

Quantitative And Qualitative Evaluation Of Methodology To Manage Victim industry demands

Manish Saxena, Malik Pratap

Assistant Professor, Babasaheb Bhimrao University, India

Article Information

Received : 02 Oct 2024
Revised : 06 Oct 2024
Accepted : 18 Oct 2024
Published : 22 Oct 2024

Abstract— The current practices of software industry demands development of a software within time and budget which is highly productive. The traditional approach of developing a software from scratch requires considerable amount of effort. To overcome the drawback a reuse drive software development approach is adopted. However there is a dire need for realizing effective software reuse. This paper presents several measures of reusability and presents a methodology of reconfiguring the victim components. The CBO measure helps in identifying the component to be reconfigured. The proposed strategy is simulated using HR portal domain specific component system.

Corresponding Author:

Manish Saxena

Keywords: *Reusability, Component Measure, Victim Component, Reconfigurable component.*

Copyright © 2024: Manish Saxena, Malik Pratap, This is an open access distribution, and reproduction in any medium, provided Access article distributed under the Creative Commons Attribution License the original work is properly cited License, which permits unrestricted use.

Citation: Manish Saxena, Malik Pratap, “Quantitative And Qualitative Evaluation Of Methodology To Manage Victim industry demands”, Journal of Science, Computing and Engineering Research, 7(10), October 2024.

I. INTRODUCTION

Effective software reuse helps in development of quality product within time and budget. This also helps in reducing the high effort needed for testing and maintenance of the software products. Effective software reuse is still a challenging task. Some of the components may not be reused effectively because of their highly cohesive nature. Such components have to be reconfigured. In this paper several quantitative measures are presented for software reusability. Further an approach is presented to identify the component(s) which is less reused. Later CBO measure is used to identify the reconfigurable components and a methodology is presented for reconfiguring the component(s). The remaining part of this paper is organized as follows: section-2 presents the related work, section-3 presents basic concepts of software reuse & domain engineering, section-4 describes the component measures which are realized on HR portal system, section -5 presents a methodology of identifying and realizing the reconfigurable component(s) and section -6 concludes the paper.

II. RELATED WORK

Reusability metrics defines an approach to measure reusable components. Several reusability metrics have been proposed in literature which has less emphasis on quantitative metrics. In [10] reusability metrics are based upon four characteristics viz. Self descriptiveness, modularity, portability and platform independence. However their weights are assumed based on assumed value which is qualitative in nature. In [11] a subset of reusability metrics are proposed. Though this approach is more efficient than non-automatable techniques, however the goal is to reuse the components interfaces only. This approach lacks the

reuse measures at the design level. Zhongjie Wang et.al [12], proposed that the deficiencies of the components which are not suitable for reuse has to be redesigned. However no such approach to identify such components is presented. In this paper certain software reusability measures are presented. Later a quantitative strategy is presented to identify the components which are to be reconfigured.

III. SOFTWARE REUSE

Software Reuse is the use of available software or to build new software from software knowledge. Reusable assets can be either reusable software or software knowledge. Reusability is a property of a software asset that indicates it's probability of reuse [1]. Software Reuse means the process that use “designed software for reuse” again and again [2]. By software reusing, we can manage complexity of software development, increase product quality and makes faster production in the organization. Recently, design reuse has become popular with (object-oriented) class libraries, application frameworks, design patterns and along with the source code [3]. Jianli et al. proposed two complementary methods for reusing existing components. Among them one allows component evolution itself, which is achieved with binary class level inheritance across component modules. The other is by defined semantic entity so that they can be assembled at compile time or bind at runtime. Although component containment still is the main reuse model that leads to contribute the software product lines development [4]. Regarding the components much information has to be collected, maintained and processed for the retrieval of the components. Maurizio has proposed a methodology to automatically build a software catalogue the tools for archiving and retrieval of information are presented

[5]. Software Reuse can be broadly divided into two categories viz. product reuse and process reuse. The product reuse includes the reuse of a software component and by producing a new component as a result of module integration and construction. The process reuse represents the reuse of legacy component from repository. These components may be either directly reused or may need a minor modification. The modified software component can be archived by versioning these components. The components may be classified and selected depending on the required domain. [6].

3.1 Domain Engineering Software Reuse can be improved by identifying objects and operations for a class of similar systems, i.e. for a particular domain. In the context of software engineering domains are application areas [7]. There are various definitions of what a domain is. Czarnecki's defines [8]:" an area of knowledge scoped to maximize the satisfaction of the requirements of stakeholders, which includes concepts

88 International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.2, March 2012 and terminology understood by practitioners in the area and the knowledge of how to build (part of) systems in the area". Domain Engineering is a process in which the reusable component is developed and organized and in which the architecture meeting requirements of the domain is designed [9]. Domain Engineering can be defined by identifying the candidate domains and performing domain analysis and domain implementation which includes both application engineering and component engineering. Domain Analysis is a continuing process of creating and maintaining the reuse infrastructure in a certain domain. The main objective of domain analysis is to make the whole information readily available. The relevant components (if available) has to be extracted from the repository rather than building the new components from the scratch for a particular domain. Domain Analysis mainly focuses on reusability of analysis and design, but not code. This can be achieved by building common architectures, generic models or specialized languages that additionally improve the software development process in the specific problem area of the domain. A vertical domain is a specific class of systems. A horizontal domain contains general software parts being used across multiple vertical domains. Mathematical functions libraries container classes and UNIX tools are the examples of horizontal reuse. The purpose of domain engineering is to identify objects and operations of a class in a particular problem domain [7]. In the process of domain analysis, each component identified can be categorized as follows.

- General-purpose components: These components can be used in various applications of different domains (horizontal reuse).
- Domain-specific components: They are more specific and can be used in various applications of one domain (vertical reuse).
- Product-specific components: They are very specific and custom-built for a certain

application, they are not reusable or only useful to a small extent.

3.2 HR Portal Application The system is designed in such a way that the client can interact with the web tier and business tier and can connect to the Data Access Object(DAO) component. The web-tier component consists of the JSP's and Servlets.The Business tier consists of the EJB's.The DAO's consists of the classes with its objects communicating to the database.The web-tier components are HttpServlet, HRProcessServlet, Login Servlet, InterviewResultServlet and RegistrationServlet classes.The Business-tier components are EmployeeBean, InterviewResultsBean, HRProcessBean are the three stateless bean classes.The DAO components are BaseDAO, EmployeeDAO, InterviewDAO, HRDAO, ProcessDAO classes.

89 International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.2, March 2012 90 Figure 1. Components of HR Portal Application

IV. COMPONENT MEASURES

Most of the reuse driven approaches often maintains a repository of reusable components. However an approach is needed to identify those components which are not reused or might have been used very less. Such components are known as Victim components. These components have to be modified or reconfigured to achieve high reusability. The Object Oriented metrics are useful for measuring the effectiveness of component reusability. These metrics are also helpful in identifying the victim components. Some of these metrics are described as follows.

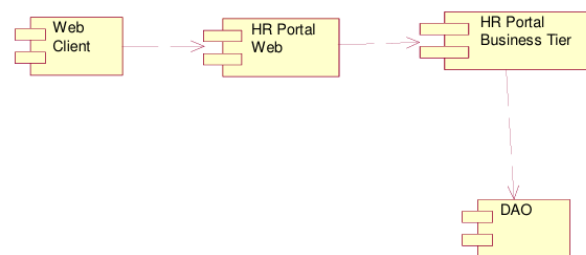


Figure 1. Components of HR Portal Application

4.1 Weighted Component Measure (WCM) Each component may have several classes and each class may have several methods $m_1, m_2, m_3, \dots, m_n$. The complexity of all the methods of a particular class is evaluated to find weighted method measure of a class (WMC). The Weighted Component Measure (WCM) is the sum of Weighted Method Measure of Classes which are composed in the given component.

$$WCM_1 = WMC_1 + WMC_2 + \dots + WMC_n$$

where WMC_i is a Weighted Measure of a classes $i=1$ to n The weighted measure of a particular class in a given component is evaluated as

$$WMC_1 = C[M_1] + C[M_2] + C[M_3] + \dots + C[M_K]$$

Where M_1 to M_K are the methods of a class. The complexity of a method is

evaluated as follows. $C[M_j]$ = Number of edges – Number of vertices + 1 (in a particular flow graph) or = Number of Decision Elements of a method + 1 The HR Portal Application consists of three components i.e. Web-Tier, Business-Tier, Data Access Object. These three components consists of 13 classes. Web tier component contains 5 HR Portal Web HR Portal Business Tier DAO Web Client International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.2, March 2012 classes, business tier contains 3 classes and DAO component 5 classes. The Weighted Component Measure (WCM) of the Webtier component is as follows $WCM_{WB} = W_{MCHR} + W_{MCINT} + W_{MCRS} + W_{MCLS}$ The Weighted Measures of a class are as follows: $W_{MCHR} = C[M_{PR}] + C[MR]$ $W_{MCINT} = C[M_{PR}] + C[MAR]$ $W_{MCRS} = C[M_{PR}] + C[MRG]$ $W_{MCLS} = C[MPS]$ The complexity of the methods is evaluated as follows: $C[M_{PR}] = 11 + 1 = 12$ $C[MR] = 10 + 1 = 11$ $C[M_{PR}] = 11 + 1 = 12$ $C[MAR] = 6 + 1 = 7$ $C[M_{PR}] = 11 + 1 = 12$ $C[MRG] = 8 + 1 = 9$ $C[MPS] = 11 + 1 = 12$ $W_{MCHR} = C[M_{PR}] + C[MR] = 12 + 11 = 23$ $W_{MCINT} = C[M_{PR}] + C[MAR] = 12 + 7 = 19$ $W_{MCRS} = C[M_{PR}] + C[MRG] = 12 + 9 = 21$ $W_{MCLS} = C[MPS] = 12$ The Weighted Component Measure(WCM) of the Webtier component is as follows: $W_{CM_{WB}} = W_{MCHR} + W_{MCINT} + W_{MCRS} + W_{MCLS} = 23 + 19 + 21 + 12 = 75$ The Weighted Component Measure(WCM) of the Business tier component is as follows: $W_{CM_{BT}} = W_{MCEMP} + W_{MCHRPB} + W_{MCIR}$ The Weighted Measures of a class are as follows: $W_{MCEMP} = C[MGS] + C[MSS] + C[MA] + C[MACP]$ $W_{MCHRPB} = C[MRC] + C[MR]$ $W_{MCIR} = C[MVR] + C[MAIR]$ The cyclomatic complexity of a method is evaluated as follows: $C[MGS] = 5 + 1 = 6$ $C[MSS] = 6 + 1 = 7$ $C[MA] = 7 + 1 = 8$ $C[MACP] = 21 + 1 = 22$ $C[MRC] = 12 + 1 = 13$ 91 International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.2, March 2012 $C[MR] = 10 + 1 = 11$ $C[MVR] = 9 + 1 = 10$ $C[MAIR] = 13 + 1 = 14$ The Weighted Measures of a class are as follows: $W_{MCEMP} = C[MGS] + C[MSS] + C[MA] + C[MACP] = 6 + 7 + 8 + 22 = 43$ $W_{MCHRPB} = C[MRC] + C[MR] = 13 + 11 = 24$ $W_{MCIR} = C[MVR] + C[MAIR] = 10 + 14 = 24$ The Weighted Component Measure(WCM) of the Business tier component is as follows: $W_{CM_{BT}} = W_{MCEMP} + W_{MCHRPB} + W_{MCIR} = 43 + 24 + 24 = 91$ The Weighted Component Measure(WCM) of the Data Access Object component is as follows: $W_{CM_{DAO}} = W_{M_{CBDO}} + W_{M_{CEDO}} + W_{M_{CIDO}} + W_{M_{CHRDO}} + W_{M_{CPDO}}$ The Weighted Measures of a class are as follows: $W_{M_{CBDO}} = C[MGC] + C[MCC]$ $W_{M_{CHRO}} = C[MRC] + C[MRE]$ $W_{M_{CEDO}} = C[MGP] + C[MGE] + C[MAE] + C[MGEE] + C[MRC]$ $W_{M_{CIDO}} = C[MVIR] + C[MAIR]$

$W_{M_{CPDO}} = C[MRC] + C[MAT]$ The cyclomatic complexity of a method is evaluated as follows: $C[MGC] = 34 + 1 = 35$ $C[MCC] = 4 + 1 = 5$ $C[MRC] = 12 + 1 = 12$ $C[MRE] = 13 + 1 = 14$ $C[MGP] = 9 + 1 = 10$ $C[MGE] = 13 + 1 = 14$ $C[MAE] = 21 + 1 = 22$ $C[MGEE] = 14 + 1 = 15$ $C[MRC] = 22 + 1 = 23$ $C[MVIR] = 9 + 1 = 10$ $C[MAIR] = 13 + 1 = 14$ $C[MRC] = 29 + 1 = 30$ $C[MAT] = 6 + 1 = 7$ The Weighted Measures of a class are as follows: $W_{M_{CBDO}} = C[MGC] + C[MCC] = 35 + 5 = 40$ $W_{M_{CHRO}} = C[MRC] + C[MRE] = 12 + 14 = 26$ $W_{M_{CEDO}} = C[MGP] + C[MGE] + C[MAE] + C[MGEE] + C[MRC] = 10 + 14 + 22 + 15 + 23 = 84$ 92 International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.2, March 2012 $W_{M_{CIDO}} = C[MVIR] + C[MAIR] = 10 + 14 = 24$ $W_{M_{CPDO}} = C[MRC] + C[MAT] = 30 + 7 = 37$ The Weighted Component Measure(WCM) of the Data Access Object component is as follows: $W_{CM_{DAO}} = W_{M_{CBDO}} + W_{M_{CEDO}} + W_{M_{CIDO}} + W_{M_{CHRDO}} + W_{M_{CPDO}} = 40 + 84 + 24 + 27 + 37 = 212$ 4.2 Depth of Inheritance Tree Measure The depth of inheritance is a length from the node where the class is located to the root of the tree. The depth of inheritance tree measures in the maximum length from the node to the root. The DIT for the Web-tier component is equal to 3. The DIT for the DAO Component is equal to 2. The DIT for the business-tier component is equal to 3. 4.3 Number of Children Measure The Number of Children (NOC) measure quantifies the number of immediate subclasses subordinated to a class in the class hierarchy.

Table 1. Component Management Relation

Component	Count of Reuse
Webtier	12
Businesstier	5
DAO	18

It measures how many subclasses are inherits the methods of the parent. The NOC for a class `HTTPServlet` in component Webtier is 4. The NOC for a class `BaseDAO` in DAO component is also 4. At any particular point of time, if the designer wants to know about which part of the system is not effectively reused then a lookup is to be performed on the component management relation. A Central repository maintains a table for managing a component reuse. This table contains two fields. One specifies the name of the component and the count specifies the number of times the component was reused by several systems. Table 1. Component Management Relation Component Count of Reuse Webtier 12 Businesstier 5 DAO 18 Table 1. specifies

the list of HR portal system components which were reused by several applications. If any component is not used frequently, they are termed as victim components. As Business-tier component was used only 5 times, it can be a candidate of victim component. The 93 International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.2, March 2012 victim component has to be reconfigured by dividing it into several parts to increase the reusability count in future. However reconfiguring the non-victim components may also enhance the reusability of the system. The idea is to reconfigure those components which are highly cohesive. Hence, a Coupling Between Object Measure (CBOM) of such components is to be evaluated.

V. COMPONENT RECONFIGURATION

Coupling represents the strength of relationship between the components of the system. A well structured system demands loose coupling and a tight cohesion. Those components which are highly cohesive in nature needs to be reconfigured. Coupling between object measure is used to identify the highly cohesive components. Coupling between object measure (CBOM) for a component is defined as the number of invocations by the specified component. Those components whose CBOM is high or those component(s) of the system whose CBOM is greater than certain scalar value are the components which needs to be reconfigured at the earliest. Hence a reconfigurable component (Cr) can be identified as $Cr = \text{Max} \{ \text{CBOM}(C_i) \}$, where C_1, C_2, \dots, C_n are components of the system (or) $Cr_j = \{ \text{CBOM}(C_i) > P \}$, where P is a scalar value whose value differs from one domain to the other. The following figure specifies the number of invocations by different components in HR portal application.

HR Portal - Method	Self time [%]	Self time	Invocations
com.mycompany.hr.dao.BaseDAO.getConnection()		1267 ms (41.4%)	50
com.mycompany.hr.dao.EmployeeDAO.addCandidateProfile		946 ms (28.9%)	20
com.mycompany.hr.dao.EmployeeDAO.addEmployeeCredentials		624 ms (19.4%)	20
com.mycompany.hr.dao.EmployeeDAO.authenticateEmployee		65.8 ms (2.0%)	10
org.apache.jsp.Login_jsp._jspService		54.6 ms (1.6%)	25
com.mycompany.hr.process.EmployeeBeanRemote.RemoteDynamicSub.addCandidateProfile		30.8 ms (0.9%)	20
com.mycompany.hr.process.EmployeeBeanRemote.RemoteDynamicSub.authenticate		15.2 ms (0.4%)	10
com.mycompany.hr.service.HRProcessService.processRequest		11.3 ms (0.3%)	22
com.mycompany.hr.process.EmployeeBeanRemote.RemoteDynamicSub.addCredentials		8.7 ms (0.2%)	10
org.apache.jsp.AddCandidate_jsp._jspService		5.47 ms (0.1%)	23
com.mycompany.hr.service.LoginService.processRequest		3.21 ms (0.1%)	10
org.apache.jsp.Welcome_jsp._jspService		1.88 ms (0.0%)	6
com.mycompany.hr.process.EmployeeBeanRemote.authenticate		1.17 ms (0.0%)	10
org.apache.jsp.ViewProfile_jsp._jspService		0.85 ms (0.0%)	3
com.mycompany.hr.service.HRProcessService.doGet		0.36 ms (0.0%)	22
com.mycompany.hr.dao.BaseDAO.doGet		0.235 ms (0.0%)	50
com.mycompany.hr.dao.EmployeeDAO.doGet		0.195 ms (0.0%)	20
com.mycompany.hr.process.EmployeeBeanRemote.addCredentials		0.185 ms (0.0%)	10
com.mycompany.hr.service.LoginService.doPost		0.177 ms (0.0%)	20
com.mycompany.hr.process.EmployeeBeanRemote.addCandidateProfile		0.175 ms (0.0%)	20
com.mycompany.hr.service.EmployeeCredentials.doGet		0.117 ms (0.0%)	60
com.mycompany.hr.process.EmployeeBeanRemote.Wrapper.addCandidateProfile		0.114 ms (0.0%)	20
com.mycompany.hr.process.EmployeeBeanRemote.Wrapper.addCredentials		0.109 ms (0.0%)	20
com.mycompany.hr.service.BaseDAO.doGet		0.104 ms (0.0%)	40
com.mycompany.hr.process.EmployeeBeanRemote.Wrapper.authenticate		0.068 ms (0.0%)	10
com.mycompany.hr.service.LoginService.doGet		0.064 ms (0.0%)	2
com.mycompany.hr.service.HRProcessService.doGet		0.033 ms (0.0%)	1
com.mycompany.hr.service.HRProcessService.doPost		0.028 ms (0.0%)	1

Figure 2. Invocations occurred for different components of HR Portal Application Based on the invocations specified in the above figure, CBOM is evaluated to identify one reconfigurable component. As there are three components of the HR Portal application viz., Web-tier, Business-tier and DAO, the component with highest CBOM is the candidate for reconfigurable component. $Cr = \text{Max} \{ \text{CBOM}(\text{WBR}), \text{CBOM}(\text{BR}), \text{CBOM}(\text{DAO}) \}$ $Cr = \text{Max} \{ 180, 95, 224 \} =$

224 Hence the DAO component has to further reconfigured to reduce the coupling. This is achieved by dividing the DAO component into two sub DAO components viz., DAO1 & DAO2. The division of the DAO component makes the DAO component less cohesive. This is realized later by evaluating the CBO measure of DAO1 & DAO2

VI. CONCLUSION

Component reusability helps in developing quality product as the component in the repository is successfully tested. Most of the reusability metrics proposed in literature or either qualitative or they realize only interface reusability metrics. In this paper an effort was made to propose reusable quantitative measures. Components which are less used are identified from the repository known as victim components. Further to achieve high reusability, certain components which are highly cohesive in nature have to be reconfigured. An approach using CBOM was proposed to identify such reconfigurable components. In future, strategies to reconfigure the components are to be realized.

REFERENCES

- [1]. P. Nirmala, T. Manimegalai, J. R. Arunkumar, S. Vimala, G. Vinoth Rajkumar, Raja Raju, "A Mechanism for Detecting the Intruder in the Network through a Stacking Dilated CNN Model", Wireless Communications and Mobile Computing, vol. 2022, Article ID 1955009, 13 pages, 2022. <https://doi.org/10.1155/2022/1955009>.
- [2]. D. Sathyanarayanan, T. S. Reddy, A. Sathish, P. Geetha, J. R. Arunkumar and S. P. K. Deepak, "American Sign Language Recognition System for Numerical and Alphabets," 2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE), Chennai, India, 2023, pp. 1-6, doi: 10.1109/RMKMATE59243.2023.10369455.
- [3]. J. R. Arunkumar, Tägele berihun Mengist, 2020" Developing Ethiopian Yirgacheffe Coffee Grading Model using a Deep Learning Classifier" International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-9 Issue-4, February 2020. DOI: 10.35940/ijitee.D1823.029420.
- [4]. Ashwini, S., Arunkumar, J.R., Prabu, R.T. et al. Diagnosis and multi-classification of lung diseases in CXR images using optimized deep convolutional neural network. Soft Comput (2023). <https://doi.org/10.1007/s00500-023-09480-3>
- [5]. J.R.Arunkumar, Dr.E.Muthukumar," A Novel Method to Improve AODV Protocol for WSN" in Journal of Engineering Sciences" ISSN NO: 0377-9254 Volume 3, Issue 1, Jul 2012.
- [6]. R. K, A. Shameem, P. Biswas, B. T. Geetha, J. R. Arunkumar and P. K. Lakineni, "Supply Chain Management Using Blockchain: Opportunities, Challenges, and Future Directions," 2023 Second International Conference on Informatics (ICI), Noida, India, 2023, pp. 1-6, doi: 10.1109/ICI60088.2023.10421633.

- [7]. Arunkumar, J. R. "Study Analysis of Cloud Security Challenges and Issues in Cloud Computing Technologies." *Journal of Science, Computing and Engineering Research* 6.8 (2023): 06-10.
- [8]. J. R. Arunkumar, R. Raman, S. Sivakumar and R. Pavithra, "Wearable Devices for Patient Monitoring System using IoT," 2023 8th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2023, pp. 381-385, doi: 10.1109/ICCES57224.2023.10192741.
- [9]. S. Sugumaran, C. Geetha, S. S, P. C. Bharath Kumar, T. D. Subha and J. R. Arunkumar, "Energy Efficient Routing Algorithm with Mobile Sink Assistance in Wireless Sensor Networks," 2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 2023, pp. 1-7, doi: 10.1109/ACCAI58221.2023.10201142.
- [10]. R. S. Vignesh, V. Chinnammal, Gururaj,D, A. K. Kumar, K. V. Karthikeyan and J. R. Arunkumar, "Secured Data Access and Control Abilities Management over Cloud Environment using Novel Cryptographic Principles," 2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 2023, pp. 1-8, doi: 10.1109/ACCAI58221.2023.10199616.
- [11]. Syamala, M., Anusuya, R., Sonkar, S.K. et al. Big data analytics for dynamic network slicing in 5G and beyond with dynamic user preferences. *Opt Quant Electron* 56, 61 (2024). <https://doi.org/10.1007/s11082-023-05663-2>
- [12]. Krishna Veni, S. R., and R. Anusuya. "Design and Study Analysis Automated Recognition system of Fake Currency Notes." *Journal of Science, Computing and Engineering Research* 6.6 (2023): 16-20.
- [13]. V. RamKumar, S. Shanthi, K. S. Kumar, S. Kanageswari, S. Mahalakshmi and R. Anusuya, "Internet of Things Assisted Remote Health and Safety Monitoring Scheme Using Intelligent Sensors," 2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 2023, pp. 1-8, doi: 10.1109/ACCAI58221.2023.10199766.
- [14]. R. S. Vignesh, R. Sankar, A. Balaji, K. S. Kumar, V. Sharmila Bhargavi and R. Anusuya, "IoT Assisted Drunk and Drive People Identification to Avoid Accidents and Ensure Road Safety Measures," 2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 2023, pp. 1-7, doi: 10.1109/ACCAI58221.2023.10200809.
- [15]. I. Chandra, G. Sowmiya, G. Charulatha, S. D, S. Gomathi and R. Anusuya, "An efficient Intelligent Systems for Low-Power Consumption Zigbee-Based Wearable Device for Voice Data Transmission," 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), Chennai, India, 2023, pp. 1-7, doi: 10.1109/ICECONF57129.2023.10083856.
- [16]. G. Karthikeyan, D. T. G, R. Anusuya, K. K. G, J. T and R. T. Prabu, "Real-Time Sidewalk Crack Identification and Classification based on Convolutional Neural Network using Thermal Images," 2022 International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 2022, pp. 1266-1274, doi: 10.1109/ICACRS55517.2022.10029202.
- [17]. R. Meena, T. Kavitha, A. K. S, D. M. Mathew, R. Anusuya and G. Karthik, "Extracting Behavioral Characteristics of College Students Using Data Mining on Big Data," 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), Chennai, India, 2023, pp. 1-7, doi: 10.1109/ICECONF57129.2023.10084276.
- [18]. S. Bharathi, A. Balaji, D. Irene. J, C. Kalaivanan and R. Anusuya, "An Efficient Liver Disease Prediction based on Deep Convolutional Neural Network using Biopsy Images," 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2022, pp. 1141-1147, doi: 10.1109/ICOSEC54921.2022.9951870.
- [19]. I. Chandra, G. Sowmiya, G. Charulatha, S. D, S. Gomathi and R. Anusuya, "An efficient Intelligent Systems for Low-Power Consumption Zigbee-Based Wearable Device for Voice Data Transmission," 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), Chennai, India, 2023, pp. 1-7, doi: 10.1109/ICECONF57129.2023.10083856.
- [20]. Revathi, S., et al. "Developing an Infant Monitoring System using IoT (INMOS)." *International Scientific Journal of Contemporary Research in Engineering Science and Management* 6.1 (2021): 111-115.
- [21]. J.R.Arunkumar, Dr.E.Muthukumar, A Novel Method to Improve AODV Protocol for WSN in *Journal of Engineering Sciences* ISSN NO: 0377-9254 Volume 3, Issue 1, Jul 2012.
- [22]. R. S. Vignesh, A. Kumar S, T. M. Amirthalakshmi, P. Delphy, J. R. Arunkumar and S. Kamatchi, "An Efficient and Intelligent Systems for Internet of Things Based Health Observance System for Covid 19 Patients," 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), Chennai, India, 2023, pp. 1-8, doi: 10.1109/ICECONF57129.2023.10084066.
- [23]. I. Chandra, K. V. Karthikeyan, R. V, S. K, M. Tamilselvi and J. R. Arunkumar, "A Robust and Efficient Computational Offloading and Task Scheduling Model in Mobile Cloud Computing," 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), Chennai, India, 2023, pp. 1-8, doi: 10.1109/ICECONF57129.2023.10084293.
- [24]. R. K. A. Shameem, P. Biswas, B. T. Geetha, J. R. Arunkumar and P. K. Lakineni, "Supply Chain Management Using Blockchain: Opportunities, Challenges, and Future Directions," 2023 Second International Conference on Informatics (ICI), Noida, India, 2023, pp. 1-6, doi: 10.1109/ICI60088.2023.10421633.
- [25]. J. R. Arunkumar, and R. Anusuya, "OCHRE: A Methodology for the Deployment of Sensor Networks." *American Journal of Computing Research Repository*, vol. 3, no. 1 (2015): 5-8.