Web Service Selection using Soft Computing Techniques

Smita Kumari

Roll. 213CS3186

under the supervision of

Prof. Santanu Kumar Rath

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela – 769008, India

Web Service Selection using Soft Computing Techniques

Thesis Submitted

to the department of

Computer Science and Engineering

of

National Institute of Technology Rourkela

in partial fulfillment of the requirements

for the degree of

Master of Technology

by

Smita kumari

(Roll. 213CS3186)

under the supervision of

Prof. Santanu Kumar Rath

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela – 769 008, India

May 2015

Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela-769 008, India. www.nitrkl.ac.in

May, 2015

Certificate

This is to certify that the work in the thesis entitled *Web Service Selection using Soft Computing Techniques* by *Smita Kumari*, bearing roll number 213CS3186, is a record of an original research work carried out by her under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of *Master of Technology* in *Computer Science and Engineering Department*. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Place: NIT Rourkela
Date: June 1, 2015

(Prof. Santanu Ku. Rath)
Professor, CSE Department
NIT Rourkela, Odisha

Acknowledgment

First of all, I would like to express my deep sense of respect and gratitude towards my supervisor Prof. Santanu Kumar Rath, who has been the guiding force behind this work. I want to thank him for introducing me to the field of Service Oriented Architecture and giving me the opportunity to work under him. His undivided faith in this topic and ability to bring out the best of analytical and practical skills in people has been invaluable in tough periods. Without his invaluable advice and assistance it would not have been possible for me to complete this thesis. I am greatly indebted to him for his constant encouragement and invaluable advice in every aspect of my academic life. I consider it my good fortune to have got an opportunity to work with such a wonderful person.

I would also like to thank all faculty members, reasearch scholars, my seniors and juniors and all colleagues to provide me their regular suggestions and encouragements during the whole work.

At last but not the least I am in debt to my family to support me regularly during my hard times.

I wish to thank all faculty members and secretarial staff of the CSE Department for their sympathetic cooperation.

Smita Kumari

Abstract

Web service selection is one of the important aspects of SOA. It helps to integrate the services to build a particular application. Web services need to be selected using appropriate interaction styles i.e., either Simple Object Access Protocol (SOAP) or Representational State Transfer Protocol (REST) because choosing web service interaction pattern is a crucial architectural concern for developing the application, and has an impact on the development process.

In this study, the performance of web services for Enterprise Application based on SOAP and REST are compared.

Since web services operate over the network, throughput and response time are considered as metrics for evaluation.

In the literature, it is observed that, emphasis is given on interaction style for selecting web services. However, as the number of services grows day by day, it is time-consuming and difficult to select services that offer similar functionalities. Web services are often described in terms of their functionalities and set of operations. If a customer chooses an application that is of low quality or have malicious content that can affect the overall performance of the application.

Hence, web services are selected based on the quality of service (QoS) attributes. In this proposed work, various models are designed using soft computing techniques such as Back Propagation Network (BPN), Radial Basis Function Network (RBFN), Probabilistic Neural Network (PNN) and hybrid Artificial Neural Network (ANN) for web service selection, and their performances are compared based on various performance parameters.

Contents

List Of Abbreviations

SOA	Service Oriented Architecture
ANN	Artificial Neural Network
BPN	Back Propagation Network
RBFN	Radial Basis Function Network
PNN	Probablistic Neural Network
GA	Genetic Algorithm
SOAP	Simple Object Access Protocol
REST	Representational State Transfer Protocol
BPEL	Business Process Execution Language
UDDI	Universal Description Discovery Integration
WSDL	Web Service Description Language
JSON	Java Script Object Notation
JSP	Java Server Pages

List of Figures

List of Tables

Chapter 1

Introduction

Service Oriented Architecture (SOA) is the concept of building an architecture for software development and application integration, which supports loose coupling, reusability, standard service contracts, interoperability, composability, discoverability, autonomous and stateless principles. The basic building block of SOA is web service. Web service is an atomic self-supporting unit of software that performs a particular task. SOA is realized with the help of services. SOA is an architectural pattern in which application components communicate with other application components via a communications protocol in heterogeneous distributed environment. Service oriented architecture is mainly consists of three basic component as shown in Figure 1.1, such as

- Service consumer
- Service provider
- Service broker

Service provider publishes services to service-repository, it just stores the service interface definition in the repository. Service- consumer is the client that searches service repository to find desired services with the help of service description. SOA uses the standards such as Web service description language (WSDL) service

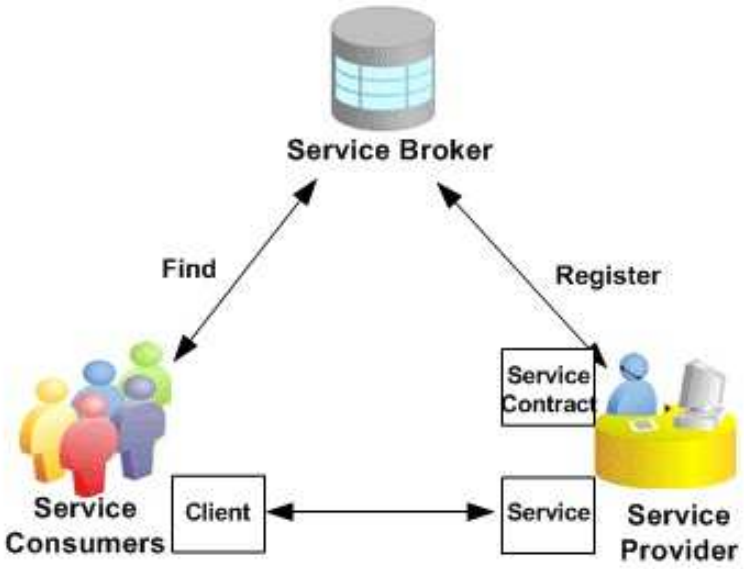

Figure 1.1: Service Oriented Architecture

interface definition, Simple Object Access Protocol (SOAP) as a communication protocol between services and universal description discovery and integration (UDDI) as a service-repository

In this work, ANN has been used for fixing up the appropriate service, because ANN has capability to learn from past experience, So network can be trained to identify or classify a particular service based on QoS attribute. For example, some application may require the high value of throughput, low response time. Based on previous learning it matches the appropriate service that satisfies the particular QoS value.

Artificial neural networks are machine learning algorithms derived from the concept of the central nervous systems of living beings, in particular, the brain [1]. Artificial neural networks (ANNs) is made up of a number of nodes. These nodes represent neuron in the brain. These neurons are interlinked with each other through which information flows, where interlinks are assigned with some weights. Weights

that are allocated to the connection between the neurons are variable until appropriate weights are obtained for which the error parameter is minimum. Weight can be updated based on learning, which make network robust to inputs [1].

In ANN, information as patterns are stored as training data, networks are trained using these data, and once the network is trained it can be used to solve the problem of classification, pattern recognition and others. Learning Algorithms can be of three types.

- Supervised Learning Algorithm
- Unsupervised Learning Algorithm
- Reinforcement Learning Algorithm

1.1 Web Services

Web Service is a software unit that provides an interoperable machine to machine communication over the network. It provides reusable, loosely coupled, autonomous, interoperable and standard way to build any application. Web services communicate with help of standard HTTP and XML based message protocol over the network [2]. In the era of information technology, it is vital for success of any organisation to seamlessly exchange information between enterprise, consumer, and provider, But most organizations use legacy applications and databases that serve critical business needs. Legacy applications communicate and exchange data which is not compatible for integration with modern software. Therefore they cannot communicate to one another efficiently.

1.1.1 Web Service Description Language (WSDL)

WSDL stands for web service description language. Web service interface is described in XML format. WSDL document structure can be given as. Service consumer desired to access Web service can analyse and comprehend WSDL file to get

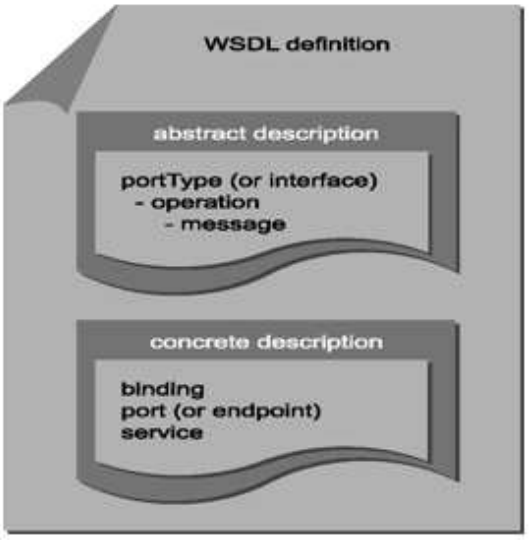

Figure 1.2: WSDL Structure

location of the service and its available functionality. WSDL is the interface description language that provides a standard way to interpret and comprehend messages and requisite parameters regardless of the network protocol or platform they use [3]. WSDL is an XML based standard description of the web services. It provides the detail about the services such as what are the related operations and where services are located.

WSDL consists of two parts namely abstract and concrete as shown in Figure1.2. There are the six elements of WSDL document.

- PortType: It describe group of operations performed by one or more service endpoint and the message involved. It is similar to function library in conventional programming language.
- Port: It specifies an address for a binding, i.e., actual target location of a service.
- Message: It describes the data elements of an operation. It consists of the part name. Part name is similar to parameter in a normal function call.

- Types: Types are data types defined by XML schema used by services for sending messages.
- Binding: It defines the data format and protocol for the group of operations provided by the service.
- Service: It specifies a collection of related endpoints enclosed in the service definitions.

1.1.2 Simple Object Access Protocol (SOAP)

Applications built on different platform and language can communicate over the network using SOAP. SOAP is a standard, extensibleformat for sending messages between a service and its client(s) [4]. SOAP is language and platform independent. Web services interoperability is supported with the help of SOAP.

The basic SOAP structure is shown in Figure 1.3. Its elements are:

- Envelope: Every SOAP message is enclosed in a container known as an envelope.
- Header: A header is optional in SOAP. It contains meta information of message such as the date on which message was sent, authentication data, etc.
- Body: It hosts the message payload. It contains call and response message information.
- Fault: If some error occurs while exchanging information, then fault message is generated. It is present in fault part of SOAP message.

1.1.3 Universal Description Discovery and Integration(UDDI)

UDDI is service repository where the service description in form of WSDL files are stored. UDDI is a standard way to create an XML-based repository that store

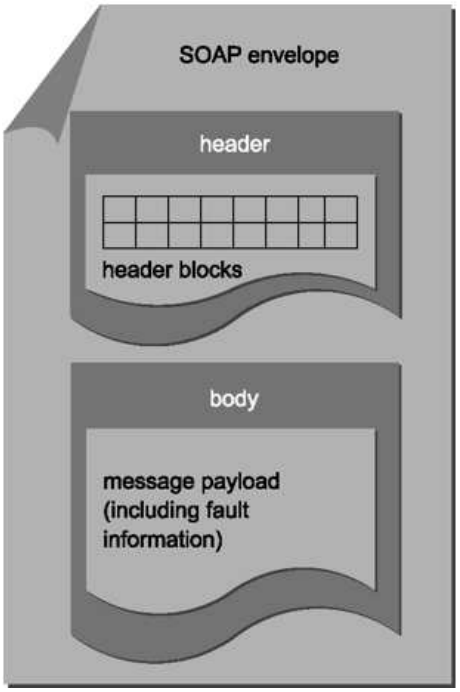

Figure 1.3: SOAP Structure

information about the Web services they offer. Investigating appropriate business applications published as Web Services in UDDI is a critical issue [5].

One of the main principles of SOA is discoverability. Even if the services are available, if services are not discoverable they cannot be utilized in any application development. Service Discoverability is achieved with the help of UDDI. UDDI stores the WSDL files and communicates with the assistance of SOAP messaging protocol [2].

UDDI stores service description, i.e., WSDL and interacts via SOAP messaging. When the web services are built, their interfaces are described using WSDL file. WSDL file is registered in UDDI registry. Registering a Web service in a UDDI registry is an optional step. UDDI registry can be of two type:

- Private UDDI- In this category registration is carried out within an organization
- Public UDDI - In this category registration is carried out among organization

1.2 Litrature Review

In this study, interaction style i.e., SOAP and REST have been considered for building web services. In this segment, the studies related to the performance of SOAP and REST based services by various author have been tabulated in Table 1.1.

Howevere, in 2013, Snehal, and Puja Padiya have compared the performance of SOAP and REST considering an example of multimedia confrencing application [20].

This section presents a review of literature on the application of soft computing techniques for web service selection. Table 1.2 shows the summary of empirical studies available in literature on web service selection.

Q. H. Al-Masri *et al.,* [6] experimentally applied the neural network for appropriate web service selection. Ezhilarasi *et al.,* [9] have proposed a framework for JIT-Oriented web service discovery using neural network. Where JIT corresponds to Just -in Time framework. Suresh *et al.,* have used various machine learning algorithms for software fault prediction using CK metric suite [12]. Lockheed Missiles*et al.,* has described Probabilistic Neural Networks [13]. Ioana Sora,*et al.,* have used fuzzy based approach for selection of web service based on QoS value [10].

Table 1.1: Selection of web services based on interaction style: SOAP and REST

Author	**Methodology**
Cesare Pautasso *et al.,* [14]	Made a technical comparison between SOAP and REST, and prescribed included architectural principles, conceptual as well as different technology decisions.
Michael zur Muehlen *et al.,* [15]	Provide a comparison which focuses on the choreography standards of the two interaction styles.
Gomez and*et al.,* [16]	Addressed the general issue of SOAP based web service multimedia conferencing in the IP Multimedia Subsystem .
David Lozano *et al.,* [17]	Developed multimedia conferencing applications using RESTful Web service.
Tekli *et al.,* [18]	The basic idea is to identify common parts of SOAP messages, so that these common parts are processed only once hence, avoiding a large amount of overhead .
Potti *et al.,* [19]	CRUD operation has been used for comparing performance of SOAP and REST. Metrics such as response time and throughput have been used to compare the performance of these Web services .

Table 1.2: Web service Selction based on soft Computing Techniques

Author	**Methodlogy**
Q. H. Al-Masri, E Mahmoud, *et al.,* [6]	Neural network used for best service discovery.
Keskes *et al.,* [7]	Web service selection based on context ontology and quality of services.
G. Ezhilarasi a*et al.,* [9]	Designed a discovery model JIT(Just-in-Time)based on neural network approach .
Ioana Sora *et al.,* [10]	Fuzzy approach used for web service selection.
Abdallah Missaoui1 *et al.,* [11]	Neuro fuzzy approach used for ranking of web services.

1.3 Problem Statement

To select and rank appropriate web services based on interaction styles and quality of service (QoS) attributes.

1.3.1 Web Service Selection

One of the main challenges in service oriented architecture is the optimal selection and ranking of web services. The process of selecting relevant services from a service repository in a heterogeneous environment is a difficult task. Use of different search engines help in selection process by efficiently searching the service repository, peer to peer networks, service portals etc. Fixing up appropriate services is necessary because composition of these services leads to the development of a particular application.

1.3.2 Objective of My Work

Web services need to be selected based on interaction styles i.e., SOAP and REST, because interaction styles play a major role in building and integrating the application.

Web services need to be selected based on QoS attributes. To enhance the qulaity of web service selection, ANN and hybrid ANN are employed for optimal selection of web service.

With the help of the requisite attributes related to quality of service, a comparative study of performance of both the techniques based on error parameter has been made in order to help in critical assessment.

The proposed model contains two soft computing techniques:

1. Artificial Neural Network

- BPN Model
- RBFN Model
- PNN Model

2. Hybrid Artificial Neural Network
   - Neuro-GA approach

1.4 Motivation

Nowdays, business requirements are changing rapidly, but the technology is not changing that quick. So it is important to have business agility. SOA provides agility to meet ever changing requirement to technology.

While building any application, it is a practice that some component can be developed some components from off the self (COTS) can be used, some functionalities may be borrowed from cloud in form of services. So there is a need to integrate all these hetrogeneous component and provide a common interface. So web services need to be selected based on interaction style for easy integration.

SOA provides seamless integration for diverse system with the help of platform independent, interoperable, loosely coupled architecture which is cost-effective.

Before integration, It is important to select the appropriate services. Web service selection based on functional properties alone are not sufficient, Non-functional properties must be considered. If client selects a service which is of poor quality then it can affect the overall performance of system. So in this work QoS parameter is considered for web service selection.

1.5 Thesis Organization

The rest of thesis is organized as follow:

- In Chapter 2, performances of web services identified for an example of Loan Broker Application developed in SOAP and REST and are compared. Since web services operate over the network, throughput and response time are considered as metrics for evaluation.

- In Chapter 3, Artificial neural network (ANN), has been used to design a model for web service selection, to classify a service as based on its QoS value. In this chapter, Back Propagation Network (BPN), Radial Basis Function Network (RBFN) and Probabilistic Neural Network (PNN) have been described and how they have been used in service selection. Also performance of these techniques are compared in order to help in critical assessment.

- In Chapter 4, hybrid approach of artificial neural network, genetics algorithm (GA) coupled with ANN has been used for web service selection.

Chapter 2

Web service selection based on interaction styles: SOAP or REST

2.1 Introduction

Web services are common way to exchange data and information over the network. It is the fundamental unit of application development in service-oriented development. Web services can be built in two different interaction styles such as SOAP and REST, while developing any application, there is a need to select the particular type of services SOAP or REST. To build an application which type of services need to be chosen from the available services is very critical.

Services can interact regardless of different platforms. Services publish its functionalities through WSDL in case of SOAP and through URI in case of REST, while their implementation details need to be kept as private [23]. Once the services are developed it can be integrated with the help of Business Process Execution Language (BPEL) for the SOAP based services. Java script object notation (JSON) can be used for composing REST based services. Building SOAP based services is relatively easier, bacause various platforms offer standard libraries to create

SOAP services or SOAP clients, such as Axis2/Java [24], But the parsing of SOAP messages can be memory and computation intensive. REST uses client-server architecture. REST is not restricted to any particular protocol. REST can have message format in XML as well as JSON. On the other, hand SOAP relies on XML only. XML is verbose in nature so it increases load on server netwok traffic. On the other, hand REST is lightweight in nature.

In the previous chapter, emphasis is given on selection of web services based on interaction styles upon which services are built, but other factor such as quality of service (QoS) attributes are also very important.

In this study, an experiment has been done for web service selection based on interaction style i.e., SOAP and REST. A case study of Loan Broker system is taken as an example. Loan Broker application is built in both SOAP and REST interaction styles. The concept of multi threading has been used for simulating concurrent clients. each user corresponds to one thread.

The performance of each application in different interaction style is analyzed. Throughput and Response time has been considered for performance evaluation parameter as the web services operate over the network.

2.2 Overview of OpenESB And LoanBroker Application

2.2.1 Tool for building SOA application

OpenESB is a commonly applied tool for developing service oriented applications [24]. ESB provides a mature way to integrate services. [24]. ESB supports and helps to realize various SOA principles. With the help of ESB services complex interlinking of services is avoided [25].

2.3 Case Study

In this section standard example of LoanBroker is discussed [26]. LoanBroker example captures the various aspect of service compositions. In this example customer request for best loan rate from the different banks. The Loan Broker BPM(Business Process Model) uses a Business Process Engine to orchestrate the requests [27].

2.3.1 Message sequence in LoanBroker System

Specification of requirement [26].

- Customer requests to LoanBroker application system by giving loan amount as input.
- Loan Broker system communicates with CreditAgency service to get the credit profile of customer.
- CreditAgency service returns the credit profile of customer to LoanBroker system as response.
- Loan Broker system uses credit profile as input to the LenderGateway service to obtain the list of banks.
- LenderGateway service returns bank list to the LoanBroker system based on the credit profile of customer and the amount of the loan requested.
- LoanBroker system requests to all the banks to return the rates.
- Bank services compute and return the loan rate as response to LoanBroker.
- LoanBroker system input different loan rate to the getBestLoanQuote service. which return the best loan rate.
- Best loan rate obtained from the getBestLoanQuote service is returned to the customer as response.

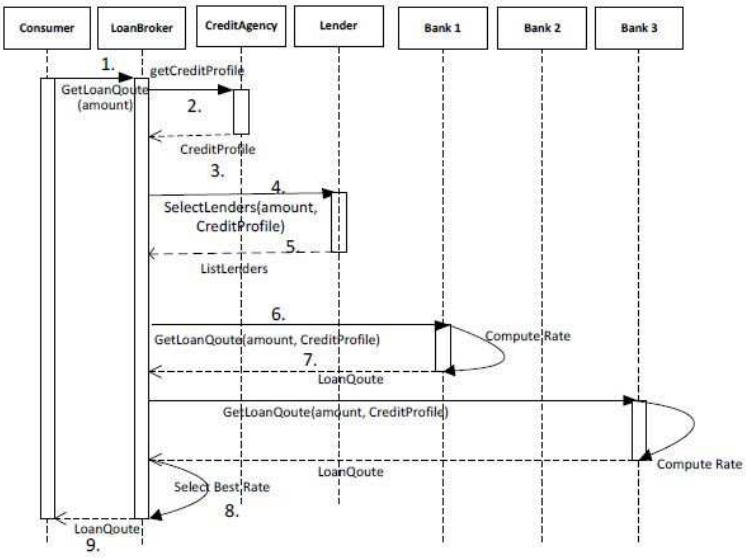

Figure 2.1: Message flow in Loan Broker application

2.3.2 Type of Services Used in building LoanBroker system

LoanBroker application consists of two types of services.

1. Primary services
2. Composite Services

Primary services are the atomic services. Services which can not be further decomposed. Loan Broker application consists of services which is combination of both atomic services as well as composite services. Table 2.1 shows the list of various services involved in it. Simultaneous service requests are managed with the help of multi threading. Total of twenty eight number of simultaneous users are simulated. Throughput and response time is calculated for each user then the average of response time and throughput is taken as final result.

Loan Broker application is built from the integration of different primary services. In SOAP based services, Integration is done with the help of BPEL engine of OpenESB tool. In case of REST, services are integrated by taking the output of one service as input to the other. In SOAP, for service integraion direct support of BPEL module is available and it composes the services with the help of WSDL

Table 2.1: Functionalities of services with requisite input and output

Services	Input	Output
CreditAgency	LoanAmount	CreditProfile
LenderGateway	CreditProfile	List of Banks
*Bank*1	CreditProfile and LoanAmount	Return Rate
*Bank*2	CreditProfile and LoanAmount	Return Rate
*Bank*3	CreditProfile and LoanAmount	Return Rate
BestLoanQuote	rates return from all banks	Return best Rate
LoanBroker	LoanAmount	Return best Rate

address. REST is resource oriented architecture in which services are integrated logically.

2.4 Proposed Work

Web service selection based on appropriate interaction is of prime importance. Web services can be either SOAP based or REST based. In this study, performance comparison of both the interaction styles (SOAP and REST) are provided to select the proper service for development of application. Metrics such as throughput and response time have been considered as performance parameters for evaluation. However the performance of these two techniques are provided to Create, Read, Update, Delete(CRUD) operation [19]. In this study, a comparison between SOAP and REST for enterprise application based on case study is provided.

2.4.1 SOAP Based Web Services

Open ESB tool has been used to find throughout of the development of application. Java platform has been used for developing individual services. In Back-end MySQL database is used. This database stores the information of customer profile such as

available balance in particular customer account. Customer Id serves as primary key. Customer makes a request to loan broker application using unique customer Id and loan amount and gets Best Loan quote as response. LoanBroker application is built in client server architecture. In this work, single machine plays the role of both client as well as server. So the network latency can be considered as zero. After the deployment of services, unique WSDL address is generated. This WSDL address are stored in UDDI registry. This act as a service contract between service provider and consumer.

2.4.2 Integration of SOAP Based Services using BPEL Engine

The Loan Broker application is considered as example of SOA/ESB [26].

It is discussed that how SOAP and REST behave for integration of large application.

The Loan Broker application illustrates a number of capabilities of ESB such as integration, orchestration, messeging and routing. [14]. LoanBroker application is integrated using the BPEL Engine. BPEL consists of two types of activities such as basic activities and structured activities [23]. Basic activities cover receive, reply, wait, switch, while, and sequence. The structured activities determine the structure, of the sequencing of the process, and the basic activities determine what happens in the process [28]. BPEL process takes the WSDL address and based on address services are accessed in the order in which they are integrated. The BPEL module shown in Figure 2.2 depicts integration of all the services. This application receives a series of input and produces desired output. Composite application is developed using the BPEL Engine and CASA(Composite Application Service Assembly). This contains SOAP messages and input output port as shown in Figure 4.

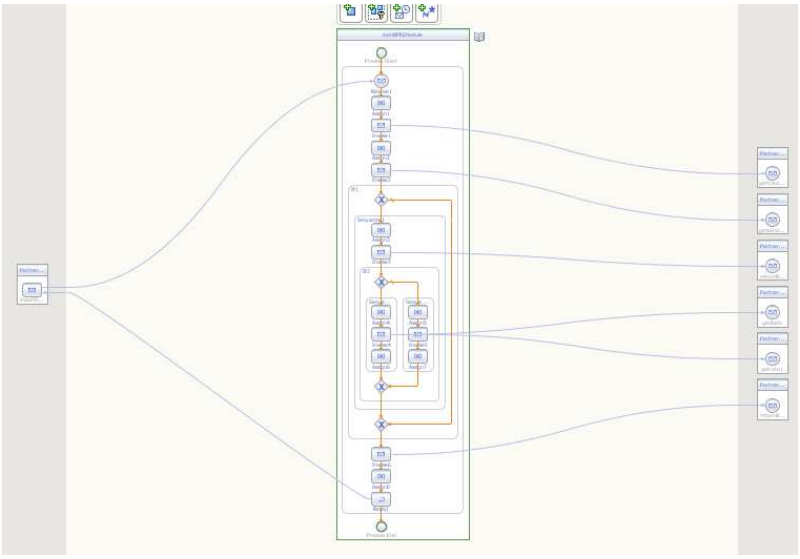

Figure 2.2: BPEL module of LoanBroker application

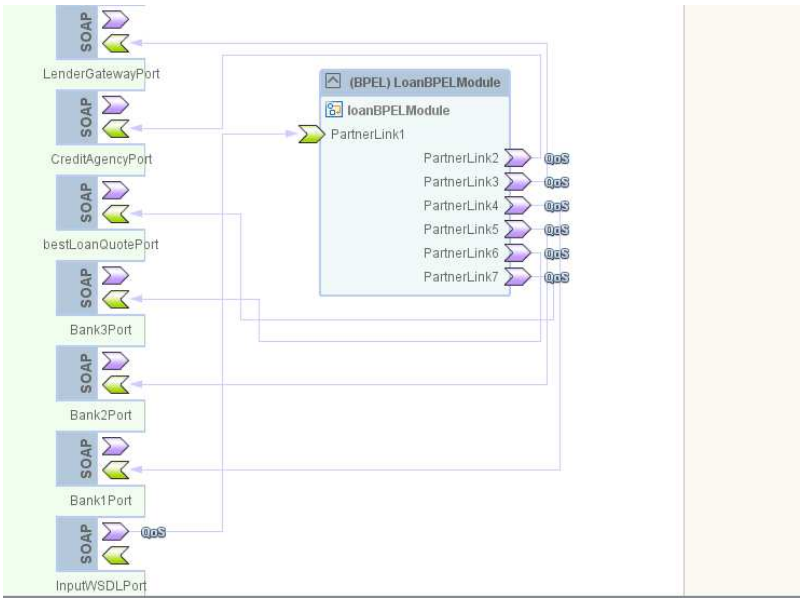

Figure 2.3: Composite application Service Assembly

2.4.3 REST based web services.

RESTful service development considers services, operations and parameter in term of resources. Each of these resources are assigned with a unique URI (uniform resource indicator). All these resources are accessed with the help of URI(uniform resource indicator). The main reason behind developing REST based services are:

- Uniform interface
- Addressability
- Statelessness
- Connectedness

In RESTful web services, these properties are embodied in resources, URIs, and representation [29]. REST based web services is developed with the help of OPEN ESB tool. It uses the same database that is used by the SOAP.

2.4.4 Integration of REST based Services

REST-based services, unlike SOAP, does not generate WSDL address. It produces WADL address. But this WADL address is not supported by the BPEL engine. So in this study, Services are integrated with JSP. The input of one service is passed as output to the other service. However, many tools are available to integrate REST based services such as MULE ESB. But the result obtained using this tool for REST can not be compared with the outcome obtained by OpenESB for SOAP style. So OPEN ESB tool is used in the development of both types of services for uniform comparison.

2.5 Performance Metrics.

Throughput and response time are measured against two parameters.

- File size
- Number of simultaneous request

Different file sizes ranging from 1000kb to 7000 kb are taken into consideration. These files are stored in the system and retrieved with the help of services. Each file has a unique id. Total of 28 concurrent clients are simulated using the concept of multithreading. For each request sent to web service, response time and throughput is evaluated. In the other analysis, response time and throughput are analysed with the help of number of service request. As the number of requests go on increasing what will be the effect on throughput and response time are found out.

2.5.1 Response Time

Response Time is defined as the time duration of the request sent and response received. When the client application sends a request to service with appropriate input, the server takes information to process the data and pass through a series of services, because an application is integrated using all these services. After the request is processed, the resulted output is sent as a reply to the client. Maximum simultaneous request that can be be sent to the server is considered as twenty-eight. Response Time is calculated as different file size and different number of service request. Response time for different file size is calculated as the total time elapsed in retrieving a file using the equation as:

$$R_t = t_1 - t_2 \tag{2.1}$$

where R_t = Response Time
t_1 = Time at which client sent Request.
t_2= Time at which client receives Response.

2.5.2 Throughput

Throughput is defined either in terms of number of the requests or terms of average data bytes per second. Throughput is defined as number of request per second. In this study, we have considered the twenty-eight number of requests. Throughput is calculated using the total no of requests divided by total time taken to process all request.

Throughput = Number of simultaneous request sent by client/Total time taken to process all the requests. Throughput can have other measurement units in terms of number of bytes per second. Suppose 'X' bytes of data transfer take 'Y' unit of time, throughput can be calculated as number o bytes sent or received in unit time. It can be computed using the following equation.

$$T = X/Y \tag{2.2}$$

Where T = Throughput
X = Data Byte transfered.
Y= Time Time taken to transfer data bytes.
To calculate throughput, files of different size ranging from 1000kb to 7000kb has been stored on disk. These files are retrieved from the disk and total time elapsed is calculated. Throughput is calculated by dividing the file of different size by the total time elapsed.

After finding the value of Response time and Throughput of SOAP and REST based web services, a comparison is made between these two techniques to have the critical assessment on the performance of both the styles.

2.6 Result Analysis

The proposed work in this paper compares SOAP and REST styles for Enterprise Application. In this work, seven number of services namely CreditAgency, LenderGate-

way, Bank1, Bank2, Bank3, getBestLoanquote, and LoanBroker Service(Composite service) are implemented. Services are compared using throughput and response time for integrated application. The main idea is to implement them in SOAP and REST paradigm separately and then compare their throughput and response time respectively.

2.6.1 Throughput comparison of SOAP and REST

The figure shown in Figure 2.4 provides the throughput comparison of SOAP and REST for composite services. i.e., for Loan Broker Application. From the Figure 2.4, it is evident that REST based application has better throughput than SOAP based application. As the number of the client increases SOAP throughput increases at a slow pace as compared to REST. Because SOAP is heavyweight, and its XML presentation is verbose. On the other hand, REST is lightweight. As the file size increases, throughput increases for both SOAP and REST but REST has better throughput than SOAP, i.e., REST transfer more number of bytes in unit time.

In Figure 2.5, comparison between SOAP and REST is provided for varying number of requests. The number of request increases SOAP throughput decreases.

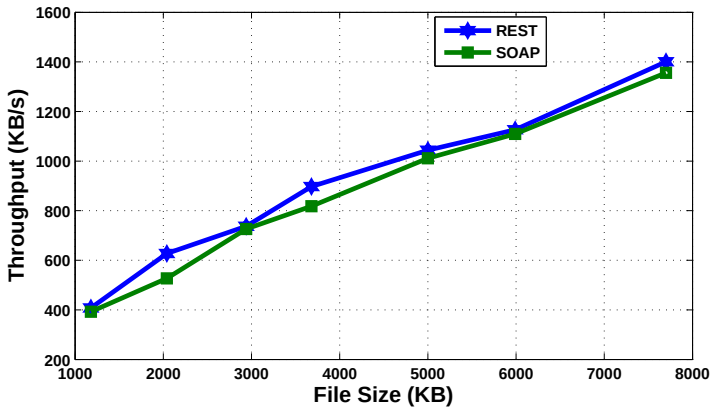

Figure 2.4: Throughput Comparison for varying number of requests

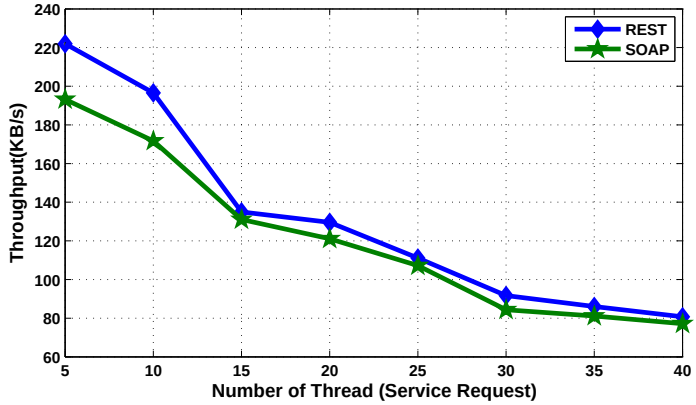

Figure 2.5: Throughput for different file size

2.6.2 ResponseTime comparison of SOAP and REST for EAI

The Figure shown in Figure 2.6 below provides the ResponseTime comparison of SOAP and REST for composite service. i.e., for Loan Broker Application. ResponseTime is measured in ms (milliseconds). For the response time, as the number of clients are increased in SOAP, response time increases but the rise in number of clients does not affect the response time of REST services drastically. REST has better response time than SOAP.

In the Figure 2.6 comparative analysis for response time is provided. Response

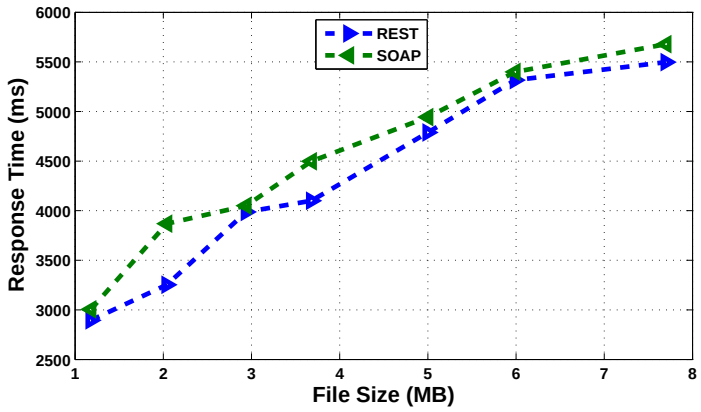

Figure 2.6: Response time for varying number of requests

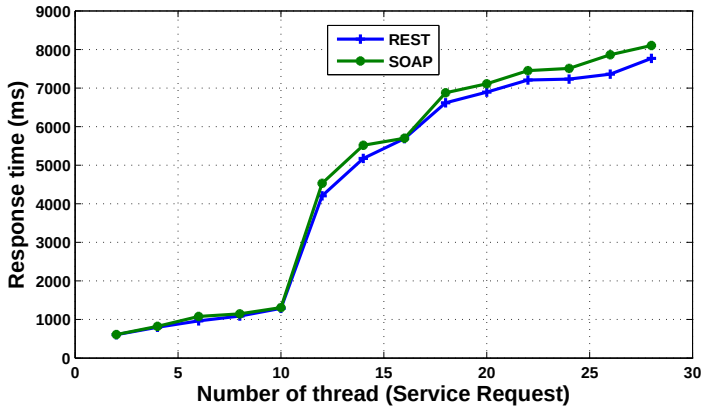

Figure 2.7: Respopnse time for different file size

time is calculated for file size ranging from 1000kb to 7000kb in milliseconds.

For SOAP, response time is initially greater as compared to REST but as the file size increases for a file of size 3 MB their response time are almost equal. For other cases, SOAP has more considerable response time than REST.

In Figure 2.7, as the number of service requests grow SOAP response time decreases more rapidly than REST. It means that REST can handle large number of request inamore efficient way than SOAP

2.7 Summary

This study provides a comparative analysis of the SOAP and REST styles For selection of web services. It is crucial to select a appropriateweb services based on its interaction style as it affects the underlying implementation and the integration of enterprise application.

The central issues in SOA are how to integrate the application that are distributed over the internet. So, selecting a appropriate interaction style can ease the integration and the application can be accessed from the network fast and efficiently.

In this study, LoanBroker application is considered for the case study. LoanBroker application is suitable for evaluating the integration of various services as the message flow between each of the services can be clearly seen.

From the experimental analysis, it is observed that REST based development has better throughput and response time that of the system developed, based on SOAP. However, it is convenient to build the application with SOAP due the adequate support of tools. But in case of REST, it 's hard to find right kind of tools. REST is an emerging technology whereas SOAP is already popular and has been already used in building good number of applications. Hence, it may be opined that use of REST style has definite advantage over the use of SOAP style.

Chapter 3

ANN Techniques for Web Service Selection

3.1 Introduction

Web Service selection is one of the important aspects of SOA. It helps to integrate the services to build a particular application. It is important to select the services which matches requirement as well as has good quality. Web services are chosen based on the quality of service attributes. ANN techniques such as Backpropagation Network (BPN), Radial Basis Function Network (RBFN) and Probabilistic Neural Network (PNN) has been used to enhance the quality of selection. In this study, nine QoS attributes are taken as input to the neural network for service classification. After the network is trained, testing data is used for testing the system. The error parameter is calculated by taking the difference between actual and predicted output. The accuracy obtained from each method compared and finally, it is concluded that BPN outperforms the other two techniques. The obtained value of accuracy of PNN is moderate, and that of RBFN has least accuracy.

3.2 Research background

The following subsection highlights about the dataset being used for web service selection and Data Normalization.

3.2.1 About the Dataset

The QWS Dataset consist of 365 number of web services. Each web service has nine QoS (Quality of Service) attributes as shown in TABLE 3.1. This Dataset has dimension of $365X9$.

3.2.2 Data Normalization

There are nine QoS attributes which lie in varying range, So it can not be directly taken as input to the ANN and Fuzzy logic. It must be normalized in the range of 0 to 1 for achieving the better accuracy.

Data is processed to transform into the linear form using min max formula. Each of the actual data d of attribute p is mapped to a normalized value d' which lies in the range of 0 to 1. The Min-Max normalization is calculated by using the equation:

$$Normalized(d) = d' = \frac{d - min(p)}{max(p) - min(p)} \tag{3.1}$$

3.3 Methodology

Web service description and discovery is realized using UDDI. As the number of services grow day by day searching web service efficiently according to requirement poses a significant challenge. To define the quality of service nine QoS parameters are described in the Table 3.1 are considered.

Based on these nine QoS parameters service classification is carried out separately by each ANN algorithm [6] .

Table 3.1: Quality of Service Attributes and Units

ID	QoS attribute	Specification	Units
1	Response time (RT)	The time between making a request and getting a response	ms
2	Availability (AV)	Number of successful requests/total number of requests	%
3	Throughput (TP)	Number of invocations per unit time	Invokes/second
4	Success ability (SA)	Number of response / number of request messages	%
5	Reliability (RB)	Ratio of the number of error messages to total messages	%
6	Compliance (CP)	The extent to which a WSDL document follows	%
7	Best Practices (BPT)	The extent to which a web service follows	%
8	Latency (LC)	Time taken for the server to process a given request	Ms
9	Documentation (Doc)	Measure of documentation (i.e. description tags) in WSDL	%

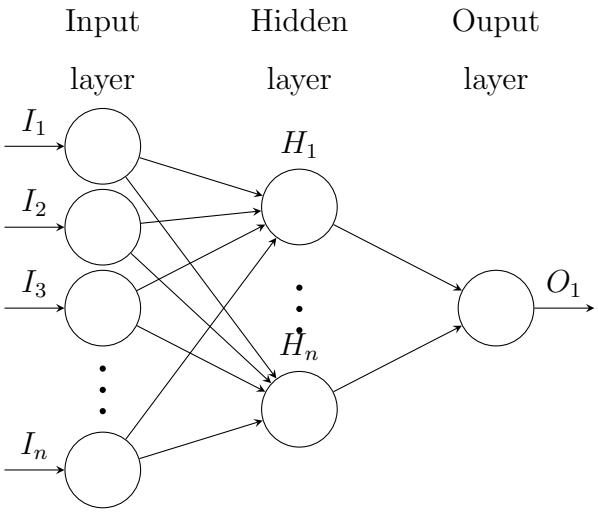

Figure 3.1: Artificial Neural Network.

3.4 Proposed work for Web Service selection

The following subsections highlight on the various ANN algotithm used for web service selection.

Back Propagation Network

The structure of Artificial Neural Network is shown in the Figure 3.1. BackPropagation network is a multilayer feed forward network having three layers of input layer, hidden layer, and output layer. Network is trained using supervised learning algorithm.

During the training period of network, input patterns are passed through the network, and small random weights are assigned to the connection link between different layers. Output of the input layer is fed to input layer. It is given by:

$$O_i = I_i \tag{3.2}$$

Net input I_{inh} to the hidden layer is calculated as:

$$I_{inh} = \sum_{i=1}^{n} I_i W_{ih} \tag{3.3}$$

In this paper, sigmoidal function is used as activation function at hidden layer and output layer. Output at hidden layer O_h for the net input I_{ih} is given by:

$$O_h = \frac{1}{1 + e^{-I_{inh}}} \tag{3.4}$$

The net input O_{ink} to the output layer O_k is given by:

$$O_{ink} = \sum_{h=1}^{P} O_h V_{hk} \tag{3.5}$$

and output is given by:

$$O_k = \frac{1}{1 + e^{-O_{ink}}} \tag{3.6}$$

The main aim of applying ANN is to reduce the MSE (Mean Square Error) which is acheived by small random change in weight for each iteration. MSE is calculated as follows:

$$MSE = \frac{1}{n} \sum_{k=1}^{n} (O'_k - O_k)^2 \tag{3.7}$$

Where O_k = actual output and O'_k = expected output. After error calculation, weight is updated. In this study, gradient descent method is used for weight updation. Gradient Descent compute the first order derivative of error function to minimize the error.

$$E_k = \frac{1}{2}(T_k - O_k)^2 \tag{3.8}$$

$$V' = \frac{\partial d}{\partial dW} \frac{1}{2}(T_k - O_k)^2 \tag{3.9}$$

where V' is the gradient vector. The small random change in the weight vector can be given as:

$$W_{k+1} = W_k - \alpha V'_k \tag{3.10}$$

where W_{k+1} = updated weight, W_k = current weight , α = learning parameter and V'_k= Gradient vector.

Radial Basis Function Network (RBFN)

In Radial Basis Function (RBFN) neural network, network architecture of Radial Basis Function (RBF) consists of three layers namely input layer, hidden layer and output layer. Each hidden layer unit is represented as radial center $C_1, C_2, C_3, C_4 ....., C_h$. as shown in Figure 3.2. Number of hidden layers in RBF is restricted to one. In this paper RBFN is used for solving classification problem.

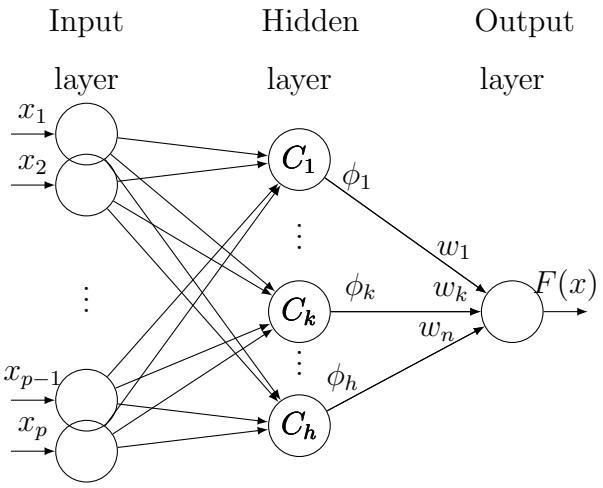

Figure 3.2: RBFN Model.

In this network, both weight as well as center are learned. There are various techniques for updating weight and centers.

- Pseudo Inverse Technique (offline).
- Gradient Descent Learning (online).
- Hybrid Learning (online).

In this paper, Pseudo Inverse Technique for updation of weight and center has been used. Euclidian distance between the each input x_i and the hidden center c_j is calculated and then radial basis function is applied to the result in order to obtain activation function. It is obtained by:

$$\phi_j = \phi(\parallel x - c_j \parallel)$$

where $\parallel x - c_j \parallel$ is euclidean distance. The output is obtained by taking sum of the product of ϕ_j and w_j.

$$\phi_j = \sum_{j=1}^{h} \phi_j w_j \tag{3.11}$$

w_j: weight of jth center. $\phi(z) = e^{-z^2/2\sigma^2}$ where $\phi(z)$: Guassian Radial Basis Function

where $z = \parallel x - c_j \parallel$ and $d = \frac{\sigma}{\sqrt{2h}}$. Gaussian Radial Basis function is most popular Radial Basis Function which is applied in this paper.

For the weight and center updation Pseudo Inverse Technique is used. In pseudo inverse tehnique width of the radial basis function is chosen randomly. For every x, ϕ_i should be normalized such that:

$$\sum_{j=1} \phi_j = 1 \tag{3.12}$$

weight vector is computed as:

$\Phi = [\phi_1, \phi_2, \phi_3, ..., \phi_h]$

W= $[w_1, w_2, ..., w_h]^T$

ΦW= y^d, y^d is the desired output

W = $(\Phi^T\Phi)^{-1}\Phi^T y^d = \Phi' y^d$

$\Phi' = (\Phi^T\Phi)^{-1}\Phi^T$ is the pseudo inverse of Φ this is possible only when $(\Phi^T\Phi)$ is non singular. If this is singular, then singular value decomposition is used. after the appropriate choice of weight is obtained desired output is calculated.

Probablistic Neural Network (PNN)

Probablistic neural network(PNN) is inherently a classifier. PNN is based on the Bayesian network and principle of statistical algorithm. Network Architecture of probablistic neural netwok is shown in the Figure 3.3. It consist of four layers. input layer, Hidden layer, summation layer or pattern layer, and output layer [13].

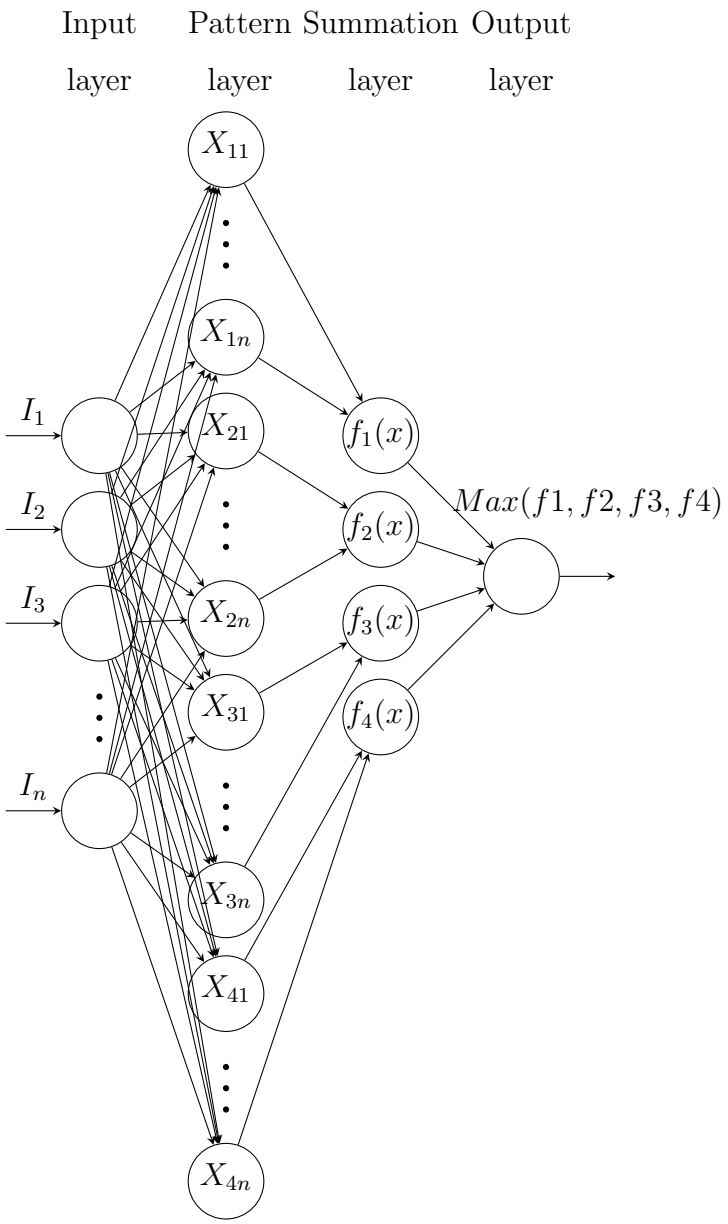

Figure 3.3: PNN Model.

The euclidian distance between input and hidden unit is calculated and activation function is applied to the result.

$$z = \| x - h_i \|$$

Summation layer sums the output obtained from each hidden unit and finally the

result obtained from each summation unit is compared. It is based on the winner-takes-all approch. The summation unit which is having the highest value, input is classified to that class. PNN is one of the fastest network, where σ = smoothing parameter. Smoothing parameter has great impact on the accuracy of classification achieved by the network.

3.5 Performance Evaluation Parameters

There are four performance evaluation parameter.

- Mean absolute error (MAE)
- Mean absolute relative error (MARE)
- Root mean square error (RMSE)
- Standard error of the mean (SEM)

3.5.1 Mean Absolute Error (MAE)

Mean Absolute Error is used to compare the actual value and predicted value i.e, how much predicted output is deviated from actual.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| \tag{3.13}$$

3.5.2 Mean Absolute Relative Error (MARE)

$$MARE = \frac{1}{n}\sum_{i=1}^{n}\frac{|f_i - y_i'|}{f_i} \tag{3.14}$$

It creates numerical overflow (Divided by zero), so that 0.05 is added in the denominator in order to avoid numerical overflow.

$$MARE = \frac{1}{n}\sum_{i=1}^{n}\frac{|f_i - y_i'|}{y_i + 0.05} \tag{3.15}$$

3.5.3 Root Mean Square Error (RMSE)

RMSE is a calculated by taking difference of actual and predicted value. it mesures the diffrence between predicted and actual output.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f_i - y_i')^2} \tag{3.16}$$

3.5.4 Standard Error of mean (SEM)

The standard error of the mean is denoted as M. It is the ratio of standard deviation to number of sample:

$$\sigma_M = \frac{\sigma}{\sqrt{N}} \tag{3.17}$$

where σ = standard deviation N = Number of sample.

3.6 Results and Analysis

3.6.1 Application of Artificial Neural Network

ANN algorithms like BPN, RBFN and PNN used in this study considers nine QoS attribute as input. These nine inputs correspond to input layer nodes in each ANN algorithm. However, number of hidden layer differs from each network.

In BPN, number of nodes present in the hidden layer must be greater than input layer nodes, so the number of hidden nodes are fifteen in BPN.

In RBFN, hidden layer consists of four radial centers. Since there are four different classes. The Euclidian distance of a web service input vector (nine QoS attribute) from these radial center is calculated.

In PNN, hidden layer consist of 4 hidden unit each hidden unit corresponds to each class. In the summation layer sum of the output from each unit is calculated. Then at the output layer maximum of all the output from summation layer is taken.

Table 3.2: Performance of Different Algorithm

Performance Evaluation Parameter	BPN	RBFN	PNN
MAE	0.0182	0.204	0.144
MARE	0.0795	0.56	0.234
RMSE	0.0779	0.26	0.164
SEM	0.0234	0.06	0.04
Accuracy	98%	80%	86%

Maximum value corresponds to output class to which a particular web service belong. Calculated value is compared with the actual value for the performance evaluation.

Python programming language is used for performance evaluation of ANN. Finally, comparison of the different technique are given in Table 3.2. From the Table 3.2 it can be inferred that BPN outperforms other methods.

3.7 Summary

In this paper methodology has been proposed for optimal web service selection using various ANN algorithm such as BPN, RBFN, PNN algorithms are used for the classification of web services. Among these neural networks accuracy obtained by the BPN is best. PNN accuracy is moderate, and RBFN accuracy is less than PNN. The problem with this approach is as the no of attribute increases, service classification becomes more complex.

Chapter 4

Hybrid ANN approch for Web Service Selection

4.1 Introduction

In this chapter, Neuro-GA approach has been applied for web service selection. The appropriate weight for ANN is generated with the help of genetic algorithm. Neural Networks can recognize only those pattern that network have already learned. It can not recognize the new patterns. The network needs to be trained sufficiently to extract and determine general features which applies to both training and testing data. However overtraining of neural networks lead to the undesired result.

BPN uses gradient descent method to minimize the error and update the weights, so there is a probability to stuck in a local minimum [21].

Genetic algorithm (GA) is an optimization technique applied to find exact or approximate solution. The solution generated by GA may or may not be the best solution. Genetic algorithm intensively used to search the solution space. GA requires population size, selection rate, initial weight range, number of training epochs to find the solution of problem [22]. In neuro GA approach, GA is used to generate weight, which is fed to the neural network. GA is continued to find the weight set

until the fitness values for 95% of the chromosomes are same. Chromosome is the collection of genes (weights) [21].

4.2 Proposed work for Web Service Selection

The following sub-sections highlight on the classification methods used for Web Service selection.

4.2.1 Genetic Algorithm

In this approach, genetic algorithm is used for weight determination for neural network. Neural network having 'x' number of input nodes, 'y' number of hidden nodes, and 'z' number of output nodes. Output of the input layer is fed to the input. It is given by the fllowing equation.

$$O_{in} = I_{in} \tag{4.1}$$

Sigmoidal function is used as activation function at hidden and output layer. The output of hidden layer 'O'_h for the net input 'I'_{inh} can be given as:

$$O_h = \frac{1}{1 + e^{-I_{inh}}} \tag{4.2}$$

The output of output layer 'O'_o for the net input of the output layer 'O'_{ink} is represented as:

$$O_o = \frac{1}{1 + e^{-o_{ink}}} \tag{4.3}$$

The number of weight that is to be determined for neural network can be calculated as:

$$N = (x + z) * y \tag{4.4}$$

Each weight is equivalent to gene which is randomly generated. each gene being a real number. Each gene is coded as binary digit and let the number of digits in weights is l. The length of the chromosome S can be computed as:

$$S = N * l = (x + y) * y * l \tag{4.5}$$

Fitness value for each chromosome is determined after weights are extracted from each chromosome using the following equation: The weight equivalent to each gene can be evaluated using following equation.

$$W_j = \begin{cases} -\frac{t_{jl+2}*10^{l-2}+t_{jl+3}*10^{l-3}+....+t_{(l+1)l}}{10^{l-2}} & \text{if } 0 <= t_{jl+1} < 5 \\ +\frac{t_{jl+2}*10^{j-2}+t_{jl+3}*10^{l-3}+....+t_{(l+1)l}}{10^{l-2}} & \text{if } 5 <= t_{jl+1} <= 9 \end{cases} \tag{4.6}$$

The fitness value for each chromosome is derived with the help of fitgen algorithm: which can be given as [21].

Algorithm to compute fitness function: FITGEN()

Input: $\bar{I}_i = (I_{1i}, I_{2i}, I_{3i}, ..., I_{li})$

Output: $\bar{T}_i = (T_{1i}, T_{2i}, T_{3i}, ..., T_{ni})$

where $\bar{I}_i$, $\bar{T}_i$ represent the input-output pairs of network.

Step 1: Weights $\bar{W}_j$ from Ch_i are calculated using equation 4.6.

Step 2: Considering $\bar{W}_j$ as a constant weight, the network is trained for N input instances and the estimate value O_{in} is found.

Step 3: For each input instance k error E_K can be evaluated using the follwing equation:

$$E_k = (T_{ki} - O_{ki})^2 \tag{4.7}$$

Step 4: Root mean square error (RMSE) for the chromosome Ch_i is given by the following equation:

$$E_i = \sqrt{\frac{\sum_{k=1}^{k=N} E_k}{N}} \tag{4.8}$$

where N = training data .

Step 5: Fitness value for chromosome C_i using the following equation is found out as:

$$\boxed{F_i = \frac{1}{E_i} = \frac{1}{\sqrt{\frac{\sum_{k=1}^{k=N} E_k}{N}}}} \tag{4.9}$$

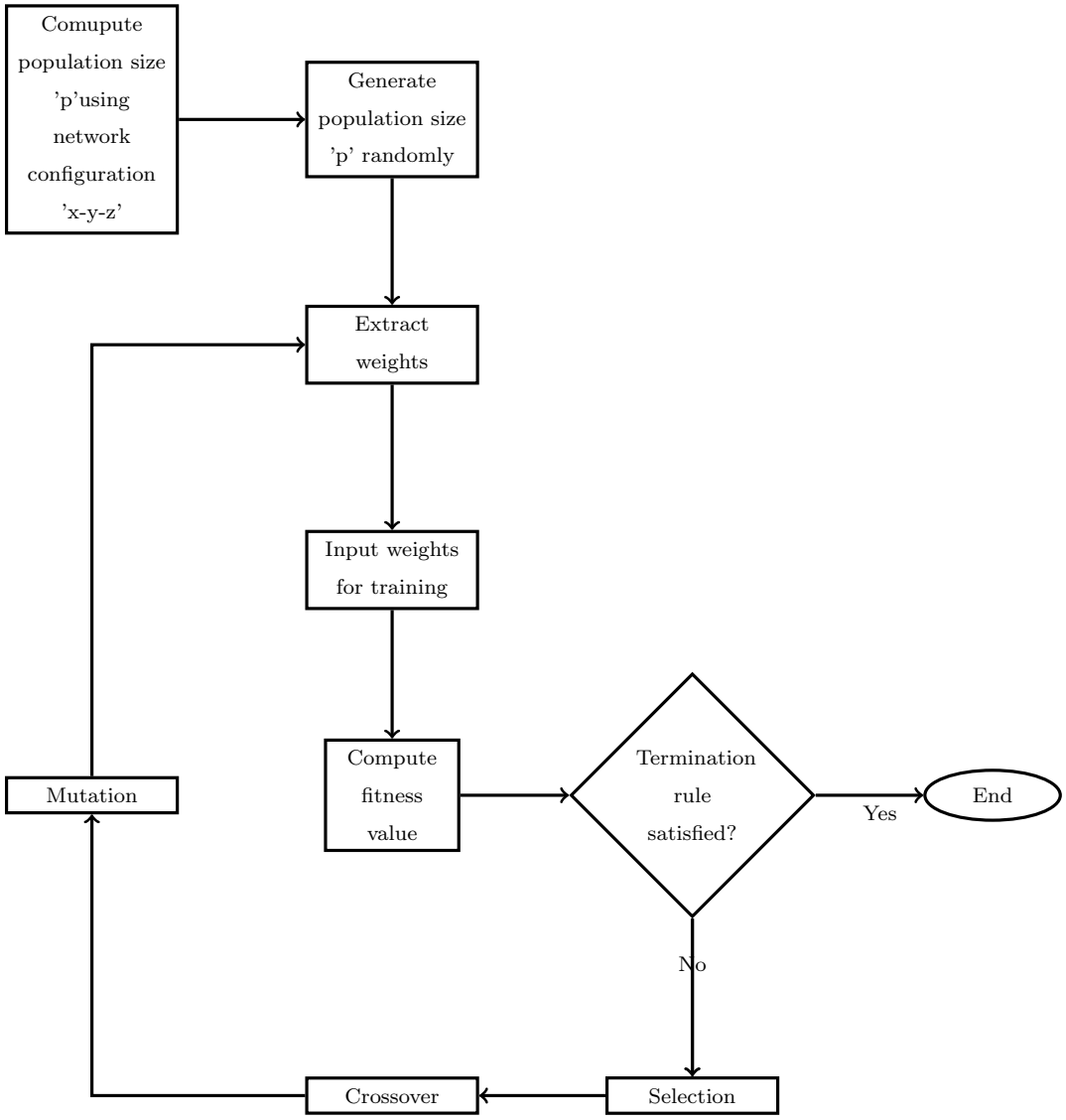

Figure 4.1: Flow chart for Neuro-GA Model

Figure 4.1 shows the block diagram for Neuro-GA approach, which represent the steps followed to design the model.

4.3 Result Analysis

In this proposed approach, a total population size of 100 is taken. The procedure converges when the fitness value of 90 % of chromosomes are same. The Neuro GA

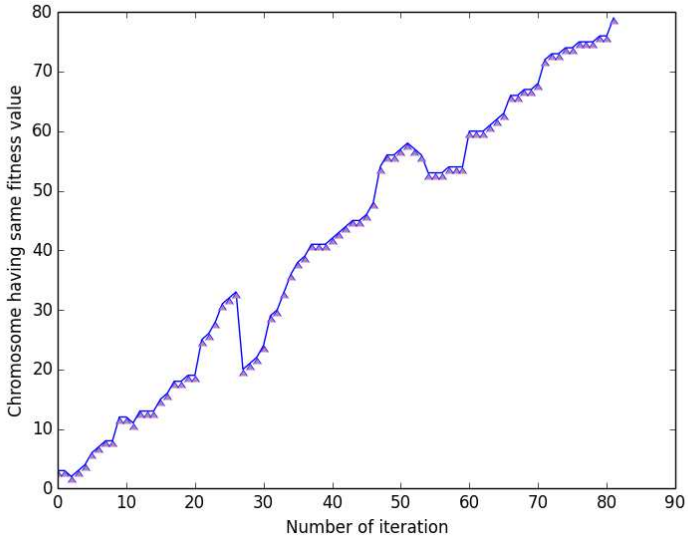

Figure 4.2: Fitness value Versus number of iteration

approach makes 300 number of iteration to have the chromosomes having similar fitness value. The result is shown in Figure 4.2 shows the plot between number of chromosome having same fitness value versus number of iteration. It can be inferred from the Figure 4.2 that as the number of iterations increase number of chromosomes having same fitness value is rising accordingly.

The weight set extracted when fitness value reaches 90% percent is used as a final weight. Testing data is used to test the network, and finally MARE is calculated to obtain the accuracy of the system. The error parameters and accuracy of neuro GA is given in the Table 4.1.

Table 4.1: Performance of Neuro GA Algorithm

Error	MAE	MARE	RMSE	SEM	Accuracy
Neuro-GA	0.063	0.089	0.085	0.037	94%

4.4 Summary

In this chapter Hybrid, ANN approach has been used for web service selection. In this approach, ANN has been used to train the network. But in case of ANN precision of output is often limited it does not admit zero error but only minimization of least square error. GA algorithm is used to determine weight set for neural network that leads to minimization of error.

Chapter 5

Conclusion

From the analysis, it is observed that REST based services have better throughput and response time than that of the system developed, based on SOAP.

However it is convenient to build the application with SOAP due the adequate support of tools. But in case of REST right kind of tools are not available. REST is an emerging technology, whereas SOAP is already popular and used in building good number of applications. Hence, it may be opined that the use of REST style has a definite advantage over the use of SOAP style.

In this study, a method has been proposed for optimal web service selection using various ANN algorithm such as BPN, RBFN, PNN. Back Propagation Network, Radial Basis Function and Probabilistic Neural Network are used for the classification of web services. Among these neural network accuracy obtained by the BPN is best. PNN accuracy is moderate, and RBFN accuracy is less than PNN. The problem with this approach is as the no of attribute increases; service classification becomes more complicated. In the next approach, GA is used for weight determination for neural network.

In this approach, ANN has been used to train the network. But in case of ANN precision of output is often limited it does not admit zero error but only minimization of least square error. GA algorithm is used to determine weight set

for neural network that leads to minimization of error.

Bibliography

[1] "Artificial neuralnetwork." `http://en.wikipedia.org/wiki`, May 2014.

[2] "Wsdl." `http://www.w3schools.com/webservices/ws_wsdl_intro.asp`, August 2014.

[3] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana, *et al.*, "Web services description language (wsdl) 1.1," 2001.

[4] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, and D. Winer, "Simple object access protocol (soap) 1.1," 2000.

[5] L.-J. Zhang, T. Chao, H. Chang, and J.-Y. Chung, "Xml-based advanced uddi search mechanism for b2b integration," *Electronic Commerce Research*, vol. 3, no. 1-2, pp. 25–42, 2003.

[6] E. Al-Masri and Q. H. Mahmoud, "Discovering the best web service: A neural network-based solution," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pp. 4250–4255, IEEE, 2009.

[7] N. Keskes, A. Lehireche, and A. Rahmoun, "Web services selection based on context ontology and quality of services.," *Int. Arab J. e-Technol.*, vol. 1, no. 3, pp. 98–105, 2010.

[8] V. X. Tran and H. Tsuji, "Qos based ranking for web services: Fuzzy approaches," in *Next Generation Web Services Practices, 2008. NWESP'08. 4th International Conference on*, pp. 77–82, Ieee, 2008.

[9] G. Ezhilarasi and P. Dhavachelvan, "Effective web service discovery model using neural network approach," *International Journal of Computer Theory and Engineering*, vol. 2, no. 5, pp. 60–64, 2010.

[10] I. Sora, G. Lazar, and S. Lung, "Mapping a fuzzy logic approach for qos-aware service selection on current web service standards," in *Computational Cybernetics and Technical Informatics (ICCC-CONTI), 2010 International Joint Conference on*, pp. 553–558, IEEE, 2010.

[11] A. Missaoui and K. Barkaoui, "A neuro-fuzzy model for qos based selection of web service," *Journal of Software Engineering and Applications*, vol. 3, no. 06, p. 588, 2010.

[12] Y. Suresh, L. Kumar, and S. K. Rath, "Statistical and machine learning methods for software fault prediction using ck metric suite: A comparative analysis," *International Scholarly Research Notices*, vol. 2014, 2014.

[13] D. F. Specht, "Probabilistic neural networks," *Neural networks*, vol. 3, no. 1, pp. 109–118, 1990.

[14] C. Pautasso and E. Wilde, "Why is the web loosely coupled?: a multi-faceted metric for service design," in *Proceedings of the 18th international conference on World wide web*, pp. 911–920, ACM, 2009.

[15] M. Zur Muehlen, J. V. Nickerson, and K. D. Swenson, "Developing web services choreography standardsthe case of rest vs. soap," *Decision Support Systems*, vol. 40, no. 1, pp. 9–29, 2005.

[16] M. Gómez and T. P. de Miguel, "Advanced ims multipoint conference management using web services," *Communications Magazine, IEEE*, vol. 45, no. 7, pp. 51–57, 2007.

[17] D. Lozano, L. A. Galindo, and L. García, "Wims 2.0: Converging ims and web 2.0. designing rest apis for the exposure of session-based ims capabilities," in *Next Generation Mobile Applications, Services and Technologies, 2008. NGMAST'08. The Second International Conference on*, pp. 18–24, IEEE, 2008.

[18] J. M. Tekli, E. Damiani, R. Chbeir, and G. Gianini, "Soap processing performance and enhancement," *Services Computing, IEEE Transactions on*, vol. 5, no. 3, pp. 387–403, 2012.

[19] P. K. Potti, S. Ahuja, K. Umapathy, and Z. Prodanoff, "Comparing performance of web service interaction styles: Soap vs. rest," in *Proceedings of the Conference on Information Systems Applied Research ISSN*, vol. 2167, p. 1508, 2012.

[20] S. Mumbaikar, P. Padiya, *et al.*, "Web services based on soap and rest principles," *International Journal of Scientific and Research Publications*, vol. 3, no. 5, 2013.

[21] S. Rajasekaran and G. V. Pai, *NEURAL NETWORKS, FUZZY LOGIC AND GENETIC ALGORITHM: SYNTHESIS AND APPLICATIONS (WITH CD)*. PHI Learning Pvt. Ltd., 2003.

[22] Y. Lee and C.-H. Wei, "A computerized feature selection method using genetic algorithms to forecast freeway accident duration times," *Computer-Aided Civil and Infrastructure Engineering*, vol. 25, no. 2, pp. 132–148, 2010.

[23] J. Gortmaker, M. Janssen, and R. W. Wagenaar, "The advantages of web service orchestration in perspective," in *Proceedings of the 6th international conference on Electronic commerce*, pp. 506–515, ACM, 2004.

[24] "Open enterprise service bus." `http://www.open-esb.net`, August 2014.

[25] C. Groba, I. Braun, T. Springer, and M. Wollschlaeger, "A service-oriented approach for increasing flexibility in manufacturing," in *Factory Communication Systems, 2008. WFCS 2008. IEEE International Workshop on*, pp. 415–422, IEEE, 2008.

[26] B. W. Gregor Hohpe, *Enterprise Integration Patterns*. Addison Wesley, 2003.

[27] P. Brebner, "Service-oriented performance modeling the mule enterprise service bus (esb) loan broker application," in *Software Engineering and Advanced Applications, 2009. SEAA'09. 35th Euromicro Conference on*, pp. 404–411, IEEE, 2009.

[28] A. Karande, M. Karande, and B. Meshram, "Choreography and orchestration using business process execution language for soa with web services," *International Journal of Computer Science Issues IJCSI*, vol. 11, pp. 224–232, 2011.

[29] L. Richardson and S. Ruby, *RESTful web services*. 2008.

Dissemination

Conference

1. Naik, Debendra Kumar, Smita Kumari, and Santanu Kumar Rath. "Application of Soft Computing Technique for Web Service Selection." Distributed Computing and Internet Technology. Springer International Publishing, 2015. 245-248.

2. Smita Kumari, and Santanu Kumar Rath , Comparing Performance of SOAP and REST based Web Services for Enterprise Application , *International Conference on Advances in Computing, Communications and Informatics (ICACCI)* , May 2015(Communicated).