ISSN 2278-3091
Volume 8, No.2, March - April 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse28822019.pdf
https://doi.org/10.30534/ijatcse/2019/28822019

OSSR-P: Ontological Service Searching and Ranking System for PaaS Services

Mohammed Gouse Galety[1] Saravana Balaji B[2] Saleem Basha M S[3]
[1]Assistant Professor, Department of Computer Networks, Lebanese French University, Iraq
[2]Assistant Professor, Department of Information Technology, Lebanese French University, Iraq
[3]Assistant Professor, Department of Computing & Informatics, Mazoon College, Oman

ABSTRACT

Cloud computing enables the use of computing resources through the web, which is delivered as software, platform and infrastructure services. With increasing demand and adoption of cloud services, cloud service user faces many issues on selecting a particular service. In this paper, we provide a system, OSSR-P, which search and rank the platform as a service type services according to user requirements. OSSR-P system is based on PaaS service ontology which is developed and populated by parsing the service description document and WSDL document of PaaS services. The user requirements are parsed and generated as SPARQL queries which then retrieve the service list from the ontology repository. The retrieved services are ranked according to the requirements matching. The performance evaluation of OSSR-P shows that it provides and maintains consistent results irrespective of the number of requirements given by the user.

Key words: Cloud Service, Ontology, Service Search, PaaS Service, SPARQL.

1. INTRODUCTION

The software and hardware usage of a user has been moved from a personal computer to the internet due to the emergence of cloud computing. Cloud computing is a paradigm which evolves from the combination of grid computing and service-oriented architecture (SOA), where grid computing emphasizes the shared use of computational resources and SOA deals with delivering functionality as web service for the users through various standards like SOAP, UDDI etc. [16]. The need for Cloud computing is to reduce the cost of computing and also to ensure availability, reliability and flexibility of services [7]. Cloud computing provides services to users through the internet under three kinds - Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Normally a cloud service is explained through service description document and WSDL document. Service description document describes components of the service under sections like services offered, cost The service etc., where WSDL documents describe technical attributes of the service[15]. Due to the diverse nature of cloud services, selecting an appropriate cloud service becomes a difficult task for an inexperienced user. To solve this issue, semantic technologies can be used. In the Semantic Web, knowledge is represented using ontology [3], which is viewed as a formal specification of a domain knowledge conceptualization. Semantic Web Services are providing a declarative, ontological structure for describing web services in a machine-readable format that facilitate the interpretation of web service description based on their meanings [8]. In this paper, we provide a semantic system OSSR-P which search for PaaS kind cloud services and rank the services that are found according to the user requirements. PaaS kind services are consist of the software and services which are used by developers to build new applications, normally through an API. PaaS kind services reduce the developer's requirements regarding deployment, configuration and scaling of applications. Normally PaaS kind services are composed of PaaS utilities along with the underlying infrastructure. In this proposed system the information about a particular PaaS kind service is updated on the PaaS service ontology which we developed, using parsing the service description document and WSDL document of that service using PoS tagging system. The service requester issues the request through the interface that is provided on the system. Then the semantic service searching system search for the PaaS kind cloud services and ranks them according to the user requirement and the results are provided to the user in a table format which lists the services according to the rank. There is a need to develop an ontology for defining cloud services and the concepts; their relationships are to be derived from the documents that are available from cloud service providers. The remainder of this paper is organized as follows. Section 2 describes various related works that are available; section 3 describes the PaaS ontology structure, section 4 describes various subsystems of the system, section 5 describes the service searching and ranking methodology of the system, section 6 describes the experimental analysis of the system and section 7 concludes the paper.

2. RELATED WORK

There have been several methods proposed by many researchers for the problem of selecting an appropriate service which involves multi-criteria decision-making situation. The method proposed in [17] developed QoS ontology and a QoS-based ranking algorithm for evaluating Web services.

The QoS- based ranking algorithm adopted the Analytic Hierarchy Process (AHP), a multiple criteria decision-making technique, as an underlying mechanism for developing a flexible and dynamic ranking algorithm. In this system, business requirements are described in OWL-T, an OWL-based task template language, in terms of a high-level task template including various component tasks. The proposed QoS ontology and the ranking algorithm can be used in various applications in order to facilitate automatic and dynamic discovery and selection of Web services. The method proposed in presented an ontology-based framework called SEMantic web services and Multi-Agent System framework (SEMMAS), which provides seamless integration of the technologies by making use of ontologies to facilitate their interoperation. As the framework becomes independent from both the domain and the actual application, it is applied to powerful distributed systems, and it is cost-effective. A weighted semantic similarity algorithm has been proposed in [13]. It is developed to support a more automated and veracity service discovery process, by distinguishing among the potentially useful and the likely irrelevant services and by ordering the potentially useful ones according to their relevance to the developer's query. The semantic similarity between matching pair of web service can be ranked manually or programmatically by selecting the objective web service. As a result, the selected web service can be bound into the business process. In a semantic information system for the advertisement, retrieval and selection of application services, and markets that trade resources, in a democratized Grid e-marketplace environment has been proposed. This system motivates the development of the Grid4All Semantic Information System (G4A-SIS), a web service providing clients with an interface for registering and querying semantically annotated market and application-specific services. In addition to the ontology-centred mechanisms provided for matchmaking and semi-automatic annotation of web services, the system provides a ranking mechanism to support the selection of markets and services. To facilitate arctic research, a Semantic-based web service discovery has been proposed in [11]. In this method, the knowledge-based approach and the spatial web portal technology were utilized to prototype and Arctic SDI (ASDI) by proposing, the buildup of a hydrology ontology to model the latent semantic relationship among the data and Smart search and integration service to chain and visualize the datasets to enable a semiautomatic science workflow. This method addressed three important research challenges, service discovery, knowledge base development, and service decomposition and chaining, when building an integrated ASDI. The proposed hybrid approach, which combines multi-catalogue searching, and active crawling mechanisms, help to collect rich resources to support scientific modelling. In the current implementation, an intuitive performance indicator in terms of availability and response time is used to rank the quality of the datasets retrieved by semantic reasoning. Decision support in e-business has been proposed in [19]. It has an ontology similarity assessment algorithm to map concepts and properties between ontologies of semantic web services. In order to get promising results they have followed a two-phase procedure, syntactic analysis measuring the difference between tokens by the edit distance and then semantic analysis based on WordNet as semantic relation and similarity assessment of tree-structured graphs with the Tversky similarity model.

3. PAAS SERVICE ONTOLOGY STRUCTURE

The traditional software development model makes the developer to stick with the development environment, where the PaaS services remove these aspects and manage the platform requirements altogether. PaaS service providers provide a computing platform that includes a set of development, middleware, and deployment capabilities. The PaaS services are also capable of supporting interoperability, migration etc. Before selecting a PaaS service, a user has to know about that service provider in many aspects. The aspects of a PaaS service are application development framework, language supported, databases supported, several parallel applications, portability etc. [4]. Normally the aspects of a PaaS service are provided through service description document and WSDL documents. As the number of PaaS service providers are increasing, searching and comparing different PaaS service providers aspects become an overhead for a user. To address this issue, we are proposing the system OSSR-P which searches for the PaaS services and ranks them according to user requirement based on PaaS service ontology. The proposed PaaS service ontology has classes and resources in the view of a developer. The classes that are defined for the PaaS ontology are framework supported, languages supported, a platform supported, development kind, PaaS stack, database supported, no of application instances etc. The properties relating to the resources of various concepts are also defined under the PaaS service ontology. The PaaS ontology structure developed based on OWL-S[18] with major concepts, and few resources is given in figure 1.

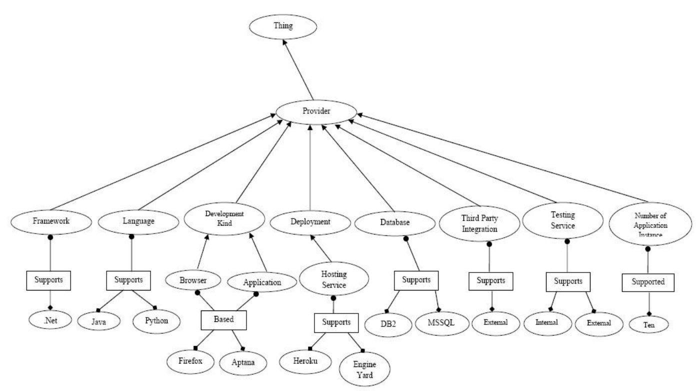

Figure 1: PaaS Service Ontology

4. SYSTEM ARCHITECTURE

The system architecture of OSSR-P is given in figure 2. This system consists of five sub systems, a user query interface and an ontology repository which stores the ontology information of PaaS services. The subsystems of OSSR-P are:

a. Service Document Analyzer
b. Ontology Updating System
c. User Query Analyzer
d. Query Generator
e. Service Ranking System

4.1 SERVICE DOCUMENT ANALYZER

This system is built up using the PoS tagger called Stanford log-linear part of speech tagger [10]. This system is made with a crawler which searches the web for PaaS service providers and will retrieve the service description documents and the WSDL documents of the corresponding service. The document portions are then separated according to the parameters that are defined under the PaaS ontology. Then these documents are parsed using the PoS tagger, and the nouns (NN, NNS, NNP and NNPS) are retrieved. The properties that relate the resources are identified based on the verbs (VB, VBN, and VBZ) of the PoS tagger. Then these values are sent to the ontology updating system where they are added with the ontology repository according to class as resources.

4.2 ONTOLOGY UPDATING SYSTEM

Ontology updating system is responsible for ontology population. Ontology population is an important part of the semantic web systems which enables the identification of text related to concepts of the domain ontology. It is performed by identifying the key terms in the text

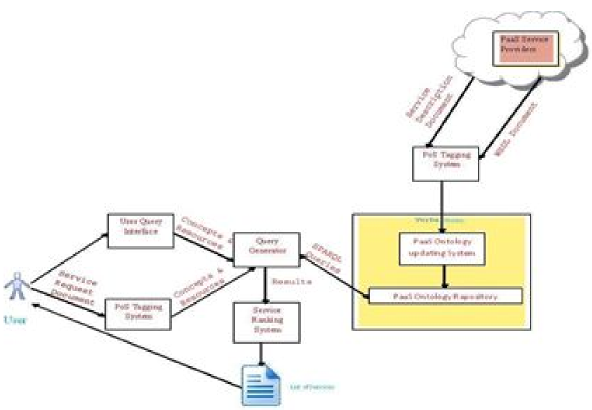

Figure 2: OSSR-P: Architecture

Moreover, relating them to the concepts of the ontology, which is called ontology-based information extraction [12]. The entities that are transferred from service document analyzer are updated in the PaaS service ontology.

4.3 USER QUERY ANALYZER

This system parses the query given by the user which then identifies the entities of the query. The query generator will use the entities from this system and SPARQL [14] queries are generated accordingly.

4.4 UERY GENERATOR

This system accepts input from user query analyzer and the query interface system. Based on the type of input given whether, from user query analyzer or query interface system, it will generate SPARQL queries accordingly. The query generation algorithms are described in the next section.

4.5 SERVICE RANKING SYSTEM

This system retrieves the results from the ontology repository and ranks the services based on the type of query issuing methodology used by the user. The service ranking algorithms are described in the next section. The query generator will use the entities from this system and SPARQL [14] queries are generated accordingly.

4.6 QUERY GENERATOR

This system accepts input from user query analyzer and the query interface system. Based on the type of input given whether, from user query analyzer or query interface system, it will generate SPARQL queries accordingly. The query generation algorithms are described in the next section.

4.7 SERVICE RANKING SYSTEM

This system retrieves the results from the ontology repository and ranks the services based on the type of query issuing methodology used by the user. The service ranking algorithms are described in the next section.

4.8 USER QUERY INTERFACE

The user query interface system consists of two kinds of query issuing systems. The first kind accepts the service request document from the user and the second kind provides the interface which consists of various concepts of systems and the resources of the system along with the weight giving the option for each resource of concepts. The resources under the concepts are updated frequently if there is an ontology updating happens at the ontology repository. If the user issues a service request document, then the Stanford PoS tagger [10] will parse the document and identifies the nouns and the verbs. Then the identified entities are transferred to the query generator. If the user uses the query interface, then the resources along with weight for each resource are selected by the user. Then the entities along with weight are transferred to the query generation system.

4.9 ONTOLOGY REPOSITORY

The PaaS service ontology repository is an OWL database which contains the OWL schema of the PaaS service ontology and the OWL file of the resources of the ontology, which is accessed through Jena for the storing the resources on the PaaS service ontology.

5. SERVICE SEARCHING AND RANKING METHODOLOGY

This section describes the PaaS service searching and Ranking methodology. The adaption of service requirements of the user by the system is in two kinds. In the first kind, the service user issues the service requirements in a text file format called the service requirement document. The searching system accepts the document, and it is analyzed by the Stanford PoS tagger system. The PoS tagger returns the nouns and verbs of the documents. The nouns and verbs are assigned as the resources and their properties under the predefined PaaS service ontology concepts. Then the identified entities are transferred to query generator which generates the SPARQL queries and the service returned are ranked by the service ranking system based on the algorithm A.. In this method the ranking of services is based on several concepts that are matched with the requirements of the user, that is, the service that matches with a higher number of concepts that are requested by the user is in the first place of the service list.

The second kind which is a user query interface used by the user for giving the requirements. This interface lists all the concepts of the PaaS service ontology along with the resources available under them. The user can select the concepts and the resources according to his need and have to give the weight age to each of the resources he selected. Then the concepts and the resources are transferred to the query generator which generates the SPARQL query and the services returned are ranked by the service ranking system based on the algorithm B.. In this method, the rank is based on the weight assigned by the user for each concept he requested. The retrieved PaaS services are returned according to their rank to the user along with their corresponding URL [1] of the service provider.

```
for every request
set concept[] = {concepts}
set resources[] = {resources from request}
set property[] = {properties from request}
set services[] = {}
for each resource and property{
SELECT ? service WHERE{
Object Property Assertion (:property?concept?resource)}
for each service returned{
if service is not in services[] then
add service to services[];
set servicevalue = 1;
else
servicevalue=servicevalue+1;
end
}
}
reorder services in services [] based on servicevalue;
return services[];
end;
```

Algorithm A

```
for every request
set concept[] = {concepts}
set resources[] = {resources from request}
set property[] = {properties from request}
set resourceweight[]={weight of resources from request}
set services[] = {}
for each resource and property{
SELECT ? service WHERE{
Object Property Assertion (:property?concept?resource)}
for each service returned{
if service is not in services[] then
add service to services[];
weight=resourceweight[resource];
set servicevalue = 1;
else
weight=weight+resourceweight[resource];
servicevalue=servicevalue+1;
end
}
}
For each service returned {
Weight=weightservicereturn;
}
reorder services in services [] based on servicevalue;
return services[];
end;
```

Algorithm B

6. EXPERIMENTAL ANALYSIS

OSSR-P system has been implemented using Jena semantic web framework [9] and SPARQL for querying the PaaS service ontology repository through ARQ. Apache Tomcat was used to host the CS- ontology repository. The systems of OSSR-P are given access through a JAVA API which has the functionality of querying. The PaaS service ontology repository has been populated by parsing the service description documents and WSDL documents of various PaaS services listed under [20] which provide details about cloud service providers under different categories along with manual searching of PaaS service providers in the WWW. The PaaS service ontology consists of 25 concepts, 73 properties and sub-properties along with 125 resources (service providers). For any information retrieval system, the effectiveness of the system is evaluated using precision and recall [2] where

$$\text{Precision} = \frac{\text{\# of relevant item retrieved}}{\text{\# of retrieved items}} \quad (1)$$

$$\text{Recall} = \frac{\text{\# of relevant items retrieved}}{\text{total \# of relevant items}} \quad (2)$$

The performance of OSSR-P system is evaluated using the following measures – precision, recall and query response time – the time required by the OSSR-P for answering a single requirement issued by the user. For evaluation purpose, we have generated 50 queries of different requirements, from few numbers of requirements to all the concepts that are listed under OSSR-P, which are issued through both the kinds of user query interface system. Figure 3 depicts the precision graph, figure 4 depicts the recall graph, and figure 5 depicts the query response time of the system. From the results, it is clear that the system provides ranked services according to user requirements with consistent precision and recall irrespective of the number of requirements issued by the user. The query response time of the system is also directly proportional to the number of requirements issue by the user.

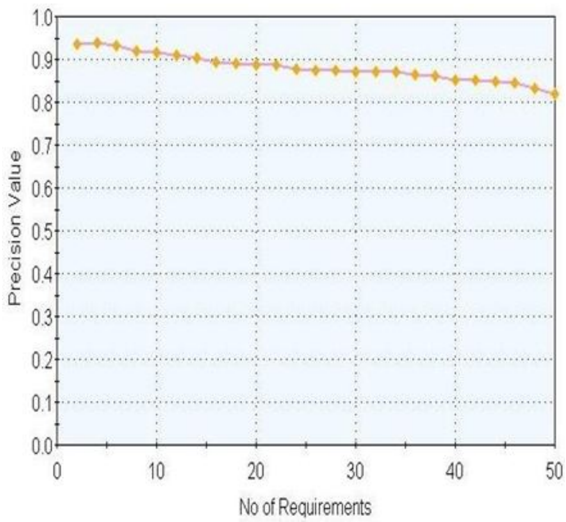

Figure 3: Precision Graph

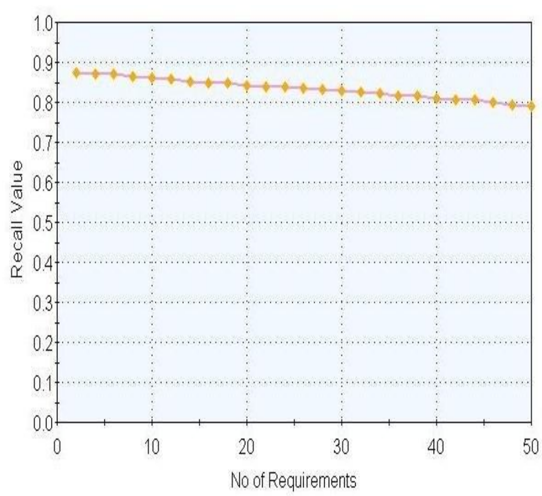

Figure 4: Recall Graph

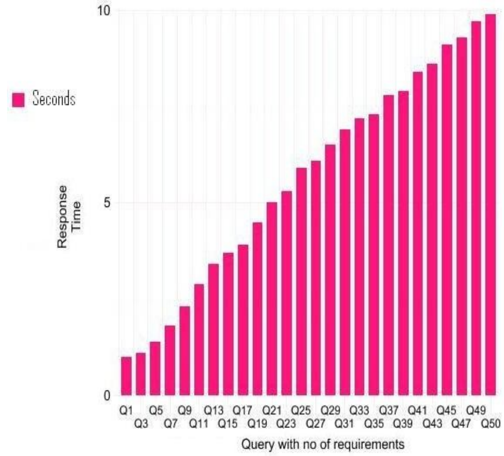

Figure 5: Query Response Time

7. CONCLUSION

Selecting an appropriate PaaS service under diverse nature of cloud service providers is a cumbersome task for a user. To address this issue, we have developed a system called OSSR-P which searches for the PaaS services according to user requirements and ranks them based on the number of requirements that are matched. This system is based on PaaS service ontology which is populated by parsing the service description document and WSDL document of the PaaS service provider. This system accepts user request through two ways, by accepting the service request document or directly issuing the request through the user query interface. Then this system searches for the appropriate services according to user requirement and ranks the services based on the type of service request issuing method used by the user. The performance evaluation of the system shows that this system retrieves the services more precisely according to the requirements of the user. This system can be extended by parsing the SLA documents of the service providers for updating the concepts of the PaaS service ontology under technical attributes like the cost of the service, the uptime of the service etc. For selecting and ranking service, considering user feedback about that service is a good practice. The OSSR-P system can be extended to update the concepts of the ontology under user feedback criteria.

REFERENCES

[1] Berners-Lee, T., Masinter, L., and M. McCahill, "Uniform Resource Locators (URL)", RFC 1738, December 1994. https://doi.org/10.17487/rfc1738

[2] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, “Introduction to Information Retrieval”, Cambridge University Press, 2009. https://doi.org/10.1017/CBO9780511809071

[3] Dieter Fensel, et al., “An Ontology Infrastructure for the Semantic Web”, IEEE Transactions on Knowledge and Data Engineering, 2001, Vol.19 No.2 pp.261272. https://doi.org/10.1109/5254.920598

[4] Dan Orlando, “Cloud computing service models, Part 2: Platform as a Service”, IBM developer works, Jan 2011.

[5] Francisco Garcıa-Sanchez, Rafael Valencia-Garcıa, Rodrigo Martınez-Bejar, Jesualdo T. Fernandez-Breis, “An ontology, an intelligent agent-based framework for the provision of semantic web services”, Expert Systems with Applications, 2009. https://doi.org/10.1016/j.eswa.2008.01.037

[6] George A. Vouros, Andreas Papasalouros, et.al, “A semantic information system for services and traded resources in Grid e-markets”, Future Generation Computer Systems, 2010. https://doi.org/10.1016/j.future.2010.03.004

[7] B. Hayes, Cloud computing, Communications of the ACM 51 (7), 2008. https://doi.org/10.1145/1364782.1364786

[8] Hai H.Wang, NickGibbins, TerryR.Payne , Domenico Redavid, “A formal model of the Semantic Web Service Ontology (WSMO)”, Information Systems 2012. https://doi.org/10.1016/j.is.2011.07.003

[9] Jena - A Semantic Web Framework for Java, http://incubator.apache.org/jena/

[10] Kristina Toutanova, et al., “Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network.”, in Proceedings of HLT-NAACL 2003, pp. 252-259. https://doi.org/10.3115/1073445.1073478

[11] W. Li, C.Yang, et al. “Semantic-based web service discovery and chaining for building an Arctic spatial data infrastructure”, Computers & Geosciences, 2011. https://doi.org/10.1016/j.cageo.2011.06.024

[12] Maynard, D, et al., “NLP Techniques for Term Extraction and Ontology Population”, in Buitelaar, P, Cimiano, P., eds.: Bridging the Gap between Text and Knowledge - Selected Contributions to Ontology Learning and Population from Text, IOS Press, 2008.

[13] Min Liu, Weiming Shen, Qi Hao, Junwei Yan, “A weighted ontology-based semantic similarity algorithm for web service”, Expert Systems with Applications, 2009. https://doi.org/10.1016/j.eswa.2009.04.034

[14] Prudhommeaux, E., and Seaborne, A. (2008). “SPARQL query language for RDF”, W3C recommendation, W3C, available at http://www.w3.org/TR/2008/REC-RDF-SPARQL-query-20080115.

[15] M. Risch, J. Altmann, “Enabling Open Cloud Markets through WS-Agreement Extensions”, Service Level Agreements in Grids Workshop, in conjunction with GRID 2009, CoreGRID Springer Series, Banff, Canada, 2009. https://doi.org/10.1007/978-1-4419-7320-7_10

[16] L. Srinivasan and J. Treadwell, "An overview of Service Oriented Architecture, Web Services and Grid Computing", http://h71028.www7.hp.com/ERC/downloads/SOA-Grid-HP-WhitePaper.pdf

[17] Vuong Xuan Tran, Hidekazu Tsuji, Ryosuke Masuda, “A new QoS ontology and its QoS-based ranking algorithm for Web services”, Simulation Modelling Practice and Theory, 2009.

[18] World Wide Web Consortium, “OWL- S: Semantic Markup for Web Services”, World Wide Web Consortium (W3C) Member Submission, 2004, available at http://www.w3.org/submission/owl-s.

[19] Yi Zhao, Zhong Li, Xia Wang, Wolfgang A. Halang, “Decision support in e-business based on assessing similarities between ontologies”, Knowledge-Based Systems, 2012. http://www.cloudbook.net/directories/product-services/cloud-computing-directory. https://doi.org/10.1016/j.knosys.2011.08.020

[20] Saravanan S., Hailu M., Gouse G.M., Lavanya M., Vijaysai R. (2019) Optimized Secure Scan Flip Flop to Thwart Side Channel Attack in Crypto-Chip. In: Zimale F., Enku Nigussie T., Fanta S. (eds) Advances of Science and Technology. ICAST 2018. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 274. Springer, Cham. https://doi.org/10.1007/978-3-030-15357-1_34

[21] Saravanan S., Hailu M., Gouse G.M., Lavanya M., Vijaysai R. (2019) Design and Analysis of Low-Transition Address Generator. In: Zimale F., Enku Nigussie T., Fanta S. (eds) Advances of Science and Technology. ICAST 2018. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 274. Springer, Cham. https://doi.org/10.1007/978-3-030-15357-1_19

[22] Dr G. Mohammed Gouse, Chiai Mohammed Haji, Dr Saravanan, Improved Reconfigurable based Lightweight Crypto Algorithms for IoT based Applications, Journal of Advanced Research in Dynamical Control Systems, Volume: 10 Issue: 12 Pages: 186-193

[23] B.Manoj, K.V.K.Sasikanth, M.V.Subbarao and V Jyothi Prakash,Analysis of Data Science with the use of Big Data, International Journal of Advanced Trends in Computer Science and Engineering, Volume 7 No. 6 (2018),Pages: 87 – 90 https://doi.org/10.30534/ijatcse/2018/02762018

[24] A.S. Elhassan and W.S. Elhassan, Assessment Instruments for Accreditation: A Data Management System Design & Implementation, International Journal of Advanced Trends in Computer Science and Engineering, Volume 8 No. 1.1 (2019) S I, Pages: 1 – 7. https://doi.org/10.30534/ijatcse/2019/0181.12019