

An Efficient Approach for Realizing RCCAR to Resolve Context Conflicts in Context-Aware Systems

Asma Abdulghani Al- Shargabi

Software Technology Research Laboratory (STRL)
Faculty of Technology
De Montfort University
Leicester, UK

Sponsored by: University of Science and Technology, Yemen
a.alshargabi@gmail.com

Francois Siewe

Software Technology Research Laboratory (STRL)
Faculty of Technology
De Montfort University
Leicester, UK
fsiewe@dmu.ac.uk

Abstract- The first step for determining the quality of context (QoC) in context aware systems (CASs) is to make sure that there are no conflicts among the values of each context element collected from different resources/sensors. This work is an extension of our approach RCCAR (Resolving Context Conflicts Using Association Rules) which proposes a solution to resolve context conflicts by exploiting the previous context to predict the valid values between conflicted ones using Association Rules (AR) technique. The contributions of this paper are twofold: firstly, an algorithm is proposed for implementing the RCCAR approach; secondly, a novel solution that improves the efficiency of RCCAR is proposed. This solution uses the decision tree technique to compute the most influential context elements before applying RCCAR. A number of experiments were conducted using the Weka tool. Results show that the enhanced RCCAR is more efficient than the original RCCAR.

Keywords- Context-Aware System (CAS); Context Conflicts; Quality of Context (QoC); Association Rules (AR); Decision Trees (DT).

1. INTRODUCTION

Ubiquitous computing is still way from Mark Weiser vision. Context-aware systems (CASs) which are a vital part of this environment face many challenges to keep high performance. A context conflict is one of these challenges which affect the quality of context (QoC) and consequently the services provided by CASs. CASs expect that context data are correct and reliable. But, what if they are not? QoC [1][2] is a precondition for a correct behavior of CASs. Context imperfection aspects have been addressed by many researches [3][4][5][6][7][18][11]. One of the important context imperfection aspects is the context conflicts.

Context conflicts reflect the contradictions within the context data [13][19]. This can be due to different reasons; conflicts may occur while collecting data from redundant context sources or while aggregating those data to compose the whole context. These conflicts could affect the produced decisions and consequently lead to undesirable actions. This situation could be serious if the CAS is critical. Resolving context conflict means selecting the valid value between some conflicted ones. Most researches tried to resolve conflicts according to QoC parameters [10][13][18][14][19]. This means that the context value which has the better QoC parameters values should be selected. QoC parameters are the parameters which reflect the level of context quality such as correctness, trust-worthiness, resolution, and up-to-dateness.

QoC parameters have been addressed by different approaches [1][2][3][4][7][8][9][12].

This paper is an improvement of our previous work presented in [22] which proposed RCCAR (Resolving Context Conflicts Using Association Rules). RCCAR resolves context conflicts by exploiting the previous history of context to predict the valid values from different conflicted ones. The technique which is used to predict the valid value is the Association Rules (AR) technique. Based on AR, RCCAR calculates the total affirmation for each conflicted value as a function of other context elements. Then, the value that has the greater affirmation is selected among the conflicted values. The contributions of this paper are twofold: firstly, an algorithm is proposed for implementing the RCCAR approach; secondly, a novel solution that improves the efficiency of RCCAR is proposed. This solution uses the decision tree technique to compute the most influential context elements before applying RCCAR. A number of experiments was conducted using the Weka tool. Results show that the enhanced RCCAR is more efficient than the original RCCAR.

The rest of this paper is as follows: Section 2 is devoted to related work. Section 3 gives an overview of RCCAR. Section 4 presents the proposed algorithm for realizing RCCAR and its computational complexity. Section 5 introduces our novel approach for enhancing the efficiency of RCCAR using decision trees. Experiments design and implementation are described in Section 6 and the results discussed in Section 7. Finally, Section 8 concludes the paper and points to future work.

2. RELATED WORK

Many literatures have addressed context conflict resolving. These researches proposed different solutions for conflicts resolving [10][13][18][14][19]. Most of these solutions have adopted context quality parameters as a basis for selecting the higher quality context element among some conflicted ones. Thus the context element which has the better quality parameters values will be selected. Some researches adopt the average of all these quality parameters to calculate the quality level of each context element while others have used just one of them. A significant addition was proposed by [14] and [19], where what called a "quality policy" has been proposed to make context conflicts resolving more flexible based on the idea that each CAS needs different context quality parameters and these parameters should be used for conflicts

resolving according to the nature of this system and also the nature of the context. A significant solution was proposed by [10], [13], and [18] where a focusing on the valid values between the conflicted context element values was indicated. This approach has adopted just two quality parameters for context conflict resolving, these parameters called the correctness and trustworthiness. Correctness parameter is used for context attributes (elements), however for higher level context; this approach adopted the trustworthiness level of the context provider for dealing with conflicts with this type of context. This solution has used the Bayesian theory to calculate the probability of correctness and then trustworthiness based on the confirmation of context elements which coming from the previous occurrences of these values under investigation. The confirmation of each context elements values for the context elements value under investigation is calculated individually and then the total confirmation is simply calculated. This paper is update of our approach which called RCCAR was introduced in [22]. In contrast of what introduced previously this approach use the different combination of context elements to calculate the affirmation for such context element. This means that we can get the total affirmation of the current context using both affirmations from each context element individually, and also by affirmation from different combinations between these context elements according to previous history using technique. This solution is proposed because there are cases in reality where each context element could confirm the context element individually but not collectively. More details for RCCAR approach is introduced in section 3.

3. AN OVERVIEW OF RCCAR

Our approach RCCAR (Resolving Context Conflicts Using Association Rules) is based on exploiting the pervious history of a context for predicting which among the context conflicted values are valid and which is not. The prediction uses Association Rules (AR) technique to get all associations which combine context elements together to get the affirmation from each association individually and then get the total affirmation by getting the summation of affirmations associated with each association. AR is a technique which is commonly used in data mining application for discovering patterns in a huge historical database or data warehousing. AR definitely discovers what goes together in data based on data occurrences in database. Thus, we found AR is an appropriate technique to get associations which affirm the different values of investigated context element and then deciding according to the affirmation values which is valid from those conflicted values. Expression 1 clarify the formula of association rule stating that the occurrence of y context element affirms the occurrence of x [20][25].

$$y \Rightarrow x \quad (1)$$

Where y could be one context element and also could be a combination of some context elements, and x is the context element under investigation. Now, how affirmation for each association rule is produced. As clarified above, RCCAR uses the association rules measure which called "confidence". It is calculated as stated by Equation 2:

$$\text{Confidence } (Y \Rightarrow X) = \frac{\text{last-occurrences of } x,y \text{ together}}{\text{last-occurrences of } y} \quad (2)$$

Confidence reflects the affirmation strength. To compute the **Total – affirmation** of context element x , the summation of the confidence for all possible associations which affirm the context element under investigation x is calculated by scanning the database of previous history. Equation 3 is used to calculate **Total – affirmation** as follows:

$$\text{Total – affirmation}(x) = \sum_{i=1}^m \text{confidence}(y_i \Rightarrow x) \quad (3)$$

Where m is the number of available possible associations which could be produced according to the different context elements, and also according to the occurrences in database. To clarify that, we will start by introducing the "itemset" concept. According to association rules analysis, a collection of zero or more elements is referred to by *itemset*. If *itemset* contains k items, it is called as k – itemset. Examples of *itemset* are shown in TABLE I.

TABLE I. EXAMPLES OF ITEMSETS

1-itemset			2-itemset			3-itemset		
id	elements	σ	id	elements	σ	id	elements	σ
$y_1(1)$	a	13	$y_1(2)$	a c	8	$y_1(3)$	a c d	4
$y_2(1)$	b	9	$y_2(2)$	a d	7			
$y_3(1)$	c	10	$y_3(2)$	b c	5			
$y_4(1)$	d	15	$y_4(2)$	d c	5			

Where a, b, c, and d are context elements. The frequency (occurrences) of context element individually and all possible combinations of them are calculated by scanning database. By scanning the database we get all possible associations and its confidence values and then simply sum them. We apply that to all conflicted values and select the context element that has the greater **Total – affirmation** value. TABLE II clarifies an example for that. Assume TABLE I contains the occurrences for some context elements. These values are concluded by scanning the context database. Assume that a and b are two conflicted values for a context element. In addition to a and b, the current situation of context is represented by other context elements c and d. According to occurrences in TABLE I, the associations illustrated in TABLE II will be produced.

TABLE II. PRODUCED ASSOCIATIONS

Associations affirm a	Confidence value	Associations affirm b	Confidence value
$c \Rightarrow a$	$\frac{8}{10} = 0.80$	$c \Rightarrow b$	$\frac{5}{10} = 0.50$
$d \Rightarrow a$	$\frac{7}{15} = 0.46$		
$c, d \Rightarrow a$	$\frac{4}{5} = 0.80$		
Total – affirmation(a)	2.06	Total – affirmation(b)	0.50

According to TABLE II, total-affirmation of a is greater than the total affirmation of b . The table shows that number of associations and the total value of their confidence indicate that a is the recommended value of context element under investigation.

RCCAR then examined according different conditions. Results showed that RCCAR has succeeded against different conditions. Using all possible combinations of context

elements in AR improve the prediction. Utilizing all context elements recorded in the previous history also improves the result but in varying degrees. More explanations and details could be found in [22].

4. RCCAR REALIZING ALGORITHM

This section is devoted for the designing of the algorithm that realizes RCCAR which is a new extension work to RCCAR approach. Then, an analysis of algorithm complexity will be introduced. This analysis is important in order to estimate the applicability of the solution for context-aware systems. To deduce the algorithm, we assumed a certain situation and then moving to generalize the algorithm. The certain situation assumes the existence of a current context for 5 elements and a previous history of 100 records. Because of the minimum level of combinations is 2, possible combinations will have 2, 3, 4 and 5 combination types. The investigated context element which has the conflicted values will be the first context element in context vector; where the associations should relate the investigated context element with other different combinations of context elements. Reordering the context vector in order to make the investigated element as the first one before starting the algorithm could be considered within the preprocessing phase. To clarify the main idea of the algorithm, we start with general steps of RCCAR where Algorithm 1 describes the main steps of resolving conflicts using AR technique.

Algorithm1: The general steps for resolving context conflicts using association rules (RCCAR)

INPUT: current context vector, previous context database, different conflicted values for investigated element

- 1: Receive current context values with conflicted ones for investigated element
- 2: Calculate the occurrences for all possible required combinations using previous-context-database
- 3: Calculate the total-affirmation for each conflicted value of investigated-context-element using Equation 7
- 4: Return the investigated context element value which has the greater total-affirmation

Details which describe how to get possible combinations for context elements are introduced by Algorithm 2.

Algorithm2: Getting all possible associations according to RCCAR

INPUT: current context vector current_ctx[5], previous-context[100][5], investigated-element-conflicted-values[3]

```

1: for (c=1 to 100)
2:   { /* the beginning of scanning the previous context elements*/
3:   for (i=2 to 5) /*getting the occurrences for Association-of-type-2*/
4:   {
5:     for (p=1 to 3)
6:       if previous-context [c][1]== investigated-element conflicted values[p] and
          previous-context[c][i]==current-context[i] then association2-
          occurrences[p][i]++
7:     end if
8:     if previous-context[c][i]==current-context[i] then
          association2-occurrences[4][i]++
9:     end if
10:  }
11: for (i=2 to 4) /*getting the occurrences for Association-of-type-3*/
12: for (j=i+1 to 5)
13: {
14:   for (p=1 to 3)
15:   if previous-context [c][1]== investigated-element-conflicted-values[p] and
      previous-context[c][i]==current-context[i] and
      previous-context[c][j]==current-context[j] then

```

```

          association3-occurrences[p][i][j]++
16: end if
17:   if previous-context[c][i]==current-context[i] and previous-
      context[c][j]==current-context[j] then association3-
      occurrences[4][i][j]++
18:   end if
19: }
20: for (i=2 to 3) /*getting the occurrences for Association-of-type-4*/
21: for (j=i+1 to 4)
22:   for (k=j+1 to 5)
23:   {
24:     for (p=1 to 3)
25:     if previous-context [c][1]== investigated-element-conflicted-values[p] and
        previous-context[c][i]==current-context[i] and previous-
        context[c][j]==current-context[j] and previous-
        context[c][k]==current-context[k] then association4-
        occurrences[p][i][j][k]++
26:     end if
27:     if previous-context[c][i]==current-context[i] and
        previous-context[c][j]==current-context[j] and
        previous-context[c][k]==current-context[k] then
        association4-occurrences[4][i][j][k]++
28:     end if
29:   }
30: for (i=2 to 2) /*getting the occurrences for Association-of-type-5*/
31: for (j=i+1 to 3)
32:   for (k=j+1 to 4)
33:   for (l=k+1 to 5)
34:   {
35:     for (p=1 to 3)
36:     if previous-context [c][1]== investigated- element-conflicted values[p]
        and previous-context[c][i]==current-context[i] and previous-
        context[c][j]==current-context[j] and previous-
        context[c][k]==current-context[k] and previous-
        context[c][l]==current-context[l] then
        association5-occurrences[p][i][j][k][l]++
37:     end if
38:     if previous-context[c][i]==current-context[i]and previous
        context[c][j]==current-context[j] and
        previous-context[c][k]==current-context[k] and previous-
        context[c][l]==current-context[l] then
        association5-occurrences[4][i][j][k][l]++
39:     end if
40:   }
41: } /* the end of scanning previous context database*/

```

An analysis of the efficiency of the proposed algorithm is very important as CASs needs fast response. The time complexity of the algorithm is affected by the nested loops used for either generating the association, or calculating the confidence or the total-affirmation for conflicted values where all them use the same loops. TABLE III summarizes the time complexity for such nested loops and deduces the final complexity of the algorithm where c is the number of records, N is the number of context elements and p is the number of conflicted values for investigated element.

TABLE III. TIME COMPLEXITY ANALYSIS

Association type	code	Time complexity	Complexity using problem variables
2	for (c=1 to 100) for (i=2 to 5) for (p=1 to 4)	$100 \times 4 \times 4$ $c \times i \times p$	$O(t \times (N-1) \times (p+1))=$ $O(c \times N \times p)=$ $O(p)$
3	for (c=1 to 100) for (i=2 to 4) for (j=i+1 to 5) for (p=1 to 4)	$100 \times 3 \times 3 \times 4$ $c \times i \times j \times p$	$O(c \times (N-2)^2 \times (m+1))=$ $O(c \times (N - (N-3))^{(N-3)} \times$ $(p+1))=$ $O(c \times (3)^{(N-3)} \times p)=$ $O(c \times 3^{(N-3)} \times p)$
4	for (c=1 to 100) for (i=2 to 3) for (j=i+1 to 4) for (K=j+1 to 5) for (p=1 to 4)	$100 \times 2 \times 2 \times 2 \times 4$ $c \times i \times j \times k \times p$	$O(c \times (N-3)^3 \times (p+1))=$ $O(c \times (N - (N-2))^{(N-2)} \times$ $(p+1))=$ $O(c \times (2)^{(N-2)} \times p)=$ $O(c \times 2^{(N-2)} \times p)=$
5	for (c=1 to 100) for (i=2 to 2) for (j=i+1 to 3) for (K=j+1 to 4) for (l=k+1 to 5) for (p=1 to 4)	$100 \times 1 \times 1 \times 1 \times 1 \times 4$ $c \times i \times j \times k \times l \times p$	$O(c \times 1^4 \times (p+1))=$ $O(c \times (N - (N-1))^{(N-1)} \times$ $(p+1))=$ $O(c \times 1^{(N-1)} \times p)=$ $O(c \times p)=$
The Overall Complexity = $O(c \times 2^{(N-2)} \times m)$			

At first glance, it is clear that the big Oh function reflects a high computation complexity. The number of context elements, size window of previous history (number of records), and conflicted values for the investigated element will affect the complexity exponentially. However, number of context elements we have surveyed during our experiments for different available datasets almost did not exceed 10. Also, the number of conflicted values is limited to the number of sensors assigned to each context element context; and this number usually does not exceed 5. Therefore, the number of a previous record which contributes in prediction should be determined carefully to reduce the complexity of the algorithm computation. Suppose that there are 10 context elements, 100 previous records and 3 conflicted values. The number of steps would be 76800 (according to $O(c \times 2^{(N-2)} \times m)$). Times on a 1-billion-steps-per second computer will be 0.0000768 sec which means 0.005 mille-seconds and 0.002 mille-seconds on some modern computers which process 3-billion-steps-per-second and this is an accepted value. Thus, the complexity of the algorithm generally is fine; however the following section introduce an enhancement of this complexity.

5. RCCAR EFFICIENCY ENHANCEMENT USING THE MOST INFLUENTIAL CONTEXT ELEMENTS

The idea which is the basis for reducing the time complexity of RCCAR algorithm is that the different context elements have not equal proportion of impact on the context element under investigation. We have to know which from these elements have the most impact on the values of the context element under investigation before applying RCCAR. Selecting the most influential context elements and not all context elements in prediction will reduce the time complexity and improve the performance of RCCAR. Suppose that there are 10 context elements (N), 100 previous records (c) and 3 conflicted values (m). Using the original RCCAR, the number of steps would be 76800 (according to $O(c \times 2^{(N-2)} \times m)$) while it will be reduced to be 600 with enhanced RCCAR. Times on a 1-billion-steps-per second computer will be reduced from 0.0000768 sec to 0.0000006 sec. That means reducing the time from 0.005 mille-seconds and 0.002 mille-seconds on some modern computers which process 3-billion-steps-per-second to 0.00003 mille-seconds and 0.000006 with modern computers. TABLE IV and Figure 1 show that time will be reduced exponentially by reducing the number of context element which contributes in prediction. According to this result, it is worthy to explore which context elements between all context elements have the most impact on the investigated context element; where that will be implemented just one time according the previous history.

TABLE IV. REDUCING TIME COMPLEXITY USING ENHANCED RCCAR

No. of context elements	10	9	8	7	6	5	4	3	2
Time complexity of RCCAR (mille-sec)	0.002	0.0008	0.0004	0.0002	0.0001	0.00005	0.00002	0.00001	0.000006

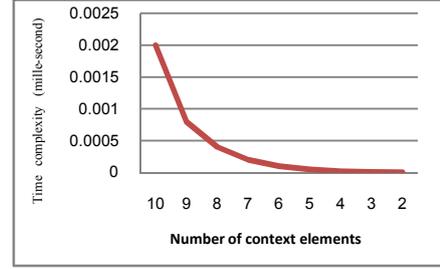


Figure 1. Reducing time complexity using enhanced RCCAR

Decision Trees (DT) technique has been used to offer the prior knowledge of the context elements which have the greater impact on the context elements under investigation. DTs are used successfully in many diverse areas such as classification, character recognition, medical diagnosis, expert systems, and speech recognition. DTs are traditionally drawn with the root at the top and leaves at the bottom. The attribute which has the greatest impact for the target attribute will be the root. Many different leaves may make the same classification, but each leaf makes that for a different reason. A decision-tree technique will start by classifying the records based on the value of a single attribute and then at the second level another attribute will be selected in order to proceed with the split and so on. There are different algorithms used for producing the decision tree depending on different statistic measures to select the root and other test nodes toward the leaves. ID3, C4.5, CART and CHAID are examples of these algorithms [21][23][24][25]. For our experiments, it is never mind which algorithm will be used; all of them will produce the required tree.

6. IMPLEMENTATION AND EXPERIMENTS

This section is devoted for addressing how is the enhanced RCCAR have been implemented and examined according different conditions. In addition, the preprocessing operations are presented and also the used tool is introduced. To examine the enhanced approach, different test cases were designed. A summary of factors forming these test cases are shown in TABLE V. According to these factors, many experiments were implemented.

TABLE V. DIFFERENT FACTORS FORMING THE TEST CASES

The Context Elements Which Contribute in Prediction	The Investigated Context Element	
	Valid Value	Invalid Value
all context elements	different	far from valid value
all influential elements	investigated context element	close from valid value
most influential elements (60% of impact)		

The dataset which was used for experiments is Southampton monthly weather historical data from the year 1855 to 2000. This data is officially collected and recorded by Southampton Weather station and published by its website [16]. This data set contains 1744 instance. The variables are: the year (YYYY), the month (MM), max temperature (TMax), min temperature (TMin), air frost (AF), rainfall (Rain), and sunshine hours (Sun). Weka product has been used to implement different experiments. It is selected as it supports solutions uses data mining techniques such as AR and DT. In addition, Weka provides a wide range of transformations to help researchers processing data before applying different

techniques [17]. In our experiments we used a version of C4.5 algorithm called J4.8 which is provided by Weka tool which we have been used for implementation.

7. RESULTS AND DISCUSSION

The experiments were divided into two stages; the first one was devoted for using the DT for knowing the influential variables for the investigated context elements, and the second one was devoted for implementing the enhanced RCCAR solution. This section introduces some examples which summarize the main results for different experiments. Then, the main remarks and recommendations according that is introduced. At first, all experiments results shows the success of the enhanced approach in prediction as there is a difference between the affirmation for the valid and invalid values of the investigated context element (Figure 2). Secondly, most results show a small difference between the original RCCAR and the enhanced RCCAR. Often this difference was an added value for enhanced RCCAR (Figure 2). Figure 3 and Figure 4 show the prediction for far/close-invalid values respectively. On the other hand, we can note that the shape of the result is very similar for both approaches and this indicates how much the enhanced approach is representative.

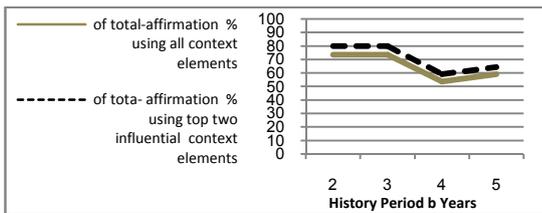


Figure 2. Results of prediction for the valid value using the original RCCAR and the enhanced RCCAR

Figure 5 shows a very important result where it shows the proportion of the most important context elements affirmation from the total affirmation using all variables for 4 examples through five years. As expected, a vital proportion of the total affirmation was due to these influential variables. This result was noted for most experiments. This result supports the success of our enhanced approach.

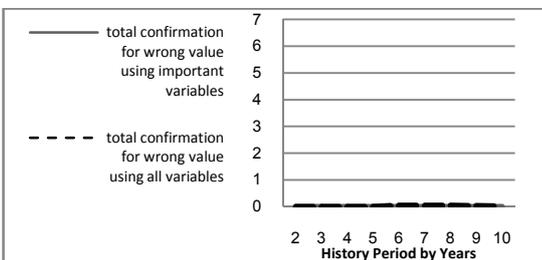


Figure 3. Results of prediction for far invalid value using the original RCCAR and the enhanced RCCAR

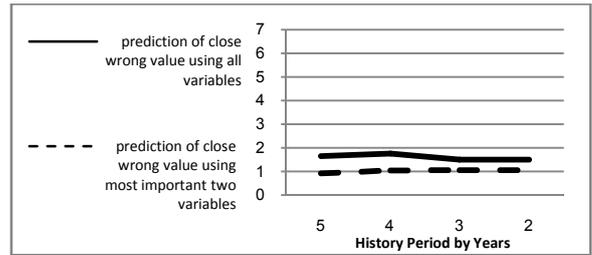


Figure 4. Results of prediction for the invalid close and far values using the enhanced RCCAR

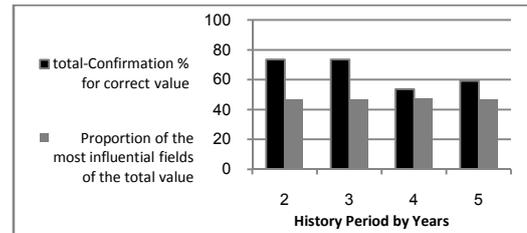


Figure 5. The proportion of the influential variables of the total affirmation using all variables

The following result is due to an example shows clearly the difference between original RCCAR and enhanced RCCAR successes and cost. According to produced association rules, the total affirmation of associations and also the cost using original and enhanced RCCAR is summarized in TABLE VI.

TABLE VI. THE COST USING RCCAR AND ENHANCED RCCAR

Context Elements Contribute in Prediction	The Maximum Total-Affirmation (No. of Associations)	The Result Total-Affirmation	% for Total-Affirmation	The Computational Cost	The Computation Time (mille-second)
				For Each Conflicted Value	
All Elements	31	18.09	58.35	18208	0.0004
All Influential Elements	7	4.1	58.57	4552	0.00009
Top Two Influential Variables	3	1.74	58	2276	0.00005

The result of this experiment was remarkable as there was almost no difference in prediction but the cost is largely reduced using enhanced RCCAR with just two variables although the recommended according to DT was three; adding the third one enhanced the prediction but not with a large percentage.

Finally, the main remarks and recommendations based on the overall experiments are as follows: (1) participating of the largest possible number of fields in classification before resolving context conflicts provides a better opportunity to explore and identify the most influential context elements accurately, (2) using long history as possible is preferred for classification when determining the influential context elements; where generalizing of results will be more valid, (3) in many experiments a remarkable results showed that there are many context elements do not affect the investigated field, (4) adding some context elements increases exponentially the cost of the prediction and similarly excluding them improve performance significantly, (5) not all

influential context elements are important for prediction; we can just use the elements which have together 60% of impact.

8. CONCLUSIONS

This paper introduces an enhancement for RCCAR (Resolving Context Conflicts using Association Rules) approach which is proposed in our previous work [22]. RCCAR works for resolving context conflicts using AR technique to discover the affirmation of previous context for different conflicted values. This paper presents the algorithm of RCCAR. Then, it introduced a novel solution for improving the efficiency of RCCAR algorithm by reducing the computational complexity of the RCCAR algorithm. This solution is based on the idea of reducing the number of context elements which affirm the conflicted values of context element under investigation. The solution adopts the prior knowledge of the context elements which have the greatest influence on the context element under investigation. This prior knowledge is achieved by exploiting the previous history of context for a long time. The technique which applied for getting this knowledge is decision trees. The enhanced RCCAR have achieved a success in resolving context conflicts with decreasing the computational cost compared to the original RCCAR. For future work, more experiments using different types of datasets is recommended. Engaging RCCAR in a complete framework for context quality management is also recommended for future work.

REFERENCES

- [1] Buchholz, Thomas et al, "QoC: What It Is And Why We Need It", In Proceedings of the 10th Workshop of the OpenView University Association: OVUA'03, 2003.
- [2] Manzoor, Atif et al, Hong-linh, Dustdar, Schahram, "QoC: Models and Applications for CASs in Pervasive Environments". Special issue on Web and Mobile Information Services, 2010.
- [3] Manzoor, Atif et al, "On the Evaluation of QoC", Proceedings of the 3rd European Conference on Smart Sensing and Context, Springer-Verlag Berlin, Heidelberg, 2008.
- [4] Bu, Yingyi Gu et al, "Managing QoC in Pervasive Computing", Sixth International Conference on Quality Software (QSIC'06), 2006.
- [5] Henricksen, K. et al, "Modeling and using imperfect context information", Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, IEEE Proceedings, 2004.
- [6] Michael Krause et al, " Challenges in Modelling and Using QoC ", MATA , Springer-Verlag Berlin Heidelberg, 2005.
- [7] Filho, José Bringel et al, "Modeling and Measuring QoC Information in Pervasive Environments", 24th IEEE International Conference on Advanced Information Networking and Applications, 2010.
- [8] Y. Kim et al, "A quality measurement method of context information in ubiquitous environments", ICHIT'06. Vol 2. International, 2006.
- [9] Hegering, H. et al, "Management Challenges of Context-Aware Services in Ubiquitous Environments", Springerlink, 2003.
- [10] Jos'e Bringel Filho et al, "A Quality-Aware Approach for Resolving Context Conflicts in Context-Aware Systems", IFIP 9th International Conference on Embedded and Ubiquitous Computing, 2011.
- [11] Alessandra Toninelli et al, "A Quality of Context-Aware Approach to Access Control in Pervasive Environments", The Second International ICST Conference on Mobile Wireless Middleware, Operating Systems, and Applications, Berlin, 28 April 2009.
- [12] Federica Paganelli et al, "A Context Model for Context-aware System Design towards the Ambient Intelligence Vision: Experiences in the eTourism Domain", Universal Access in Ambient Intelligence Environments,, 2006.
- [13] Jos'e Bringel Filho et al, "A quality-aware approach for selecting context information from redundant context sources". 7th Network Operations and management Sumposium, IEEE Proceedings, 2011.
- [14] Atif Manzoor, "Quality Aware Context Information Aggregation System for Pervasive Environments", International Conference on Advanced Information Networking and Applications Workshops, 2009.
- [15] Eunhoe Kim, "A CMF for Supporting Context-Aware Applications", Proceedings of the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing-, 2008.
- [16] <http://www.southamptonweather.co.uk/sotonhist.php>
- [17] Witten, I.H. and Frank, E., "Data Mining: Practical machine learning tools and techniques", Morgan Kaufmann, 2nd Edition, 2005.
- [18] Nermin Brgulja et al, "Measuring the Probability of Correctness of Contextual Information in Context Aware Systems", Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, 2009.
- [19] Atif Manzoor et al, "Using QoC to Resolve Conflicts in Context-Aware Systems", Proceedings of the 1st international conference on Quality of context, Springer-Verlag Berlin, Heidelberg, 2009.
- [20] Tan, Pang-Ning; Michael, Steinbach; Kumar, Vipin, "Chapter 6. Association Analysis: Basic Concepts and Algorithms". Introduction to Data Mining. Addison-Wesley., 2005.
- [21] Michael J. A. Berry, Gordon Linoff, "Data Mining Techniques", Wiley Computer Publishing, 1997.
- [22] Asma A. Al- Shargabi et al, "Resolving Context Conflicts Using Association Rules (RCCAR) to Improve Quality of Context-Aware Systems", A final Accepted paper for ICCSE 2013, IEEE conference, Colombo, Sri Lanka.
- [23] S. Rasoul Safavian et al, " A Survey of Decision Tree Classifier Methodology" , , IEEE Trans SMC, Vol. 21, No. 3, May ,1991.
- [24] Tom M. Mitchell, "Machine Learning", McGRAW-Hill, 1997.
- [25] Lior Rokach, " DATA MINING AND KNOWLEDGE DISCOVERY HANDBOOK", Chapter 9-DECISION, Second Edition ,ISBN 978-0-387-09822-7 e-ISBN 978-0-387-09823-4, DOI 10.1007/978-0-387-09823-4, Springer New York Dordrecht Heidelberg London, 2010.