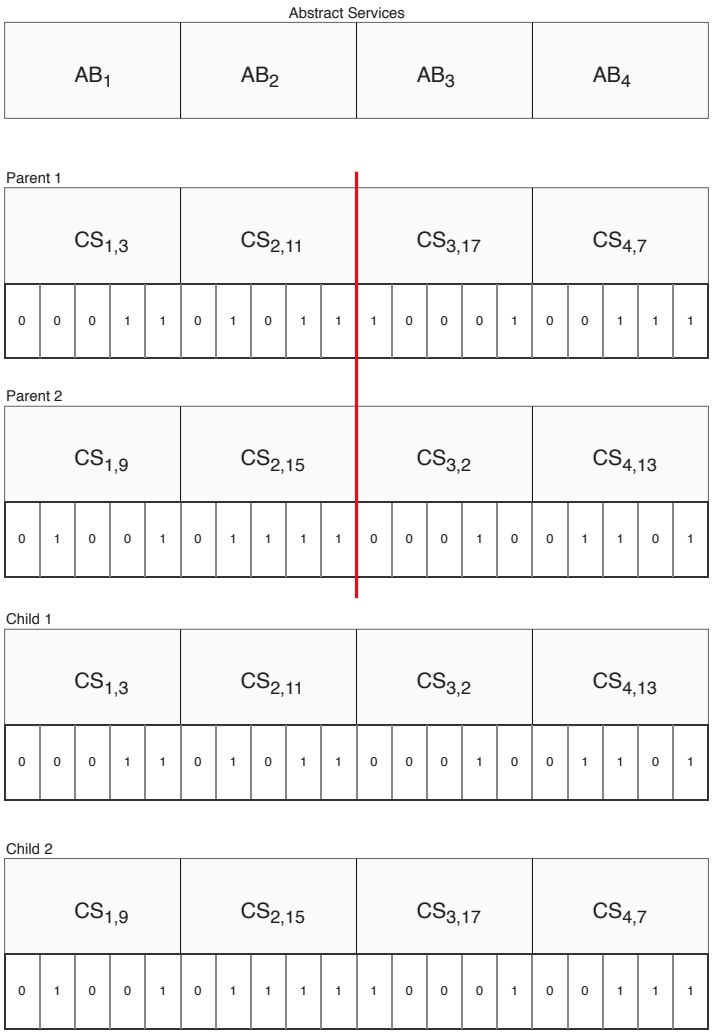


Figure 7.10 : Genetic crossover operator for Cloud SaaS Composition

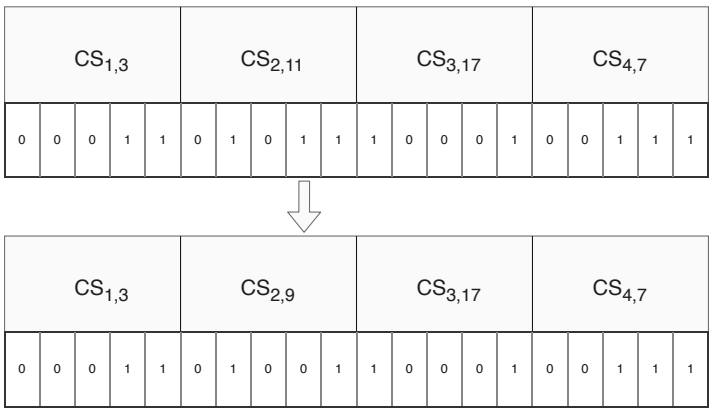


Figure 7.11 : Genetic mutation operator for Cloud SaaS Composition

consumer's constraints regarding the cost of composite services and the RbSR level. Our proposed method is using multi-criteria decision making for helping the consumer to select the optimum composite service match with consumer's preferences. TOPSIS is a multi-criteria decision-making approach developed by Hwang and Yoon in (1981) (Yoon & Hwang, 1995). The basic functionality of the TOPSIS method is to sort and rank all options or alternatives, beginning with the option that has the shortest distance from the positive ideal solution to farthest distance from a negative ideal solution. TOPSIS has the following steps in order to find and rank composite services to its consumer request:

Step 1: Compute the normalised decision matrix.

There are two types of criteria/non-functional : (1) Equation (7.8) is used for normalized the positive vector such as service availability. (2) Equation (7.9) is used for normalized the negative vector such as service price.

$$r_ij = \frac{x_ij}{\sqrt{\sum_{i=1}^{m} x_ij^2}} \tag{7.8}$$

$$r_ij = \frac{1/x_ij}{\sqrt{\sum_{i=1}^{m} 1/x_ij^2}} \tag{7.9}$$

Step 2: Compute the weighted normalised decision matrix

The following calculation formula (Equation 7.10) is used for this purpose.

$$v_ij = w_j r_ij \tag{7.10}$$

Step 3: Identify the positive-ideal (A^*) and negative-deal (A^-) solutions

The (A^*) and (A^-) are determined in terms of weighted normalized vector.

$$\begin{aligned} &A^* = v_1^*, v_2^*, ..., v_j^*, ..., v_n^* = \\ &\{(\underset{i}{max}\ v_{ij} | j \in J_1), (\underset{i}{min}\ v_{ij} | j \in J_2) |\ i = 1, ..., m\} \end{aligned} \tag{7.11}$$

$$\begin{aligned} &A^- = v_1^-, v_2^*, ..., v_j^-, ..., v_n^- = \\ &\{(\underset{i}{min}\ v_{ij} | j \in J_1), (\underset{i}{max}\ v_{ij} | j \in J_2) |\ i = 1, ..., m\} \end{aligned} \tag{7.12}$$

Step 4: Compute the distance of each alternative to the positive-ideal solution vector (S_i^*), as well as to the negative-ideal solution vector (S_i^-). Equation (7.13) is used to measure the distance to the positive ideal solution vector. Moreover, equation (7.14) is used to measure the distance to the negative-ideal solution.

$$S_i^* = \sqrt{\sum_{j=1}^{n} (v_{ij} - v_j^*)}, i = 1, ..., m \tag{7.13}$$

$$S_i^- = \sqrt{\sum_{j=1}^{n} (v_{ij} - v_j^-)}, i = 1, ..., m \tag{7.14}$$

Step 5: Compute the similarities to positive-ideal solution (C_i^*) by using equation (7.15)

$$C_i^* = \frac{S_i^-}{S_i^* + S_i^-} \tag{7.15}$$

Note that: $C_i^* \in [0,1]$. ($C_i^* = 0$ when $A_i = A^-$), and (C_i^*= 1 when $A_i = A^*$)

Step 6: Rank composite services beginning with the composite service that has a maximum C_i^*.

The result of this final step is the list of composite services RSC_i.

7.7 Experiment

This section presents the experimental evaluation of Find SaaS GA TOPSIS. It includes a case study of composite selection service for business sector called EMAR Ltd Company *. We have collected the services in this experiment from Capterra web site †. Also, R language has been used to develop coding and obtain the result graphs. The experiments were conducted on machine running MAC OS X4 640 3.00GHz CPU and 8 GB RAM.

7.7.1 Case study: Emar Ltd Company

EMAR is a Saudi civil and electrical construction company founded in 2005 by Rami Ekram. Emar has multiple sectors dependent on its construction work: energy, project management, maintenance, power planet, air conditioning and firefighting. Emar uses on-premise software service hosted locally. This software has four main services as follows: CRM, project management, energy and architecture (see figure 7.12). Therefore, four Cloud SaaS service providers should be composed in order to adopt cloud services for the Emar company. The selection of service composition should match with Emar's preferences of service cost, service rating and service

*www.saudiemar.com

†www.capterra.com

reviewers. Also, the selection of composite service should minimize the cost of composite and maximize the RbSR level.

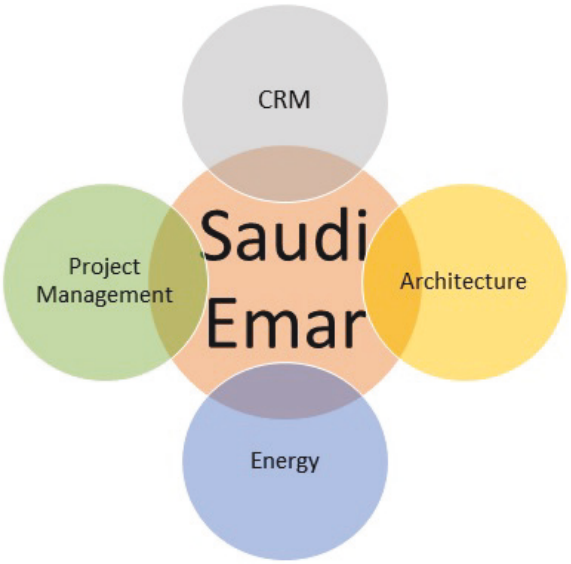


Figure 7.12 : The first main page of ranking services of Find SaaS

7.7.2 Experiment Setup

We evaluate Find SaaS GA TOPSIS using a real scenario of composite selection of Cloud SaaS service providers for Emar company. The dataset is collected from Capterra web service with the top 20-services from each functional category (Project management, Architecture software, energy management software and customer relationship management software (CRM)). Therefore, the number of execution paths for our problem is $20^4 = 160,000$ for all composite services. Figure (7.13) shows the the execution paths of RbSR level for all available composite services. Also, Figure (7.14) shows the execution paths of the total cost for all composite services. Our methods reduce all those paths for selecting the composite services that have maximum RbSR and minimum cost. All experiments were taken on MAC OS 3 GHz Intel Core i5, and RStudio is used to develop all experiments and the results.

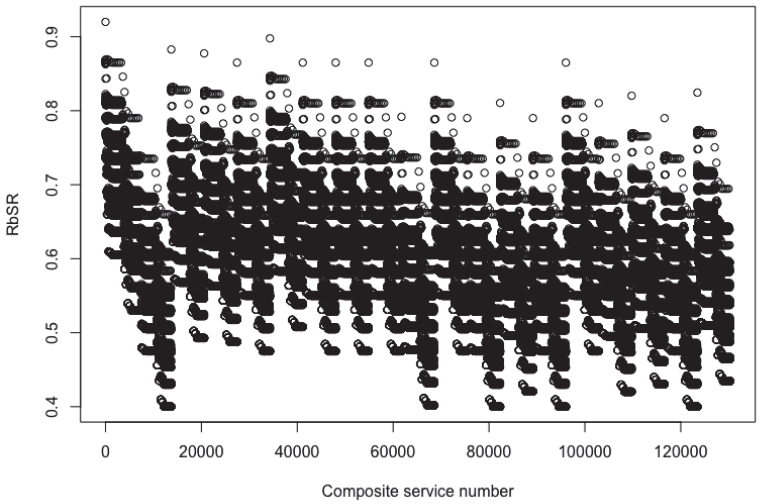


Figure 7.13 : The execution paths of RbSR

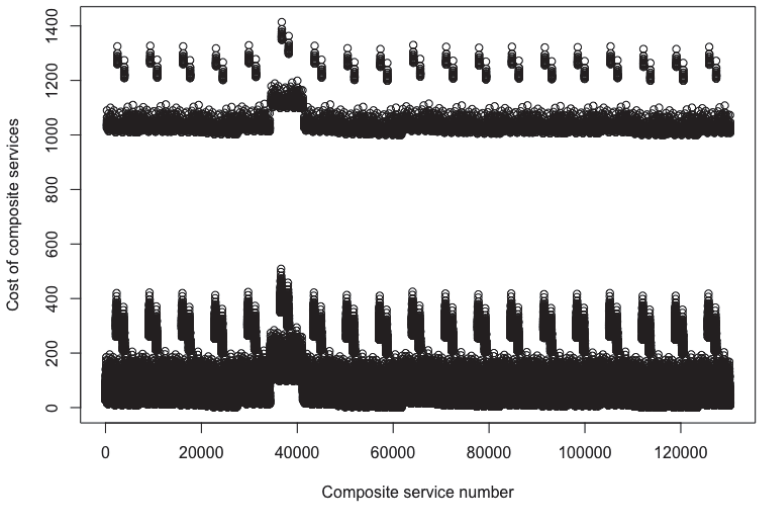


Figure 7.14 : The execution paths of composite costs

The list of services is presented in the Appendix section at the end of this thesis. Also, we have developed a fuzzy logic system using the Fuzzy logic toolbox by Matlab. In our experiment, we consider the following non-functional quality attributes: service rating, service reviewers, service cost, service availability, service response time and service throughput. Table 7.9 shows Emar preferences among all criteria. Also, table 7.10 shows the GA parameters including Emars constraints for the cost of the composite as well as the RbSR level, in addition to this, the crossover and mutation probabilities and the population size.

Table 7.9 : Non-functional preferences input

Criteria	Linguistic Weight	Crisp Weighting
Composite Rating	I	0.1640625
Composite Reviewers	SI	0.1171875
Composite Cost	EI	0.2265625
Composite Availability	I	0.1640625
Composite Response Time	I	0.1640625
Composite Throughput	I	0.1640625

7.7.3 Experiment Result

As shown in figure (7.15) the fitness function value for each iteration is around 0.22. Therefore, in the first step and in order to obtain all composite services that have minimum cost and maximum RbSR and based on GA algorithm all unwanted paths that do not meet our parameter will be excluded from the list. The result of this method is the Petro set solution.

Table 7.10 : Consumer Constraints

Genetic Algorithm Parameters	
Cons(RbSR(max))	0.9
Cons (RbSR(min))	0.5
Cons(Cost(max))	80
Cons(Cost(min))	50
Mutation	0.1
Crossover	0.8
Population Size	100

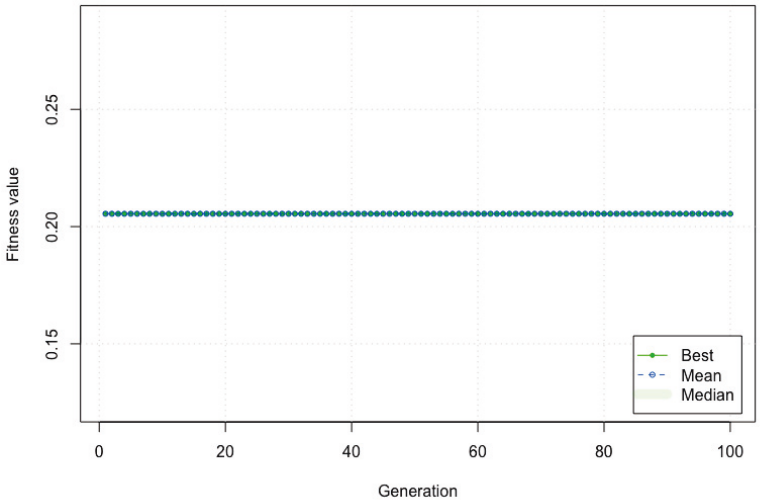


Figure 7.15 : The Fitness function

As you can see from the table 7.11, the top 20-composite services that are obtained by the genetic algorithm. The last column represents the utility value calculated by measuring the differences between the consumer's constraints with each composite parameter. The equation (7.16) is used to calculate the utility value for each composite service. Moreover, the Euclidean distance is calculated to measure the distance to the consumer's constraints of RbSR and the total cost of service composition.

$$utility(f_i) = 1 - | \sum_{j=1}^{n} (x_i, j \times w_j) - (dist(cons_{max}^{RbSR}, RbSR_i) + dist(cons_{min}^{Cost}, Cost_i)) | \tag{7.16}$$

After taking into account the other non-functional criteria of the consumer's preferences, the table 7.12 shows the top-20 composite services obtained by using the GA optimization method plus the TOPSIS MCDM method. As you can see the positions of composite services for the two tables are different. That is because other non-functional attributes such as service rating and service availability are taken into account in the second table. For example, the composite service number (i.e. execution path) (20578) has a position on the top of composite services ranked #1 by the GA TOPSIS approach. On the other hand, the same composite number ranked #18 in the optimization method (GA).

In order to compare our result with other approaches, the decision-maker two common methods SAW and TOPSIS are used to find the top 20 composite services. The result of these two methods is obtained after the aggregation mechanism. That means there are no consumer's constraints involved in these two results. Table 7.13 shows the result of composite services using the SAW with their utility value for each composite path. Moreover, the table 7.14 represents the top-20 composite services using the TOPSIS method alone. The results show that both methods are showing

Table 7.11 : The Top-20 Composite services generated by GA

Composite Number	RbSR	Cost of Composite	Composite Rating	Composite Reviwers	Utility Value
48014	0.865	68	4.5	3769.5	0.767498725
1084	0.86525	67.94	4.5	6181.25	0.769217278
3	0.865	66.99	4.375	4949.75	0.777937646
5	0.865	69.99	4.5	4652.25	0.748053507
2889	0.865	69.94	4.375	6153.25	0.748803882
3250	0.865	60.09	4.5	6149.5	0.847200889
1806	0.865	60.09	4.5	6167.75	0.847317953
6	0.865	59.99	4.5	4577.5	0.84802815
4	0.865	55.09	4.5	4671.75	0.896780776
8	0.865	55.09	4.375	4242.5	0.89692224
10	0.865	55.09	4.375	4239	0.896983611
27437	0.865	79	4.5	3814.75	0.657569234
7	0.865	77.99	4.375	4379.5	0.667922454
41155	0.865	77	4.5	3746.25	0.677380705
1445	0.865	74.99	4.5	6177.5	0.69853955
13719	0.88275	74.99	4.5	5445.5	0.7159854
1	0.92	74.99	4.5	6891.5	0.753497164
20578	0.8775	67.7	4.5	5083	0.783940891
54873	0.865	65.1	4.5	3796.25	0.79654423
2528	0.865	64.97333333	4.5	6153.25	0.798594765

Table 7.12 : The Top-20 Composite services generated by GA TOPSIS

Ranking	Composite Number	Composite Rating	Composite Reviewers	Composite Cost	s_composite_positive	s_composite_negative	consumer_driven_composite	Utility Value
1	20578	4.5	5083	67.7	0.018908991	0.061625128	0.76520522	0.783940891
2	1445	4.5	6177.5	74.99	0.035158434	0.03798214	0.519303276	0.69853955
3	1084	4.5	6181.25	67.94	0.033843784	0.035931051	0.514957161	0.769217278
4	2528	4.5	6153.25	64.97333333	0.033667521	0.035306176	0.511878834	0.798594765
5	10	4.375	4239	55.09	0.0361439	0.035186917	0.493291937	0.896983611
6	1806	4.5	6167.75	60.09	0.035286203	0.034257356	0.492602855	0.847317953
7	8	4.375	4242.5	55.09	0.03671756	0.034637168	0.485422189	0.89692224
8	6	4.5	4577.5	59.99	0.035791994	0.033201835	0.481229054	0.84802815
9	1	4.5	6891.5	74.99	0.037045073	0.033984103	0.47845273	0.753497164
10	2889	4.375	6153.25	69.94	0.036801784	0.032365797	0.46793305	0.748803882
11	5	4.5	4652.25	69.99	0.037116226	0.031482666	0.458938406	0.748053507
12	3	4.375	4949.75	66.99	0.040669238	0.030251862	0.426556584	0.777937646
13	7	4.375	4379.5	77.99	0.040428174	0.029846438	0.424711526	0.667922454
14	13719	4.5	5445.5	74.99	0.041955841	0.029659714	0.414151844	0.7159854
15	3250	4.5	6149.5	60.09	0.04317438	0.027355404	0.387856059	0.847200889
16	54873	4.5	3796.25	65.1	0.047825158	0.021568963	0.310818306	0.79654423
17	4	4.5	4671.75	55.09	0.055375687	0.02233168	0.287381765	0.896780776
18	27437	4.5	3814.75	79	0.051352716	0.018400276	0.263791927	0.657569234
19	48014	4.5	3769.5	68	0.053438174	0.015866745	0.228941111	0.767498725
20	41155	4.5	3746.25	77	0.065069619	0.011629386	0.151623682	0.677380705

similar results.

The next section provides the evaluation of the result of Find SaaS GA TOPSIS and comparing with other approaches GA, SAW and TOPSIS.

7.7.4 Experiment Evaluation and Discussion

In order to evaluate the Find SaaS GA TOPSIS with other approaches, the nDCG metric is used for this purpose. The nDCG metric is a ranking performance metric commonly used for measuring performance for the ranking system and decision-maker system. Table 7.15 displays the evaluation performance for Find SaaS GA TOPSIS, GA, SAW and TOPSIS approaches. The evaluation performance is measured for the top 5,10,15 and 20 services. We can summarize the main findings into the following points:

1. The result shows that the GA and GA TOPSIS achieves a higher ranking performance compared with the other two approaches SAW and TOPSIS.

Table 7.13 : The Top-20 Composite services generated by SAW

Ranking	Composite.Number	RbSR	Composite Rating	Composite Reviewers	Composite Cost	SAW	Utility Value
1	115813	0.5355	4.5	154.5	0.424166667	0.011190358	0.130924957
2	115604	0.58025	4.5	161.25	0.424166667	0.011182512	0.175667111
3	115794	0.5355	4.5	154.5	0.424166667	0.011166223	0.130900823
4	115737	0.58025	4.25	155.25	0.424166667	0.011162897	0.175647496
5	115756	0.559	4.5	155	0.424166667	0.011161201	0.1543958
6	115566	0.5835	4.5	165.25	0.424166667	0.011130735	0.178865334
7	114730	0.54175	4.5	157.25	0.424166667	0.011120029	0.137104628
8	116535	0.5355	4.5	153.75	0.424166667	0.011118533	0.130853132
9	114521	0.5865	4.5	164	0.424166667	0.011107898	0.181842497
10	116326	0.58025	4.5	160.5	0.424166667	0.011106423	0.175591022
11	114711	0.54175	4.5	157.25	0.424166667	0.011102129	0.137086728
12	116516	0.5355	4.5	153.75	0.424166667	0.011100601	0.1308352
13	114673	0.56525	4.5	157.75	0.424166667	0.011098287	0.160582886
14	116478	0.559	4.5	154.25	0.424166667	0.011096753	0.154331353
15	114654	0.5865	4.25	158	0.424166667	0.011093687	0.181828287
16	116459	0.58025	4.25	154.5	0.424166667	0.011092185	0.175576785
17	114483	0.58975	4.5	168	0.424166667	0.011074855	0.185059455
18	116288	0.5835	4.5	164.5	0.424166667	0.011073286	0.185059455
19	115585	0.58125	4.375	162.25	0.424166667	0.011057195	0.176541794
20	115623	0.58025	4.375	159	0.424166667	0.011056757	0.175541357

Table 7.14 : The Top-20 Composite services generated by TOPSIS

Ranking	Composite.Number	RbSR	Service.Rating	Service.Reviwers	Cost	s_composite_positive	s_composite_negative	consumer_driven_composite	Utility Value
1	115604	0.58025	4.5	161.25	0.424166667	0.00232329	0.008699447	0.789227478	0.175667111
2	115813	0.5355	4.5	154.5	0.424166667	0.002331929	0.008701878	0.788656002	0.130924957
3	115737	0.58025	4.25	155.25	0.424166667	0.002335713	0.008701642	0.788381112	0.175647496
4	115794	0.5355	4.5	154.5	0.424166667	0.002351637	0.008700984	0.787232655	0.130900823
5	115756	0.559	4.5	155	0.424166667	0.002355414	0.008700819	0.78696056	0.1543958
6	61000	0.55	4.375	353.25	0.4	0.002494352	0.00917282	0.786207646	0.145008173
7	115566	0.5835	4.5	165.25	0.424166667	0.002377818	0.00869988	0.78535089	0.178865334
8	114521	0.5865	4.5	164	0.424166667	0.002379025	0.008696958	0.785208654	0.181842497
9	116326	0.58025	4.5	160.5	0.424166667	0.002379157	0.008696968	0.785199494	0.175591022
10	60734	0.625	4.5	360.25	0.4	0.00250964	0.009172183	0.785167102	0.219998107
11	114730	0.54175	4.5	157.25	0.424166667	0.002384772	0.008699657	0.78485386	0.137104628
12	116535	0.5355	4.5	153.75	0.424166667	0.002384919	0.008699666	0.784843661	0.130853132
13	114654	0.5865	4.25	158	0.424166667	0.002387763	0.008699489	0.784638887	0.181828287
14	116459	0.58025	4.25	154.5	0.424166667	0.002387912	0.008699498	0.784628515	0.175576785
15	115870	0.50525	4.375	154.25	0.424166667	0.002380766	0.008672957	0.784618648	0.100380216
16	60943	0.58025	4.5	353.5	0.4	0.002520338	0.009171783	0.784441303	0.175231561
17	60867	0.625	4.25	354.25	0.4	0.002523284	0.009171614	0.784240639	0.219955037
18	114711	0.54175	4.5	157.25	0.424166667	0.002400083	0.008699131	0.783761013	0.137086728
19	116516	0.5355	4.5	153.75	0.424166667	0.002400248	0.008699138	0.783749492	0.1308352
20	114673	0.56525	4.5	157.75	0.424166667	0.002403023	0.008699034	0.783551536	0.160582886

Table 7.15 : The Evaluation of Find SaaS GA TOPSIS Compared with other approaches using nDCG metric

Method	nDCG			
	$N = 5$	$N = 10$	$N = 15$	$N = 20$
GA +TOPSIS	0.9564674	0.9506507	0.9537824	0.9450543
GA	0.9935952	0.9569123	0.9334658	0.9450543
TOPSIS	0.9187491	0.9209988	0.9014942	0.8878753
SAW	0.9187491	0.9284374	0.9225578	0.9297252

2. In terms of GA and our proposed approach, because we used the same composite services that were generated by the GA, therefore, our proposed approach has a similar result to that obtained using only the optimization method which is the GA. However, based on the results if we have more than 20 paths, our Find SaaS GA TOPSIS has a better performance metric in comparison to the GA. This is because, in terms of the top 15 composite services, the Find SaaS GA TOPSIS obtains around 95.3% comparing with 93.3% for the GA.

3. The SAW and TOPSIS have similar performance metric since both two methods are the decision-maker approaches.

4. In conclusion, the proposed Find SaaS GA TOPSIS decision system of composite Cloud SaaS taking into account the consumer's preferences provides a good result to assist a Cloud SaaS consumer to find and select the best composite services.

7.8 Chapter Discussion

We can summarize the limitations of our proposed approach (Find SaaS GA TOPSIS) as follows:

- We tested our proposed approach considering a few numbers of abstract services, with a small number of candidate services. Therefore, we will test and evaluate Find SaaS GA TOPSIS with many sequence abstract services, also with many candidate services. In addition, we will compare the length of time taken using this approach against that taken by existing methods.

In addition, the future directions of Cloud SaaS service composition can be summarized as the following points:

- Interoperability and usability factors for selecting the composite Cloud SaaS can be other issues that should be considered in the future. In order to address this issue, we will develop a new system model to select the optimum composite services to have maximum interoperability and usability. We can solve this issue using the fuzzy logic concept by measuring the satisfaction degree or by using the machine learning to match the selecting composite service with the other existing consumers who already used the same composite service.

- The second phase of testing the performance of our proposed approach will be undertaken using the Azure Active Directory (AAD). AAD is the cloud Azure service that has the service Cloud SaaS providers repository [‡]. The reason for this is to test the capability of Find SaaS GA TOPSIS with a real scenario deployment. Azure active directory is the Microsoft cloud service used for managing users in the organization domain. It has a marketplace for all the SaaS services registered in its system.

7.9 Conclusion

In this chapter, we present the service composition of Cloud SaaS called the Find SaaS GA TOPSIS. The main functionality of the proposed approach is to

[‡]http://azure.microsoft.com/en-au/services/active-directory/

select the best composite service based on the combination of two methods: the Genetic algorithm (GA) and TOPSIS MCDM. The Genetic algorithm is to address the optimization problem by minimizing the total cost of service composition and maximize the reputation factor of composite service. Also, the TOPSIS is implemented to select the best composite service generated by the Genetic algorithm known as the Pareto set. Therefore, our proposed approach is concerned with the consumer's constraints of the cost of composite service and the reputation factor, in addition to the consumer's preferences by weighting the most important non-functional attributes.

The reputation factor has been discussed earlier in this thesis. The reputation factor is based on service reviews and service rating. We term this factor as reputation- based service reviewers (RbSR). Moreover, the fuzzy logic is applied to obtain RbSR factor by using the fuzzy rule system called the fuzzy associative matrix (FAM).

In the next chapter, we present the system prototype of Find SaaS framework. The system prototype is generated by the R Shiny library and uses the BluePage Cloud SaaS dataset introduced by (Alkalbani et al., 2015).

Chapter 8

Find SaaS System Prototype

In this chapter, the effectiveness of Find SaaS is farther demonstrated by developing three different system prototype functionality to provide simple, and composite Cloud SaaS service selection. It contains the following functionalities: Ranking the Cloud SaaS service provider (Find SaaS SNFCP, M2NFCP and LNFCP), Evaluation Cloud SaaS Service Provider (Find SaaS Evaluation Linguistic-based System and Rule-based System) and Cloud SaaS service composition (Find SaaS GA TOPSIS). Different case studies have been proposed to assist with understanding how the proposed system works. The R shiny library is used to generate the system prototype to provide proof of the concept system methodology. In addition, the Blue SaaS dataset (Alkalbani et al., 2015) has been implemented to generate the list of services after requesting the types of service required.

8.1 Introduction

The system prototype of Find SaaS service broker is introduced in this chapter. The system prototype includes three functionalities that we have developed to address the simple service selection and composite service selection of Cloud SaaS. This work has been undertaken to achieve our objective (4) to obtain the proof of concept for our proposed system. It includes three cases to demonstrate each functionality of the proposed approach. Firstly, consumer X needs to select an optimum CRM cloud SaaS service provider based on their preferences service rating, service reviewers and the date of founding. The three options that have been proposed in Chapter 5 are presented. Secondly, the evaluation system is demonstrated based

on service consumers' experiences for the following criteria: software support, vendor reputation, security, recoverability, usability and offline support. We start with the linguistic evaluation system followed by percentage evaluation (i.e. Fuzzy rule-based system). Finally, the service composition approach proposed is demonstrated based on the service consumer's preferences and service consumer's constraints of composite cost and composite RbSR.

The R Shiny library is used to demonstrate and to develop the system prototype of Find SaaS approach. The R shiny library is a high-tech library used to demonstrate the data view and recently became the first tool to provide data vision. Also, the blue SaaS dataset (Alkalbani et al., 2015) is used for testing the system prototype.

The reminder of the chapter is organized as follows: Section 8.2 presents the ranking system prototype of Find SaaS (SNFCP, M2NFCP and LNFCP); Section 8.3 provides the evaluation of the system prototype of Find SaaS Evaluation for both linguistic and percentage evaluation of Cloud SaaS service providers. The service composition system prototype of Find SaaS GA TOPSIS is presented in Section 8.4. Section 8.5 concludes this chapter.

8.2 Simple Cloud SaaS Service Selection Ranking Service Providers Prototype

To better understand the working of our ranking services approach work, we present the following case to demonstrate the system prototypes. A consumer X is looking for CRM Cloud SaaS service provider for selection. The first page will be displayed as Figure (8.1). After choosing the CRM Service category, the consumer X needs to choose how they would provide their preferences of non-functional parameters. In this case the consumer X chooses SNFCP explained in chapter 5. The

second page as Figure (8.2) will be displayed.

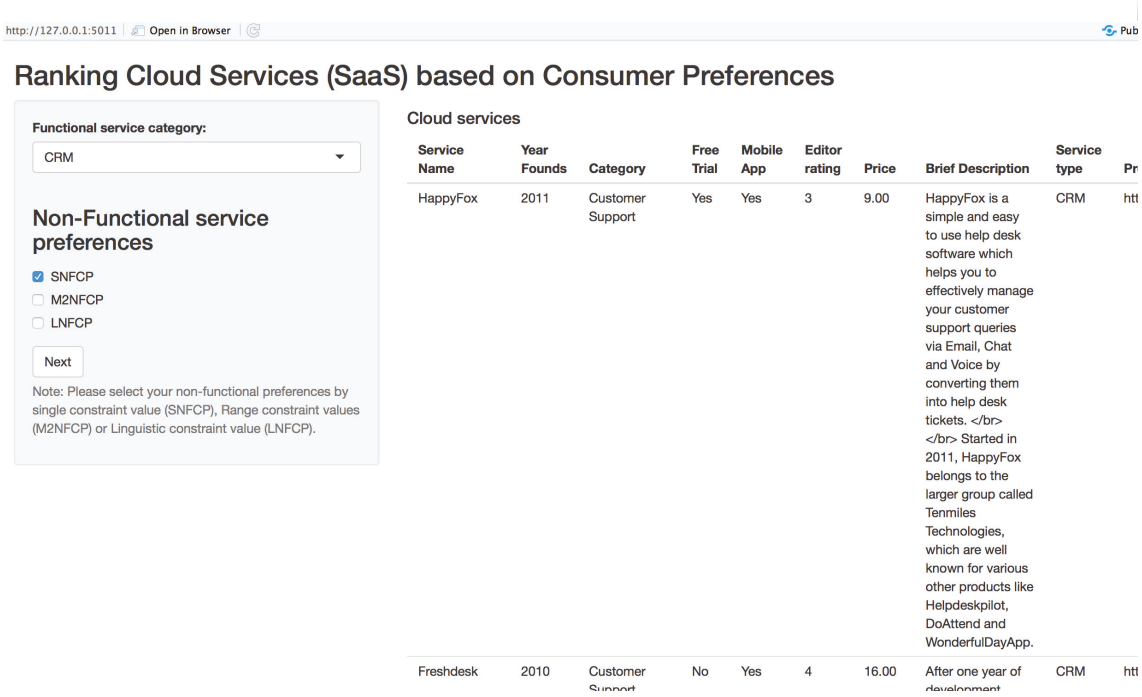


Figure 8.1 : The first main page of ranking services of Find SaaS

The Figure (8.2), a consumer X needs to choose the criteria preferences for ranking the services. After pressing the (Next) button, the page as Figure (8.3) will be displayed.

In this page Figure (8.3), the consumer X needs to describe their non-functional preferences as a single criteria constraint and weighting criteria. For example, the consumer X selects a service price like 30 dollars, with Very Important - (VI) weight, the service founded in 2008 with Important - (I) weight and finally rating service (3) with Extremely Important - (EI) weight. After that, the consumer X presses (rank the services) the button to rank the services. The consumer can also download the ranking list.

In addition, in this case the consumer chooses M2NFCP to rank the services. The working of M2NFCP is explained in chapter 5. The Figure (8.4) will be displayed.

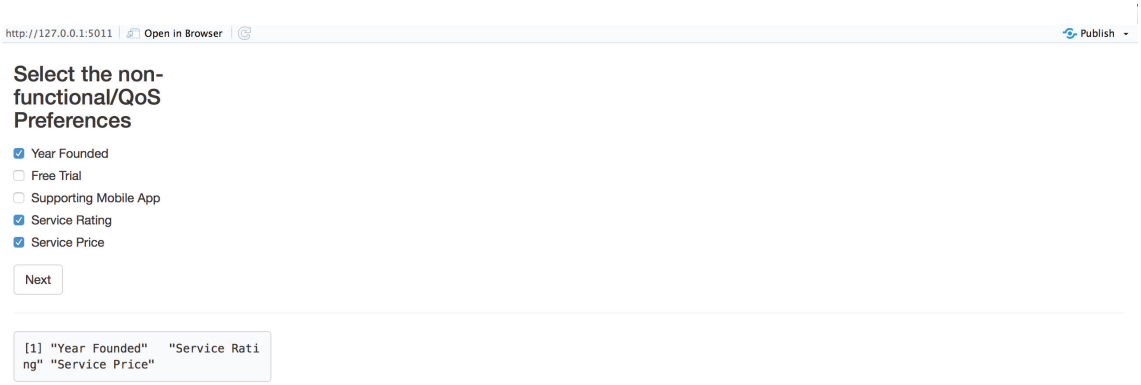


Figure 8.2 : Selected the non-functional preferences for ranking the services

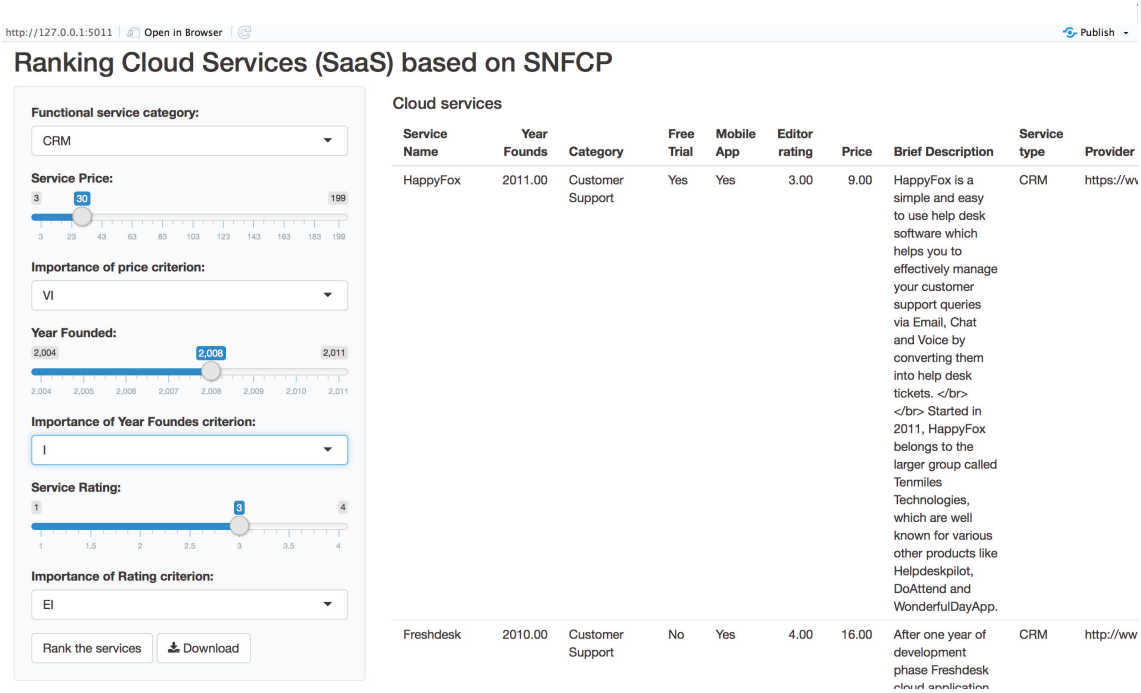


Figure 8.3 : The main page of ranking the cloud services using the SNFCP ranking approach

The consumer X chooses the service price as a range from $20 to $62 dollars with Very Important - (VI) weight, the service founded (2005 - 2010) with Very Important - (VI) weight and service rating (2.5-3.5) with Extremely Important - (EI) weight. Then, a consumer presses the (rank the services) button to rank the services. Once again, it is possible for the consumer to download the ranking list.

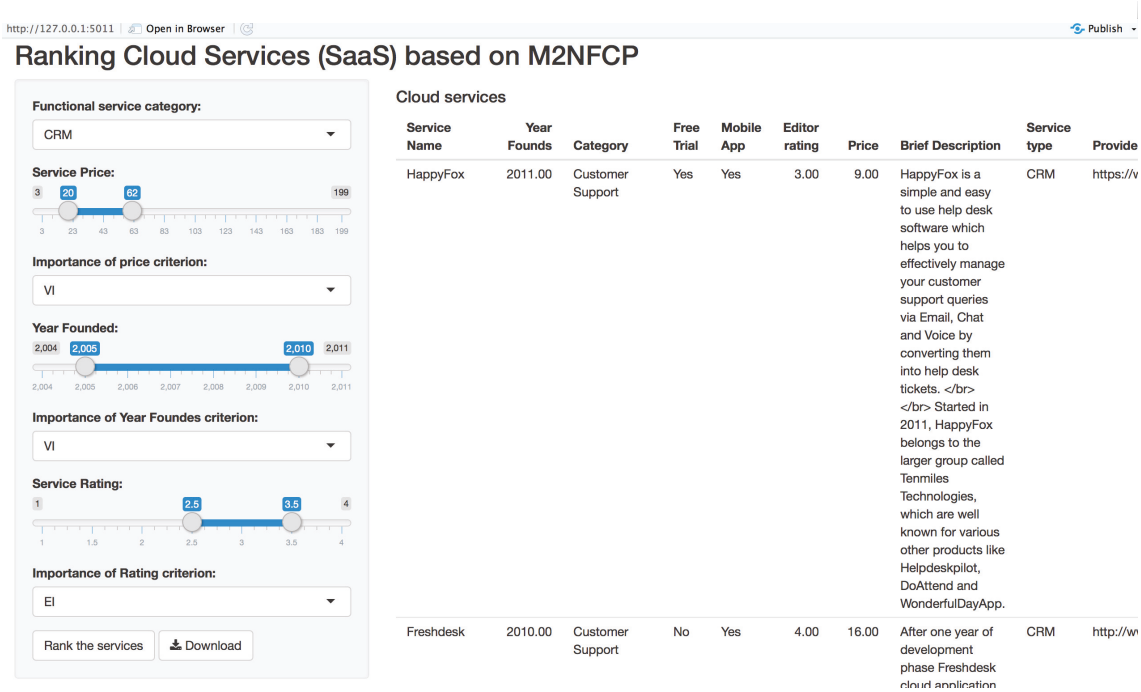


Figure 8.4 : The main page of ranking the cloud services using the M2NFCP ranking approach

Finally, consumer X selects LNFCP for ranking the services. The working of LNFCP is explained in chapter 5. The ranking service page as shown in Figure (8.5) will be displayed. The consumer X selects an affordable price with Very Important - (VI) weight, new service type with Not Important - (NI) weight and High rating service with Extremely Important - (EI) weight criteria. Again, consumer X can download the ranking list.

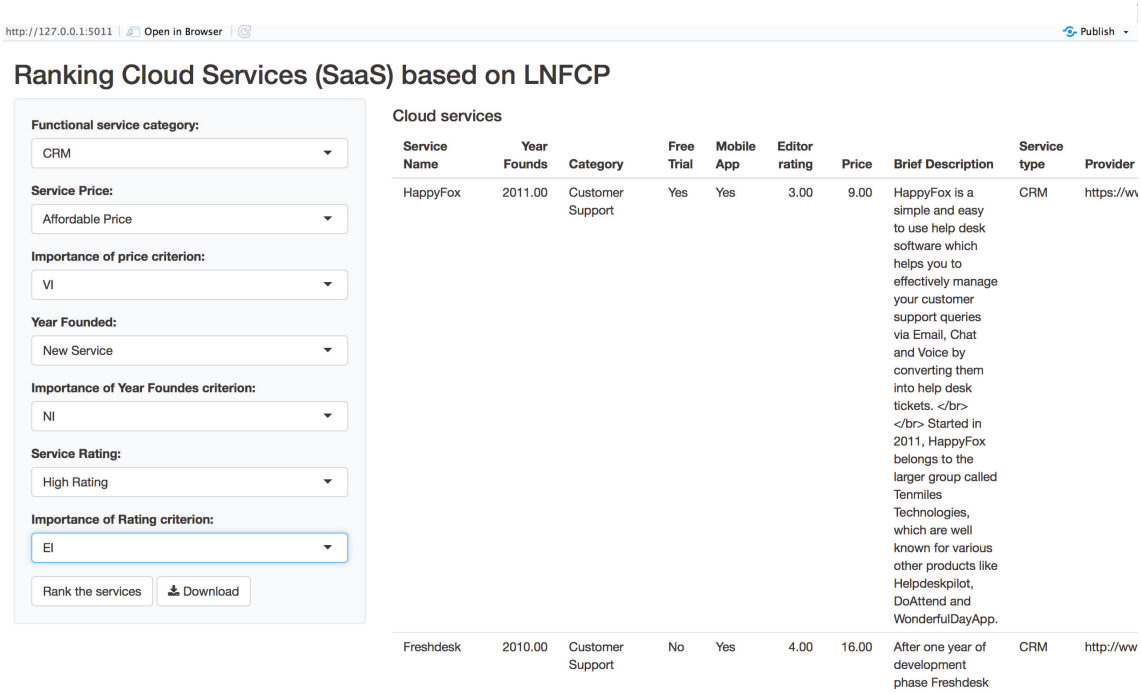


Figure 8.5 : The main page of ranking the cloud services using the LNFCP ranking approach

8.3 System Prototype of Cloud SaaS Evaluation System

It is known that the final service selection of Cloud SaaS service provider is based on the service evaluation by a Cloud SaaS service consumer. The Find SaaS Evaluation proposed approach is basically based on two techniques: (1) Based on the linguistic terms to describe the consumer's evaluation of quality attributes. (2) Based on percentage to describe the consumer's evaluation of quality attributes. In the case where a service consumer of Cloud SaaS used the ranking services approach (as described in the previous section) in order to find the services that meet their preferences of service cost, service rating and service availability. The service consumer can evaluate each service provider free of charge for one month prior to selecting the service that best meets with their requirements and their desired experience.

Our system helps a Cloud SaaS service consumer to evaluate selected services for testing and assess their performance based on service consumer's experiences. Figure (8.6) shows the main page of evaluation services system. The Cloud SaaS service consumer selects the list of services that they wish to investigate. The Cloud SaaS service consumer can find these services from the service repository in our system. Then, the list of experiences is selected to obtain all the qualities attributes that will be under investigation during the test period. Here, the Cloud SaaS service consumer can select the following attributes: software support, vendor reputation, security, recoverability, usability and offline service. Then, the Cloud SaaS service consumer selects which approach will be used for evaluating the service provider. They can choose either the Linguistic Evaluation technique or the Percentage Evaluation technique. Lets start with the Linguistic Evaluation technique. The Cloud SaaS also needs to identify the priority of these attributes.

As seen in Figure (8.7), there are six Cloud SaaS service providers to evaluate. Each service will be evaluated against the performance of selected quality attributes. For example, the first service, in terms of software support, the service is evaluated as Excellent. The Cloud SaaS service consumer can evaluate these attributes based on their experience with the service provider. Figure (8.8) shows the result of evaluation of the Cloud SaaS providers. As seen that service number 6 selected first with evaluation score (0.74). That means it has a similar match with the service consumer's experiences with 75%.

Figure (8.9) shows the user interface page when the Cloud SaaS service consumer selects the second technique for evaluating the cloud SaaS service providers. The percentage-based evaluation technique uses the fuzzy rule system for evaluating the service consumer's experiences. It is seen that service number 1 evaluates as 50% performance in terms of software support. In the final Cloud SaaS service providers list described in Figure (8.10). As seen in this figure, the service number one rank

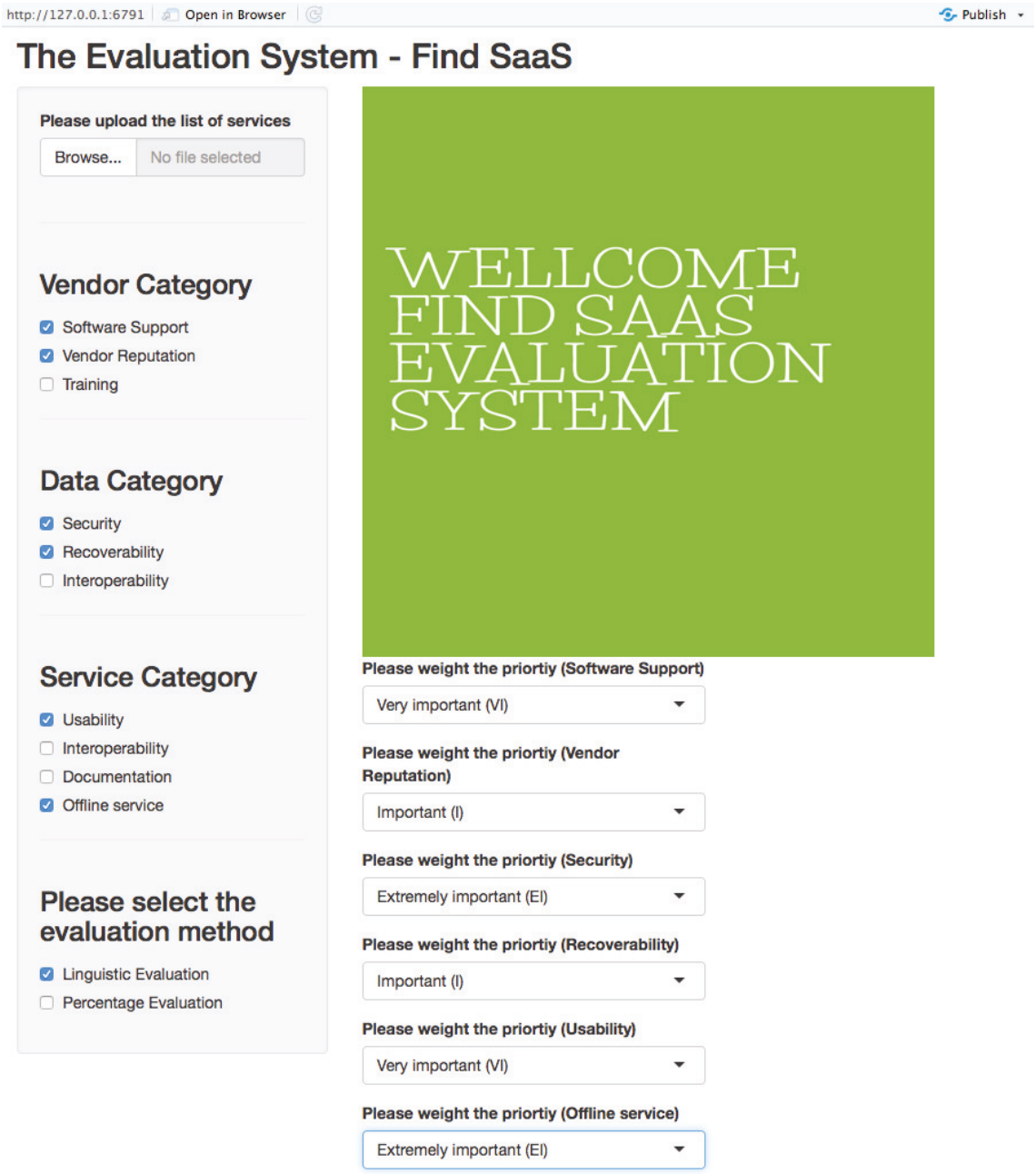


Figure 8.6 : The main page of evaluation system of Find SaaS

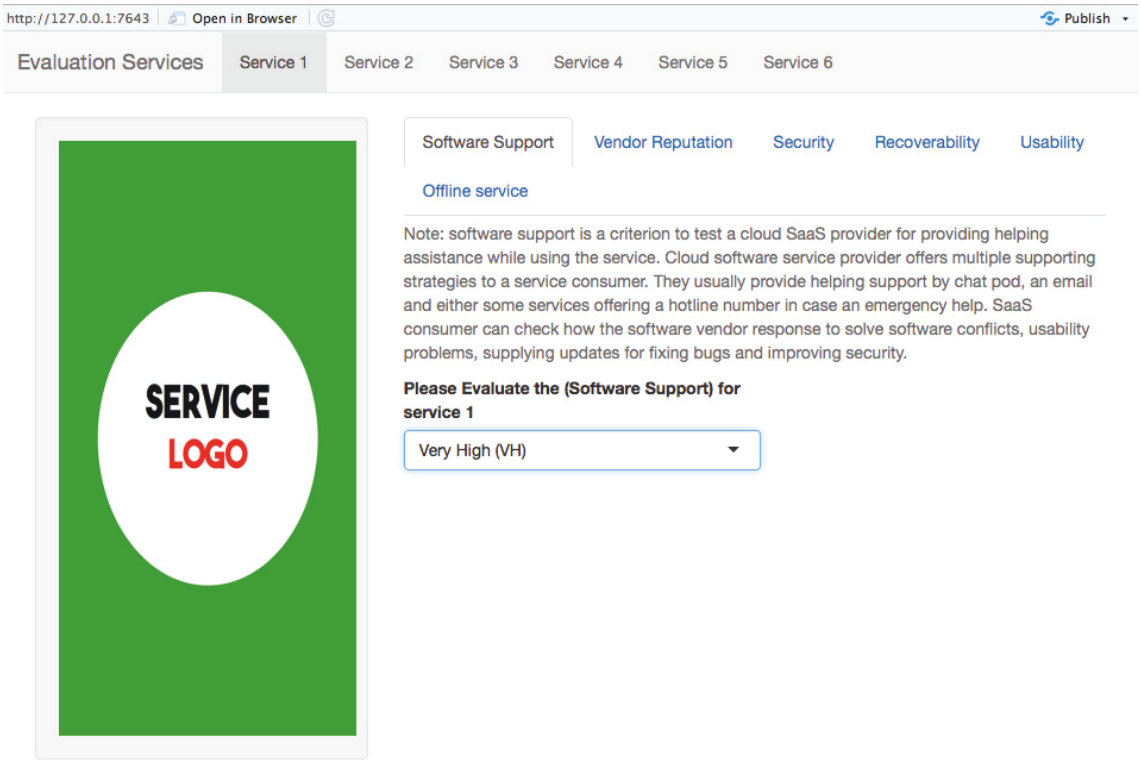


Figure 8.7 : The main page of cloud SaaS services using the the linguistic for evaluation the criteria

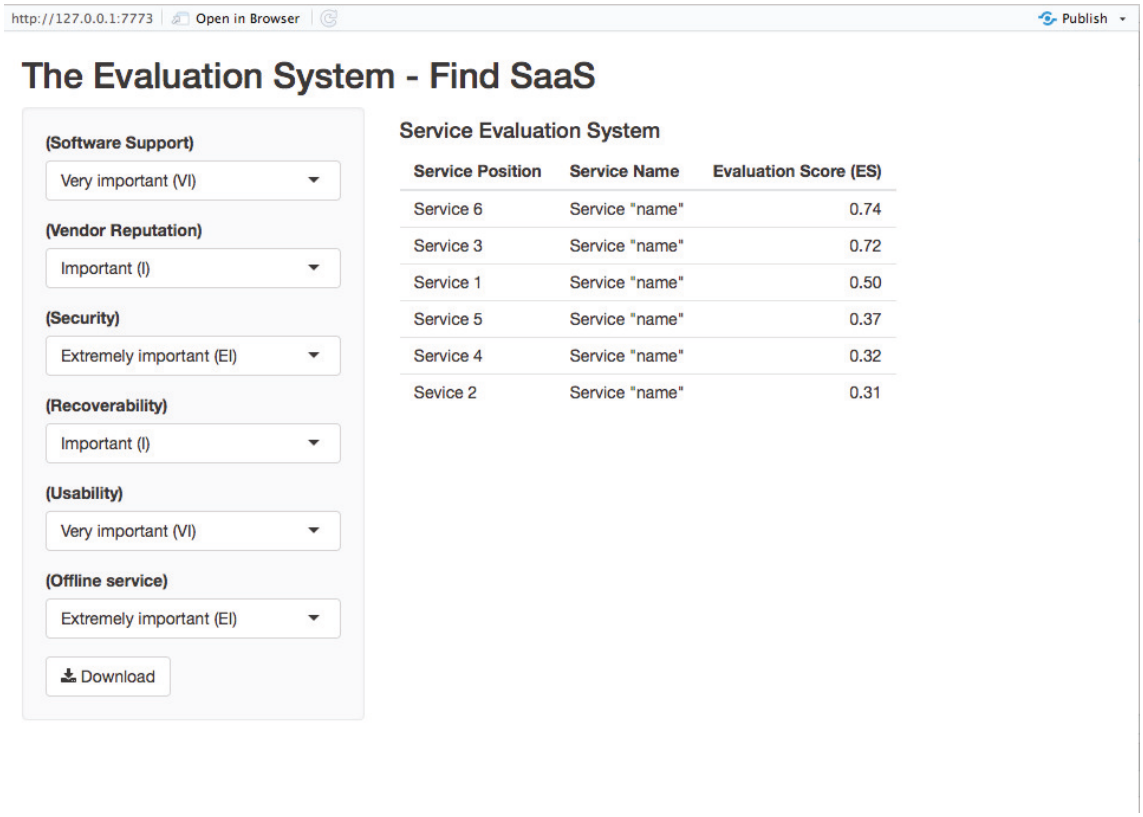


Figure 8.8 : The decision maker for selection the best cloud software service after the testing the service

in the first position with the similarity of 71.28%. The last service is service number 2 with the 46.72% similarity.

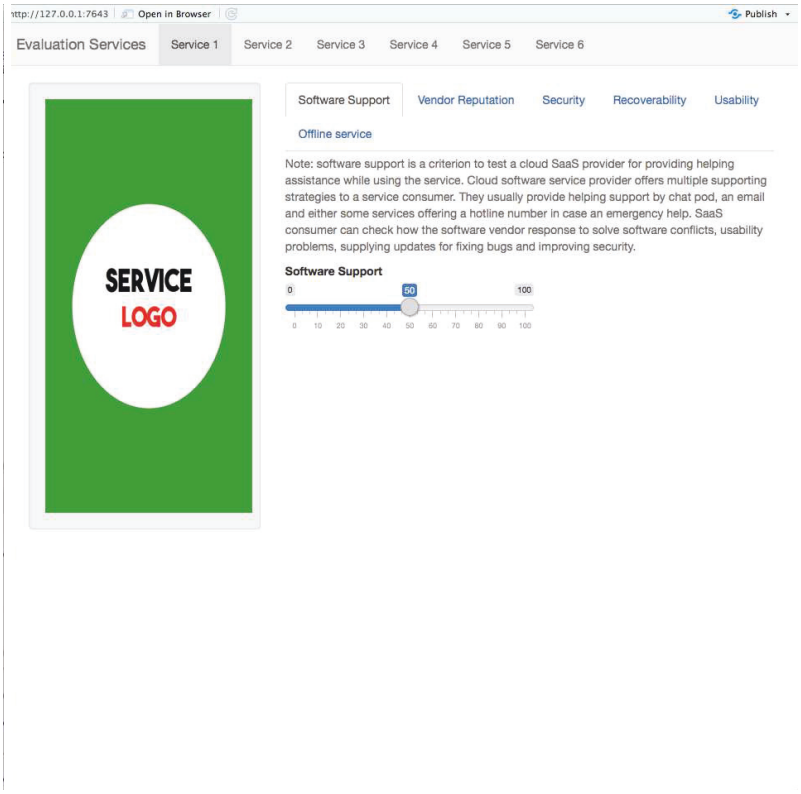


Figure 8.9 : The main page for rule based evaluation system of cloud software services

8.4 System Prototype of Cloud SaaS Service Composition Decision Making

Finally, the service composition approach using the Find SaaS GA TOPSIS is presented in this section. In service composition multiple service providers are selected to fulfil the service consumer's workflow requirements. The service consumer declares their workflow by described their tasks as an abstract service. In each abstract service, multiple services that match their functionality are called candidate concrete services. The Find SaaS GA TOPSIS selects optimal candidate service

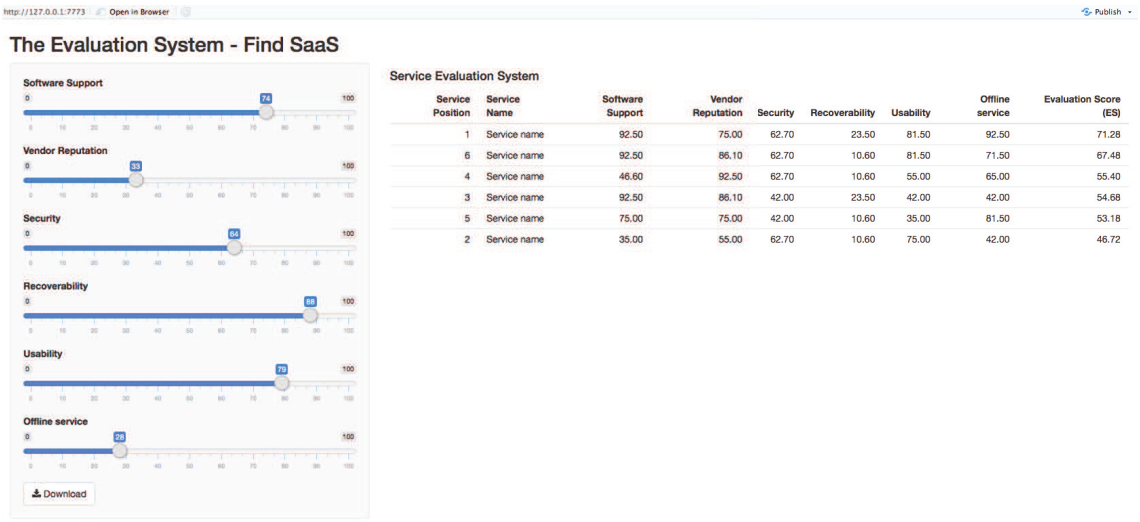


Figure 8.10 : The main page of selection after the decision using the percentage of selection or using the rule based evaluation system

providers from each abstract service. Figure (8.11) shows the user interface page for Find SaaS GA TOPSIS. Firstly, the Cloud SaaS service consumer notifies the service functionality required that match with their functional requirements. In a previous case study, Emmar company needs to select four functional services: Project service, Architecture service, Energy service and CRM service. Then, the Cloud SaaS service consumer weights their preferences from all the quality attributes such as composite cost weighted as extremely important. After that, the Cloud SaaS service consumer quantifies the constraints for the composite cost and composite reputation. In this case, the Cloud SaaS service consumer's constraints for the composite cost is in the range from 50 to 80. Also, the composite reputation is identified in the range of 0.5 to 0.9. The result is the list of composite services that match the consumer's preferences and constraints. The composite number 20578 is ranked first with a matching score of 0.77. This composite number includes Evernote, Freshbooks, Prontoforms and Mailchimps services. These top 20 composite services are selected

from around 16000 composite services.

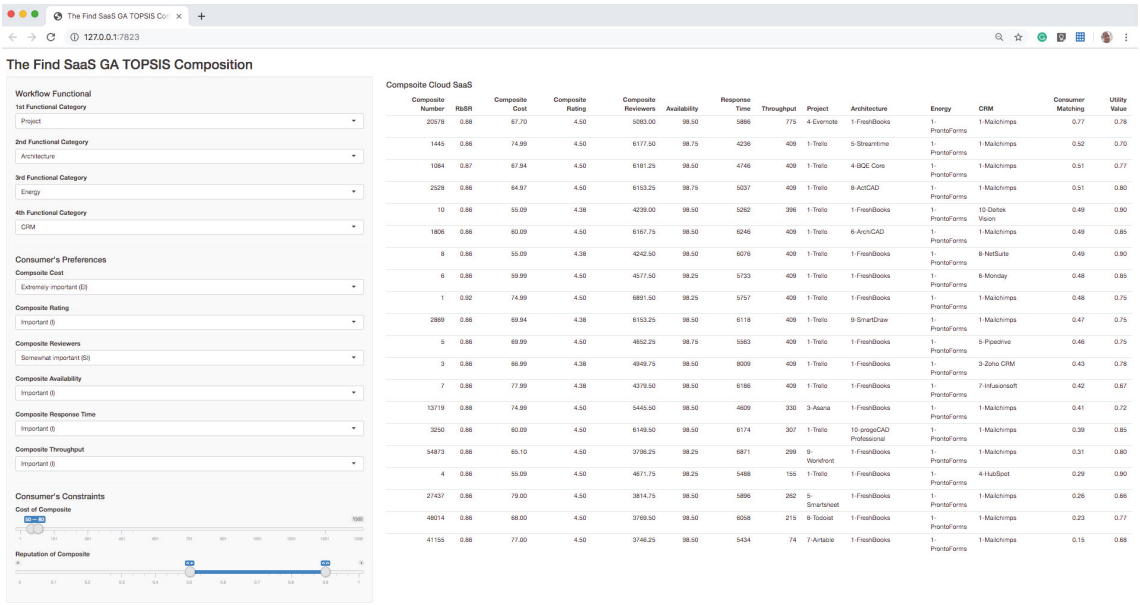


Figure 8.11 : The Find SaaS GA TOPSIS Composite solution

8.5 Conclusion

In this section, the Find SaaS system prototype is presented. The reason for this prototype is to further illustrate our proposed approach. The R Shiny library is the tool used to validate and build the system prototype. Three cases are presented to demonstrate the system prototype. Also, the Blue SaaS dataset used to validate some part of validation such as ranking services with Find SaaS SNFCP, M2NFCP and LNFCP.

The next chapter concludes the thesis by describing the main contributions made by this thesis. Also, the limitations of this work are described. Finally, potential future research directions are included in the last chapter.

Chapter 9

Recapitulation and Future Research Direction

This chapter concludes the thesis by summarizing the main research contributions, limitations, and future research directions.

9.1 Introduction

The field of selection Cloud SaaS either simple or composite service has recently been given much attention by researchers. As was mentioned in the literature review carried out in Chapter 2, several proposals have been made by various authors to address the selection of simple and composite of Cloud SaaS. However, the major shortcoming of these existing methods is that none of these provides a comprehensive service broker of Cloud SaaS to deal with Cloud SaaS simple and composite selection based on consumer's preferences. Our proposed service broker called the Find SaaS works as an agent in between the service consumer and service providers.

In terms of simple Cloud SaaS selection, the ranking system approach is proposed by measuring the shortest distance to the consumer's requirements. In order to deal with the consumer's requirements, multiple options have been proposed in this thesis presented in Chapter 5. Single (e.g. $20), minimum and maximum ($15 - $20), or linguistic (e.g. Affordable price) based service requirements are three options available to the service consumer to present their requirements. In addition to this, the service consumer can assign their priority of individual non-functional attributes by linguistic values (e.g. Highly important). Several methods have been used to address all these requirements in order to rank and sort all service providers

that match the service request.

Furthermore, in order to carry out the evaluation system of Cloud SaaS providers, post the trial duration and taking into account the feedback from the Cloud SaaS consumer, in this research we developed two approaches. Our system can assess the services based on two approaches; Linguistic or percentage. Both systems developed use fuzzy logic in order to obtain a service evaluation score for each service provider. Moreover, the quality of experiences (QoEs) is used to evaluate the service providers that were recommended using the literature review in the field of Cloud SaaS selection as well as asking people with expertise.

The selection mechanism of Cloud SaaS service composition introduced in this thesis uses the genetic algorithm (GA) and the TOPSIS MCDM. Firstly, the GA is used to address the optimization problem in order to aggregate the services that have a maximum reputation factor and minimum cost of service composition. The reputation factor is proposed using the service rating and service reviewers by using fuzzy logic to obtain the reputation factor which is called the reputation-based service reviewers (RbSR). Then, the TOPSIS method is used by the decision-maker to select the best composite service that is generated from the GA. The result is the list of suggested composite Cloud SaaS services that match the consumer's requirements.

Finally, the system prototype of Find SaaS service broker is proposed in chapter 8. The proposed system was developed using the RShiny library and using the dataset introduced by (Alkalbani et al., 2015). Three case studies are introduced to demonstrate the Find SaaS functionalities beginning with the ranking and sorting services, before continuing to service evaluation and service composition respectively.

The remainder of this chapter is organized as follows: Section (9.2) problems that have been addressed in this thesis, Section (9.3) provides the contributions of

this thesis to the existing literature and discuss them. The limitations are discussed in Section (9.4). Finally, Section (9.5) shows the future work directions for Cloud Software as a Service (SaaS).

9.2 Problems addressed in this thesis

This thesis addressed four research issues associated with the simple and composite Cloud SaaS selection. The major issues addressed in this thesis as follows:

1. Proposal of a new ranking system for Cloud SaaS service providers that takes into account the consumers' during the process of select the optimal service based on the consumer's preferences.

2. Proposal of a new evaluation system based on the service consumer's experiences during the trial period and taking into account the linguistic terms for evaluation services. Moreover, the new evaluation system based on quantifying the quality attributes as a percentage has been proposed. Both proposed methods are based on fuzzy logic.

3. Proposal of a new composite selection of Cloud SaaS based on the genetic algorithm and TOPSIS MCDM.Our proposed method takes into account the reputation of the Cloud SaaS provide during the process of composite SaaS selection.

4. Proposal of a comprehensive evaluation for our system using multiple approaches and by demonstrating the system using a prototype at the end of this thesis.

9.3 Contributions of this thesis to the existing literature

In this thesis, the problem of simple and composite Cloud SaaS service selection is studied. The Cloud SaaS service selection, either simple or composite service

becomes active research since the grow of the services recently. The simple Cloud SaaS selection is based on the consumer's preferences. Also, the selection of the Cloud SaaS needs to go through multiple stages, such as ranking the services in the beginning to meet with the consumer's requirements.

Most Cloud SaaS service providers offer a one-month trial version. This provides a consumer with an opportunity to evaluate the Cloud SaaS before entering a contract. The consumer's data is important for completing the selection process.

Most Cloud SaaS service providers ensure that their services are able to communicate with other services and can be integrated with other services to satisfy a customers requirements. This means that it is important that a service selection process is available to assist consumers in selecting the best composite selection. The research methodology presented in this thesis aims to address the research issues is divided into five chapters based on research directions involved to address simple and composite Cloud SaaS selection.

9.3.1 Contribution 1: State-of-the-art survey of present literature

In chapter 2, we investigated previous research on simple and composite of Cloud SaaS service selection. The comprehensive reviews on Cloud SaaS simple service selection are started first. We classified the previous research into three approaches based on the solutions provided: (1) Multi-criteria decision-making approaches. (2) Multi-objective decision-making approach. (3) The important of non-functional quality attributes concerns when selecting the service. The advantages and disadvantages of each proposed approach are discussed in the chapter (2). We then moved on to discuss evaluation system approaches for Cloud SaaS service providers.

Moreover, we discussed the main benefits and limitations of previous research in that field. Finally, we identified the main shortcomings of simple Cloud SaaS selection to identify the scope of our proposed approach for simple Cloud SaaS selection.

On the other hand, we explained the main shortcomings of previous approaches to selection of composite Cloud SaaS along with the advantages and disadvantages of each. Finally, we identified the main weaknesses in order to refine the scope of this research. Namely, to address the shortcomings of previous approaches for composite Cloud SaaS selection. The dataset is also discussed in the chapter (2) in order to compare and evaluate the proposed approach with existing methods.

9.3.2 Contribution 2: Simple Cloud SaaS Service Selection Model based on consumer's preferences (Find SaaS SNFCP, M2NFCP and LNFCP)

In chapter 5, the ranking system of Cloud SaaS is proposed to help Cloud SaaS service consumer for selecting the best service. The Find SaaS SNFCP, M2NFCP and LNFCP are three ranking system approaches considering the consumer's preferences. Previous works in this filed consider only weighting the important of non-functional attributes for making a decision and selecting a service. In addition to this, if we consider the matching approach to meet with the consumer's requirements, in some cases we retrieve zero services. That means no services comply with the consumer's requirements. Therefore, our proposed system takes into account weighting the attributes with a requesting value requirement for particular non-functional attributes. We also divide the consumer's requirements into three types: single requirements (e.g. $20), minimum and maximum requirements (e.g. $15 - $20), and linguistic requirements (e.g. Affordable price). The idea for the ranking system is to rank the list of service based on measuring the distance to the consumer's request of Cloud SaaS services. We also defined the new non-functional quality attributes based on the resources available to our system and based on three resources: (1) Service provider. (2) Monitoring tool. (3) Previous consumer feedback.

9.3.3 Contribution 3: Evaluation of Cloud SaaS Service Provider

In chapter 6, an evaluation system is proposed to evaluate the Cloud SaaS service providers for making a final decision to select the service based on the consumer's experiences. As was mentioned previously, the majority of Cloud SaaS service provider offer a one-month trial version to help a consumer to assess the service before selection. However, the main shortcomings of previous works is that they only developed their approaches using pairwise comparisons for each service. Hence, our proposed approach is based on two separate methodologies. Firstly, an evaluation system based on linguistic terms to evaluate the service provider. The idea for this approach is to use a decision-maker system after evaluation of the services by a consumer based on their experiences, then making a decision for selecting the service using the MCDM approach. We called this approach Find SaaS Linguistic-based Evaluation System. Secondly, the evaluation system is based on the percentage level for the evaluation of the service provider. The idea for this approach is using a fuzzy associative matrix (FAM) in order to generate the fuzzy rule and to make a decision. This approach is called the Find SaaS Rule-based Evaluation System. Both approaches generate an evaluation service score to help a Cloud SaaS service consumer to select the service. Furthermore, the quality of experiences (QoE) that are of concern when evaluating the Cloud SaaS service provider are introduced in this chapter. We revisit the literature review and use expert advice to select Cloud SaaS services are used for this purpose. They are also used and to create the quality of experiences (QoE).

9.3.4 Contribution 4: Composite Cloud SaaS Service Selection (Find SaaS GA TOPSIS)

In chapter 7, the composite of Cloud SaaS service selection is proposed called Find SaaS GA TOPSIS. The previous researchers in this field address service compo-

sition based on objective optimization to find suitable composite services for a Cloud SaaS service consumer. The shortcoming in this approach is that they addressed the selection issue by aggregating the service providers by reducing the search space for selecting a composite service. None of the previous works considers the decision maker after reducing the search space using the optimization method.

Hence, in this work, we proposed the decision-maker system approach after using the optimization method. The genetic algorithm (GA) has been used in our approach to address the multi-objective optimization problem by minimizing the total cost of composite services as well as maximizing the reputation factor. In addition, the reputation factor is proposed in this chapter using service rating and service reviewers, we term that factor as the reputation-based service reviewers (RbSR). The concept of fuzzy logic is used for this purpose.

9.3.5 Contribution 5: System prototype to demonstrate the Find SaaS service porker framework

In chapter 8, the Find SaaS system prototype is proposed. The RShiny library is used to test and demonstrate the Find SaaS prototype. Also, the Cloud SaaS dataset introduced in the article (Alkalbani et al., 2015) is used for testing purposes. It begins with ranking Cloud SaaS service providers matching the consumer's requests. Then, the consumer can select some services for testing and evaluation purposes, taking advantage of the opportunity provided by service providers of a one-month trial period. The service consumer can choose which approach they would like to use for evaluation, such as linguistic-based or rule-based. After entering all the evaluations parameters, our system makes a decision for selecting the leading service suitable to store consumer data. If a service consumer would like to apply to select the composite service of Cloud SaaS, our system has an option to choose composite Cloud SaaS. The service consumer enters all requesting parameters describes their

preferences. Then our system makes a decision and presents all the composite services matching the consumer's request. After that, the service consumer can select the optimum composite service for deploying IT services with full cloud advantages such as elasticity and availability.

In summary, this thesis has made the following contributions to the field of simple and composite of Cloud SaaS service selection:

1. Software as a Service (SaaS) Service Selection Based on Measuring the Shortest Distance to the Consumer's Preferences, (Ikram & Hussain, 2018).

2. Dynamic Ranking System of Cloud SaaS Based on Consumer Preferences - Find SaaS M2NFCP, (Ikram et al., 2019).

3. The Non-Functional Cloud Software as a Service (SaaS) classification based on Quality of Service (QoS) and Quality of Experience (QoE).

4. Linguistic-based Evaluation System for Cloud Software as a Service (SaaS).

5. Service Broker for Ranking, Evaluation and Selection of the Cloud Software Service (SaaS) based on Consumer Non-Functional Preferences.

6. Decision Maker of Composite Cloud Software as a Service (SaaS) using Genetic Algorithm and TOPSIS MCDM.

9.4 Limitations

This research work focused on service selection for Cloud SaaS simple and composite service based on consumer's preferences. In addition, our proposed approaches provided novel and efficient solutions, but there are some limitations that need to improve in the future as follows:

Regarding our proposed approach to select simple Cloud SaaS service provider:

1. The selection work focused on developing multiple approaches based on non-functional quality attributes for making a decision. However, selection of the Cloud SaaS service provider based on matching functional service is still an active research topic. Therefore, the selection of Cloud SaaS service provider can be improved by using several semantic or syntactic approaches taking into account our proposed approach for ranking the services.

2. The ranking approach based on linguistic requirements can be improved by translating the linguistic value into a crisp number using any defuzzification methods.

3. The selection method for Cloud SaaS service provider can be improved using the aggregation mechanism of the quality of service (QoS) with quality of experiences (QoE). Therefore, the new approach concerns both these quality attributes when selecting a Cloud SaaS service provider.

Regarding the evaluation system of Cloud SaaS service provider:

- The evaluation of some quality of experiences can be assessed automatically based on the recommender system in order to address the unknown for measuring the quality. That means the service provider assess based on previous consumer experience that used a certain service. The solution addresses the problem in case the service consumer is not able to evaluate the service provider accurately.

Finally, according to the Cloud SaaS service composition:

- The interoperability and usability factors have not been considered in the current selection mechanism of composite Cloud SaaS. These two factors play important roles when selecting a composite service. It is worth noting that our

composite service selection considers only the reputation factor and the cost of composite. There are some challenges which need to be addressed, such as how interoperability and usability be measured in a numerical quantity.

9.5 Future work

9.5.1 Service provider instead of Service consumer

The current research focused on Cloud SaaS consumers for ranking cloud services in order to match consumer requirements for service selection. The next step of this research is focusing on the service providers in order to help a service provider to match consumer requirements in terms of functional requirements as well as non-functional/QoS requirements. For example, building a framework to obtain the shortcoming of existing services available to the consumer, and then suggesting new capabilities to the provider which would enhance the consumer experience. Thus, when a consumer writes the review of some service it is submitted to a module. This module can analyze comments from a wide range of consumers, then send suggestions or recommendations for a new service platform which better meets consumer needs.

It should be noted that new businesses will evolve. Therefore, people will change their interests. It is known that customer relationship management (CRM) software and enterprise resource planning (ERP) software run the majority of business infrastructure. Soon however, the possibility of a new business module will emerge that engages the social profiles of employees, customers or even third-party brokers in the recommender system. It would then be necessary to build a new module to analyze future business module and trends, and to embrace new technologies that evolve in the future. This can include but is not limited to the application of computer vision and artificial intelligence to predict business and people needs.

In addition, most Cloud SaaS providers launch their cost based on the futures they have and based on the cost used to run a service such as the cost of infras-

tructure and cost of technicians to run the service. Moreover, a service consumer that locates out of the datacenter where the service is situated, may need a different price to a consumer who is in the same place of a datacenter, since the performance of service and the data policy differ. For example, based on the Capterra web site, which is the most best known platform for finding Cloud SaaS provider the price range of CRM services starting from few dollars reaching to the hundreds of dollars per month. Also, some of the more economical services that have high rating reviews and a good reputation. Therefore, it is necessary to build a recommender system for a service provider to predict their price based on different factors to match the consumer view.

9.5.2 Cloud SaaS selection technique based on economical and environmental value

This future research direction helps a Cloud SaaS consumer to find and select a service based on the economic and environmental value. Building on-premises software is costly due to the infrastructure requirements, salary costs for a technician to run the service, and maintenance. Therefore, the selection of a Cloud SaaS seems to be less costly. However, on-premises software has some advantages such as privacy and security. Cloud SaaS service providers give a guarantee that data is safe, and will be available for a long time. However, the data are not owned by a Cloud SaaS consumer, the data is stored in the datacenter of a service provider. Environmental issues are becoming increasingly of concern. On-premise software needs a lot of maintenance and server upgrades to keep the system updated and running with full efficiency to address increases in the need for data storage. Cloud SaaS providers offer elasticity of service and offers full storage of data which means a Cloud SaaS consumer does not need to upgrade their device regularly.

Therefore, the future research direction of this research will consider the eco-

nomic and environmental values to select a cloud service. For example, consider several services available for selection, compared with an on-premise software plan. The system addresses the consumer requirements such as the cost of infrastructure, maintenance, and the expected number of years that service will run. Moreover, it compares this with all the cloud services available that seems matching the requirements. Then, the system will predict the economic and environmental value for using each SaaS compared with other cloud services as well as building on-premises software. This study may need a combination of multiple techniques including objective optimization solution methods such as like Evolutionary algorithm (EA), Mixed Integer Programming (MIP) or Linear programming (LP). This research also requires some multi-criteria decision making (MCDCM), Artificial Intelligence (AI) algorithms and machine learning to obtain and predict the economic and environmental values of selecting a service among other services.

9.5.3 The interoperability of on-premises software and cloud Software as a service (SaaS)

Selecting based on interoperability between the Cloud SaaS and on-premises software is a significant issue facing current service consumers of Cloud SaaS. For example, an organization has on-premises software and wants to transfer some of their business software into Cloud SaaS services. Challenges and issues may be raised by this strategy, such as datatype, usability, and operability between each technology. To address these issues and select an optimum Cloud SaaS the interoperability level between the cloud service and the on-premises software must be taken into account including multiple factors such as datatype, software functionality, and usability.

9.5.4 The risk assessment module for selecting Cloud SaaS

Security and privacy play an essential factor to select the Cloud SaaS service provider. The majority of Cloud SaaS service provider guarantee all data will be kept safe and managed with expert security engineering. However, the data will not save in the client servers, which could give rise to security issues when using Cloud SaaS service providers. The future direction of Cloud SaaS service providers should consider the risk assessment procedures that Cloud SaaS service consumers conduct in order to select the best service provider that has minimum risk when deploying cloud services.

There are multiple approaches to risk assessment which identify and analyze the potential negative impact for deploying Cloud SaaS instead of using on-premise software applications. Dynamic risk assessment is one example that can be used in the future to calculate these risks for using a Cloud SaaS provider.

9.5.5 The IoT and Cloud SaaS

The internet of Things (IoT) is likely to becomes the new era of IT services in the future. IoT makes physical devices communicate with each other (i.e. Machine to Machine (M2M)), or by using software applications to control physical devices. Recently, multiple Cloud SaaS applications have started and are available online to take control, for instance, of home devices or city devices (as known as Smart home or smart city). However, there are some issues raised with the growth of these kinds of applications that may lead to difficulty in selecting the best service. These applications work under cloud services which make these applications work as part of the Cloud SaaS concept.

Our future direction may focus on selecting the best Cloud SaaS under the IoT technology concept. The process could involve firstly understanding the characteristics of IoT devices, and how to deal with the quality of services for IoT that match

with the quality of cloud services.

References

Afify, Y. M., Moawad, I. F., Badr, N. L., & Tolba, M. F. (2017). A personalized recommender system for saas services. *Concurrency and Computation: Practice and Experience*, *29*(4), e3877.

Alkalbani, A. M., Ghamry, A. M., Hussain, F. K., & Hussain, O. K. (2015a, Nov.). Blue pages: Software as a service data set. In *2015 10th international conference on broadband and wireless computing, communication and applications (bwcca)* (Vol. 00, p. 269-274). Retrieved from doi.ieeecomputersociety.org/10.1109/BWCCA.2015.83 doi: 10.1109/BWCCA.2015.83

Alkalbani, A. M., Ghamry, A. M., Hussain, F. K., & Hussain, O. K. (2015b, Nov). Blue pages: Software as a service data set. In *2015 10th international conference on broadband and wireless computing, communication and applications (bwcca)* (p. 269-274). doi: 10.1109/BWCCA.2015.83

Alkalbani, A. M., Ghamry, A. M., Hussain, F. K., & Hussain, O. K. (2015, Nov). Blue pages: Software as a service data set. In *2015 10th international conference on broadband and wireless computing, communication and applications (bwcca)* (p. 269-274). doi: 10.1109/BWCCA.2015.83

Al-Masri, E., & Mahmoud, Q. H. (2007). Qos-based discovery and ranking of web services. In *Computer communications and networks, 2007. icccn 2007. proceedings of 16th international conference on* (pp. 529–534).

Alrifai, M., & Risse, T. (2009). Combining global optimization with local selection for efficient qos-aware service composition. In *Proceedings of the 18th*

international conference on world wide web (pp. 881–890). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/1526709.1526828 doi: 10.1145/1526709.1526828

Al-Shammari, S., & Al-Yasiri, A. (2014). Defining a metric for measuring qoe of saas cloud computing. *Proceedings of PGNET*, 251–256.

Arman, A., Foresti, S., Livraga, G., & Samarati, P. (2016). A consensus-based approach for selecting cloud plans. In *Research and technologies for society and industry leveraging a better tomorrow (rtsi), 2016 ieee 2nd international forum on* (pp. 1–6).

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... Zaharia, M. (2010, April). A view of cloud computing. *Commun. ACM*, *53*(4), 50–58. Retrieved from http://doi.acm.org/10.1145/1721654.1721672 doi: 10.1145/1721654.1721672

Asanka, P. D., & Perera, A. S. (2017). Defining fuzzy membership function using box plot.

Azzam, I. A., Leung, C. H. C., & Horwood, J. F. (2005, Jan). A fuzzy expert system for concept-based image indexing and retrieval. In *11th international multimedia modelling conference* (p. 452-457). doi: 10.1109/MMMC.2005.8

Badidi, E. (2013). A framework for software-as-a-service selection and provisioning. *arXiv preprint arXiv:1306.1888*.

Benjamin McGrath, R. P. M. (2015). *Worldwide saas and cloud software 20152019 forecast and 2014 vendor shares* (Report).

Bentaleb, A., & Ettalbi, A. (2016, May). Toward cloud saas for web service composition optimization based on genetic algortihm. In *2016 2nd international conference on cloud computing technologies and applications*

(cloudtech) (p. 147-152). doi: 10.1109/CloudTech.2016.7847692

Boussoualim, N., & Aklouf, Y. (2015a). Evaluation and selection of saas product based on user preferences. In *Technological advances in electrical, electronics and computer engineering (taeece), 2015 third international conference on* (pp. 299–308).

Boussoualim, N., & Aklouf, Y. (2015b, April). Evaluation and selection of saas product based on user preferences. In *2015 third international conference on technological advances in electrical, electronics and computer engineering (taeece)* (p. 299-308). doi: 10.1109/TAEECE.2015.7113644

Burkon, L. (2013). Quality of service attributes for software as a service. *Journal of Systems Integration*, *4*(3), 38.

Cai, J. J., & Tong, C. J. (2009, May 5). *Method, system, and web service broker for dynamic web service invocation.* Google Patents. (US Patent 7,529,833)

Chamodrakas, I., Leftheriotis, I., & Martakos, D. (2011). In-depth analysis and simulation study of an innovative fuzzy approach for ranking alternatives in multiple attribute decision making problems based on topsis. *Applied Soft Computing*, *11*(1), 900–907.

Chauhan, A., & Singh, A. (2016). A hybrid multi-criteria decision making method approach for selecting a sustainable location of healthcare waste disposal facility. *Journal of Cleaner Production*, *139*, 1001 - 1010. Retrieved from `http://www.sciencedirect.com/science/article/pii/S095965261631246X` doi: https://doi.org/10.1016/j.jclepro.2016.08.098

Cortzar, G. O., Zapater, J. J. S., & Snchez, F. G. (2012, May). Adding semantics to cloud computing to enhance service discovery and access. In *2012 6th euro american conference on telematics and information systems (eatis)* (p. 1-6). doi: 10.1145/2261605.2261639

Delac, G., Silic, M., & Srbljic, S. (2015, March). A reliability improvement method

for soa-based applications. *IEEE Transactions on Dependable and Secure Computing*, *12*(2), 136-149. doi: 10.1109/TDSC.2014.2327971

Garg, S. K., Versteeg, S., & Buyya, R. (2013). A framework for ranking of cloud computing services. *Future Generation Computer Systems*, *29*(4), 1012 - 1023. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0167739X12001422` (Special Section: Utility and Cloud Computing) doi: https://doi.org/10.1016/j.future.2012.06.006

Ghani, A. A. A., & Sultan, A. B. M. (2014). Saas quality of service attributes. *J. Appl. Sci*, *14*(24), 3613–3619.

Godse, M., & Mulik, S. (2009). An approach for selecting software-as-a-service (saas) product. In *Cloud computing, 2009. cloud'09. ieee international conference on* (pp. 155–158).

Haselmann, T., & Vossen, G. (2011). Software-as-a-service in small and medium enterprises: an empirical attitude assessment. In *International conference on web information systems engineering* (pp. 43–56).

He, Q., Han, J., Yang, Y., Grundy, J., & Jin, H. (2012). Qos-driven service selection for multi-tenant saas. In *2012 ieee fifth international conference on cloud computing* (pp. 566–573).

Herrera, F., Lozano, M., & Verdegay, J. L. (1995). Tuning fuzzy logic controllers by genetic algorithms. *International Journal of Approximate Reasoning*, *12*(3-4), 299–315.

Holland, J. H., et al. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

Ikram, M. A., & Hussain, F. K. (2018). Software as a service (saas) service selection based on measuring the shortest distance to the consumer's

preferences. In *Advances in internet, data & web technologies, the 6th international conference on emerging internet, data & web technologies, eidwt-2018, tirana, albania, march 15-17, 2018.* (pp. 403–415). Retrieved from https://doi.org/10.1007/978-3-319-75928-9_36 doi: 10.1007/978-3-319-75928-9_36

Ikram, M. A., Sharma, N., Raza, M., & Hussain, F. K. (2019). Dynamic ranking system of cloud saas based on consumer preferences - find saas m2nfcp. In L. Barolli, M. Takizawa, F. Xhafa, & T. Enokido (Eds.), *Advanced information networking and applications* (pp. 1000–1010). Cham: Springer International Publishing.

Jagli, D., Purohit, S., & Chandra, N. S. (2016). Evaluating service customizability of saas on the cloud computing environment. *Int. J. Comput. Appl.*, *141*(9).

Jian, X., Zhu, Q., & Xia, Y. (2016). An interval-based fuzzy ranking approach for qos uncertainty-aware service composition. *Optik-International Journal for Light and Electron Optics*, *127*(4), 2102–2110.

Jordan, D., Evdemon, J., Alves, A., Arkin, A., Askary, S., Barreto, C., ... others (2007). Web services business process execution language version 2.0. *OASIS standard*, *11*(120), 5.

Joshi, A. V. (2020). Ranking. In *Machine learning and artificial intelligence* (pp. 193–198). Cham: Springer International Publishing. Retrieved from https://doi.org/10.1007/978-3-030-26622-6$_2$0 doi: 10.1007/978-3-030-26622-6$_2$0

Jula, A., Sundararajan, E., & Othman, Z. (2014). Cloud computing service composition: A systematic literature review. *Expert Systems with Applications*, *41*(8), 3809–3824.

Kang, J., & Sim, K. M. (2011). Ontology and search engine for cloud computing system. In *System science and engineering (icsse), 2011 international*

conference on (pp. 276–281).

Karsak, E. E., & Tolga, E. (2001). Fuzzy multi-criteria decision-making procedure for evaluating advanced manufacturing system investments. *International journal of production economics*, *69*(1), 49–64.

Klir, G. J. (1997). Fuzzy arithmetic with requisite constraints. *Fuzzy Sets and Systems*, *91*(2), 165 - 175. Retrieved from http://www.sciencedirect.com/science/article/pii/S0165011497001383 (Fuzzy Arithmetic) doi: https://doi.org/10.1016/S0165-0114(97)00138-3

Kumar, S., Bahsoon, R., Chen, T., Li, K., & Buyya, R. (2018). Multi-tenant cloud service composition using evolutionary optimization. In *2018 ieee 24th international conference on parallel and distributed systems (icpads)* (pp. 972–979).

Liu, Y., Esseghir, M., & Boulahia, L. M. (2016). Evaluation of parameters importance in cloud service selection using rough sets. *Applied Mathematics*, *7*(06), 527.

Marian, M., & Hamburg, I. (2012). Guidelines for increasing the adoption of cloud computing within smes. In *The third international conference on cloud computing, grids, and virtualization* (pp. 7–10).

McGrath, B. (2015, Aug.). Worldwide saas and cloud software 20152019 forecast and 2014 vendor shares. In *Idc publisher* (Vol. 00, p. 0).

Mell, P., Grance, T., et al. (2011). The nist definition of cloud computing.

Mital, M., Pani, A., & Ramesh, R. (2014). Determinants of choice of semantic web based software as a service: An integrative framework in the context of e-procurement and erp. *Computers in Industry*, *65*(5), 821–827.

Naseer, M., & Nazar, M. (2016, Aug). A framework for selection of saas by evaluating the quality of freemium model. In *2016 sixth international conference on innovative computing technology (intech)* (p. 78-82). doi:

10.1109/INTECH.2016.7845029

Negnevitsky, M. (2005). *Artificial intelligence: a guide to intelligent systems*. Pearson education.

Oliveira, L. S., Sabourin, R., Bortolozzi, F., & Suen, C. Y. (2002). Feature selection using multi-objective genetic algorithms for handwritten digit recognition. In *Object recognition supported by user interaction for service robots* (Vol. 1, pp. 568–571).

Papazoglou, M. P. (2003, Dec). Service-oriented computing: concepts, characteristics and directions. In *Proceedings of the fourth international conference on web information systems engineering, 2003. wise 2003.* (p. 3-12). doi: 10.1109/WISE.2003.1254461

Peffers, K., Tuunanen, T., Rothenberger, M., & Chatterjee, S. (2007, December). A design science research methodology for information systems research. *J. Manage. Inf. Syst.*, *24*(3), 45–77. Retrieved from http://dx.doi.org/10.2753/MIS0742-1222240302 doi: 10.2753/MIS0742-1222240302

Reixa, M., Costa, C., & Aparicio, M. (2012). Cloud services evaluation framework. In *Proceedings of the workshop on open source and design of communication* (pp. 61–69).

Repschlaeger, J., Wind, S., Zarnekow, R., & Turowski, K. (2012). Selection criteria for software as a service: an explorative analysis of provider requirements.

Rezaei, R., Chiew, T. K., Lee, S. P., & Aliee, Z. S. (2014). A semantic interoperability framework for software as a service systems in cloud computing environments. *Expert Systems with Applications*, *41*(13), 5751–5770.

Rodrigues, J., Ruivo, P., & Oliveira, T. (2014). Software as a service value and firm performance-a literature review synthesis in small and medium

enterprises. *Procedia Technology*, *16*, 206–211.

Safari, F., Safari, N., & Hasanzadeh, A. (2015). The adoption of software-as-a-service (saas): ranking the determinants. *Journal of Enterprise Information Management*, *28*(3), 400–422.

Sajjanhar, A., Hou, J., & Zhang, Y. (2004). Algorithm for web services matching. In J. X. Yu, X. Lin, H. Lu, & Y. Zhang (Eds.), *Advanced web technologies and applications* (pp. 665–670). Berlin, Heidelberg: Springer Berlin Heidelberg.

Schroeter, J., Cech, S., Götz, S., Wilke, C., & Aßmann, U. (2012). Towards modeling a variable architecture for multi-tenant saas-applications. In *Proceedings of the sixth international workshop on variability modeling of software-intensive systems* (pp. 111–120).

Sultan, N. A. (2011). Reaching for the cloud: How smes can manage. *International journal of information management*, *31*(3), 272–278.

Sun, L., Dong, H., Hussain, F. K., Hussain, O. K., & Chang, E. (2014). Cloud service selection: State-of-the-art and future research directions. *Journal of Network and Computer Applications*, *45*, 134–150.

Sun, L., Dong, H., Hussain, F. K., Hussain, O. K., Ma, J., & Zhang, Y. (2014). A hybrid fuzzy framework for cloud service selection. In *Web services (icws), 2014 ieee international conference on* (pp. 313–320).

Sundarraj, R., et al. (2011). A selection framework for saas-based enterprise resource planning applications.

Tahamtan, A., Beheshti, S. A., Anjomshoaa, A., & Tjoa, A. M. (2012, June). A cloud repository and discovery framework based on a unified business and cloud service ontology. In *2012 ieee eighth world congress on services* (p. 203-210). doi: 10.1109/SERVICES.2012.42

Toh, S. Y., & Tang, M. (2018). Software-as-a-service composition in cloud

computing using genetic algorithm. In *International conference on neural information processing* (pp. 542–551).

Tsai, W., Bai, X., & Huang, Y. (2014). Software-as-a-service (saas): perspectives and challenges. *Science China Information Sciences*, *57*(5), 1–15.

Turksen, I., Tian, Y., & Berg, M. (1992). A fuzzy expert system for a service centre of

spare parts. *Expert Systems with Applications*, *5*(3), 447 - 464. Retrieved from `http://www.sciencedirect.com/science/article/pii/095741749290029R` (Special Issue: The World Congress on Expert System) doi: https://doi.org/10.1016/0957-4174(92)90029-R

Tyrväinen, P., & Selin, J. (2011). How to sell saas: a model for main factors of marketing and selling software-as-a-service. *Software Business*, 2–16.

ur Rehman, Z., Hussain, F. K., & Hussain, O. K. (2011). Towards multi-criteria cloud service selection. In *Innovative mobile and internet services in ubiquitous computing (imis), 2011 fifth international conference on* (pp. 44–48).

ur Rehman, Z., Hussain, O. K., & Hussain, F. K. (2014). Parallel cloud service selection and ranking based on qos history. *International Journal of Parallel Programming*, *42*(5), 820–852.

Van Leekwijck, W., & Kerre, E. E. (1999). Defuzzification: criteria and classification. *Fuzzy sets and systems*, *108*(2), 159–178.

Vijaya, A., & Neelanarayanan, V. (2015). Framework for platform agnostic enterprise application development supporting multiple clouds. *Procedia Computer Science*, *50*, 73–80.

Wagle, S. S., Guzek, M., & Bouvry, P. (2015). Cloud service providers ranking based on service delivery and consumer experience. In *Cloud networking (cloudnet), 2015 ieee 4th international conference on* (pp. 209–212).

Wang, D., Yang, Y., & Mi, Z. (2015, April). A genetic-based approach to web service composition in geo-distributed cloud environment. *Comput. Electr. Eng.*, *43*(C), 129–141. Retrieved from http://dx.doi.org/10.1016/j.compeleceng.2014.10.008 doi: 10.1016/j.compeleceng.2014.10.008

Wang, Y., He, Q., & Yang, Y. (2015). Qos-aware service recommendation for multi-tenant saas on the cloud. In *2015 ieee international conference on services computing* (pp. 178–185).

Waters, B. (2005). Software as a service: A look at the customer benefits. *Journal of Digital Asset Management*, *1*(1), 32–39.

Wu, C.-S., & Khoury, I. (2012). Tree-based search algorithm for web service composition in saas. In *2012 ninth international conference on information technology-new generations* (pp. 132–138).

Ye, Z., Zhou, X., & Bouguettaya, A. (2011). Genetic algorithm based qos-aware service compositions in cloud computing. In *Database systems for advanced applications* (pp. 321–334).

Yoon, K. P., & Hwang, C.-L. (1995). *Multiple attribute decision making: an introduction* (Vol. 104). Sage publications.

Zadeh, L. A. (1999). Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, *100*, 9–34.

Zhang, W., Cao, K., Liu, S., & Huang, B. (2016). A multi-objective optimization approach for health-care facility location-allocation problems in highly developed cities such as hong kong. *Computers, Environment and Urban Systems*, *59*, 220–230.

Zheng, Z., Zhang, Y., et al. (2010). Cloudrank: A qos-driven component ranking framework for cloud computing. In *Reliable distributed systems, 2010 29th ieee symposium on* (pp. 184–193).

Appendix A

Chapter 5 data results

Table A.1 : Ranking Cloud SaaS top-20 CRM providers using Find SaaS SNFCP considering two non-functional preferences

Ranking	Service Name	Supporting mobile	Free trial version	Rating	reviews	price	Language	Service location	Year Founds	Provider Link	AV	RT	TH	DB	DW	CSB
5	OroCRM	Yes	Yes	5	7	295	English	USA	2012	www.oroinc.com/	99.92	508	118	0.01450999	0.044148568	0.752636442
20	bpm'online	Yes	Yes	5	67	25	English	UK	2002	www.bpmonline.com/	100	1009	214	0.014534699	0.044121732	0.75220621
6	HubSpot CRM	Yes	Yes	4.5	865	50	English	USA	2006	www.hubspot.com/	100	360	1194	0.015116777	0.034078203	0.692717082
13	Apptivo CRM	Yes	Yes	4.5	167	10	English	USA	2010	www.apptivo.com	100	255	555	0.016509563	0.033540136	0.670136617
18	Base	Yes	Yes	4.5	93	45	English	USA	2008	www.getbase.com	100	229	517	0.016615557	0.033592428	0.669065443
8	Pipedrive	Yes	Yes	4.5	1317	15	English	USA	2010	www.pipedrive.com	99.81	673	126	0.016929628	0.033206519	0.66232691
19	Salesboom	Yes	Yes	4.5	22	14	English	Canada	2003	www.salesboom.com	100	352	262	0.017577962	0.033252544	0.65418479
1	Salesforce	Yes	Yes	4	5869	25	English	USA	1999	www.salesforce.com	100	583	681	0.022995087	0.025174798	0.522625254
16	SuiteCRM	Yes	Yes	4	10	1.00E-04	English	UK	2006	www.suitecrm.com	100	3230	30	0.484085084	0.483835912	0.499871285
10	SAP	NA	NA	4	58	60	English	Germany	1972	www.sap.com	100	362	1345	0.024993086	0.02379392	0.487710194
4	Microsoft Dynamics	Yes	Yes	4	1545	65	English	USA	1975	www.microsoft.com	100	335	596	0.024288122	0.022778511	0.483963039
12	Oracle Sales Cloud	NA	NA	4	43	600	English	USA	1977	www.oracle.com	100	174	345	0.025550985	0.023127158	0.475103542
7	Act!	Yes	Yes	4	182	10	English	USA	1987	www.swiftpage.com	100	183	199	0.025688394	0.022968577	0.472051108
2	Zoho	Yes	Yes	4	881	12	English	USA	1996	www.zoho.com	100	420	104	0.025662096	0.022264033	0.464548948
14	Infusionsoft	Yes	Yes	4	533	99	English	USA	2001	www.infusionsoft.com	100	815	91	0.026209088	0.022104395	0.457520209
15	SugarCRM	Yes	Yes	4	183	40	English	USA	2004	www.sugarcrm.com	100	654	43	0.026499976	0.022102554	0.454761389
11	Insightly	Yes	Yes	4	245	29	English	USA	2010	www.insightly.com	100	1597	37	0.026702506	0.022057184	0.452365139
3	Odoo	Yes	Yes	4	110	20	English	USA	2002	www.odoo.com	99.86	1499	35	0.026801067	0.022054245	0.451419588
9	Maximizer	Yes	Yes	4	49	60	English	Canada	1987	www.maximizer.com	100	1452	52	0.026809811	0.022054514	0.451341832
17	Sage CRM	Yes	Yes	3	25	45	English	UK	1981	www.sagesoftware.com	99.39	1839	79	0.046665652	0.000430718	0.009145462

Table A.2 : Ranking Cloud SaaS top-20 CRM providers using Find SaaS SNFCP considering four non-functional preferences

Ranking	Service Name	Supporting mobile	Free trial version	Rating	reviews	price	Language	Service location	Year Founds	Provider Link	AV	RT	TH	DB	DW	CSB
7	Act!	Yes	Yes	4	182	10	English	USA	1987	www.swiftpage.com	100	183	199	0.011460974	0.070236301	0.859714125
12	Oracle Sales Cloud	NA	NA	4	43	600	English	USA	1977	www.oracle.com	100	174	345	0.013842746	0.07409908	0.842592008
18	Base	Yes	Yes	4.5	93	45	English	USA	2008	www.getbase.com	100	229	517	0.012400195	0.055325846	0.81690654
13	Apptivo CRM	Yes	Yes	4.5	167	10	English	USA	2010	www.apptivo.com	100	255	555	0.017095404	0.04927554	0.742426388
4	Microsoft Dynamics	Yes	Yes	4	1545	65	English	USA	1975	www.microsoft.com	100	335	596	0.028331562	0.036624109	0.563832352
19	Salesboom	Yes	Yes	4.5	22	14	English	Canada	2003	www.salesboom.com	100	352	262	0.030927942	0.03451446	0.527402096
6	HubSpot CRM	Yes	Yes	4.5	865	50	English	USA	2006	www.hubspot.com/	100	360	1194	0.031072978	0.034087232	0.523129556
10	SAP	NA	NA	4	58	60	English	Germany	1972	www.sap.com	100	362	1345	0.031554313	0.03398837	0.518568487
16	SuiteCRM	Yes	Yes	4	10	1.00E-04	English	UK	2006	www.suitecrm.com	100	3230	30	0.355337371	0.349398142	0.4957862
2	Zoho	Yes	Yes	4	881	12	English	USA	1996	www.zoho.com	100	420	104	0.036804831	0.028256867	0.434308789
5	OroCRM	Yes	Yes	5	7	295	English	USA	2012	www.oroinc.com/	99.92	508	118	0.042477713	0.022636687	0.347644869
1	Salesforce	Yes	Yes	4	5869	25	English	USA	1999	www.salesforce.com	100	583	681	0.044858042	0.021042989	0.319311984
15	SugarCRM	Yes	Yes	4	183	40	English	USA	2004	www.sugarcrm.com	100	654	43	0.048323258	0.016628083	0.256008305
8	Pipedrive	Yes	Yes	4.5	1317	15	English	USA	2010	www.pipedrive.com	99.81	673	126	0.048619917	0.016160561	0.24946653
14	Infusionsoft	Yes	Yes	4	533	99	English	USA	2001	www.infusionsoft.com	100	815	91	0.052260263	0.01254181	0.193540259
20	bpm'online	Yes	Yes	5	67	25	English	UK	2002	www.bpmonline.com/	100	1009	214	0.055454755	0.009408667	0.145053506
9	Maximizer	Yes	Yes	4	49	60	English	Canada	1987	www.maximizer.com	100	1452	52	0.059594808	0.005248068	0.080935147
3	Odoo	Yes	Yes	4	110	20	English	USA	2002	www.odoo.com	99.86	1499	35	0.059879533	0.004919406	0.075917996
11	Insightly	Yes	Yes	4	245	29	English	USA	2010	www.insightly.com	100	1597	37	0.060403988	0.004371971	0.067493733
17	Sage CRM	Yes	Yes	3	25	45	English	UK	1981	www.sagesoftware.com	99.39	1839	79	0.061545016	0.003307304	0.050997464

Table A.3 : Ranking Cloud SaaS top-20 CRM providers using Find SaaS SNFCP considering six non-functional preferences

Ranking	Service Name	Supporting mobile	Free trial version	Rating	reviews	price	Language	Service location	Year Founds	Provider Link	AV	RT	TH	DB	DW	CSB
7	Act!	Yes	Yes	4	182	10	English	USA	1987	www.swiftpage.com	100	183	199	0.014607509	0.060677749	0.805971186
18	Base	Yes	Yes	4.5	93	45	English	USA	2008	www.getbase.com	100	229	517	0.013538592	0.049648025	0.785736396
12	Oracle Sales Cloud	NA	NA	4	43	600	English	USA	1977	www.oracle.com	100	174	345	0.01785229	0.063759446	0.781253394
13	Apptivo CRM	Yes	Yes	4.5	167	10	English	USA	2010	www.apptivo.com	100	255	555	0.015907175	0.044942268	0.73858142
6	HubSpot CRM	Yes	Yes	4.5	865	50	English	USA	2006	www.hubspot.com/	100	360	1194	0.027662324	0.038844072	0.584065206
4	Microsoft Dynamics	Yes	Yes	4	1545	65	English	USA	1975	www.microsoft.com	100	335	596	0.035933652	0.049358115	0.578697294
19	Salesboom	Yes	Yes	4.5	22	14	English	Canada	2003	www.salesboom.com	100	352	262	0.028145253	0.033423079	0.542861529
1	Salesforce	Yes	Yes	4	5869	25	English	USA	1999	www.salesforce.com	100	583	681	0.134740457	0.142006084	0.513126861
10	SAP	NA	NA	4	58	60	English	Germany	1972	www.sap.com	100	362	1345	0.029950732	0.030571606	0.505129299
2	Zoho	Yes	Yes	4	881	12	English	USA	1996	www.zoho.com	100	420	104	0.033578367	0.033596338	0.500133765
16	SuiteCRM	Yes	Yes	4	10	1.00E-04	English	UK	2006	www.suitecrm.com	100	3230	30	0.163404802	0.15366273	0.484637228
8	Pipedrive	Yes	Yes	4.5	1317	15	English	USA	2010	www.pipedrive.com	99.81	673	126	0.04553817	0.037865279	0.454001359
5	OroCRM	Yes	Yes	5	7	295	English	USA	2012	www.oroinc.com/	99.92	508	118	0.037131632	0.02883637	0.437126618
20	bpm'online	Yes	Yes	5	67	25	English	UK	2002	www.bpmonline.com/	100	1009	214	0.047578728	0.022987442	0.325757261
14	Infusionsoft	Yes	Yes	4	533	99	English	USA	2001	www.infusionsoft.com	100	815	91	0.045051412	0.019690958	0.304143297
15	SugarCRM	Yes	Yes	4	183	40	English	USA	2004	www.sugarcrm.com	100	654	43	0.042375489	0.018232578	0.300827573
11	Insightly	Yes	Yes	4	245	29	English	USA	2010	www.insightly.com	100	1597	37	0.052163714	0.012729218	0.196157235
3	Odoo	Yes	Yes	4	110	20	English	USA	2002	www.odoo.com	99.86	1499	35	0.052188369	0.011798671	0.184391575
9	Maximizer	Yes	Yes	4	49	60	English	Canada	1987	www.maximizer.com	100	1452	52	0.052229384	0.01167626	0.182710933
17	Sage CRM	Yes	Yes	3	25	45	English	UK	1981	www.sagesoftware.com	99.39	1839	79	0.057078348	0.002768897	0.046266078

Table A.4 : Ranking Cloud SaaS top-20 CRM providers using Find SaaS M2NFCP considering two non-functional preferences

Ranking	Service Name	Supporting mobile	Free trial version	Rating	reviews	price	Language	Service location	Year Founds	Provider Link	AV	RT	TH	DB	DW	CSB
20	bpm'online	Yes	Yes	5	67	25	English	UK	2002	www.bpmonline.com/	100	1009	214	0.016412813	0.053797143	0.766232397
5	OroCRM	Yes	Yes	5	7	295	English	USA	2012	www.oroinc.com/	99.92	508	118	0.01646358	0.053707181	0.765378352
18	Base	Yes	Yes	4.5	93	45	English	USA	2008	www.getbase.com	100	229	517	0.019716966	0.041068519	0.675630363
13	Apptivo CRM	Yes	Yes	4.5	167	10	English	USA	2010	www.apptivo.com	100	255	555	0.020757431	0.041800166	0.668186883
19	Salesboom	Yes	Yes	4.5	22	14	English	Canada	2003	www.salesboom.com	100	352	262	0.020505025	0.040427172	0.663477999
6	HubSpot CRM	Yes	Yes	4.5	865	50	English	USA	2006	www.hubspot.com/	100	360	1194	0.052433257	0.065848237	0.556707855
8	Pipedrive	Yes	Yes	4.5	1317	15	English	USA	2010	www.pipedrive.com	99.81	673	126	0.07886294	0.088205801	0.527961128
4	Microsoft Dynamics	Yes	Yes	4	1545	65	English	USA	1975	www.microsoft.com	100	335	596	0.09464809	0.096114683	0.503844023
16	SuiteCRM	Yes	Yes	4	10	1.00E-04	English	UK	2006	www.suitecrm.com	100	3230	30	0.029377345	0.029750431	0.503154912
1	Salesforce	Yes	Yes	4	5869	25	English	USA	1999	www.salesforce.com	100	583	681	0.35017529	0.352411555	0.501591451
2	Zoho	Yes	Yes	4	881	12	English	USA	1996	www.zoho.com	100	420	104	0.058841092	0.05887833	0.500158165
14	Infusionsoft	Yes	Yes	4	533	99	English	USA	2001	www.infusionsoft.com	100	815	91	0.042858954	0.041376375	0.491199776
10	SAP	NA	NA	4	58	60	English	Germany	1972	www.sap.com	100	362	1345	0.030021765	0.028704656	0.488786053
7	Act!	Yes	Yes	4	182	10	English	USA	1987	www.swiftpage.com	100	183	199	0.031810568	0.029626377	0.482224128
12	Oracle Sales Cloud	NA	NA	4	43	600	English	USA	1977	www.oracle.com	100	174	345	0.030562105	0.027966927	0.47783
11	Insightly	Yes	Yes	4	245	29	English	USA	2010	www.insightly.com	100	1597	37	0.033945825	0.030361554	0.47213173
15	SugarCRM	Yes	Yes	4	183	40	English	USA	2004	www.sugarcrm.com	100	654	43	0.032608639	0.028853472	0.469451372
3	Odoo	Yes	Yes	4	110	20	English	USA	2002	www.odoo.com	99.86	1499	35	0.032064109	0.027520988	0.461877031
9	Maximizer	Yes	Yes	4	49	60	English	Canada	1987	www.maximizer.com	100	1452	52	0.031821679	0.026946423	0.458521236
17	Sage CRM	Yes	Yes	3	25	45	English	UK	1981	www.sagesoftware.com	99.39	1839	79	0.056333949	0.001121555	0.019520404

Table A.5 : Ranking Cloud SaaS top-20 CRM providers using Find SaaS M2NFCP considering four non-functional preferences

Ranking	Service Name	Supporting mobile	Free trial version	Rating	reviews	price	Language	Service location	Year Founds	Provider Link	AV	RT	TH	DB	DW	CSB
6	HubSpot CRM	Yes	Yes	4.5	865	50	English	USA	2006	www.hubspot.com/	100	360	1194	0.008046367	0.021611904	0.728697381
8	Pipedrive	Yes	Yes	4.5	1317	15	English	USA	2010	www.pipedrive.com	99.81	673	126	0.015677674	0.031454178	0.66736562
4	Microsoft Dynamics	Yes	Yes	4	1545	65	English	USA	1975	www.microsoft.com	100	335	596	0.023726298	0.036150722	0.603749523
2	Zoho	Yes	Yes	4	881	12	English	USA	1996	www.zoho.com	100	420	104	0.016180594	0.020085355	0.553835082
1	Salesforce	Yes	Yes	4	5869	25	English	USA	1999	www.salesforce.com	100	583	681	0.123105877	0.140208036	0.532474849
16	SuiteCRM	Yes	Yes	4	10	1.00E-04	English	UK	2006	www.suitecrm.com	100	3230	30	0.300024027	0.298970041	0.499120204
20	bpm'online	Yes	Yes	5	67	25	English	UK	2002	www.bpmonline.com/	100	1009	214	0.018572513	0.015109357	0.448590202
5	OroCRM	Yes	Yes	5	7	295	English	USA	2012	www.oroinc.com/	99.92	508	118	0.019921842	0.015168322	0.432267068
14	Infusionsoft	Yes	Yes	4	533	99	English	USA	2001	www.infusionsoft.com	100	815	91	0.017468035	0.011660703	0.400316114
13	Apptivo CRM	Yes	Yes	4.5	167	10	English	USA	2010	www.apptivo.com	100	255	555	0.01735289	0.008753638	0.335304573
18	Base	Yes	Yes	4.5	93	45	English	USA	2008	www.getbase.com	100	229	517	0.018941742	0.008433629	0.308073596
17	Sage CRM	Yes	Yes	3	25	45	English	UK	1981	www.sagesoftware.com	99.39	1839	79	0.036037539	0.015105462	0.295357362
19	Salesboom	Yes	Yes	4.5	22	14	English	Canada	2003	www.salesboom.com	100	352	262	0.020769447	0.007864411	0.274654268
10	SAP	NA	NA	4	58	60	English	Germany	1972	www.sap.com	100	362	1345	0.023502657	0.005605369	0.192571258
7	Act!	Yes	Yes	4	182	10	English	USA	1987	www.swiftpage.com	100	183	199	0.021671944	0.005117581	0.19102917
11	Insightly	Yes	Yes	4	245	29	English	USA	2010	www.insightly.com	100	1597	37	0.021168027	0.004703819	0.181812252
12	Oracle Sales Cloud	NA	NA	4	43	600	English	USA	1977	www.oracle.com	100	174	345	0.023996343	0.004387315	0.154571871
15	SugarCRM	Yes	Yes	4	183	40	English	USA	2004	www.sugarcrm.com	100	654	43	0.022024864	0.003338571	0.131629302
3	Odoo	Yes	Yes	4	110	20	English	USA	2002	www.odoo.com	99.86	1499	35	0.02335412	0.001482271	0.05968141
9	Maximizer	Yes	Yes	4	49	60	English	Canada	1987	www.maximizer.com	100	1452	52	0.024406861	0.000611073	0.02442539

Table A.6 : Ranking Cloud SaaS top-20 CRM providers using Find SaaS M2NFCP considering six non-functional preferences

Rankig	Service Name	Supporting mobile	Free trial version	Rating	reviews	price	Language	Service location	Year Founds	Provider Link	AV	RT	TH	DB	DW	CSB
1	Salesforce	Yes	Yes	4	5869	25	English	USA	1999	www.salesforce.com	100	583	681	0.064383816	0.085142857	0.569415845
12	Oracle Sales Cloud	NA	NA	4	43	600	English	USA	1977	www.oracle.com	100	174	345	0.070495583	0.078665622	0.527386611
7	Act!	Yes	Yes	4	182	10	English	USA	1987	www.swiftpage.com	100	183	199	0.07163869	0.076612354	0.516774465
16	SuiteCRM	Yes	Yes	4	10	1.00E-04	English	UK	2006	www.suitecrm.com	100	3230	30	0.234347948	0.212260977	0.475272582
18	Base	Yes	Yes	4.5	93	45	English	USA	2008	www.getbase.com	100	229	517	0.065542822	0.056274867	0.461959729
13	Apptivo CRM	Yes	Yes	4.5	167	10	English	USA	2010	www.apptivo.com	100	255	555	0.065371143	0.049269672	0.429774264
4	Microsoft Dynamics	Yes	Yes	4	1545	65	English	USA	1975	www.microsoft.com	100	335	596	0.056489229	0.040921842	0.420094367
6	HubSpot CRM	Yes	Yes	4.5	865	50	English	USA	2006	www.hubspot.com/	100	360	1194	0.057448784	0.03159925	0.354856238
2	Zoho	Yes	Yes	4	881	12	English	USA	1996	www.zoho.com	100	420	104	0.078155716	0.042591795	0.35273435
19	Salesboom	Yes	Yes	4.5	22	14	English	Canada	2003	www.salesboom.com	100	352	262	0.078673477	0.040958805	0.342372513
8	Pipedrive	Yes	Yes	4.5	1317	15	English	USA	2010	www.pipedrive.com	99.81	673	126	0.082306226	0.039503063	0.32430255
10	SAP	NA	NA	4	58	60	English	Germany	1972	www.sap.com	100	362	1345	0.067394535	0.030708836	0.313025287
5	OroCRM	Yes	Yes	5	7	295	English	USA	2012	www.oroinc.com/	99.92	508	118	0.088312317	0.038822328	0.305363875
15	SugarCRM	Yes	Yes	4	183	40	English	USA	2004	www.sugarcrm.com	100	654	43	0.092811743	0.038005484	0.290523538
14	Infusionsoft	Yes	Yes	4	533	99	English	USA	2001	www.infusionsoft.com	100	815	91	0.09189027	0.036034017	0.281682376
11	Insightly	Yes	Yes	4	245	29	English	USA	2010	www.insightly.com	100	1597	37	0.101408879	0.037833361	0.271708935
3	Odoo	Yes	Yes	4	110	20	English	USA	2002	www.odoo.com	99.86	1499	35	0.1020605	0.037722	0.269862109
9	Maximizer	Yes	Yes	4	49	60	English	Canada	1987	www.maximizer.com	100	1452	52	0.10199333	0.037016102	0.266284821
17	Sage CRM	Yes	Yes	3	25	45	English	UK	1981	www.sagesoftware.com	99.39	1839	79	0.104800286	0.037853292	0.26535116
20	bpm'online	Yes	Yes	5	67	25	English	UK	2002	www.bpmonline.com/	100	1009	214	0.09515981	0.032190165	0.252769311

Table A.7 : Ranking Cloud SaaS top-20 CRM providers using Find SaaS LNFCP considering two non-functional preferences

Ranking	Service Name	Supporting mobile	Free trial version	Rating	reviews	price	Language	Service location	Year Founds	Provider Link	AV	RT	TH	DB	DW	CSB
16	SuiteCRM	Yes	Yes	4	10	1.00E-04	English	UK	2006	www.suitecrm.com	100	3230	30	0.258534184	0.372881214	0.590548179
1	Salesforce	Yes	Yes	4	5869	25	English	USA	1999	www.salesforce.com	100	583	681	0.372905789	0.252651809	0.403882568
4	Microsoft Dynamics	Yes	Yes	4	1545	65	English	USA	1975	www.microsoft.com	100	335	596	0.418783187	0.062235208	0.129382179
8	Pipedrive	Yes	Yes	4.5	1317	15	English	USA	2010	www.pipedrive.com	99.81	673	126	0.423525198	0.051995403	0.109344164
6	HubSpot CRM	Yes	Yes	4.5	865	50	English	USA	2006	www.hubspot.com/	100	360	1194	0.433219869	0.033234591	0.071249379
2	Zoho	Yes	Yes	4	881	12	English	USA	1996	www.zoho.com	100	420	104	0.43295062	0.032864276	0.070552221
14	Infusionsoft	Yes	Yes	4	533	99	English	USA	2001	www.infusionsoft.com	100	815	91	0.440961662	0.017462855	0.038093196
10	SAP	NA	NA	4	58	60	English	Germany	1972	www.sap.com	100	362	1345	0.452369101	0.010422881	0.022521742
12	Oracle Sales Cloud	NA	NA	4	43	600	English	USA	1977	www.oracle.com	100	174	345	0.452780501	0.008748713	0.018955924
7	Act!	Yes	Yes	4	182	10	English	USA	1987	www.swiftpage.com	100	183	199	0.449347713	0.00732576	0.016041571
18	Base	Yes	Yes	4.5	93	45	English	USA	2008	www.getbase.com	100	229	517	0.451519522	0.006862437	0.014971
19	Salesboom	Yes	Yes	4.5	22	14	English	Canada	2003	www.salesboom.com	100	352	262	0.453329378	0.006514269	0.014166269
5	OroCRM	Yes	Yes	5	7	295	English	USA	2012	www.oroinc.com/	99.92	508	118	0.453736911	0.006507358	0.014138923
13	Apptivo CRM	Yes	Yes	4.5	167	10	English	USA	2010	www.apptivo.com	100	255	555	0.449685323	0.006355362	0.013935953
17	Sage CRM	Yes	Yes	3	25	45	English	UK	1981	www.sagesoftware.com	99.39	1839	79	0.453319304	0.005024982	0.010963335
11	Insightly	Yes	Yes	4	245	29	English	USA	2010	www.insightly.com	100	1597	37	0.447880496	0.004803925	0.010612084
9	Maximizer	Yes	Yes	4	49	60	English	Canada	1987	www.maximizer.com	100	1452	52	0.452717065	0.004087185	0.008947344
20	bpm'online	Yes	Yes	5	67	25	English	UK	2002	www.bpmonline.com/	100	1009	214	0.452239167	0.003980468	0.008724894
15	SugarCRM	Yes	Yes	4	183	40	English	USA	2004	www.sugarcrm.com	100	654	43	0.449381586	0.002717384	0.006010594
3	Odoo	Yes	Yes	4	110	20	English	USA	2002	www.odoo.com	99.86	1499	35	0.451200554	0.001632688	0.003605495

Table A.8 : Ranking Cloud SaaS top-20 CRM providers using Find SaaS LNFCP considering four non-functional preferences

Ranking	Service Name	Supporting mobile	Free trial version	Rating	reviews	price	Language	Service location	Year Founds	Provider Link	AV	RT	TH	DB	DW	CSB
16	SuiteCRM	Yes	Yes	4	10	1.00E-04	English	UK	2006	www.suitecrm.com	100	3230	30	0.171151814	0.247691673	0.591370478
1	Salesforce	Yes	Yes	4	5869	25	English	USA	1999	www.salesforce.com	100	583	681	0.247274039	0.153107151	0.382403456
4	Microsoft Dynamics	Yes	Yes	4	1545	65	English	USA	1975	www.microsoft.com	100	335	596	0.277518544	0.027310512	0.089592875
8	Pipedrive	Yes	Yes	4.5	1317	15	English	USA	2010	www.pipedrive.com	99.81	673	126	0.280536473	0.020959979	0.06951982
5	OroCRM	Yes	Yes	5	7	295	English	USA	2012	www.oroinc.com/	99.92	508	118	0.300436942	0.02000798	0.062438125
17	Sage CRM	Yes	Yes	3	25	45	English	UK	1981	www.sagesoftware.com	99.39	1839	79	0.300710871	0.019495991	0.060885614
20	bpm'online	Yes	Yes	5	67	25	English	UK	2002	www.bpmonline.com/	100	1009	214	0.299448025	0.018442144	0.058014201
19	Salesboom	Yes	Yes	4.5	22	14	English	Canada	2003	www.salesboom.com	100	352	262	0.300201909	0.018072273	0.056782089
10	SAP	NA	NA	4	58	60	English	Germany	1972	www.sap.com	100	362	1345	0.299670891	0.017515357	0.055221047
12	Oracle Sales Cloud	NA	NA	4	43	600	English	USA	1977	www.oracle.com	100	174	345	0.299942419	0.017455784	0.054996481
9	Maximizer	Yes	Yes	4	49	60	English	Canada	1987	www.maximizer.com	100	1452	52	0.299900973	0.01653311	0.052248195
18	Base	Yes	Yes	4.5	93	45	English	USA	2008	www.getbase.com	100	229	517	0.299007651	0.016465104	0.052191842
3	Odoo	Yes	Yes	4	110	20	English	USA	2002	www.odoo.com	99.86	1499	35	0.298900988	0.014749498	0.047025266
13	Apptivo CRM	Yes	Yes	4.5	167	10	English	USA	2010	www.apptivo.com	100	255	555	0.29779734	0.014427365	0.046208274
7	Act!	Yes	Yes	4	182	10	English	USA	1987	www.swiftpage.com	100	183	199	0.297678059	0.013481306	0.043326048
15	SugarCRM	Yes	Yes	4	183	40	English	USA	2004	www.sugarcrm.com	100	654	43	0.297701093	0.012665075	0.040806879
11	Insightly	Yes	Yes	4	245	29	English	USA	2010	www.insightly.com	100	1597	37	0.2967112	0.010814707	0.035166817
6	HubSpot CRM	Yes	Yes	4.5	865	50	English	USA	2006	www.hubspot.com/	100	360	1194	0.286932304	0.010356575	0.034836737
2	Zoho	Yes	Yes	4	881	12	English	USA	1996	www.zoho.com	100	420	104	0.286862658	0.007951525	0.026971309
14	Infusionsoft	Yes	Yes	4	533	99	English	USA	2001	www.infusionsoft.com	100	815	91	0.292146866	0.002576387	0.008741715

Table A.9 : Ranking Cloud SaaS top-20 CRM providers using Find SaaS LNFCP considering six non-functional preferences

Ranking	Service Name	Supporting mobile	Free trial version	Rating	reviews	price	Language	Service location	Year Founds	Provider Link	AV	RT	TH	DB	DW	CSB
16	SuiteCRM	Yes	Yes	4	10	1.00E-04	English	UK	2006	www.suitecrm.com	100	3230	30	0.11795163	0.157264575	0.571421931
1	Salesforce	Yes	Yes	4	5869	25	English	USA	1999	www.salesforce.com	100	583	681	0.160349087	0.108829251	0.404301668
12	Oracle Sales Cloud	NA	NA	4	43	600	English	USA	1977	www.oracle.com	100	174	345	0.19028103	0.033773389	0.150737439
4	Microsoft Dynamics	Yes	Yes	4	1545	65	English	USA	1975	www.microsoft.com	100	335	596	0.177522182	0.031192702	0.149451256
7	Act!	Yes	Yes	4	182	10	English	USA	1987	www.swiftpage.com	100	183	199	0.188855851	0.0316315	0.143461745
8	Pipedrive	Yes	Yes	4.5	1317	15	English	USA	2010	www.pipedrive.com	99.81	673	126	0.1813924	0.027316405	0.130882859
18	Base	Yes	Yes	4.5	93	45	English	USA	2008	www.getbase.com	100	229	517	0.189972447	0.024570193	0.11452359
6	HubSpot CRM	Yes	Yes	4.5	865	50	English	USA	2006	www.hubspot.com/	100	360	1194	0.183620545	0.021818137	0.106202671
13	Apptivo CRM	Yes	Yes	4.5	167	10	English	USA	2010	www.apptivo.com	100	255	555	0.189466728	0.021591777	0.102302329
2	Zoho	Yes	Yes	4	881	12	English	USA	1996	www.zoho.com	100	420	104	0.184119672	0.018689731	0.092154165
20	bpm'online	Yes	Yes	5	67	25	English	UK	2002	www.bpmonline.com/	100	1009	214	0.193987196	0.017812804	0.084102004
5	OroCRM	Yes	Yes	5	7	295	English	USA	2012	www.oroinc.com/	99.92	508	118	0.193104378	0.01635634	0.078087862
19	Salesboom	Yes	Yes	4.5	22	14	English	Canada	2003	www.salesboom.com	100	352	262	0.191921676	0.014877209	0.071940469
14	Infusionsoft	Yes	Yes	4	533	99	English	USA	2001	www.infusionsoft.com	100	815	91	0.189126452	0.013677637	0.067442609
11	Insightly	Yes	Yes	4	245	29	English	USA	2010	www.insightly.com	100	1597	37	0.193021108	0.01371007	0.066318346
3	Odoo	Yes	Yes	4	110	20	English	USA	2002	www.odoo.com	99.86	1499	35	0.194304011	0.012859212	0.06207285
9	Maximizer	Yes	Yes	4	49	60	English	Canada	1987	www.maximizer.com	100	1452	52	0.194884847	0.012608136	0.060764156
10	SAP	NA	NA	4	58	60	English	Germany	1972	www.sap.com	100	362	1345	0.191734306	0.01196916	0.058757764
17	Sage CRM	Yes	Yes	3	25	45	English	UK	1981	www.sagesoftware.com	99.39	1839	79	0.195905695	0.010807258	0.052281474
15	SugarCRM	Yes	Yes	4	183	40	English	USA	2004	www.sugarcrm.com	100	654	43	0.192111295	0.009144552	0.045437448

Table A.10 : Ranking top 20 services for QWS Dataset using Find SaaS SNFCP considering two non-functional preferences

Ranking	Service position	price	response time	availability	throughput	successability	reliability	compliance	best practice	latency	documentation	DB	DW	CSB
1	407	30	49.53	53	15	54	60	89	69	0.82	32	0.003337245	0.022213528	0.869387704
2	1237	27	49	72	21.6	72	78	89	89	2.5	3	0.00341179	0.022673555	0.869206634
3	1292	30	46.05	89	20.7	96	67	100	77	4.48	8	0.004398384	0.023830976	0.844191145
4	84	20	47.27	61	20.3	62	67	78	82	2	3	0.004632846	0.024695673	0.842036153
5	367	30	50.75	56	8.6	56	80	89	83	6	33	0.004082791	0.021668447	0.841452636
6	86	25	43	47	3.6	47	80	100	83	5.17	95	0.004977045	0.025905764	0.838840919
7	1029	30	53	97	11.6	99	73	78	84	2	38	0.004001579	0.020784368	0.838554539
8	1155	22	42.64	64	6.1	64	67	89	72	1.78	39	0.005220714	0.026567179	0.835764086
9	1564	22	42.7	67	7.6	68	67	89	72	2.3	41	0.005247505	0.026531894	0.834877146
10	1310	33	44.97	51	7.8	52	60	89	69	1.59	35	0.00499311	0.024243216	0.82921555
11	1160	27	41	97	43.1	99	73	100	84	1	5	0.005564305	0.026944796	0.828838552
12	159	21	42.5	72	13.2	72	73	78	84	2	12	0.00554795	0.026856872	0.828792451
13	1345	31	53	83	21	84	60	89	69	4	6	0.004331728	0.020718293	0.827076855
14	74	30	42.5	72	13.1	72	73	78	84	3.5	8	0.005536482	0.025772707	0.823167504
15	1201	29	56.2	97	22.5	99	67	78	72	7	78	0.004496352	0.019728497	0.814390921
16	1367	24	40	72	19.8	72	73	78	84	3.5	5	0.00642128	0.02787239	0.812756126
17	1834	21	57.32	54	10.4	54	67	89	82	1.76	40	0.004830385	0.020690563	0.810728624
18	919	27	57.42	66	12.5	66	67	100	77	5.31	10	0.004570358	0.019510757	0.81020987
19	2461	38	49.5	85	1.8	95	67	78	72	1.5	34	0.005313176	0.021904445	0.804789112
20	1392	36	43	72	15.2	72	73	78	84	4.5	5	0.006240476	0.025254071	0.801855352

Table A.11 : Ranking top 20 services for QWS Dataset using Find SaaS SNFCP considering four non-functional preferences

Ranking	Service Position	price	response time	availability	throughput	successability	reliability	compliance	best practice	latency	documentation	DB	DW	CSB
1	1605	23	67	86	41	95	73	100	84	5	3	0.001180191	0.015941852	0.93107182
2	613	31	89	83	40.4	84	78	89	89	3	3	0.002193562	0.015741898	0.877696903
3	2413	24	107	91	36.9	97	83	89	91	3	12	0.002401921	0.015148213	0.863139433
4	1609	39	80	99	36.3	100	80	100	87	13	3	0.00241007	0.014875806	0.860575762
5	1454	44	178	98	34.4	99	83	100	91	8	9	0.002817403	0.014322906	0.835627083
6	447	46	122	99	34.3	100	73	78	84	8	12	0.00282977	0.014346253	0.835248826
7	776	34	66	83	36.1	84	78	89	89	3	11	0.002908621	0.014477608	0.832705473
8	1585	25	82	83	41.2	84	67	89	77	2	10	0.003515773	0.016445859	0.823873478
9	950	27	82	83	38.7	84	80	89	57	2	3	0.003686072	0.015722903	0.810084149
10	404	29	149	96	33.9	99	58	100	75	43	12	0.003409711	0.014067698	0.804907546
11	985	33	125	91	32.3	97	58	89	80	6	9	0.003461366	0.013540685	0.796414794
12	455	22	195	83	34.7	84	80	78	83	107	6	0.003594661	0.013634343	0.791359903
13	574	25	138	85	33	95	73	100	84	13	12	0.003561379	0.013441991	0.790548651
14	1621	45	77	83	33.3	84	78	89	89	3	11	0.003645796	0.01367233	0.789480895
15	1261	23	82	83	35.9	84	73	78	75	2	3	0.004080599	0.014929804	0.785349163
16	1623	30	101	87	34.7	95	73	100	84	2	34	0.004143143	0.014899464	0.782427755
17	1495	21	103	93	34.2	98	67	100	85	2	7	0.004183744	0.014977885	0.7816603
18	2235	22	196	83	32.8	84	83	89	91	3	57	0.003799916	0.013529746	0.780727652
19	435	48	126	97	31.5	99	58	100	75	14	62	0.00388301	0.013454382	0.776032645
20	1555	37	65	99	31.6	100	80	89	65	21	42	0.003917022	0.013558967	0.775862639

Table A.12 : Ranking top 20 services for QWS Dataset using Find SaaS SNFCP considering six non-functional preferences

Ranking	Service position	price	response time	availability	throughput	successability	reliability	compliance	best practice	latency	documentation	DB	DW	CSB
1	1422	34	59	88	28.6	96	73	89	62	2	93	0.002424109	0.013087966	0.843727605
2	831	44	57	98	34.6	100	73	100	84	1	95	0.003013214	0.014276585	0.825723022
3	1201	29	56.2	97	22.5	99	67	78	72	7	78	0.002803086	0.012249926	0.813785682
4	1455	39	64.5	99	24.3	100	73	78	80	22	88	0.003461426	0.012043349	0.776750957
5	524	45	41	83	24	84	60	89	74	3	65	0.004149634	0.013933438	0.770523832
6	1476	20	73.5	87	23.8	95	73	89	62	18.75	95	0.003902466	0.01220775	0.757764534
7	169	25	63.8	99	18.1	100	73	78	84	1.8	95	0.003851824	0.011890312	0.755317576
8	2199	26	78	100	35.3	100	67	78	72	1	63	0.004157217	0.012656259	0.75274495
9	2122	27	69.33	95	19.4	99	67	78	77	25.22	93	0.004017563	0.011484529	0.740837354
10	938	49	78	92	36.9	97	73	100	84	1	89	0.004669243	0.013322973	0.740485399
11	448	20	90	85	31.1	95	60	89	79	1	89	0.004460938	0.012527281	0.737409893
12	519	35	79	83	32	84	78	89	89	3	67	0.004331203	0.011636974	0.728760339
13	1504	48	69.71	91	22.6	97	67	89	58	15.47	92	0.0043328	0.011530361	0.726864034
14	1555	37	65	99	31.6	100	80	89	65	21	42	0.004511772	0.011981875	0.726453969
15	1488	24	95	96	28.3	99	60	89	79	1	88	0.00450527	0.01177438	0.723257581
16	759	49	68.83	100	21.4	100	73	100	80	7.33	92	0.00443643	0.011550619	0.722498509
17	1027	29	79.25	99	20.8	100	73	78	80	25.5	92	0.004422456	0.011153514	0.716071862
18	631	35	82.25	99	22.7	100	73	78	80	6.75	95	0.004548976	0.011298813	0.712958324
19	1096	44	65	86	18	86	73	89	84	4.5	91	0.004622754	0.01109717	0.705930238
20	1160	27	41	97	43.1	99	73	100	84	1	5	0.007095578	0.016419221	0.698250543

Table A.13 : Ranking top 20 services for QWS Dataset using Find SaaS SNFCP considering eight non-functional preferences

Ranking	Service position	price	response time	availability	throughput	successability	reliability	compliance	best practice	latency	documentation	DB	DW	CSB
1	822	23	101	93	17.9	98	73	78	80	0.83	93	0.000862121	0.008513631	0.908047849
2	1764	22	108.8	90	18.1	97	73	78	80	0.8	92	0.000913735	0.008355524	0.901423105
3	1767	23	111.2	93	16.9	98	67	78	77	0.4	96	0.000972144	0.008364824	0.895882232
4	8	21	102.62	91	15.3	97	67	78	82	0.93	91	0.001197166	0.008318522	0.874190329
5	2178	22	105.56	92	14.6	97	73	78	80	0.67	95	0.001208568	0.008343822	0.873480001
6	869	25	103	85	16.1	95	73	78	80	0.89	91	0.00121616	0.00812151	0.869757648
7	2268	30	105.6	100	19.2	100	73	78	84	0.6	95	0.001284318	0.008529656	0.869133778
8	405	26	102	86	17.3	86	80	89	87	59.5	92	0.001386086	0.008242084	0.856038521
9	775	20	123.7	93	15.9	98	67	100	82	0.7	96	0.001415504	0.008313047	0.85450004
10	487	29	115.4	96	18.7	99	80	78	87	1.4	93	0.001422963	0.008272693	0.853237067
11	2358	30	107.4	85	19	95	73	78	80	0.8	92	0.001413981	0.008152958	0.85220132
12	1094	27	85.25	97	19.3	99	73	89	62	28.25	92	0.001571901	0.008843846	0.849084207
13	40	27	106.75	90	16.2	96	73	78	80	24.13	93	0.001462256	0.008134424	0.847628956
14	2134	28	116	98	16.7	100	67	78	77	1.33	92	0.00147147	0.008050411	0.845464334
15	2355	33	103.4	85	18.8	95	73	78	80	0.4	87	0.001511164	0.008022137	0.841485789
16	1488	24	95	96	28.3	99	60	89	79	1	88	0.00174581	0.009218985	0.840780391
17	121	31	109.6	99	19.3	100	73	78	62	1	94	0.001580973	0.008329417	0.840473194
18	1027	29	79.25	99	20.8	100	73	78	80	25.5	92	0.001769724	0.009196198	0.838615998
19	2261	25	114	97	15.7	99	73	89	62	50.5	93	0.001561196	0.008086762	0.838183796
20	726	33	104	87	18.5	95	73	78	80	1	97	0.001665109	0.008348023	0.83370748

Table A.14 : Ranking top 20 services for QWS Dataset using Find SaaS SNFCP considering ten non-functional preferences

Ranking	Service position	price	response time	availability	throughput	successability	reliability	compliance	best practice	latency	documentation	DB	DW	CSB
1	519	35	79	83	32	84	78	89	89	3	67	0.002901218	0.009191821	0.760091915
2	435	48	126	97	31.5	99	58	100	75	14	62	0.002750454	0.008095109	0.746398198
3	2413	24	107	91	36.9	97	83	89	91	3	12	0.003163502	0.008959599	0.739051738
4	1623	30	101	87	34.7	95	73	100	84	2	34	0.003136974	0.008856871	0.738451375
5	1585	25	82	83	41.2	84	67	89	77	2	10	0.00359643	0.010035537	0.736176732
6	1605	23	67	86	41	95	73	100	84	5	3	0.003833277	0.010693468	0.736122789
7	613	31	89	83	40.4	84	78	89	89	3	3	0.003488278	0.009613594	0.733757262
8	1555	37	65	99	31.6	100	80	89	65	21	42	0.00365932	0.009898423	0.730093701
9	736	23	107	83	27.6	84	83	89	91	9	64	0.00290521	0.007829192	0.729355201
10	1609	39	80	99	36.3	100	80	100	87	13	3	0.003560402	0.009577133	0.728990108
11	305	46	126	83	31.4	84	83	89	91	3	67	0.00301027	0.008029967	0.72733646
12	741	43	111	83	29	84	83	89	91	8	64	0.002981388	0.007839417	0.724476282
13	1199	32	110	83	27.6	84	83	89	91	10	67	0.003003487	0.007735363	0.720315768
14	1114	43	108	83	28.2	84	83	89	91	8	64	0.003081156	0.007802313	0.716895796
15	1447	20	124	91	26.9	97	73	78	84	8	66	0.003037841	0.00766062	0.716048793
16	787	38	142	96	29.3	99	73	100	84	17	60	0.00305933	0.007683706	0.715226708
17	950	27	82	83	38.7	84	80	89	57	2	3	0.003885447	0.009693095	0.713853896
18	938	49	78	92	36.9	97	73	100	84	1	89	0.004382024	0.010799777	0.711363376
19	776	34	66	83	36.1	84	78	89	89	3	11	0.004106233	0.010102424	0.711004842
20	1476	20	73.5	87	23.8	95	73	89	62	18.75	95	0.003718766	0.008994037	0.707478666

Table A.15 : Ranking top 20 services for QWS Dataset using Find SaaS M2NFCP considering two non-functional preferences

Ranking	Service position	price	response time	availability	throughput	successability	reliability	compliance	best practice	latency	documentation	DB	DW	CSB
1	1237	27	49	72	21.6	72	78	89	89	2.5	3	0.003411764	0.012553017	0.786294351
2	407	30	49.53	53	15	54	60	89	69	0.82	32	0.003337239	0.011917133	0.781227353
3	84	20	47.27	61	20.3	62	67	78	82	2	3	0.004632691	0.015360921	0.768291453
4	1155	22	42.64	64	6.1	64	67	89	72	1.78	39	0.005219486	0.016683994	0.761705158
5	1564	22	42.7	67	7.6	68	67	89	72	2.3	41	0.005246307	0.016650833	0.760411303
6	86	25	43	47	3.6	47	80	100	83	5.17	95	0.0049759	0.015755194	0.759978889
7	159	21	42.5	72	13.2	72	73	78	84	2	12	0.005546743	0.017088927	0.754955639
8	1292	30	46.05	89	20.7	96	67	100	77	4.48	8	0.004398025	0.013451202	0.753601382
9	1160	27	41	97	43.1	99	73	100	84	1	5	0.005562443	0.016634132	0.749400859
10	367	30	50.75	56	8.6	56	80	89	83	6	33	0.004082779	0.011375807	0.735889219
11	74	30	42.5	72	13.1	72	73	78	84	3.5	8	0.005535273	0.015328805	0.734698427
12	1367	24	40	72	19.8	72	73	78	84	3.5	5	0.006419187	0.017701046	0.73386714
13	1310	33	44.97	51	7.8	52	60	89	69	1.59	35	0.004992571	0.013733085	0.733383379
14	1029	30	53	97	11.6	99	73	78	84	2	38	0.004001406	0.010553171	0.725075735
15	1731	24	38	71	1.9	71	67	89	77	0.88	38	0.007442065	0.019049495	0.719077887
16	1834	21	57.32	54	10.4	54	67	89	82	1.76	40	0.004829659	0.011857932	0.710583801
17	1345	31	53	83	21	84	60	89	69	4	6	0.004331569	0.010422433	0.706413936
18	181	20	58	95	16	98	73	78	80	1	5	0.005157141	0.012292763	0.704460207
19	1392	36	43	72	15.2	72	73	78	84	4.5	5	0.006239563	0.014658978	0.70143548
20	1560	29	37	71	21.3	72	73	78	84	3	4	0.008516183	0.019109564	0.691730222

Table A.16 : Ranking top 20 services for QWS Dataset using Find SaaS M2NFCP considering four non-functional preferences

Ranking	Service position	price	response time	availability	throughput	successability	reliability	compliance	best practice	latency	documentation	DB	DW	CSB
1	1605	23	67	86	41	95	73	100	84	5	3	0.001179876	0.00428974	0.784285451
2	613	31	89	83	40.4	84	78	89	89	3	3	0.002193165	0.004941112	0.692587582
3	1585	25	82	83	41.2	84	67	89	77	2	10	0.003515605	0.006413455	0.64592771
4	2413	24	107	91	36.9	97	83	89	91	3	12	0.00240029	0.004144526	0.633253314
5	950	27	82	83	38.7	84	80	89	57	2	3	0.003685503	0.005944666	0.617296134
6	776	34	66	83	36.1	84	78	89	89	3	11	0.00290669	0.004029997	0.580968523
7	1261	23	82	83	35.9	84	73	78	75	2	3	0.004079136	0.005530183	0.575502075
8	1160	27	41	97	43.1	99	73	100	84	1	5	0.0083491	0.011279447	0.574645023
9	1623	30	101	87	34.7	95	73	100	84	2	34	0.004141203	0.005372297	0.564702521
10	1494	44	57	86	40.1	95	73	100	84	1	10	0.008425906	0.010887153	0.563719765
11	1495	21	103	93	34.2	98	67	100	85	2	7	0.004181592	0.005335927	0.560642671
12	938	49	78	92	36.9	97	73	100	84	1	89	0.008538431	0.010594773	0.553737534
13	1159	41	112	83	37.7	84	73	78	75	1	3	0.008612124	0.010655399	0.55302381
14	786	48	103	94	36.6	98	80	100	87	1	8	0.008564883	0.010568092	0.552349665
15	82	47	100	94	36.3	98	73	100	84	1	1	0.008588465	0.01054666	0.55116755
16	2199	26	78	100	35.3	100	67	78	72	1	63	0.008647562	0.010512268	0.548661859
17	639	45	81	93	35.6	98	58	89	75	1	6	0.008644741	0.010500372	0.548462265
18	1314	38	106	90	35.7	97	80	100	87	1	1	0.00865529	0.010500846	0.548171399
19	831	44	57	98	34.6	100	73	100	84	1	95	0.008708982	0.010476604	0.546066391
20	2181	24	120	85	34.4	95	83	89	91	1	12	0.00881023	0.01042931	0.542076889

Table A.17 : Ranking top 20 services for QWS Dataset using Find SaaS M2NFCP considering six non-functional preferences

Ranking	Service position	price	response time	availability	throughput	successability	reliability	compliance	best practice	latency	documentation	DB	DW	CSB
1	1160	27	41	97	43.1	99	73	100	84	1	5	0.006219489	0.009041912	0.592469319
2	831	44	57	98	34.6	100	73	100	84	1	95	0.00292115	0.004132344	0.585857694
3	524	45	41	83	24	84	60	89	74	3	65	0.005451546	0.007000145	0.562184265
4	1560	29	37	71	21.3	72	73	78	84	3	4	0.008378352	0.009652521	0.535332986
5	1652	41	38	71	21.1	72	73	78	84	2	2	0.008602375	0.00955177	0.52614815
6	1367	24	40	72	19.8	72	73	78	84	3.5	5	0.008216988	0.009045354	0.523993439
7	1494	44	57	86	40.1	95	73	100	84	1	10	0.006056955	0.006618136	0.522137177
8	1605	23	67	86	41	95	73	100	84	5	3	0.006426532	0.0065341	0.504149778
9	1292	30	46.05	89	20.7	96	67	100	77	4.48	8	0.007558145	0.007676602	0.503887723
10	908	45	40	72	14	72	73	78	84	2	8	0.009379261	0.009516872	0.503641249
11	159	21	42.5	72	13.2	72	73	78	84	2	12	0.008981132	0.009069134	0.502437692
12	1731	24	38	71	1.9	71	67	89	77	0.88	38	0.010842112	0.010823434	0.499568968
13	84	20	47.27	61	20.3	62	67	78	82	2	3	0.008146306	0.008127886	0.499434056
14	74	30	42.5	72	13.1	72	73	78	84	3.5	8	0.009151074	0.009030613	0.496687306
15	1392	36	43	72	15.2	72	73	78	84	4.5	5	0.008969206	0.008847644	0.496588584
16	1544	41	38	95	2.1	99	73	78	84	1.5	6	0.011780419	0.011568979	0.49547227
17	428	44	42.5	72	14.1	72	73	78	84	3.5	6	0.009285959	0.009095042	0.494806684
18	1351	43	42.5	72	13.5	72	73	78	84	2.5	11	0.009174156	0.008970307	0.494382637
19	1564	22	42.7	67	7.6	68	67	89	72	2.3	41	0.009221178	0.008985501	0.493527734
20	1237	27	49	72	21.6	72	78	89	89	2.5	3	0.007714995	0.007460903	0.491628452

Table A.18 : Ranking top 20 services for QWS Dataset using Find SaaS M2NFCP considering eight non-functional preferences

Ranking	Service position	price	response time	availability	throughput	successability	reliability	compliance	best practice	latency	documentation	DB	DW	CSB
1	448	20	90	85	31.1	95	60	89	79	1	89	0.001342373	0.003943741	0.746056783
2	1488	24	95	96	28.3	99	60	89	79	1	88	0.001193269	0.003185841	0.727508826
3	1476	20	73.5	87	23.8	95	73	89	62	18.75	95	0.00238929	0.004433357	0.649800089
4	2199	26	78	100	35.3	100	67	78	72	1	63	0.002631773	0.004523429	0.632187488
5	938	49	78	92	36.9	97	73	100	84	1	89	0.002896429	0.004454434	0.60597431
6	831	44	57	98	34.6	100	73	100	84	1	95	0.003913939	0.005946472	0.603065337
7	1422	34	59	88	28.6	96	73	89	62	2	93	0.003540973	0.005371577	0.602698109
8	169	25	63.8	99	18.1	100	73	78	84	1.8	95	0.003293065	0.004817845	0.593995589
9	519	35	79	83	32	84	78	89	89	3	67	0.00270234	0.003935121	0.592865362
10	1027	29	79.25	99	20.8	100	73	78	80	25.5	92	0.002387701	0.003439933	0.59027946
11	2122	27	69.33	95	19.4	99	67	78	77	25.22	93	0.002930524	0.004211483	0.589677838
12	1455	39	64.5	99	24.3	100	73	78	80	22	88	0.003251766	0.004553582	0.583392565
13	1201	29	56.2	97	22.5	99	67	78	72	7	78	0.003964221	0.005514258	0.581766155
14	736	23	107	83	27.6	84	83	89	91	9	64	0.002165162	0.002969527	0.578326602
15	631	35	82.25	99	22.7	100	73	78	80	6.75	95	0.002362479	0.003237657	0.578138962
16	1447	20	124	91	26.9	97	73	78	84	8	66	0.002199257	0.002939325	0.572010911
17	1094	27	85.25	97	19.3	99	73	89	62	28.25	92	0.002520263	0.0032124	0.560367811
18	822	23	101	93	17.9	98	73	78	80	0.83	93	0.002226857	0.002799791	0.556989677
19	1555	37	65	99	31.6	100	80	89	65	21	42	0.004295592	0.005331002	0.553778645
20	1160	27	41	97	43.1	99	73	100	84	1	5	0.008294784	0.010090187	0.548827993

Table A.19 : Ranking top 20 services for QWS Dataset using Find SaaS M2NFCP considering ten non-functional preferences

Ranking	Service position	price	response time	availability	throughput	successability	reliability	compliance	best practice	latency	documentation	DB	DW	CSB
1	519	35	79	83	32	84	78	89	89	3	67	0.00264124	0.005260477	0.665738472
2	1605	23	67	86	41	95	73	100	84	5	3	0.003765731	0.007226494	0.657418666
3	1585	25	82	83	41.2	84	67	89	77	2	10	0.003500146	0.006467139	0.648836611
4	613	31	89	83	40.4	84	78	89	89	3	3	0.003274531	0.005981029	0.646209298
5	1555	37	65	99	31.6	100	80	89	65	21	42	0.003450635	0.006286206	0.645610409
6	1609	39	80	99	36.3	100	80	100	87	13	3	0.003297553	0.005891762	0.641153596
7	776	34	66	83	36.1	84	78	89	89	3	11	0.00386871	0.006794873	0.637203543
8	950	27	82	83	38.7	84	80	89	57	2	3	0.003779055	0.006321448	0.625854767
9	1623	30	101	87	34.7	95	73	100	84	2	34	0.002945662	0.004910019	0.625027804
10	2413	24	107	91	36.9	97	83	89	91	3	12	0.003078335	0.005100285	0.623611982
11	938	49	78	92	36.9	97	73	100	84	1	89	0.004112786	0.006783414	0.622548612
12	1422	34	59	88	28.6	96	73	89	62	2	93	0.004317371	0.007002878	0.618615168
13	1621	45	77	83	33.3	84	78	89	89	3	11	0.003655396	0.005817345	0.614114219
14	831	44	57	98	34.6	100	73	100	84	1	95	0.005164752	0.008217973	0.614073225
15	1261	23	82	83	35.9	84	73	78	75	2	3	0.003871778	0.006114328	0.612283491
16	2199	26	78	100	35.3	100	67	78	72	1	63	0.004243417	0.006697912	0.612166198
17	446	26	66	83	33.4	84	78	89	89	2	5	0.004381132	0.006873912	0.61074058
18	1494	44	57	86	40.1	95	73	100	84	1	10	0.005696884	0.008741369	0.605431205
19	1455	39	64.5	99	24.3	100	73	78	80	22	88	0.003956867	0.00598265	0.601905496
20	2176	29	84	91	30.9	97	73	89	62	21	8	0.003548997	0.005234148	0.595930946

Appendix B

Chapter 7 data results

Table B.1 : Top 20 Cloud SaaS providers of Customer Relationship Management (CRM) based on Capttera

Service Name	RV	RA	RbSR	Cost
Mailchimps	11040	4.5	0.92	20
Microsoft Dynamics	3393	4.5	0.716	65
Zoho CRM	3273	4	0.7	12
HubSpot	2161	4.5	0.7	0.1
Pipedrive	2083	4.5	0.7	15
monday	1784	4.5	0.7	5
Infusionsoft	992	4	0.7	23
NetSuite	444	4	0.7	0.1
Insightly	443	4	0.7	29
Deltek Vision	430	4	0.7	0.1
Freshsales	426	4.5	0.691	12
Favro	391	4.5	0.613	6.8
ActiveCampaign	370	4.5	0.53	9
Bitrix24	368	4	0.519	39
Streak	351	4.5	0.4	39
Agile CRM	334	4	0.4	8.99
Daylite	282	4.5	0.4	29
Really Simple	243	4.5	0.4	15
Vtiger	239	4.5	0.4	10

Table B.2 : Top 20 Cloud SaaS providers of Energy Software based on Capttera

Service Name	RV	RA	RbSR	Cost
ProntoForms	155	4.5	0.92	30
Wattics	48	5	0.713	50
Entronix	47	5	0.713	0.1
eSight	35	4.5	0.704	0.1
Eniscope	31	5	0.7	0.1
SyAM	22	4.5	0.7	0.1
digitalenergy	12	4.5	0.7	0.1
Energy Elephant	10	5	0.7	83.25
Fabriq OS	8	5	0.7	95
InThrMa EMS	7	5	0.7	0.1
EEM Suite	7	4.5	0.7	0.1
Sustainability Performance Management	7	4	0.7	0.1
PowerPlug Pro	6	5	0.615	0.1
ePortal	4.5	5	0.521	41.66666667
EEM	4	5	0.521	0.1
EnergyLogicIQ	4	5	0.521	0.1
Utilibill	4	4.5	0.4	1000
BuildingOS	4	4	0.4	0.1
eco—Driver	3	4.5	0.4	0.1

Table B.3 : Top 20 Cloud SaaS providers of Architecture Software based on Capttera

Service Name	RV	RA	RbSR	Cost
FreshBooks	3023	4.5	0.92	15
Bluebeam's	641	4.5	0.718	4.15
SketchUp	497	4.5	0.713	24.91666667
BQE Core	182	4.5	0.701	7.95
Streamtime	167	4.5	0.7	15
ArchiCAD	128	4.5	0.7	0.1
MicroStation's	95	4.5	0.7	250
ActCAD	70	4.5	0.7	4.983333333
SmartDraw	70	4	0.7	9.95
progeCAD Professional	55	4.5	0.7	0.1
Chief Architect	45	4.5	0.624	199
CorelCAD	35	4.5	0.541	13.31666667
Vectorworks Architect	25	4.5	0.438	28.16666667
Space Designer 3D	24	4.5	0.425	0.1
ConceptDraw	22	4.5	0.4	3.316666667
Praesto AE	20	4	0.4	35
Concrete Contractors	13	4.5	0.4	0.1
FloorPlanner	11	4.5	0.4	1.245833333
Sweet Home 3D	10	4.5	0.4	0.1

Table B.4 : Top 20 Cloud SaaS providers of Project Software based on Capttera

Service Name	RV	RA	RbSR	Cost
Trello	13348	4.5	0.92	9.99
Hive	71	4.5	0.4	12
Asana	7564	4.5	0.771	9.99
Evernote	6114	4.5	0.75	2.7
Smartsheet	1041	4.5	0.7	14
Basecamp	10656	4.5	0.831	99
Airtable	767	4.5	0.7	12
Todoist	860	4.5	0.7	3
Workfront	967	4.5	0.7	0.1
Pivotal Tracker	95	4.5	0.407	15
MeisterTask	595	4.5	0.7	8.25
KanbanFlow	45	4.5	0.4	5
Clubhouse	134	4.5	0.482	8.5
Ora	67	4.5	0.4	6
Favro	391	4.5	0.7	6.8
Miro	133	5	0.481	10
Freedcamp	157	4.5	0.521	0.124166667
Quire	82	4.5	0.4	0.1
TeamGantt	167	4.5	0.538	7.9