Scalable and Self-Adaptive Service Selection Method for the Internet of Things

Manel Mejri
Research Laboratory SMART LAB Tunis, Tunisia

Nadia Ben Azzouna
Research Laboratory SMART LAB Tunis, Tunisia

ABSTRACT

The Internet of Things goes beyond the regular Internet by offering new functionalities and creating new range of services provided by the deployed objects. Therefore, one of the most challenging issues is to select the best service among similar functionally available ones. In this paper, we propose to involve both artifcial intelligence through the use of Artifcial Neural Network (ANN) and multi criteria analysis through the use of Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) model in order to return the best service to the requestor. First, The ANN is introduced as a predictive model to estimate the Qualities of services (QoS) according to user context, service context and network context. Second, the TOPSIS model evaluates, then aggregates these QoS values in order to provide the best service according to user preferences. To improve the scalability of the proposed service selection system we conduct a parallel implementation of the prototype.

General Terms

Service selection in the IoT

Keywords

Internet of Things, non-functional properties, QoS, Contextual attributes, preferences

1. INTRODUCTION

The Internet of Things (IoT) is based on the idea that all objects can be connected to the Internet and are able to transmit information and possibly receive commands. These objects are recognized according to their identities, physical attributes and their virtual personalities and use intelligent interfaces [2]. Indeed, the IoT opens up a new area of ubiquity where anyone or any thing can be connected at any time in any place. Therefore, besides people to people communication, it allows communication between objects to people and between objects [23]. The increasing number of heterogeneous existent objects that can meet the IoT applications and the complexity related to manage the huge amount of information provided by these objects, represent a crucial challenges [28]. Hence, the number of available services and of their sources increase proportionally.
In this paper, we shed more light on solving the problem of service selection in the IoT, particulary, how to offer the most suitable service among several functionally similar one. In fact, in the context of the IoT, there are different sources connected to the internet providing services like cameras, sensors, actuators, mobiles, homes, hospital, city etc. Hence as for a user, it is overriding to be delivered the right service, at the right time and in the right place.
The service selection method is, essentially, based on different criteria known as non-functional attributes which are Quality of Service (QoS) attributes and contextual information. Mainly, there are three challenges to be adressed : First, the service selection method must be adaptable according to the context, current QoS and user preferences. Second, It must return the most appropriate service for the requestor [19]. For instance, the best service can be the nearest one from the user. Hence, the user location refers to a contextual information that has an influence in delivering the most suitable service. Third, in the context of the IoT, scalability is an important issue to be solved; as the number of deployed objects and offered services is steadily increasing. Hence, the service selection method must be able to continue its function following the increase of available services.
In order to overcome these issues, we take advantages from both artificial intelligent methods through the use of Artificial Neural Network (ANN) and Multi-Criteria Decision Making (MCDM) method through the use of Technique for Order Preference by Similarity to Ideal Solution (TOPSIS). First, we exploit the ANN as being among the most powerful and universal predictor [11]. Its objective is to provide at each time a QoS value according to the current situation of both service provider, service requestors and the network state. Taking those contextual information as ANN input data, it predicts the corresponding QoS values. Second, according to user preferences, the most appropriate one is selected by applying TOPSIS [24]. Moreover, to improve scalability, we conducted a parallel implementation based on OpenMP model.
The layout of this paper is as follows: Section 2, presents an overview of service selection methods. Section 3, addresses related work and identifies some limitations. Section 4, describes the new service selection method. Section 5 is dedicated to the experimental study. Finally, Section 6, provides the concluding remarks.

2. OVERVIEW OF SERVICE SELECTION

2.1 Service selection based on non-functional properties

The non-functional properties are the criteria used to differentiate between similar services in terms of performance. They are defined by QoS and contextual information.
According to W3C group, QoS refers to various attributes namely : performance, reliability, scalability, robustness, availability, etc. [16]. Those QoS attributes are influenced by contextual informa-

tion. In fact, context-aware systems use the context information related to the execution environment of the user in order to return relevant information or services. Such systems are able to better meet the users' needs [7]. Gao et al. used a fully-connected network to predict the performance of services which is influenced by contextual attributes [11]. In fact, contextual information is related to the implicated parties like service provider, user requestor and the network. Unfortunately, this work considers only the network context to perform prediction. Similarly, Baraki et al. applied an MLPNN to predict three QoS values that are respectively, the response time, the throughput and the reputation for each Web Service. The predicted values are personalized for users since the context data are taken into account [4]. This work focuses only on the user context like geographical distance to perform predictions. None of the aforementioned works has used the user context, the service context and the network context in order to perform the seletion of the best service. Moreover, a lot of approaches have framed the service selection problem as a search problem of the best service according to the non-functional attributes which is classified in the class of NP-hard problems [5]. In this case, one challenging issue is the scalability of the selection method. It should not only provide the best service in an acceptable timescale but also keep the same behavior with the increasing number of services. Hence, it is not a wisdom choice to formulate service selection problem in the IoT as a QoS search based problem since algorithms complexity increase as long as services and QoS attributs are available.

2.2 Service selection based on user feedback

In order to achieve the service selection based on user feedback, reputation mechanism was widely used. It consists in using previous feedbacks of users which mean a set of historical interactions to identify the best services or service providers among several ones[21]. there are mainly two categories of reputation systems : (1) centralized and (2) decentralized [10].

(1) Centralized reputation systems rely on a single and central entity to manage reputations like collecting, calculating and updating scores for all implicated parties [27].

Some applications of centralized reputation systems for service selection are provided in [18] and in [17]. They employed the typical architecture of web service which has a central UDDI server. On one hand this architecture allows service providers to publish their services and on the other hand, it allows users to search for services. Moreover, they used a central QoS registry that has the responsibility of gathering and recording QoS information from web service consumers.

Generally, such systems are less complicated and easy to implement but they have some limitations like the single point of failure and the bottleneck problems because of the centralized UDDI and they must be equipped with powerful servers [29]. That is why, decentralized reputation systems were introduced.

(2) Decentralized reputation systems are characterized by the absence of the central authority node unlike centralized ones. Hence, in order to control reputation, all members must cooperate, coordinate and communicate with each others. The suitable examples in the most cases are multi-agent and Peer to Peer (P2P) systems [27]. In this context, Vu et al. in [26], have proposed the use of QoS registries to store consumer's feedbacks for QoS. Every registry manage reputation of a set of service providers. Furthermore, P2P system is deployed to organize QoS registries with the specific P-Grid structure. In fact, the P-Grid is employed to manage distributed data. It is a totally decentralized P2P structure that does not need any central coordination. It has a binary search tree form that spreads replication over a community of peers and handles efficient search [27].

Generally such systems are more scalable than centralized ones with a strong gain of bandwidth. Moreover, when a failure occurs in the system, stored data are always accessible. But, the limitation is that they are more complicated to design and to implement.

Overall, the aforementioned approaches allow service requestors to safely select a service or a service provider with higher reputation. Unfortunately, there are some drawbacks. First, they must deal with trust issue since the dishonest of malicious users and service providers is unavoidable which makes the reputation mechanism more and more complex. Second, the establishment of reputation community and the calculation of the necessary measurements for reputation assessment can be time consuming [27].Hence, as a deduction, the adoption of reputation system in the context of service selection with the consideration of both trust and QoS aspects is hard in real world applications [29].

3. RELATED WORKS

There are few works that tackle service selection issue in the IoT. In [3], an ANN algorithm was proposed to solve service selection in ubiquitous IoT. This algorithm takes into account contextual information as ANN inputs to predict a new QoS value of the service. The algorithm ensures the IoT challenge related to scalability. However, the service selection was performed based on just one QoS attribute and the user preferences were not considered. In [20], a flexible algorithm for service selection has been proposed. The particularity of this algorithm is that service requestors can express their preferences in an easy way because of the ontological reasoning that the authors have adopted to evaluate the subjective information of the user. Unfortunately, authors focused on the reliability of their algorithm for the service selection and neglected the fact that it can be time consuming since it involves a lot of computations. Jin et al. [14], proposed an algorithm for the selection of physical services in the IoT. This work focuses on just the selection of devices and cannot be generalized to all kinds of services in the IoT.

To the best of our knowledge, in the IoT, there is no work which take both context awareness and user preferences into consideration, in the selection method.

Table 1. Comparison of IoT service selection methods

	context-awareness	User preferences	Scalability
Bao et al., 2012 [3]	YES	NO	YES
NWE et al., 2014 [20]	NO	YES	NO
Jin et al., 2014 [14]	NO	YES	NO

4. PROPOSED SERVICE SELECTION METHOD FOR THE IOT

4.1 Problem formalization

As illustrated in figure 1, our approach is composed, mainly, of two phases : the prediction phase and the selection phase. First, in the prediction phase, an ANN predictive model is employed to estimate the QoS values of services. Thereby, it takes as inputs the contextual attributes of both users and services, those are extracted from the user request and from the description of the service in the registry. Moreover, since, there are several different QoS attributes, several instances of the ANN run simultaneously to forecast the values of each QoS. Second, in the selection phase, we adopt a

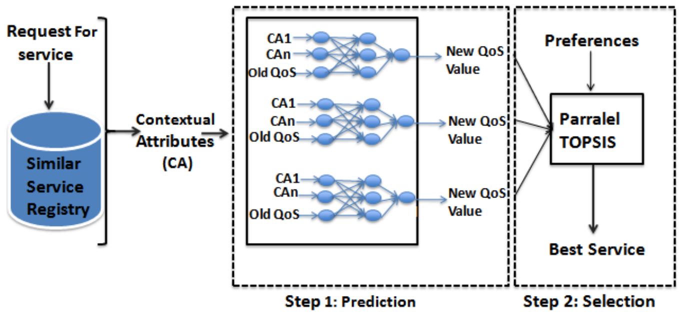

Fig. 1. Approach overview : the service selection model.

MCDM method TOPSIS with parallel implementation, that relies on the QoS predicted values in order to return the most suitable service for the requestors. It allows the service requestors to express their preferences regarding to QoS attributes.

4.2 Artificial Neural Network Adaptation

Since the Multi-Layer Perceptron Neural Network (MLPNN) is devoted to solve complex situations including prediction, we choose it to predict QoS values of services. In fact, it is the most used ANN type in the literature [9].

4.2.1 Training Set and Data Normalization. Contextual attributes for each QoS represent the networks training examples. These examples are collected, progressively, during the search process in the training set. Contextual information include user context, service context and network context. We denote by T the training set; it is composed of a set of couples which are composed of a sequence of inputs and its output. T can be represented as follows : $T = \{(X_1, a_1), (X_2, a_2), (X_3, a_3), ..., (X_n, a_n)\}$, where inputs $X_1, X_2, X_3, .., X_n$ are n-dimensional vectors (n is the number of QoS criteria) and outputs $a_1, a_2, a_3, .., a_n$ are the new QoS values which are real numbers. A single input vector is presented as follows (vector 1) :

$$X_i = \left(CA_1i \; CA_2i \; CA_3i \; ... \; CA_ni \; OldQoSValue_i \right) \tag{1}$$

where:
$CA_1, CA_2, CA_3, .., CA_n$ are contextuel attribut related to QoS criterion i.
$OldQoSValue_i$ is the old QoS values to QoS criterion i.

In order to normalize input data, we used the Min-max technique because it is among the most accurate existing techniques [13]. It is expressed by the formula 2 where input is the variable to normalize, Min and Max are, respectively, the minimum and the maximum of all the values in the training set.

$$MinMax = \frac{input - Min}{Max - Min} \tag{2}$$

4.2.2 Network Training. The training step take several training iterations, each training iteration needs two steps :

—Computing the gradient of the cost function.
—Adjusting the parameters of the gradient to minimize the error rate. We have opted for the backpropagation algorithm for the calculation of the cost function for QoS prediction since it is one of the most used learning algorithms [9].

4.3 Technique for Order Preference by Similarity to Ideal Solution Adaptation

After predicting all QoS values, it is necessary to find the most suitable service among functionally similar services to fulfil the user needs. According to a comparison study of MCDM [24], it was proved that TOPSIS is the most suitable for large scale alternatives and criteria. Moreover, it takes into account requestors preferences. Hence, TOPSIS is adopted; It assume that we have a decision matrix where, m is the number of services, n the number of QoS criteria and x_{ij} is the score of the service i with respect of the QoS criterion j.
According to [25], the steps of the TOPSIS method are the following:
Step1 : Construct the normalized decision matrix For this purpose, the above formula is applied to obtain new matrix inputs r_{ij}.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{M} x_{ij}^2}} \tag{3}$$

In this case, step 1 is neglected, the inputs for TOPSIS are the predicted QoS values generated by the neural network which are already between 0 and 1.
Step2 : Construct the weighted normalized decision matrix
A set of weights w_j for j=1,..,n is associated to criteria. They are defined by the decision maker, who is the service requestor in this case, to express his preferences. So, in this step, all inputs of the normalized matrix are multiplied by the weight associated to the criteria. We denote by V the new weighted normalized decision matrix and v_{ij} an element of it (cf. matrix 4).

$$V = \begin{bmatrix} v_{11} & v_{12} & v_{13} & \cdots & v_{1n} \\ v_{21} & v_{22} & v_{23} & \cdots & v_{2n} \\ v_{31} & v_{32} & v_{33} & \cdots & v_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ v_{m1} & v_{m2} & v_{m3} & \cdots & v_{mn} \end{bmatrix} \tag{4}$$

Step3 : Determine the ideal and the negative ideal solutions
Step3-a : for each criterion, we calculate the most favorable associated value according to the nature of the criterion (favorable or unfavorable). If the criterion is favorable, we choose the highest value of each column. If the criterion is unfavorable, we choose the lowest value of each column. For example a small response time is unfavorable and a high reliability is favorable.
Thus, the ideal solution is presented as follows :

$$A^* = \{v_1^*, v_2^*, ..., v_n^*\}, \; where \; v_j^* = \left\{ max_i(v_{ij}) \mid j\epsilon J; min_i(v_{ij}) \mid j\epsilon J' \right\}, \; for \; j = 1,...,n \tag{5}$$

$where:$
$J = \{j \in [1, n] | j$ is associated to a favorable criterion$\}$
$J' = \{j \in [1, n] | j$ is associated to a unfavorable criterion $\}$

Step3-b : for each criterion, we calculate the less favorable associated value according to the nature of the criterion (favorable or unfavorable). If the criterion is favorable, we choose the lowest value of each column. If the criterion is unfavorable, we choose the highest value of each column.

Thus, the negative-ideal solution is presented as follows :

$$\text{A}^- = \{v_1^-, v_2^-, ..., v_n^-\}\,,\ where\ v_j^- = \left\{min_i(v_{ij}) \mid j\epsilon J; max_i(v_{ij}) \mid j\epsilon J^{'}\right\},\ for\ j=1,...,n \tag{6}$$

Step4 : Calculate deviations from the ideal and negative-ideal solutions

Step4-a : For each service, we calculate the deviation from the ideal solution (evaluated by step3-a). Euclidean distance is applied for each service to calculate its deviation from the ideal service. The set of deviations is expressed by the formula 7 as follows:

$$S_i^* = \left(\sum_{j=1}^{n} \left(v_{ij} - v_j^*\right)^2\right)^{1/2}, i = 1, 2, 3, ..., m. \tag{7}$$

Step4-b : This step is analogous to the previous step, we just use the component of the vector A^- from the step3-b. Thus, the set of deviations from the negative-ideal solution is expressed by the formula 8 as follows:

$$S_i^- = \left(\sum_{j=1}^{n} \left(v_{ij} - v_j^-\right)^2\right)^{1/2}, i = 1, 2, 3, ..., m. \tag{8}$$

Step5 : Calculate the relative closeness to the ideal solution

This is the final step of TOPSIS, the relative closeness of a service A_i with respect to the ideal solution A^* is expressed by the formula 9 as follows:

$$C_i^* = \frac{S_i^-}{S_i^* + S_i^-}, 0 \le C_i^* \le 1, i = 1, 2, 3, ..., m. \tag{9}$$

Hence, the service that has the closest C_i^* to 1 is selected. Seemingly, C_i^* = 1, if A_i= A^*, and C_i^- = 0, if A_i= A^-.

5. EXPERIMENTAL STUDY

In order to evaluate the scalability of the proposed approach for service selection in the IoT, we conducted an experimentation based on parallel implementation. Hence, we used the Open Multi-Processing (OpenMP) framework which is a parallel programming framework dedicated to systems with shared memory [6]. According to a comparative study, OpenMP has showed a better result in terms of scalability than Message Massing Interface(MPI) [22] and MapReduce [8]. Moreover, authors of [15] concluded that, this latter is a good option, when the problem requires intensive computation and the amount of data is moderate.

5.1 Implementation

In order to test the two algorithms, we have developed programs in Java. The outputs of the programs are basically :

—The QoS values predicted from the ANN. We consider two QoS attributes : Response time and Reliability.

—The best service selected after analyzing QoS values by TOPSIS.

—The execution time spent in the construction of neural network and the selection of the best service in order to evaluate the scalability of the proposed method.

For the neural network, we ran parallel instances of it with different data sets. A single instance is not able to approximate multiple QoS criteria. In fact, each QoS attribute is influenced by a specific set of contextual attributes.

Response Time : is the necessary time for a service to process and complete a service request [1]. The neural network mathematical model for the response time is as follows :

$$\text{QoSRT}\ (t + \Delta t) = f[PS\,(t)\,, MS\,(t)\,, NL\,(t)\,, d\,(t)\,, Cnn\,(t)\,, oldQoSRT\,(t)] \tag{10}$$

—Δt : refers to the sampling period. We set it to 1s.

—$QoSRT\,(t + \Delta t)$: refers to the estimated value of the response time denoted by QoSRT through the neural network. It is the result of the output node.

—$PS\,(t)$: refers to the current processor speed of the devise that offers the service.

—$MS\,(t)$: refers to the current memory size of the devise that offers the service.

—$NL\,(t)$: refers to the current network load.

—$d\,(t)$: refers to the propagation delay between the location of the service's requestor and the location of the service provider.

—$Cnn\,(t)$: refers to the current connection rate.

—$oldQoSRT\,(t)$: refers to the previous estimated response time value.

Reliability : reflects the capacity of a service during a period of time to accomplish its expected function under the stated conditions [1]. The neural network mathematical model for reliability is as follows :

$$\text{QoSR}\ (t + \Delta t) = f[NL\,(t)\,, TR\,(t)\,, Cnn\,(t)\,, BL\,(t)\,, oldQoSR\,(t)] \tag{11}$$

—Δt : refers to the sampling period. We set it to 1s.

—$QoSR\,(t + \Delta t)$: refers to the estimated value of the reliability denoted by QoSR through the the neural network. It is the result of the output node.

—$NL\,(t)$: refers to the current network load.

—$TR\,(t)$: refers to the current Time for Repair which represents the time required to repair a service that has failed.

—$BL\,(t)$: refers to the current battery level of the user device.

—$Cnn\,(t)$: refers to the current connection rate.

—$oldQoSR\,(t)$: refers to the previous estimated reliability value.

In order to optimize the ANN implementation, we used the framework Encog [12]. In fact, this latter is a practical tool in java that allows to create a neural network for artificial intelligence applications. The required parameters for the ANN are shown in the table 2. Two termination criteria are used. If the error rate measured between the test set and examples in, the training set is lower than 0.05 then the training is stopped else, 1000 iterations are performed. The tuning of the learning and the momentum parameters had to ensure the rapidity of regularizing weights while escaping local minima.

Table 2. MLPNN: parameters settings

Parameters	MLPNN
Maximum number of iterations	1000
Error rate	0.005
Number of hidden neurons	10
Momentum	0.9
Learning rate	0.05

5.2 Experimental results

the algorithm is applied to a training set that we have built through our research step. We did not opt for standard benchmarks since, they do not satisfy requirements of the IoT.

5.2.1 ANN Prediction performance. First of all, the data set is divided into training data (80%) and testing data (20%). For 100 experimental data, the first 80 present the training ones and the remaining 20 present the test ones. Response time (QoSRT) data and the Reliability (QoSR) are predicted by executing simultaneously two ANN instances. The prediction results for the test set for QoSRT and QoSR for each service are shown respectively in the figures 2 and 3.

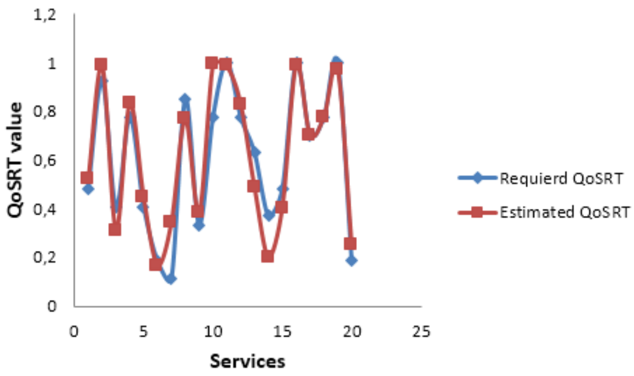

Fig. 2. Comparison between predicted QoSRT values and estimated QoSRT values.

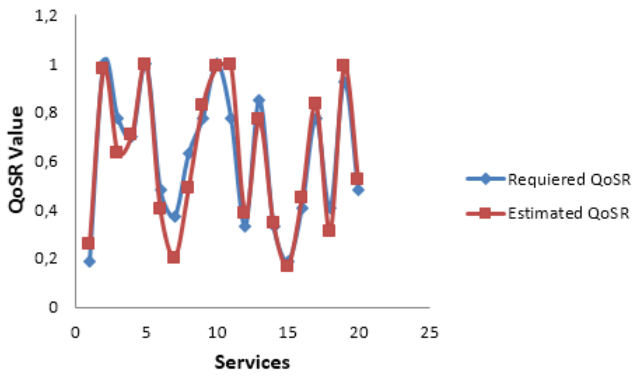

Fig. 3. Comparison between required QoSRT values and estimated QoSRT values.

According to the two above figures, we observe that, there is not a major difference between the required and the estimated values. Nevertheless, this is not enough to make conclusions about the quality of the prediction model. Hence, we used the Mean Absolute Error (MAE) as a mathematical equation defined as follows :

Table 3. Performance of ANN algorithm on various number of services

Number of available services	ANN Execution time (s)
20	0.2
200	0.25
2000	0.32

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - y'_i| \quad (12)$$

n represents the number of predictions, y'_i represents the predicted value, y_i represents the required value. The smaller the MAE value is, the higher is the accuracy of the prediction model.
To highlight the quality of the prediction , we run for each ANN instance 3 tests with different data set sizes. Results for the testing data sets are shown as histograms in figure 4:

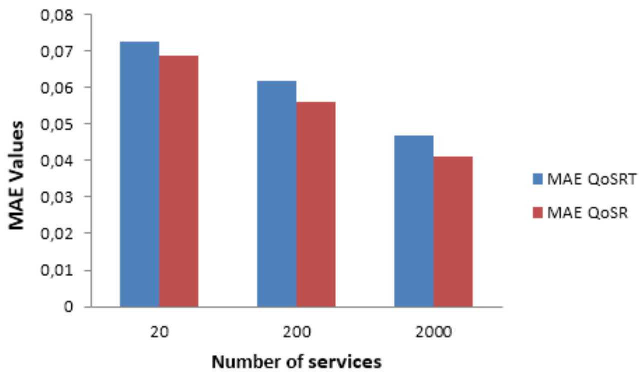

Fig. 4. Accuracy performance on various number of services.

We note that for both cases, the MAE values are inversely proportional to the number of services. The more we increase the number of services , the less are the MAE values and thus the more the accuracy increases. As a result, the quality of predictions is improved with large data.
The execution time to make predictions for ANN for both instances, does not significantly arise as shown in the table 3.

5.2.2 Parallel TOPSIS performance. To evaluate parallel TOPSIS with OpenMP framework, we compared its performance in terms of execution time with no parallel TOPSIS. The results are shown in the figure 5.

Fig. 5. Comparison of TOPSIS and parallel TOPSIS.

By increasing the number of services, we notice that the execution time for parallel TOPSIS increases slightly. However the execution time for no parallel TOPSIS is increasing exponentially.
Scalability test: In order to evaluate our algorithm, we perform a test of scalability which refers to the ability of the proposed algorithm to keep a good performance when the number of services increases. Therefore, we compare the execution time of both ANN and TOPSIS as shown in the table 4.

Table 4. Our algorithm performance on various number of services

Number of available services	Execution time (s)
20	2
200	2.1
2000	2.9

By increasing the number of services, we notice that the execution time does not significantly increase.
To the best of our knowldge this is the first work that combines MLPNN and TOPSIS.

6. CONCLUSION

this paper presents a novel essay to tackle some issues related to service selection. We are interested, in the integration of QoS attributes, contextual information and user preferences in order to deliver the most suitable service for the requestor. The use of both ANN and TOPSIS based on parallel computing can cope with time consuming issue. Hence, the scalability is improved. Moreover, the learning aspect of our approach, is of great importance since it enables automatic recognition from past results and situations. Hence, our method is self-adaptive. Finally, the proposed method is able to support all QoS features' types , generic as well as domain specific attributes. For Future research, we aim to tackle the problem of composite service selection with the use of evolutionary techniques and pruning methods to minimize searching space and select a suitable composite service efficiently.

7. REFERENCES

[1] Zainab Aljazzaf. Bootstrapping quality of web services. *Journal of King Saud University-Computer and Information Sciences*, 27(3):323–333, 2015.

[2] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer Networks*, 54:2787–2805, 2010.

[3] Jianmin Bao, Yiming Ding, and Haifeng Hu. A new service selection algorithm in uspiot. In *Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on*, volume 2, pages 22–26. IEEE, 2012.

[4] Harun Baraki, Diana Comes, and Kurt Geihs. Context-aware prediction of qos and qoe properties for web services. In *Networked Systems (NetSys), 2013 Conference on*, pages 102–109, 2013.

[5] Peter Paul Beran, Elisabeth Vinek, Erich Schikuta, and Maria Leitner. An adaptive heuristic approach to service selection problems in dynamic distributed systems. *2012 ACM/IEEE 13th International Conference on Grid Computing*, pages 66–75, 2012.

[6] Rohit Chandra. *Parallel programming in OpenMP*. Morgan Kaufmann, 2001.

[7] Alfonso Garcia de Prado and Guadalupe Ortiz. Context-aware services: A survey on current proposals. *The Third International Conferences on Advanced Service Computing*, pages 104–109, 2011.

[8] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[9] Gérard Dreyfus. *Neural Networks*. Springer Science, Paris, France, 2005.

[10] Randall Farmer and Bryce Glass. *Building Web Reputation Systems*. O'Reilly Media, Inc, 2010.

[11] Zhengdong Gao and Gengfeng Wu. Combining qos-based service selection with performance prediction. In *e-Business Engineering, 2005. ICEBE 2005. IEEE International Conference on*, pages 611–614. IEEE, 2005.

[12] Jeff Heaton. Encog: Library of interchangeable machine learning models for java and c#. *arXiv preprint arXiv:1506.04776*, 2015.

[13] T Jayalakshmi and A Santhakumaran. Statistical normalization and back propagation for classification. *International Journal of Computer Theory and Engineering*, 3(1):1793–8201, 2011.

[14] Xiongnan Jin, Sejin Chun, Jooik Jung, and Kyong-Ho Lee. Iot selection based on physical service model and absolute dominance relationship. In *Service-Oriented Computing and Applications (SOCA), 2014 IEEE 7th International Conference on*, pages 65–72. IEEE, 2014.

[15] Sol Ji Kang, Sang Yeon Lee, and Keon Myung Lee. Performance comparison of openmp, mpi, and mapreduce in practical problems. *Advances in Multimedia*, 2014.

[16] KangChan Lee, JongHong Jeon, WonSeok Lee, Seong-Ho Jeong, and Sang-Won Park. QoS for Web Services: Requirements and Possible Approaches.World Wide Web Consortium (W3C) note. `http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/`, 2008. Accessed on September 27, 2015.

[17] Umardand Shripad Manikrao and T.V.Prabhakar. Dynamic selection ofweb services with recommendation system. *Proceedings - International Conference on Next Generation Web Services Practices, NWeSP*, pages 117–121, 2005.

[18] E. Michael Maximilien and Munindar P. Singh. Conceptual model of web service reputation. *ACM SIGMOD Record*, 31:36, 2002.

[19] M.Swarnamugi. Taxonomy of web service selection approaches. *International Conference on Computing and information Technology (IC2IT-2013)*, pages 18–22, 2013.

[20] Nwe Htay Win NWE, Jian-min BAO, and CUI Gang. Flexible user-centric service selection algorithm for internet of things services. *The Journal of China Universities of Posts and Telecommunications*, 21:64–70, 2014.

[21] Maryam Saeedi, Zeqian Shen, and Neel Sundaresan. The value of feedback:an analysis of reputation system. pages 1–41, 2013.

[22] Marc Snir. *MPI–the Complete Reference: The MPI core*, volume 1. MIT press, 1998.

[23] Lu Tan and Neng Wang. Future internet: The internet of things. *3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, 5:376–380, 2010.

[24] Jureen Thor, Siew-Hong Ding, and Shahrul Kamaruddin. Comparison of multi criteria decision making methods from the maintenance alternative selection perspective. *The International Journal of Engineering And Science (IJES)*, 2(6):27–34, 2013.

[25] Evangelos Triantaphyllou, B Shu, S Nieto Sanchez, and Tony Ray. Multi-criteria decision making: an operations research approach. *Encyclopedia of electrical and electronics engineering*, 15:175–186, 1998.

[26] Le-Hung Vu, Manfred Hauswirth, and Karl Aberer. Qos-based service selection and ranking with trust and reputation management. In *On the Move to Meaningful Internet Systems*

2005: CoopIS, DOA, and ODBASE, pages 466–483. Springer, 2005.

[27] Yao Wang and Julita Vassileva. Toward trust and reputation based web service selection: A survey. *International Transactions on Systems Science and Applications (ITSSA)*, 2007.

[28] Qihui Wu, Guoru Ding, Yuhua Xu, Shuo Feng, Zhiyong Du, Jinlong Wang, and Keping Long. Cognitive internet of things: A new paradigm beyond connection. *Internet of Things Journal, IEEE*, 1:129–143, 2014.

[29] Hong Qing Yu and Stephan Reiff-Marganiec. Non-functional property based service selection: A survey and classification of approaches. *In: Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop co-located with The 6th IEEE European Conference on Web Services, 12 - 14 Nov 2008, Ireland, Dublin*, 2008.