



Distributed Context Awareness for Remote Diagnosis

EL GHAYAM Yassine¹, Nadim Ismail², SADIQ Abdelalim²

¹EMSI-Rabat, Honoris University, Morocco, yassine.elgh@gmail.com

²Faculty of Science, Ibn Tofail University, Morocco, sadiq.alim@gmail.com

ABSTRACT

One of the important applications of mobile collaboration is the remote diagnostics. Such applications are exposed to multiple interruptions due to the terminal limited resources of one of the participants of this mobile collaboration. Such terminal could not support the processing load related to its own context-awareness decisions. However, in those environments, on the all available terminals resources may be quite enough to support required processing load. Therefore, the context-awareness management constitutes a challenging task in a distributed collaborative environment. The context-awareness management has the mission to capture the context information of collaborative users and adapts the reply of their interactions according to their context. Subsequently, managing the context information is a supplement charge which may be a tedious one, especially for large distributed collaboration activities. In this work, the context adaptation task is distributed among several entities by integrating the mobile agent paradigm. We suggest an approach to adapt the user interactions, within a collaborative application, to their collected context parameters. A context-aware algorithm is given to choose the appropriate terminal where the context-awareness decisions will be taken. An illustration of a healthcare application is given to explain the suggested approach.

Key words : Adaptation, Distribution, Collaboration, Context-awareness.

1. INTRODUCTION

The co-evolution of mobile terminals (smartphone, laptop, etc.), mobile access networks (WLAN, PAN, etc) and the development environment (Android, ...) provides new chances for the creation of collaborative systems over mobile environments [1]. Such systems allow to distant members to collaborate in a shared space, composed of various multimedia resources (video, audio, Instant Messages, etc.) and tools (applications, shared whiteboard, etc.) to reach a common objective.

For instance, in the healthcare domain, a collaborative tele-diagnosis system may profit from these technologies to

offer to distributed doctors and specialists the possibility to share their know-how to reach the appropriate diagnosis.

Sharing resources must managed by a policy of control to achieve both, good and profitable group work. The management relate to: creating collaborative sessions, join/exit a created session, invite/remove users to/from the current collaborative session and close a session. The control covers the services allowing to use the shared resources in a collaborative session, such as: granting / retrieving the authorization to use a resource, liberate a previously given authorization and requesting / rejecting the authorization. Many works have worked on management and control policies in collaborative environments [2,11,13,14,29,30,31]. These policies work well for centralized systems on a wired network. Such systems are characterized by a stable state and described by a fixed set of parameters. However, in a mobile environment, which is realized by wireless access networks and the use of mobiles devices, these parameters vary dynamically and permanently. Consequently, policies to control and manage a collaboration must be reconstructed while considering the mobile environment features [12].

The mobile environment is described by the ability to move of its participants and by its restricted resources, which implies supplements consideration in the management and control policies. This character may generate multiple new contextual situations. Such environment can generate many new contextual situations. Subsequently, Collaborative applications need to be integrated by new components that can detect context variation, and based on this change, these components must decide on the proper adaptation for collaboration activities. In this case, a mobile collaborative environment needs to be context-aware in order to allow such dynamic context adaptation.

Multiple work treats the context awareness [3, 4, 5], However, the majority have proposed approaches based on a centralized architecture, where the context management concerns is made by an unique architectural entity. In such architecture the context-awareness actions may be a loud heavy tasks on the collaboration tasks management. The adaptation process requires more resources on the devices (CPU and memory space). In addition, the central-based-architecture management needs additional response time to ensure collaboration activities of remote participants.

New systems may be created while using the mobile agent

concept, by which it may improve the management of the mobile collaboration by associating the user's state and its resources to a mobile agent. To each mobile agent on a separated device, it is possible to assign the task to capture the context changes relating to the user, and to decide on the convenient adaptation. This individually mission on each device may realize a decentralized-oriented-management of the contextual collaboration and subsequently the calculation load will be distributed between several entities. Thereby, as the adaptation management is addressed locally, the response time will be considerably reduced compared to client/server model, especially in a large distributed environment.

In a previous work [6], we discussed a formal model of a context-aware collaborative mobile system by adopting a decision tree-based concept. Note that in mobile distributed collaborative applications, the available resources for a particular mobile device could be not sufficient to deal with the collaborative sessions load. However, the entire accessible "distributed" resources could be quite enough to support such processing load. The point is how to procure a "fair" distribution of the processing load throughout an active collaborative session.

Therefore, we propose a distributed mobile agent system which can transfer the processing of a given action from a mobile device with less resources to another device characterized by more sufficient resources (memory, CPU, etc.). Such system permits to separate processing task from network topology. The mobile agents may be positioned on different locations. In each place the mobile agent may ensure the context-aware tasks : collect the context information and assume decisions autonomously. The mobile agent needs to hold the decision tree algorithm to ensure the context adaptation.

The paper is organized as follows: Section II describes the requirements imposed by the context-aware systems and distributed environments. Section III presents the mobile agent-based approach. We present the architecture and the decision tree model. Section V describes the context-aware collaboration algorithm. Section VI describes an e-health case study in order to illustrate the proposed approach. A conclusion is given in Section V.

2. REQUIREMENTS

In the following, we elicit the main requirements related to context-awareness. In what follows, we address the essential requirements related to context-awareness and to the mobile distributed collaborative environment:

- Adaptation rules: adaptation to context can be considered as a collection of rules that controls the context change occurring in an environment. Hence, in the construction of the adaptation rules, the number of the context parameters is huge and then, it is not evident to enumerate all feasible situations that will be occurred. Therefore a method to construct an adaptation rules is required, which may deal with

the diversity of the possible situations that are constructed based on these context parameters.

- Lightweight code: in the mobile environment, the devices used, such as PDA and Smartphone, are characterized by their limited capabilities. Generally, these terminals are noted by low battery power. Moreover, the mobile networks, such as, WLAN and PAN, are also characterized by the low bandwidth in addition to frequent connection fluctuations. Therefore, the mobile applications must be aware of this poor-resources character by partitioning the processing of the workload of the collaboration on several entities.

- Hidden Mobility and heterogeneity: the users move frequently and may differ in their devices characteristics and in the forms of the network connectivity used. The heterogeneity may concern also the development environments (J2ME, Android, etc.) which run the applications and the services. Therefore, the collaborative applications have to absorb this heterogeneity and deal with the mobility of the users to guarantee an efficient collaboration processes.

- Loosely coupled: the users are geographically dispersed in the mobile collaboration application such as the collaborative telediagnosis. The collaborators can interact with each other by sharing concurrently, a consistent environment in terms of space, presence and time. In a centralized control and management of the repartitioned collaborative applications the control and the management of all users and resources will be complicate, particularly in the involuntary disconnection cases [11]. Hence, such applications require allowing the autonomy to distributed entities in order to establish loosely coupled treatments [25] which eliminate the blocked situations.

There are many open questions that need to be answered; how to formalize the context information, how to adapt the collaboration services to context and how to remedy to the mobile environment constraints when establishing distribution. The context formalization is discussed in a previous work [6] and in the following sections we answer the others questions.

3. THE MOBILE AGENT-BASED ARCHITECTURE

We suggest an architecture structured around this mobile agent model. This architecture consists in the following components as shown in figure 1: User Session, Mobile Agent (MA) component, MA Collaborative Session and Mobile Agent Manager components which will be described as follows:

User Session: represents the set of the functions and tools available to the user to join and to participate to a collaborative session. The session provides a user interface using the user terminal. Through this interface the user could Join/Leave a session and Ask/Release a resource. The tools allow the use of the shared resources such as the video, audio,

instant message and email by the user through a collaborative session.

Mobile agent: is an active element that may detect the context parameters (connexion, availability and location parameters) from the user terminal during a user session to allow adaptive shared resources. For that, inside the mobile agent, a Client Side Probes component is used to detect, at each moment, the context parameters of the user. The location of the user may be detected by using one of the following location sensing technologies: Active badge, RFID, GPS. The Connectivity and the Availability parameters are determined from the User Session interface which are specified by the user.

The mobile agent contains a Context Adapter component that takes the context parameters and the decision tree-based rules as inputs in order to provide adapted resources. The rules based on the decision tree are constructed previously.

Collaborative Session: the component that realizes the management and the control of collaborative sessions. It contains the Group Manager and the Resources Manager components. The Group Manager allows initiating and creating a new collaborative session, join an initiated session, invite users to the current collaborative session and terminate a session. The Resources Manager allows services to exploit the shared resources (video, audio, Instant Message, e-mail) in a collaborative session, such as: “Give” or “Refuse” permission to use such resources and “Grab” previously given permission to use a shared resource.

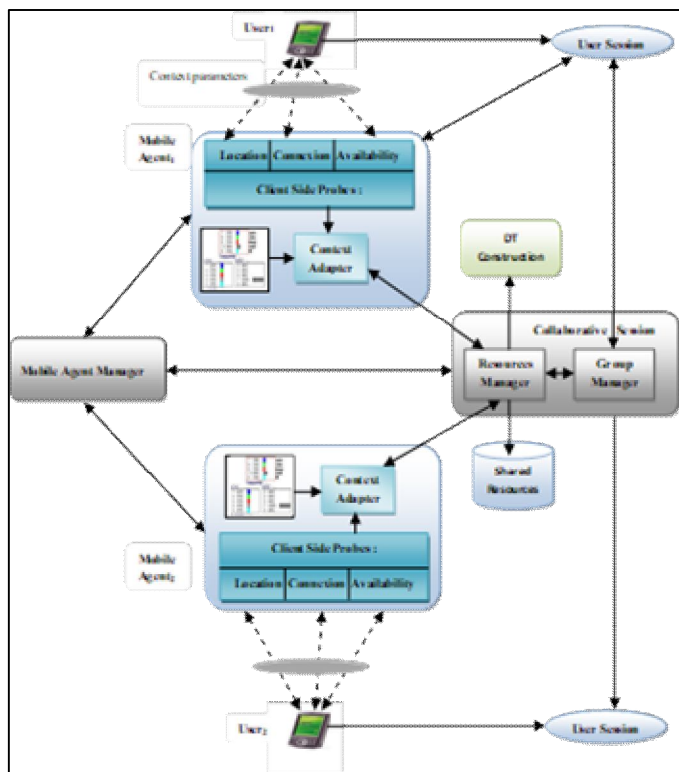


Figure 1: The mobile agent architecture

Mobile Agent Manager (MAM): represents the component that is responsible to create and delete a mobile agent, it maintains information about created agents, such as identity and location. The MAM may also establish a collaborative session between created agents. A mobile agent may communicate with the MAM to consult the list of joined mobile agents.

In the following, we present the algorithm that maintains the MAs collaborative session according to their associating terminals context information (CPU, Memory and Battery).

4. CONTEXT-AWARE SELECTION ALGORITHM

An MA has to be continuously aware of the resources status of its local terminal such as Battery, Memory and CPU resources status. Accordingly, the local terminal is dotted by a set of probes to obtain periodically the status of the terminal resources. When detecting a resource under the specified threshold, the MA must compare its own tasks requirements with the characteristics of the others machines. The comparison is based on the battery, memory and CPU parameters.

When the concerned MA (MA_c) is in a critical situation (one of the resources parameters of its local terminal is under the threshold), as shown in figure 2, it contacts the MAM to obtain the list of the available MAs (the MA_{av} array). Then, the MA_c sends a request to each one of the MA_{av} list. We suppose that the communication between the MAs is reliable. The request contains the MA identity and the type of the concerned resource (Rsc).

We consider that a MA may verify autonomously whether its own resources may support supplement MA tasks. For that, only the MAs that have sufficient resources may send a response (the MA_{res} array). The response contains the resources status of the terminals responder (Rsc_status). Therefore, several responses may be arrived (MA_{pre} array). By using the *selectBestResources* function, the MA_c selects the terminal that has the best resources (we suppose that all of the MAs have resources status that are superior to the specified thresholds). Finally, the algorithm decides to clone MA_c to the selected terminal T_{sel} .

The *selectBestResources* function is based on the algorithm shown on the figure 3. The terminal selection is based on the resources priority. Principally, the priority considered is as follows: the battery level, the CPU power and the memory remaining. However, in the case where a MA has a problem in the CPU resource, the priority will be respectively as: CPU, Battery, and Memory. And in the case where the memory is insufficient, the priority on which is based the algorithm is respectively as follows: Memory, Battery and CPU.

```

Rsc : resource variable
MAc:concerned MA
MAav=NULL: array of the available MAs
MAres=NULL: array of the MAs responders
Tsel =NULL: the selected terminal
IF (Rsc < threshold) THEN
  MAav = Available MAs from MAM
  For each MA from MAav
    send_Request(MA, Rsc)
  end For
  MAres = receive(Rsc_status, MA)
  MAres = selectBestResources(MAres, Rsc)
  Tsel = select_Trminal(MAres)
  Clone(MAc, Tsel)
END IF

```

Figure 2: Collaborative MAs algorithm

Furthermore, the algorithm has as an input a bi-dimensional array (Rsc[nb_MA][3]) that represents the resources status of the responders MAs (nb_MA is the number of these MAs and the number 3 is the three considered resources: battery, CPU and memory). These resources are sorted following the priority indicated previously. Initially, the algorithm seeks the terminal that has the best resources status following the resource that has the important priority. Accordingly, we consider the first resource (Rsc[i=0][0]) as the best one (bestRsc) and we compare it to each other one Rsc[i>0][0] (currentRsc).

```

int i=0, j=0, selectedIndex=0
float currentRsc=0, bestRsc=0, diff=0
boolean choice=0
while (j<3) And (choice=false) Do
{
  bestRsc = Rsc[i][j]
  while (i < nb_MA) Do
  {
    i = i+1
    currentRsc = Rsc[i][j]
    diff = compare(bestRsc, currentRsc)

    if (diff > 10%) then
      bestRsc = currentRsc
      selectedIndex = i
      choice = true
    }
  }
  j = j+1
  i = 0
}

```

Figure 3: Terminal selection algorithm

If, after comparing all the elements in the first column, then the algorithm finds an element that is different from the others and the algorithm selects the terminal associated to this element and the process is terminated. The difference (diff) considered is 10%. This difference constitutes the margin to support a supplement MA.

Contrary, if the algorithm does not find an element that is distinguished by the 10% as a supplementary, the algorithm passes to the next column and performs the same procedure while considering the next resource. The algorithm

terminates by selecting one appropriate terminal. In the case where no terminal is selected, an error message is sent to the user. Otherwise the cloning is performed to the selected terminal.

The cloned MA will perform the designed tasks in a transparent way for the original MA. A cloned MA is a new instance of a MA with its run-time state at another host. After cloning, the cloned agent provides the same service corporately with the original one. In our approach, the cloned MA provides a method to ensure the continuity of a MA tasks that its resources are not in good conditions. Principally, an MA collects the local context information of the user's terminal (connexion, availability and location) and assigns to the user the appropriate shared resources following the decision tree algorithm. Accordingly, by introducing the cloning concept, the original MA continuously updates the cloned MA by the context information of the user's terminal and the cloned MA performs the context adaptation tasks.

5. CASE STUDY

Medical activities in healthcare environments are characterized by the need for coordination and collaboration among specialists with different areas of expertise. Hence, the necessity of integration of tools facilitating to intercommunicate and share data in order to achieve a common goal, regardless of the circumstances of the doctors (location, availability, etc.).

When a case such as the Cerebro-Vascular Accident (CVA) is presented in a hospital, it supports by an emergency physician. This doctor contacts the Emergency Medical Service center (EMS), in order to inform him about the emergency admission of a patient who presents symptoms suspecting a CVA (trouble speaking or understanding, loss of balance or coordination, etc.).

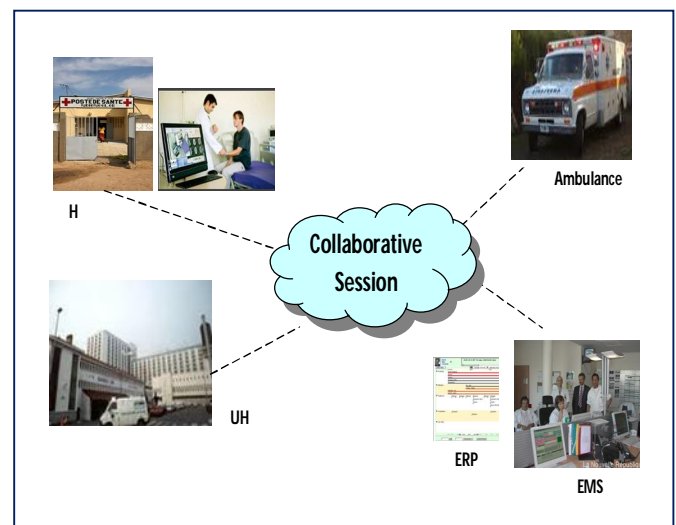


Figure 4: Distributed e-health session

The EMS's responder creates an Electronic Patient Record (EPR), such as shown in figure 4, based on initial received

physical examination, such as GCS, pulse, respiration, temperature, etc. While maintaining the contact with the emergency physician, the EMS's responder initiates a collaborative session with a neurologist on call of the University Hospital (UH) in order to establish the diagnosis. During the collaborative session between the hospital, the UH and the EMS collaborators, the neurologist assists and supports the emergency physician on establishing the clinical examinations and the para-clinical examinations. When necessary, the neurologist invokes the radiologist or other specialists to locally assist him the diagnosis in the remote site.

Thereafter, based on examination results, a decision is made: the neurologist supports remotely the patient while he is staying in the hospital, or the patient is transported in an ambulance into an appropriate place (hospital center with a neurovascular unit). In this last case, the ambulance may contain an emergency physician or a resuscitator. While transferring the patient a medical monitoring will be prevented by the neurovascular unit. A more detailed description and a formal specification of the telediagnosis scenarios have been presented in [27].

6. CONCLUSION

We have presented an approach including an architecture of a distributed context management based on the use of mobile agents. Such approach allows the use of the available distributed resources to process the context information and allow for a dynamic adaptation within a distributed collaborative application. We adopted the mobile agent paradigm where a mobile agent may be located on various places. In each place the mobile agent may take adaptation decisions autonomously. Such decisions are based on the decision tree method. Furthermore, an algorithm is proposed to manage the overall available resources in the partitioned in order to allow a "fair" distribution of the processing load among the mobile terminals.

REFERENCES

1. D. Dhoutaut. **Etude du standard IEEE 802.11 dans le cadre des réseaux ad hoc : de la simulation à l'expérimentation**, Ph.D. dissertation, *Institut National des Sciences Appliquées de Lyon*, 2003.
2. M. Ouzif, M. Erradi, H. Mountassir, **Description of a Teleconferencing Floor Control Protocol and Its Implementation**, Elsevier. *Journal of Engineering Applications of Artificial Intelligence*, EAAI, Editor, 2008.
3. T Gu , H K Pung , D Q Zhang , X H Wang, **A Middleware for Building Context-Aware Mobile Services**, in *Proc. Vehicular Technology Conference, VTC . 2004 IEEE 59th*.
4. L Capra, W Emmerich and C Mascolo, **Reactive Middleware Solutions for Context-Aware**

- Applications**, in *Proc. the Third International Conference on Metalevel Architectures and Separation of Crosscutting Concerns*, p.126-133, September 25-28, 2001.
5. L Daniele, P D Costa, and L F Pires, **Towards a Rule-Based Approach for Context-Aware Applications**, In *Proc. the 13th Open European Summer School and IFIP TC6.6 Workshop (EUNICE'07), Enschede, The Netherlands, pp. 33–43*.
6. Y. El Ghayam and M. Erradi, **A Decision Tree Based Context Management in a Collaborative Environment**, In *Proc. NOTERE'10, Tozeur, tunis*.
7. N. Bricon-Souf C. R Newman, **Context awareness in health care: A review**, *International Journal of Medical Informatics*, 2007
8. D. Guan, W. Yuan, S. Lee and Y.-K. Lee, **Context Selection and Reasoning in Ubiquitous Computing**, *International Conference on Intelligent Pervasive Computing*, 2007.
9. V. Sacramento, M. Endler, H. K. Rubinsztein, L. S. Lima, K. Gonçalves, F. N. Nascimento, G. A. Bueno, **MoCA: A Middleware for Developing Collaborative Applications for Mobile Users**, *IEEE Distributed Systems Online*, 2004.
10. A. Ranganathan and R. H. Campbell, **An infrastructure for context-awareness based on first order logic**, *Personal Ubiquitous Comput.*, 353–364, 2003.
11. Y. Elghayam, M. Ouzzif and M. Erradi. **An LTL Specification and Verification of a Mobile Teleconferencing System**, *International Workshop on Verification and Evaluation of Computer and Communication Systems, Leeds*, 2008.
12. Dario Bottazzi, Antonio Corradi, Rebecca Montanari. **Context-Awareness for Impromptu Collaboration in MANETs**, *WONS 2005*.
13. Kausar, N., Crowcroft, J. **An architecture of conference control functions**. In *Proc of Photonics East, Boston, Massachusetts, September 20–22, 1999*.
14. Dommel, H.-P, Garcia-Luna-Aceves, J.J., 1999. **Network support for turn-taking in multimedia collaboration**, In *Proc. of the IS& T/SPIE Symposium on Electronic Imaging: Multimedia Computing and Networking, San Jose, CA, February 1999*.
15. Karen Henriksen, Jadwiga Indulska., **Developing context-aware pervasive computing applications: Models and approaches**, *Pervasive and Mobile Computing 2006*.
16. Dockhorn Costa, P. and Almeida, J.P.A. and Ferreira Pires, L. and van Sinderen, M.J. **Situation specification and realization in rule-based context-aware applications**, In *Proc. 7th IFIP WG6.1 Int'l Conf. Distributed Applications and Interoperable Systems, DAIS, Paphos, Cyprus, ser. Lecture Notes in Computer Science, vol. 4531. Springer, 2007, pp. 32–47*.
17. B.-H. Lee, H.-N. Kim, J.-G. Jung1, and G.-S. Jo **Location-Based Service with Context Data for a**

- Restaurant Recommendation**, *DEXA 2006, LNCS 4080*, pp. 430 – 438, 2006.
18. J. Favela, M. Tentori, L. A. Castro, V. M. Gonzalez, E. B. Moran, A. I. Martínez-García, **Activity Recognition for Context aware Hospital Applications: Issues and Opportunities for the Deployment of Pervasive Networks**, *Mobile Netw Application*, 2007.
 19. Y. Bouzida, F. Cuppens, **Neural networks vs. decision trees for intrusion detection**, in *Proc. IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation (MonAM)*, Germany, 2006.
 20. Jakob E. Bardram and Thomas R. Hansen, **The AWARE Architecture: Supporting Context-Mediated Social Awareness in Mobile Cooperation**, *CSCW'04, November 6–10, 2004, Chicago, Illinois, USA*.
 21. T. Broens, A. v. Halteren and M. v. Sinderen, **Towards an application framework for context-aware m-health applications**, In *Proc. 11th Open European Summer School (EUNICE 2005)*, 6-8 July 2005, Colmenarejo, Spain
 22. M. A. G. Salvo, R. M. A. Mateo, J. Lee, and M. Lee, **Health Monitor Agent Based on Neural Networks for Ubiquitous Healthcare Environment**, *KES-AMSTA 2009, LNAI 5559*, pp. 380–388, 2009.
 23. Marcela Rodríguez and Alfredo Preciado, **An Agent Based System for the Contextual Retrieval of Medical Information**, *AWIC 2004*, pp 64-73.
 24. David Pinelle and Carl Gutwin, **A Groupware Design Framework for Loosely Coupled Workgroups**, In *Proc. the European Conference on Computer-Supported Cooperative Work*, 2005.
 25. Shiao, D. **Mobile agents: A new model of intelligent distributed computing**. Technical report, *IBM DeveloperWorks, China (2004)*
 26. M. A. Haddar, A. Hadj-Kacem, Y. Métivier, M. Mosbah, and M. Jmaiel. **A distributed computational model for mobile agents**. In *Proc. of 10th International Conference on Principals of Practice in Multi-Agent Systems PRIMA 2007, volume 5044 of Lecture Notes in Artificial Intelligence, pages 416-421. Springer, 2009*.
 27. R. Edaoudi, M. Hamdani, K. Bouragba, M. Ouzzif, H. Mountassir and M. Erradi, **Architecture Formalisée d'une Application Collaborative Cas d'une Situation d'Urgence en Neurologie**, *AFADL 2010, France*.
 28. N. Phung, M.M. Gaber and U. Roehm, **Resource-aware Distributed Online Data Mining for Wireless Sensor Networks**, In *Proc. of the International Workshop on Knowledge Discovery from Ubiquitous Data Streams (IWKDUDS07)*, Warsaw, Poland, 2007.
 29. M. Akour, O. A Qasem, H. Alsghaier , K. Al-Radaideh, **The Effectiveness of Using Deep Learning Algorithms in Predicting Daily Activities**, *International Journal of Advanced Trends in Computer Science and Engineering, IJATCSE, ISSN 2278-3091, Volume 8, No.5, pp.2231 – 2235, September - October 2019*.
 30. S. A. Shinde, P. R. Rajeswari, **A Novel Hybrid Framework for Cuff-Less Blood Pressure Estimation based On Vital Bio Signals processing using Machine Learning**, *International Journal of Advanced Trends in Computer Science and Engineering, IJATCSE, ISSN 2278-3091, Volume 9, No.2, pp. 1556 – 1561, March - April 2020*.
 31. M. T. Abu Bakar, A. A. Jamal, **Latency Issues in Internet of Things: A Review of Literature and Solution**, *International Journal of Advanced Trends in Computer Science and Engineering, IJATCSE, ISSN 2278-3091, Volume 9, No.1.3, pp. 83 - 91, 2020*.