Neural Network Solution For Service Level Agreement

Sarmad Al-Aloussi,Researcher
AABFS
Amman-Jordan

***Abstract*— Service Oriented Computing is playing an important role in sharing the industry and the way business is conducted and services are delivered and managed. This paradigm is expected to have major impact on service economy; the service sector includes health services, financial services, government services, etc. This involves significant interaction between clients and service providers[1]. This paper is pointed in addressing the problem of enabling Service Level Agreement (SLA) oriented resources allocation in data centers to satisfy competing applications demand for computing services. A QoS report designed to compare performance variables to QoS parameters and indicate when a threshold has been crossed. This paper was suggested a methodology which helps in SLA evaluation and comparison. The methodology was found on the adoption of policies both for service behavior and SLA description and on the definition of a metric function for evaluation and comparison of policies. In addition, this paper contributes a new philosophy to evaluate the agreements between user and service provider by monitoring the measurable and immeasurable qualities to extract the decision by using artificial neural networks (ANN).**

Keywords*— *Service Oriented Architecture; Service Level Agreements; QoS; and Neural Network.

I. Introduction

Service-level agreements are, by their nature, "output" based the result of the service as received by the User is the subject of the "agreement." The (expert) service provider can demonstrate their value by organizing themselves with ingenuity, capability, and knowledge to deliver the service required, perhaps in an innovative way. Organizations can also specify the way the service is to be delivered, through a specification (a service-level specification) and using subordinate "objectives" other than those related to the level of service.

This type of agreement is known as an "input" SLA. This latter type of requirement is becoming obsolete as organizations become more demanding and shift the delivery methodology risk on to the service provider [2].

The development of SOA, organization is able to compose complex applications from distributed services supported by third party providers. Service providers and User negotiation based service level agreement (SLA) to determine different activities (security, cost, penalty,…..etc) on the achieved performance level. The service providers need to manage their resources to maximize the profits [3].

To maximize the SLA revenues in shared data environments, it can be formulated as the dual problem of minimizing the response time and maximizing throughput. That proposal considers the problem of hosting multiple web sites.

In service oriented architecture the problem is guaranteeing the "quality" of services to final users, in terms of functional and non-functional requisites like performance (measurable qualities) or security (immeasurable qualities). In general, a service provider is able to guarantee a predefined service level and a certain security level (supposing it can be measured).

II. The Evaluation methodology

Having formalized and expressed SLAs by a policy form, in this paper we need an evaluation methodology to compare them and decide to request a service from the server or changing the server to provide the request service [1].

The proposed methodology is based on a Reference Evaluation Model (REM) to evaluate and compare different security policies and behavior policies, quantifying their security level by either value 0 (represent not existing sub provision for each immeasurable provisions) or value 1 (represent existing provisions) . The model will define how to express in a rigorous way the security policy (formalization), how to evaluate a formalized policy, and what is its service level. In particular the REM is made of three different components:

A. The policy formalization

The formalization policy is a way to express all the qualities parameters for SLA in a rigorous technique to define either the qualities exist or not, or for measurable qualities with defined values.

B. The evaluation technique

We propose REM includes the definition of a technique to compare and evaluate the policies; we have called this component the REM Evaluation technique. Different evaluation techniques represent and characterize the measurable and immeasurable level associated to a policy in different ways, for example with a numerical value, a fuzzy number [4][5] or a verbal judgment representing its security level.

C. The reference levels

The last component of the REM is the set of SLAs levels that could be used as a reference scale for the numerical evaluation of SLA. When references are not available, the REM could be used for direct comparison among two or more policies.

III. SLA Structure

The service providers and the users often negotiate by the qualities to base the Service Level Agreements (SLAs) by behavior aspects based on the achieved performance levels. The service provider needs to manage its resource to maximize its profits. The optimization approaches are commonly used to provide the service load balancing and to obtain the optimal classifications for quality of service levels. By the above are also used as guidelines and for realizing high level trends. One main issue of these systems is the high variability of the workload according to values of measurable and immeasurable qualities [6].

By such model, the service can dynamically be allocated among the service providers depending on the service availability. Fig. 1 shows the architecture of SLA model implementing an autonomic infrastructure. Service providers are allocated and de-allocated on demand on servers. The server level agreement model can monitor the qualities and by predictor phase, it can allocate the server to provide the service.

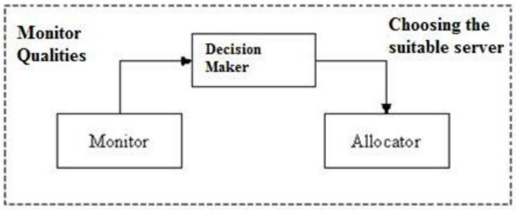


Fig. 1. SLA Architecture Of Data Center

The main components of the SLA model [7] include a monitor, a decision maker and a server allocator. The system monitors the qualities and performance matrices of each form, identifies requested classes and estimates requested service time. The decision maker can evaluate the system performance from the trace values. The allocator chooses the best system configuration [8].

IV. Service Level Agreement Categories

SLA was defined as a contract between the users and service providers, and then the contract has many rules. Their rules can be established by different ways either manually or automatically. It can be either static, which means the system will be fixed without any modifications or upgrading all the service providing or the contract rules be dynamic and changing all the time, the SLA can be classified according to the inputs values collection way and the SLA, the inputs can define as qualities can classify as below:

D. Measurable Qualities

There are many measurable qualities; it can measure for each user, the definitions of the measurable qualities are shown *below:*

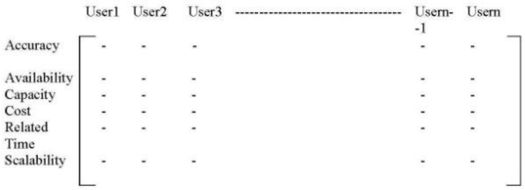
- Accuracy is concerned with the error rate of the service. It is possible to specify the average number of errors over a given time period.
- Availability is concerned with the mean time to failure for services, and the SLAs typically describe the consequences associated with these failures. Availability is typically measured by the probability that the system will be operational when needed. It is possible to specify the system’s response when a failure occurs − the time it takes to recognize a malfunction.
- Capacity is the number of concurrent requests that can be handled by the service in a given time period. It is possible to specify the maximum number of concurrent requests that can be handled by a service in a set block of time.
- Cost is concerned with the cost of each service request. It is possible to specify the cost per request the cost based on the size of the data − cost differences related to peak usage times.
- Latency is concerned with the maximum amount of time between the arrival of a request and the completion of that request.
- Provisioning-related time (e.g., the time it takes for a new client’s account to become operational).
- Reliable messaging is concerned with the guarantee of message delivery. It is possible to specify how message delivery is guaranteed (e.g., exactly once, at most once) whether the service supports delivering messages in the proper order.
- Scalability is concerned with the ability of the service to increase the number of successful operations completed over a given time period. It is possible to specify the maximum number of such operations.

E. Immeasurable Qualities

- There are three main immeasurable qualities that can be defined by main provision and sub provision for each quality, in the following we define the immeasurable qualities:
- Interoperability is concerned with the ability of a collection of communicating entities to share specific information and operate on it according to an agreed upon operational semantics. It is possible to specify the standards supported by the service and to verify them at runtime. Significant challenges still need to be overcome to achieve semantic interoperability at runtime.
- Modifiability is concerned with how often a service is likely to change. It is possible to specify how often the service’s Interface changes or Implementation changes.
- Security is concerned with the system’s ability to resist unauthorized usage, while providing legitimate users with access to the service. Security is also characterized as a system providing non-repudiation, confidentiality, integrity, assurance, and auditing. It is possible to specify the methods for
  - ➢ authenticating services or users
  - ➢ encrypting the data

F. Policy Formalization

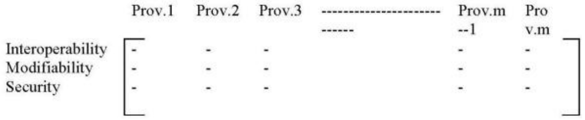
It will depend on measurable and immeasurable qualities which are mentioned is previous section. It will form two unsymmetrical matrices; first one describes the immeasurable qualities for n users, as well as the output matrix describe the decided output. The second matrix represent the measurable matrix, each row represent the provision and the state of each column represent if the sub provision exist or not, the output matrix represent the trace value of measurable matrix. Trace is the value of each matrix by maximum value of matrices.



$$\begin{array}{l|ccccccc|} & \text{User1} & \text{User2} & \text{User3} & \text{------------------------------} & \text{Usern-1} & \text{Usern} \\ \text{Accuracy} & - & - & - & & - & - \\ \text{Availability} & - & - & - & & - & - \\ \text{Capacity} & - & - & - & & - & - \\ \text{Cost} & - & - & - & & - & - \\ \text{Related} & - & - & - & & - & - \\ \text{Time} & & & & & & \\ \text{Scalability} & - & - & - & & - & - \end{array}$$

Measurable Matrix

Output user = [Trace, Request, Delay Request, Wait, Change Provider, New User]



$$\begin{array}{l|cccccc|} & \text{Prov.1} & \text{Prov.2} & \text{Prov.3} & \text{---------------------------} & \text{Prov.m--1} & \text{Prov.m} \\ \text{Interoperability} & - & - & - & & - & - \\ \text{Modifiability} & - & - & - & & - & - \\ \text{Security} & - & - & - & & - & - \end{array}$$

Immeasurable Matrix

Output= [Trace, Activated output]

V. Mathematical Evaluation Technique

To adopt WS-policy framework and to express policies for security-SLA the framework is structured as a hierarchical tree to express all sub provisions. We have started the formalization by considering the set of items proposed by [9].

Given the qualities matrices for measurable matrices as shown in section III, the evaluation process takes into account just the provisions of the policy which represent the sub provisions status for immeasurable qualities and the values of measurable qualities for each user.

With the formalization, each provision is represented by a real data type; the policy space "P" is defined as the vector product of all n provisions Ki

i.e. P = K1 x K2 x…... x Kn.

The policy space "P" has been transformed into an homogeneous one, denoted "PS" thanks to a family of threshold functions (F-functions) which allow us to associate a Local Security Level (LSL for short) to each provision. "PS" is represented by a n x 4 matrix whose n rows represent the single provisions Ki and 4 is the chosen number of LSLs admissible for each provision. For example, if the LSL associated to a provision is 3, the vector corresponding to its row in the matrix is: (1, 1, 1, 0).

The distance criteria for the definition of the metric space is the Euclidean distance among matrices, defined as:

$d(A, B) = \sqrt{\sigma(A - B, A - B)}$

Where: σ (A - B, A - B) =T r ((A - B) (A - B) T)

The T r ((A - B) (A - B) T) represents the trace value to know the invariant between any base and other qualities matrix. Below we show a sample of immeasurable matrix, the main provision was defined and sub-provisions were represented by existing status:

Provision Name	1	1	0	1
Digital Certificate Management	0	1	0	1
Policy applicability	-	-	-	-
Time between certificate request and issuance	-	-	-	-
Notification of certificate issuance and revocation	-	-	-	-
Local Registration Authorities (LRAs)	-	-	-	-
Repositories	-	-	-	-

SLA-P, SLA-X, and SLA-Y are supposed as different immeasurable matrices with the same provisions and different values of sub-provisions. SLA-X and SLA-Y; are supposed globally stronger than SLA-P since they have a lot of sub-provisions for each provision. Each policy in the example has just 10 provisions; this is just a simplification which does not affect the validity of the method.

$$\mathrm{SLAP} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathrm{SLAX} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \mathrm{SLAY} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Example 1: SLA-X is a policy that appears stronger than SLA-P, just looking at the levels of the single provisions; we first calculate the trace:

Tr((SLAX-SLAP)(SLAX-SLAP)T) = 6

Where: $d(X, P) = \sqrt{\sigma(X - P, X - P)}$ d(X, P) is the distance among matrices

Where: σ (X - P, X - P) =T r ((X - P) (X - P) T)

The distance between SLA-X and SLA-P is: d-2.45.That mirrors the fact that SLA-X is just a little stronger than SLA-P.

Example 2: SLA-Y is a policy that appears stronger than SLA-X and much stronger than policy SLA-P, while SLA-P is the same as that of the example 1; the trace is:

Tr((Y-P)(Y-P)T) = 19

The distance between SLA-Y and SLA-P is: d-4.36 .This result mirrors the evident difference between the two cases. These examples show how it is very simple to evaluate the distance between policies, once they have been represented as a matrix. The distance will be adopted to define the metric function.

VI. Proposed ANNSLA Model

In this section we will propose an artificial service level agreement model to be the base for qualities testing and comparing. The main idea of this model is to be able to decide if the services can supply by service provider or not depending on the value of the activated outputs. Because of this model was proposed depending on artificial neural network then the ANN needs to train by input and output data sets to be able to extract the outputs for other input data sets. Fig. 2 shows the general concepts of Neural Network Service Level Agreement in SOA.

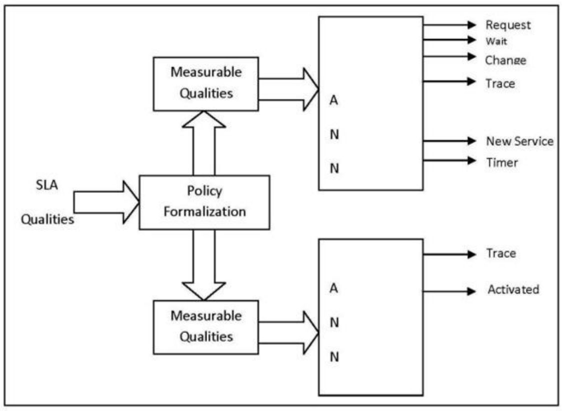


Fig. 2. ANN Service Level Agreement

The training input and output data sets will be represented according to the mathematical models. The input sets of ANN represent the measurable and immeasurable matrices which are used in mathematical model while the output sets represents the extracted outputs for the traces and the activated outputs from the mathematical calculations model. To build the artificial service level agreement model, there are two phases should execute to extract the correct decision. In the following we will discuss the phases of ANNSLA model:

- ANN Training phase: in this phase, the extracted data sets (formalized inputs and calculated outputs) from the SLA mathematical model will apply to ANN to train it. Section V.B will explain the simulation method of training phase for ANN. The training phase should apply for both part of ANNSLA model, measurable and immeasurable qualities and trained according for input/output matrices. It's very important to notice when any changing happen for the contact between the user and the service provider (i.e. the contract rules) will lead to change the SLA measurable and immeasurable qualities, the trace value will change directly as well as the outputs matrix may change. All those lead to reconfigure the ANN architecture and retrain it according to the new values of inputs/ outputs matrix values. From all the above we can consider the ANNSLA model is dynamic not static model and can be suitable for future extending, there are no limitations of users number or qualities number. The accuracy of output values ANNSLA depending on data sets and the training procedures. Fig. 3 shows the general ANNSLA model in the training phase. The ANN will train in 1st set of qualities and outputs. There are two training model one of them for measurable qualities ANN model and second one for immeasurable qualities ANN model, it can train separately because of the first one represent a logical model and the second one represent the behavior model between the user and service provider. The architecture of the ANN is very easy to build, for this reason we propose only one activated output for immeasurable formalization while there are more than one output for measurable formalization to mention that the ANN architecture may not be fixed and modifiable depending on the requirements and the rules in the contract between the user and the service providers [10].
- Operation Phase: when the ANN trained correctly according to inputs/outputs data sets, it will be ready to use it as decision maker for different input sets to extract the trace value and activated output or outputs for either immeasurable qualities or measurable qualities. To evaluate the output values, it can compare the extracted outputs with calculated outputs from mathematical model, the evaluation will explain in details in 8. Fig. 4 shows the general trained ANNSLA model. By this model can extract the outputs for different inputs by applying it to trained ANN. As we mentioned in the training phase, the trained ANN be as a decision maker to extract the outputs. When applying the immeasurable or measurable matrices to trained ANN it can extract the trace and activated output or outputs. The operation phase can be fixed for long period except if there is any rules in the SLA contact will be changed, it will lead to go again to training phase to modify the ANN architecture according to new measurable and immeasurable qualities.

From the entire above if there is any modifications in input qualities or/and the active outputs will lead to reconfigure the ANN model and retrain it according to new values of the input qualities and outputs. The evaluation method will not change when the architecture of the ANN model changed because all the time we will calculate the difference between the calculated values and extracted outputs.

The process of ANNSLA will show in 6.1.The validation of the activated outputs and the reasons of defining activated outputs for both formalizations (measurable and immeasurable) will discuss in section VI. There are many limitations in ANNSLA operation phase, one of important operation phase limitation that can't test this model online because of the servers' operation restrictions and no experiments lab to evaluate this model practically.

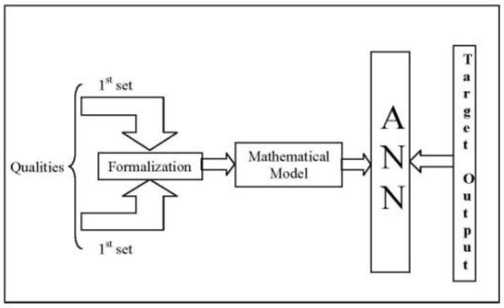


Fig. 3. ANNSLA training phase

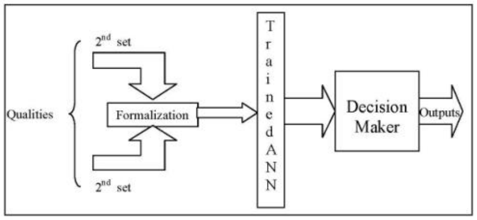


Fig. 4. ANNSLA operating phase

G. ANNSLA Process

- The ANNSLA process for two parts of the model, the first part represents measurable part with their extracted and target outputs and the second part represents immeasurable part with their extracted and target outputs, each phase has many steps will discuss in the following:
- The training phase process consist the following steps:
- Define the ANN architecture, number of hidden layers, number of input nodes in input layer, and number of output nodes in output layer. The process initiated sets of measurable and immeasurable qualities as inputs for ANNSLA model. Formalize the inputs into measurable and immeasurable input matrices to apply it to ANN. The target outputs will define for the both measurable and immeasurable ANN. The training process will initiate to train the ANNs. The training process will continue till the extracted outputs reached to target outputs. When the error percent reached to acceptable value, the training process will stop and the ANN will consider trained.
- The operation phase process consist the following steps:
- The process initiated a set of measurable and immeasurable qualities as input for trained ANNSLA model. Formalize the input into measurable and immeasurable input matrices to apply it to ANN
- Apply formalize measurable and immeasurable matrices to trained ANN. The trained ANN will run and stop when the ANN produces the outputs. The extracted outputs represent the outputs for both measurable and immeasurable parts.

From all the above, it consider that the ANNSLA model can modify the ANN architecture by changing the number of inputs and/or the number of outputs, this will lead to repeat the ANNSLA process in both training and operation phases.

H. ANNSLA Simulation

The ANNSLA will build using ANN architecture, it's very important to select the architecture by defining the number of layers (hidden layers), the number of input nodes in input layer, the output nodes in output layer, the values of inputs, the values of target outputs, and the error value. These architectures can be done by using MATHLAB package version 7.7.0. The simulation procedure will introduce the ANN architecture, the experimental data sets, training process and the running process of ANNSLA by MATHLAB model in next sections.

I. ANNSLA MATLAB Architecture

As we mentioned in the above sections the ANNSLA model architecture has two part one for measurable qualities and the second one for immeasurable qualities, this means there are two ANN.

The measurable ANN part consists the following: Input layer with consist 6 input neurons for each user. The inputs represent the measurable qualities which are the accuracy, availability, capacity, cost, related time, and scalability. The values of these inputs represent numeric value all the time ≤ 1 [11].One or two hidden layers to test the effect of increasing the hidden layers upon the performance of the architecture. The output layer consists either one neuron if we want to calculate trace of measurable matrix or five neurons represent the outputs (Request, Wait, Change, New Service, and Timer) for measurable part in ANNSLA model.

The immeasurable ANN part consists the following: Input layer with consist 40 input neurons for each user. The inputs represent the immeasurable qualities represent interoperability, modifiability, and security. All the values of these inputs will be represented either 1 (exist) or 0 (not exist).One or two hidden layers to test the effect of increasing the hidden layers upon the performance of the architecture. The output layer consist either one neuron if we want to calculate trace of immeasurable matrix or one neuron represents the activated output for immeasurable part in ANNSLA model[12].

The above ANN architecture can modify according to the measurable and immeasurable requirements in ANNSLA model.

VII. Data Collection and Experimental Results

In our test scenario, we used a sample of different matrices for measurable and immeasurable qualities; there are two experimental results as following: The trace and the distance for each matrix will calculate by mathematical model. The values of the mathematical traces will apply to train the ANNSLA. The ANN will train for input matrices and for target outputs. The trained ANN can run to extract the outputs when applying any new immeasurable or measurable matrices. Below we will mention the training input data sets and the target output which represent the trace vales and the activated output/outputs by applying it to trained ANN for both measurable and immeasurable parts for ANNSLA model. In next section, we will run the trained measurable and immeasurable ANN to extract the actual outputs for any data sets.

J. Immeasurable Experimental Results:

To train the immeasurable ANN to extract the trace value and activated output for each immeasurable matrix, the training data sets with target outputs (trace and activated output) will apply and train ANN by MATHLAB. TABLE I mentions the data sets of the immeasurable matrices with calculated trace (by mathematical SLA model which is mentioned in section IV) and the activated output (proposed simulation values).

These matrices are used to train the immeasurable ANN, it used as inputs while the calculated traces and the activated outputs are used as target outputs of immeasurable ANN. The trace ANN is used to extract the trace value for any measurable or immeasurable matrices. The trace value represents the invariant with respect to a change of basis. It means choosing a base matrix then by calculating distance between the choosing one with any other selected matrix. It easy to calculate the trace mathematically, and by applying the qualities to trained trace ANN (shows in Fig. 5) can extract the trace value, in the next section will show the comparison between the two values.

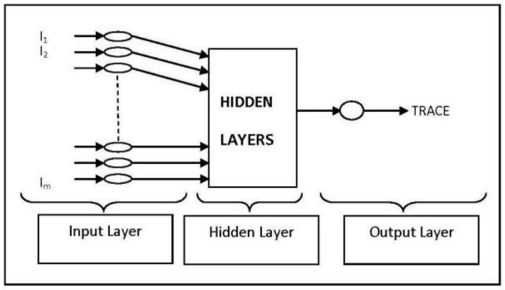


Fig. 5. Trace ANN Architecture

The trace value will extract by individual ANN represented in Fig. 5, the trace ANN should train by applying the immeasurable matrices as inputs and the target output is represented by calculated trace values. To train the ANN immeasurable ANN, the immeasurable qualities will apply with their target activated output; the number of training matrices is about 17. TABLE I shows the training set for immeasurable ANN. Fig. 6 shows the general ANN architecture for immeasurable ANN part, the number of input neurons is 40 neurons for each immeasurable matrix in input layer, and only one neuron in output layer represents the target output.

The values of the target output are representing the activated output values. To train the immeasurable ANN, we will apply 17 immeasurable matrices (M1, M2,……….Mn) with their proposed activated outputs. It represents 10 provisions with 4 sub-provisions for each provision; the provision represents some of security level tree structure, Interoperability, and Modifiability with their sub-provision. After training the immeasurable ANN, it will be ready to run and extract the activated output for any immeasurable matrix. Feed forward network architecture with back propagation momentum training algorithm was used. The back propagation algorithm was considered since it is the most successful algorithm for the design of multilayer feed forward networks

TABLE I. TRAINED IMMEASURABLE DATA SETS FOR THE TARGET TRACE VALUE

Matrix No.	Calculated Trace	Proposed Activated Output
M1	3.581	0.5
M2	3.771	0.7
M3	4.123	0.6
M4	3.5541	0.9
M5	4.321	0.6
M6	3.5487	0.8
M7	4.242	0.65
M8	5.1832	0.75
M9	3.6111	0.8
M10	4.0231	0.5
M11	4.1101	0.7
M12	4.3465	0.85
M13	4.5732	0.7
M14	4.1021	0.55
M15	4.0231	0.65
M16	3.945	0.7
M17	4.1001	0.5

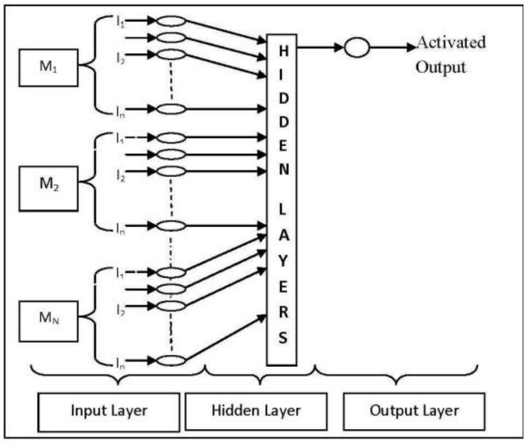


Fig. 6. Immeasurable ANN Architecture

K. Measurable Experimental Results:

The measurable qualities can be represented by the measurable matrices as input and the target outputs (representing the values Accuracy, Availability, Capacity, Cost, Related Time, and Scalability) for each user, it already defined previously.

To train the measurable ANN part in ANNSLA, we will apply the measurable matrices as input data sets for the ANN and the activated outputs (Request, Delay Request, Wait, Change Provider, and New Customer) and trace as a target outputs for ANN. The values of simulation inputs are represented analogy; it's all the time ≤1. The activated outputs are represented logically either 0 or 1. TABLE II shows the proposed activated outputs for each measurable matrix with calculated trace by mathematical model. These input data sets with proposed activated outputs and calculated trace will apply to train the measurable ANN part in ANNSLA.

TABLE II. TRAINED MEASURABLE DATA SETS FOR PROPOSED ACTIVATED OUTPUTS AND TARGET TRACE VALUE

Matrix No.	Activated Output	Calculated Trace
MM1	1,1,0,1,0	0
MM2	1,1,0,0,0	7.2621
MM3	1,0,1,1,0	8.3946
MM4	1,1,0,1,0	7.4403
MM5	1,1,0,1,0	6.6554
MM6	1,1,1,1,0	6.8806
MM7	1,0,1,1,0	6.5035
MM8	1,1,0,1,1	5.6391
MM9	1,1,1,1,0	7.1493
MM10	1,0,1,1,0	9.0777
MM11	1,1,0,1,1	5.6391
TMM1	1,0,1,1,0	6.5035
TMM2	1,1,0,1,1	5.6391
TMM3	1,1,1,1,0	7.1493

Fig 7. shows the general measurable ANN part for ANNSLA model, it represents the input, hidden, and output layers. To train the measurable ANN can be done by applying 12 inputs/outputs measurable qualities/ target outputs.

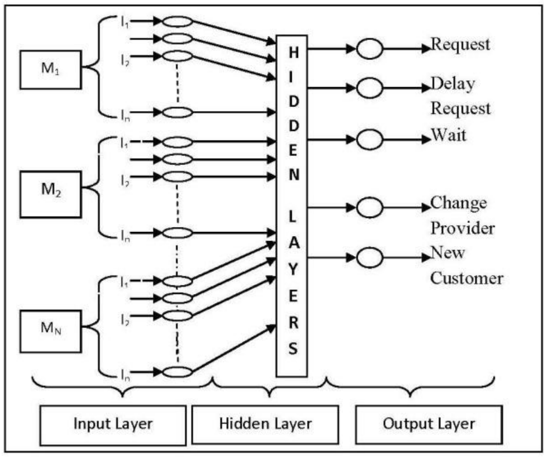


Fig. 7. Measurable ANN Architecture

VIII. SIMULATION RESULT DISCUSSION

After the trace, measurable and immeasurable ANNs in ANNSLA are trained, and then it can be operated as a decision maker. The scenario to run the trace and the immeasurable ANN, we apply 20 immeasurable matrices with by trace ANN, it can extract the trace value for each measurable matrix by ANN is shown in Fig. 5. The immeasurable ANN extracts the activated output for each matrix (ANN is shown in Fig. 6). In TABLE III the set of immeasurable tested matrices are shown, it shows the calculated trace and proposed activated output for each tested immeasurable matrices to compare the calculated trace with extracted output from trace ANN and the proposed activated output with extracted activated output from immeasurable ANN.

The scenario to run the trace and the measurable ANN, we apply 3 measurable matrices with its activated outputs. By trace ANN and measurable ANN can extract the trace and the activated outputs for each matrix. TABLE IV shows the set of measurable tested matrices, the calculated trace, proposed activated output for each tested measurable matrices, and extracted activated outputs.

IX. RESULTS EVALUATION

From TABLEs III and IV, we can conclude that the error value between the calculated activated output and ANN activated output is about +/- 0.04 and the error between the calculated trace and extracted ANN trace is between 0.1141 and -0.0088 for both measurable and immeasurable ANN running, this means that ANNSLA can specify the SLA requirements specially the value of trace for measurable and immeasurable matrices. From the extracted trace value, by ANNSLA can decide the suitable SLA between the user and service providers. The activated outputs/output for measurable/immeasurable ANN can justify the SLA work flow.

From all the above, it can consider the ANN SLA can work properly as policy selector and as a decision maker between users and service providers. The figures below mention charts represent different comparison between the values which are extracted from ANNSLA and other calculated or supposed values. Fig. 8 shows the chart for calculated and ANN trace for different immeasurable qualities matrices, it shows the extracted values are very close to calculated values, the error as mentioned in above not more +/- 0.12. Fig. 9 shows the chart for activated and ANN output for immeasurable ANN part, the error between two values is between +/-0.04.

For measurable ANN part in ANNSLA model, Fig. 10 shows the chart for calculated trace and ANN trace for different measurable qualities matrices set.

From above, the ANNSLA model can use as a decision maker to extract the best policy and the suitable activated output/outputs value; it means, the ANNSLA can define the roles of the contract between the users and service providers.

The model can modify easily by reconstruct the ANN architecture to be suitable for the approximate number of the users and providers. It can very secure if each user has his key to be part of the immeasurable input qualities then the model can train to accept or refuse the user in other hand the model can accept or refuse the key for the provider.

The error value of the extracted outputs of the model was very small according to the testing sets; it’s very simple to modify the model. The ANN can be extended simply and the training rules are very efficient and know for different users.

TABLE III. TESTING IMMEASURABLE MATRICES SETS

Matrix No.	Proposed Activated Output	ANN Activated Output	Output error	Calculated Trace	ANN Trace	Trace error
Y1	0.6	0.56	0.04	3.6056	3.593	0.0126
Y2	0.8	0.81	-0.01	3.873	3.866	0.007
Y3	0.5	0.52	-0.02	4.2426	4.222	0.0206
Y4	0.8	0.79	0.01	3.4641	3.353	0.1111
Y5	0.7	0.69	0.01	4.2426	4.213	0.0296
Y6	0.7	0.685	0.015	3.5651	3.451	0.1141
Y7	0.7	0.71	-0.01	4.246	4.233	0.013
Y8	0.7	0.72	-0.02	5.1962	5.1871	0.0091
Y9	0.7	0.68	0.02	3.6056	3.5942	0.0114
Y10	0.6	0.61	-0.01	4.1231	4.1021	0.021
Y11	0.8	0.78	0.02	4.1231	4.1111	0.012
Y12	0.75	0.73	0.02	4.3589	4.3462	0.0127
Y13	0.8	0.78	0.02	4.5826	4.5721	0.0105
Y14	0.6	0.62	-0.02	4.1231	4.0231	0.1
Y15	0.65	0.63	0.02	4.1231	4.1001	0.023
Y16	0.75	0.77	-0.02	4	3.995	0.005
Y17	0.6	0.63	-0.03	4.1231	4.0956	0.0275
Y18	0.7	0.72	-0.02	4.1231	4.201	-0.0779
Y19	0.8	0.81	-0.01	4.1231	4.0331	0.09
Y20	0.7	0.67	0.03	3.4641	3.453	0.0111

TABLE IV. TESTING MEASURABLE MATRICES SETS

Matrix No.	Activated Output	ANN Activated Output	Output error	Calculated Trace	ANN Trace	Trace error
TMM1	1,0,1,1,0	0.965,0,0.98,0.97,0	0.035,0,0.02,0.03,0	6.5035	6.5123	-0.0088
TMM2	1,1,0,1,1	0.96,0.97,0,0.99,0.98	0.04,0.03,0,0.01,0.02	5.6391	5.6458	-0.0067
TMM3	1,1,1,1,0	0.971,0.96,0.97,0.98,0	0.029,0.04,0.03,0.02,0	7.1493	7.1578	-0.0085

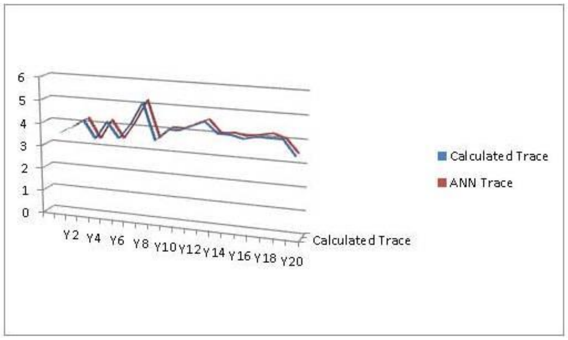


Fig. 8. Calculated and ANN Trace Chart for immeasurable matrices

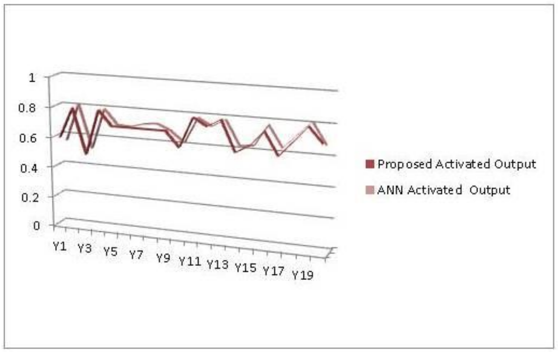


Fig. 9. Activated and ANN Output Chart for immeasurable matrices

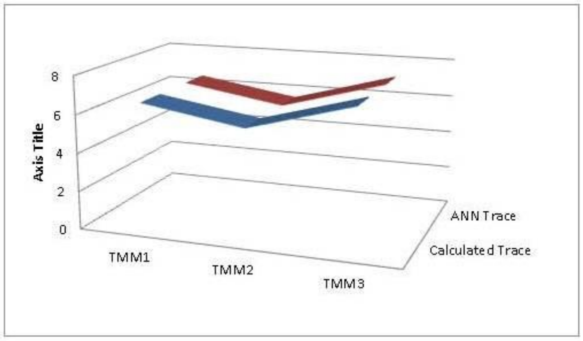


Fig. 10. Calculated and Trace Chart for measurable matrices

The ANN takes the decision according to how the user train it and what the data types give to it, then the main important should manage carefully from the beginning before constructing the model is as the following:

- Number of users.
- Number of providers.
- Define certain policy formalization for the qualities.

The main immeasurable and immeasurable qualities (it's very important to define the security level for each user and providers).The architecture of trace, measurable, and immeasurable ANN. The presented approach aims to use the neural network as the intelligent part of service level agreement model. The global architecture will permit the provider to provide a new type methodology of providing a service by checking the system completely with all other users and service providers. This will help to reduce the cost while preserving the required QoS. The ANNSLA can provide monitoring of traffic for service request by service waiting or service reallocation to other service provider. Then the deployed architecture will be evaluated with respect to the mathematical model

X. CONCLUSION

In this research we have introduced a theoretical methodology to evaluate Service Level Agreement in SOA. The methodology is based on two fundamental features; the first one is the SLA formalization through the use of standard policy while the second one is the formalization of "qualifiable service levels" against which we could measure the SLA.

In particular, we have adopted a Reference Evaluation Model, developed for different methodology, to evaluate and compare different policies and quantifying their levels. The application of the methodology in different samples of measurable and immeasurable qualities and we adopted it in the integration of mathematical model and artificial model to guarantee the same perceived service level to the end-user. SLAs have been applied in service organizations in general, including IT organizations, to formalize the level of service between a service provider and service user. While SLAs are well understood in these domains, they are less understood for services in the SOA context.

SOA enables the integration of automated services from multiple organizations. External providers may offer services that were not initially implemented to meet the quality attribute requirements of the service consumer organization. Defining an SLA and establishing SLA management mechanisms are important factors when clarifying the quality requirements for achieving the business and mission goals of SOA systems.

Standardized SLAs are going to be an important element for organizations moving to automation, SLA systems characterized by the dynamic discovery, composition, and invocation of services based on QoS and other contextual information.

There are efforts to standardize the SLAs for web services; it's an effort to make SLAs machine adaptable. However, there is no established standard for SLA specification

REFERENCES

[1] AberdeenGroup "Enterprise Service Bus and SOA Middleware", Boston, AberdeenGroup Massachusetts, 2006.

[2] K.Alexander .and L. Heiko "The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services", Journal of Network and Systems Management,11(1), pp. 57-81,2003

[3] T. Luo and L. Meng, "SLA foundation template library: reusable-component repository for SLA", Communication Technology Proceedings, ICCT 2003. International Conference on , 2 , ,pp. 1739 – 1743, 9-11 April, 2003.

[4] B. Wetzstein, D. Karastoyanova, and F. Leymann, "Towards Management of SLA-Aware Business Processes Based on Key Performance Indicators", BPMDS'08, Institute of Architecture of Application Systems, University of Stuttgart, Germany,Springer, June2008.

[5] Service Level Agreements in Service-Oriented Architecture Environments; http://www.slideshare.net/Zubin67/service-level-agreements-in-serviceoriented-architecture

[6] M. Bishop "Developing Web Services" Proceedings 17th International Conference on Data Engineering, Los Alamitos CA : IEEE Computer Society, pp. 477-481, 2003.

[7] C. Abrams and W. Roy "service-Oriented Architecture Overview and Guide to SOA Research", G00154463, Stamford: Gartner Research, 2008.

[8] K. Yuen, and H. Lau "A Distributed Fuzzy Qualitative Evaluation System", IAT '06 Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology, Hong Kong : IEEE, pp. 560-563,2006.

[9] IBM "WS-policy specification Web Services Policy Framework", IBM, BEA Systems, Microsoft, SAP AG, Sonic Software, VeriSign,2006.

[10] N.A. Abdullah "An Architecture for Augmenting the SCORM Run-Time Environment As a Service", Phd, Learning Societies Group,2006.

[11] V.Caola, A. Mazzeo, N. Mazzocc and M. Rak"A SLA evaluation methodology in Service Oriented Architectures" Advances in Information security', Advances in Information Security,23, pp.119-130,2006.

[12] L.Lu "A Novel SOA-Oriented Federate SLA Management Architecture", IEEC '09. International Symposium on Information Engineering and Electronic Commerce , Ternopil: IEEEE, pp. 630-634,2009.