International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
***Article Received:** 25 November 2023 **Revised:** 12 December 2023 **Accepted:** 30 January 2024*

Analysis on Agile Software Development Using Cloud Computing based on Agile Methodology and Scrum Framework

Prabu Ravichandran,
Sr. Data Architect, AWS, Amazon, Raleigh, NC, USA
Prabu.ravichandran07@gmail.com

ABSTRACT

The advent of agile software development processes was a response to the problems with more traditional approaches. Popular agile approaches utilized in software development include Scrum, Extreme Programming, and Kanban.A key component of an Agile methodology is the emphasis on customer-developer cooperation and the promotion of self-organization within development teams. Teams accomplish this by implementing various Agile techniques into their projects. There are teams that stick to a single practice and others that mix it up. Scrum, user stories, pair programming, burndown charts, and stand-ups are the most popular methods. The use of cloud computing has several benefits, including the potential to scale, improve communication, and lower overall costs. Here we take a look at the ADCC framework, which was suggested in a previous study, and see how well it mixes Agile Development with Cloud Computing. In order to put the design into action, the Malaysia Research and Education Network, or MyRen cloud, is used. One way to check the notion is using a case study. Before diving into the case study, participants get a rundown of the ADCC framework. Case study results demonstrate that ADCC framework usage improves agile method performance. Results from both centralized and decentralized agile development environments are used to evaluate the evolution.

Key words: Cloud, AGILE methodology, Scrum framework.

1. INTRODUCTION

Adaptability is key when it comes to software development, and a plan-driven approach isn't as good as agile software development. The capacity to make changes to requirements at any point in the software development cycle is an example of the flexibility that may be observed [1]. A self-organizing development team is encouraged to be a part of an agile methodology, which places an emphasis on collaboration between clients and developers [2]. This goal can be accomplished through the implementation of various Agile techniques in projects. As software development progresses, it is anticipated that agile approaches will lead to enhanced formal and informal communication [3]. Some techniques that impact communication are iteration planning meetings, product backlogs, sprint planning, daily meetings, iteration retrospectives, and open office spaces [4,5].

Reviews of iterations are another example. Pikkarainen and colleagues [6,7,8] identified practices that are likely to impact team communication as one of their primary areas of research attention. There is a lack of data regarding how various methods and practices when combined affect the total time and money needed to complete the project. There is a limited quantity of evidence regarding this impact. All of this speaks to the fact that it is essential to carry out research on the several approaches and procedures that are utilized in Agile initiatives. The scope of this study encompasses the investigation of the several methodologies that are utilized in Agile software development projects, as well as the impact that the combination of these methodologies has on the outcomes of the project.

With the Agile methodology that has dominated software development for the past two decades, the Scrum framework has emerged as the most popular approach to the Agile methodology. There is a huge majority of software development teams who utilize a project management technique that can be defined as some kind of "Agile." This is despite the fact that the extremely fragmented nature of the business makes it challenging to obtain meaningful figures.

In the most current assessment on the state of agile, which was published on the 15th, it was discovered that 94% of the firms that the 1382 international respondents worked for had implemented Agile. Sixty-six percent of those that utilized a Scrum framework to implement a variant of the Agile

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
Article Received: *25 November 2023* ***Revised:*** *12 December 2023* ***Accepted:*** *30 January 2024*

methodology at the team level were among the majority of those who did so.

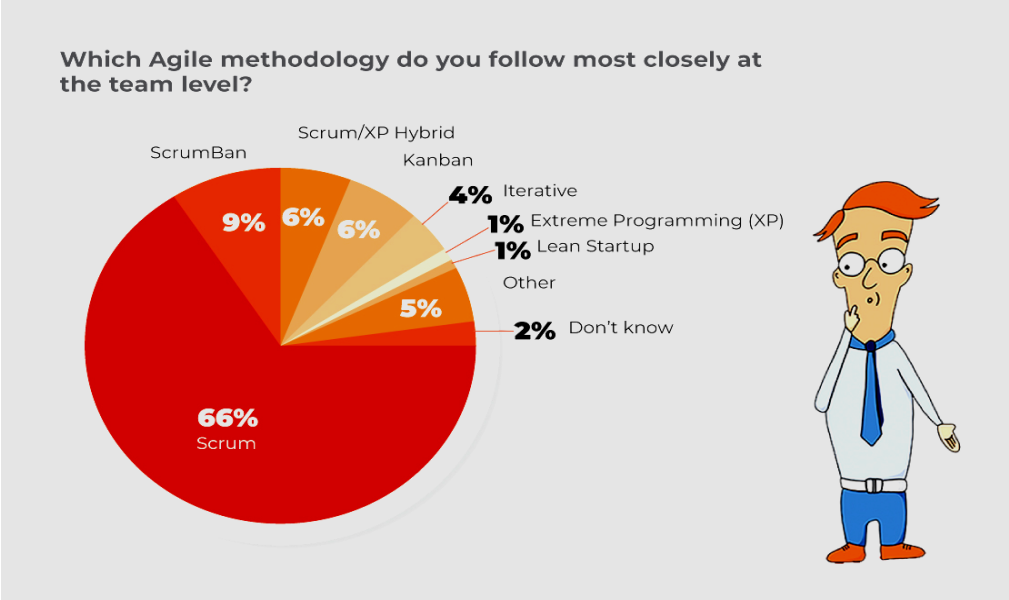

Figure 1: Agile methodology at team level.

The combination of Agile and Scrum is, by a significant margin, the method to software development that is most widely adopted in the year 2021. Further highlighting the significance of the Scrum framework to modern software development is the fact that an additional fifteen percent of respondents followed other close derivations of Scrum. This brings the significance of the Scrum framework into even sharper perspective.

Nine percent of respondents stated that their team used ScrumBan, which is a hybrid methodology that integrates aspects of both Scrum and Kanban. With an additional 6% being Scrum/XP, which is a combination of aspects of both Scrum and Extreme Programming (XP).

In the event that you were only able to select one method for gaining a grasp of the software development process and project management, the Scrum framework would be the one that you would choose. Because of this, Scrum is an excellent location to begin investigating the philosophy behind Agile software development as well as the more specific processes and actions that are given by the many frameworks that are included under the tent of Agile techniques.

2. LITERATURE REVIEW

In accordance with this manifesto, a total of twelve (12) core principles of agile development were also proposed. The principles of agile software development provide more context and support for why agility is so important when creating software. However, due to the terms' implementation in a distributed environment and the market's fast changing requirements, the agile standards are not rigorously enforced. Scalability, transparency, face-to-face contact, availability of experts, seamless control of development, capacity to design applications from remote places, and resource management are some of the obstacles that need to be solved[9,11].As a result of the shifting requirements, an environment is required to test new concepts. Providing resources for the purpose of testing new ideas contributes to a rise in the cost of development.

An environment that allows for the rapid testing of new ideas in the market is provided by cloud computing [12,13,14], which is a solution to the challenges that have been presented. The use of cloud computing offers the potential to lower the costs linked to agile development. This is due to the fact that it enables the sharing of data, the distribution of applications, the prioritization of activities, and the provision of infrastructure (which includes both hardware and software). Cloud computing makes the process of software creation more efficient because it eliminates the need for software patching, re-installation, and traditional installation procedures. On a pay-per-use basis, cloud services give cloud service providers with access to a repository of computational and storage resources. The agile process can be extended through the use of cloud computing, which allows for the delivery of software in a more timely manner, the reduction of expenses, and the improvement of the program's overall efficiency [15,16,17].

The concepts of technical excellence, frequent delivery, a strong user-developer connection, and the ability to accept changes at any stage of the development process are the cornerstones of agile software development. These principles

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
Article Received: *25 November 2023* ***Revised:*** *12 December 2023* ***Accepted:*** *30 January 2024*

were described earlier. The topic of how these agile features can be implemented in a cloud computing environment is one of the concerns that has to be answered. The review research will evaluate the usage of cloud computing in conjunction with agile management and development methodologies [18,19,20]. This will allow for the provision of a solution to the topic that has been posed. Taking into consideration the fact that there are already a number of studies that have been carried out in this area, the primary focus of this inquiry will be on the interoperability of cloud services and the reusability of a number of different tools. In light of this, the SLR study initially shows how research in this area has grown by extending previous findings. [21,22,23,24] In the context of agile and cloud computing, the research aims to find different techniques and compare, evaluate, and use their tools. This SLR study looks at how cloud computing has changed agile software development, what issues have come up in primary research, and what the future holds for this area of study. [25,26].

The combination of agile with cloud computing is advantageous for the creation of distributed applications, the sharing of data, the prioritization of tasks, transparency, and the construction of infrastructure. Specifically, this is due to the fact that cloud computing is having an increasing influence on the ecosystem of agile software development. When referring to agile methodologies, the phrase "ecosystem" is used to describe a system or a collection of pieces that are interconnected. Examples of ecosystems include the development environment, the interconnectedness of teams, and the entire system. One of the many advantages of cloud computing is the improvement in performance it brings to agile software development, as well as the reduction of expenses and the capacity to scale. [27,28,29] These are just some of the advantages. In addition, the agile methodology utilizes social technologies that are hosted in the cloud in order to facilitate the implementation of the communication practice that is a component of the agile methodology.[30-31] On the other hand, when it comes to academic research, there is not a lot of study that examines such discoveries, shares research gaps, and analyzes relevant concerns in both of these fields together. In their own right, agile and cloud computing contain sufficient amounts of content that has been peer-reviewed, but not both. Consequently, the focus of this review paper is on the research that has been conducted on both the agile approach and cloud computing.

3. RESEARCH METHODOLOGY

3.1 Execution of the adcc framework

The ADCC framework utilizes existing processes and technology to guarantee the creation of environments that are ideal for agile development in cloud computing. These environments foster the reusability of solutions that are already in existence and actively advocate their reuse. In Figure 2, the conceptual architecture of the ADCC framework is broken down in greater detail. When it comes to managing agile development activities, the cloud provider offers assistance in cloud services administration. Figure 2 is a representation of the workflow used by several roles, including users, testers, and developers. The red, green, and blue connecting lines show that users, testers, project managers, and developers all have separate responsibilities and roles. The red lines in Figure 2 represent one of the initial components of the ADCC framework: agile software project management.

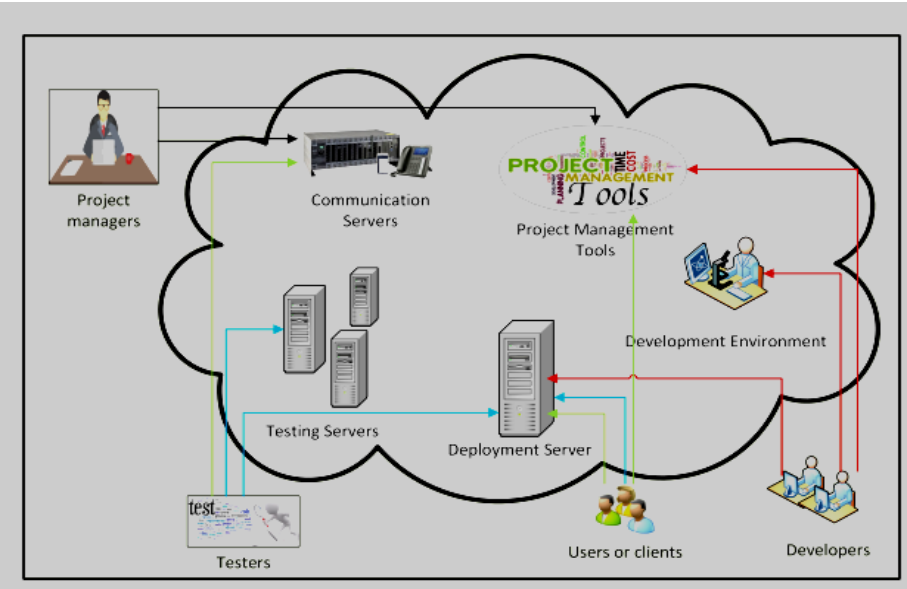

Figure 2. Designing the framework conceptually.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
Article Received: *25 November 2023* ***Revised:*** *12 December 2023* ***Accepted:*** *30 January 2024*

The second part is picking a cloud platform, which comprises servers for testing, development, and deployment. The blue lines seen in Figure 2 indicate these cloud platforms. Figure 2 shows the third component of the ADCC architecture, which includes mediums for communication and cooperation such as Google Drive, Skype, and email. The fourth component, on the other hand, is made up of code management tools like GitHub and various other similar platforms. The procedures that are involved in the implementation of the ADCC framework are delineated in the following paragraphs, as seen in Figure 3.

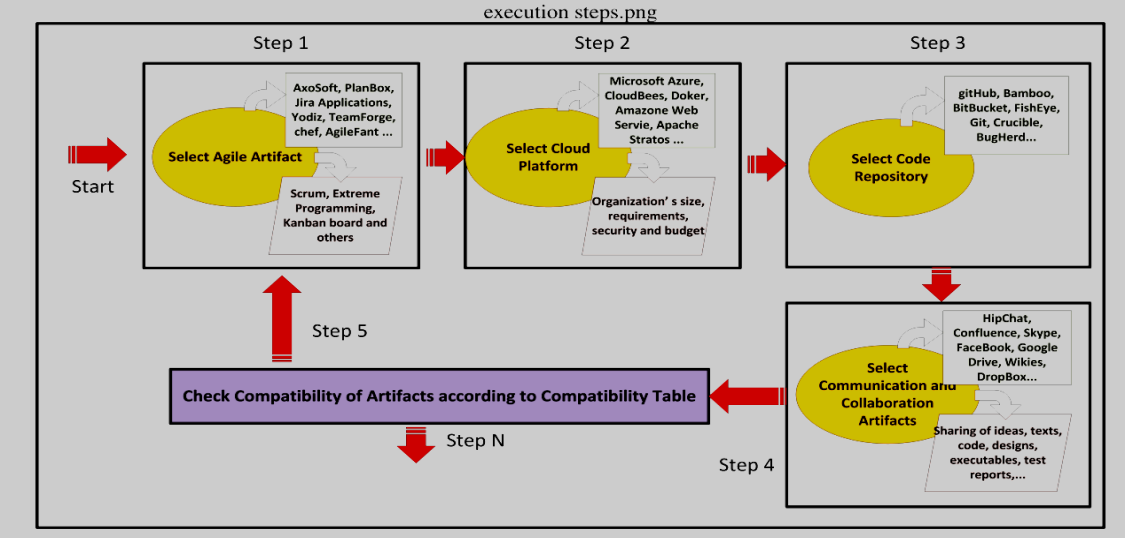

Figure 3. Systematic approach to the implementation of the conceptual design of the framework.

In order to construct an ADCC environment, the research uses the approach that is outlined in Figure 3, the specifics of which are laid forth in the following lines. The ADCC framework can be implemented in four steps, as shown in Figure 3. The first thing that has to be done is to choose agile tools, which are determined by the agile methodology and the features that are utilized by the development team. The selection of the cloud computing platform is the second phase, and it is determined by the nature of the project together with its budget, size, and level of security. When working in an agile distributed setting, the third stage involves selecting the code repository for the purpose of managing code and versioning instances. The choice of communication tools is made at the fourth step of the process. To select all of these tools in just four steps, it is important to check the compatibility of the tools, which is done in the fifth phase of the procedure. This is done in order to ensure that the tools are compatible with one another. Concerns regarding compatibility can be addressed and ultimately resolved by the utilization of the comparability table in Study. In the event that the choice that was made did not achieve the desired results, it is important to repeat steps one through four until all of the compatibility concerns have been resolved.

4. RESULTS AND STUDY

Table1. The efficiency with which on-premises or locally-based teams apply agile development or ADCC.

Development Phases	No. of Days					
	Team 1-UTM		Team 2-GCUF		Average Days	
	Using Agile Development	Using ADCC	Using Agile Development	Using ADCC	Using Agile Development	Using ADCC
Requirement elicitation	1.5	1	2.5	2	2	1.5
Planning, design and coding	33	30	39	36	36	33
Testing and deployment	7.5	6	8.5	7	8	6.5

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
Article Received: *25 November 2023* ***Revised:*** *12 December 2023* ***Accepted:*** *30 January 2024*

Within the context of the local environment, a group of personnel drawn from both GCUF and UTM worked together in a manner that was distinct and independent. Table 1 shows the outcomes for a GCUF group and a UTM group in terms of the total number of days needed to finish the development phases. The participants in both sets were first-year college students.

Table 2. The effectiveness of the team in a distributed setting through the application of agile development and ADCC.

Development Phases	No. of Days					
	Team 1-UTM-GCUF		Team 2-UTM-GCUF		Average Days	
	Using Agile Development	Using ADCC	Using Agile Development	Using ADCC	Using Agile Development	Using ADCC
Requirement elicitation	3	1.5	3	1.5	3	1.5
Planning, design and coding	37	31	43	35	40	33
Testing and deployment	10	7	8	6	9	6.5

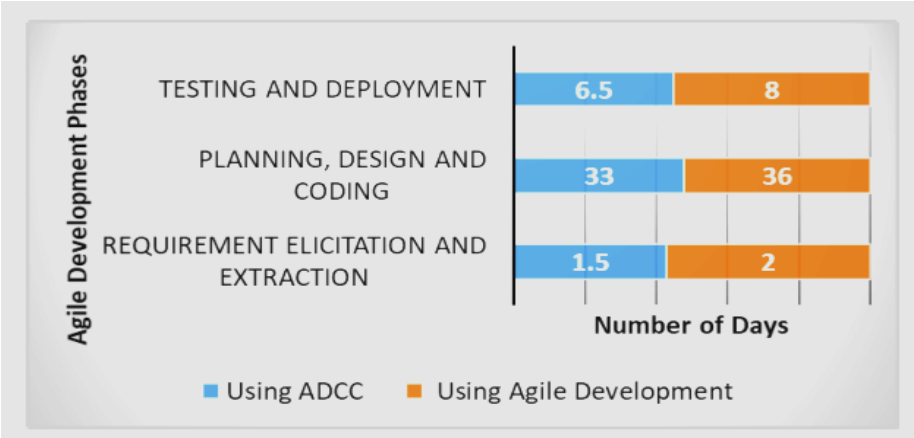

Figure4. Comparative analysis of agile phases in a local development environment, both with and without the utilization of ADCC.

The level of expertise possessed by each team is almost identical for both of the scenarios. In addition, the average of both teams is computed in order to moderate the influence of the competency. Each and every student is capable of doing the tasks of creating, coding, and testing. The role of the product owner and end user can be assumed by the class instructor and his aide in an emergency. Finding the mean is as simple as averaging the results across all teams in relation to the three phases of development. Figure 4 shows that when comparing the two scenarios, the amount of days spent on completing agile development activities is significantly different. When compared to a straightforward agile environment, Figure 6 demonstrates that the use of ADCC results in better utilization of development time in a local setting.

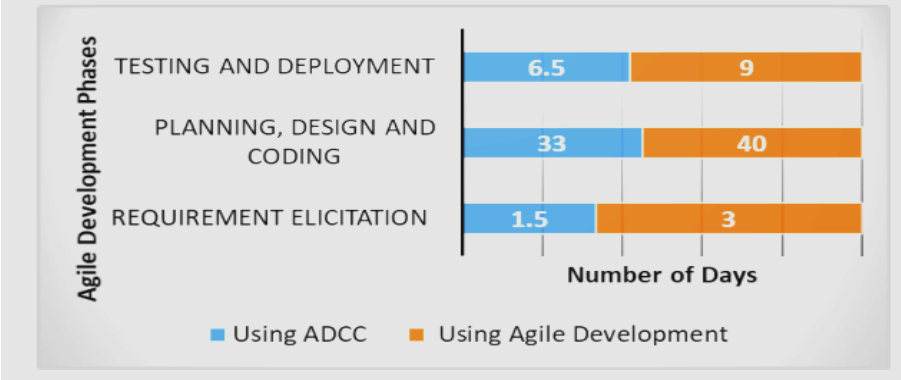

Figure5. Agile development life cycle (LCL) comparisons using and without ADCC methods in a distributed setting.

Table 2 shows the results of averaging two composite teams in an effort to lessen the effect of team competences on the organization's overall performance. Figure 5 shows that compared to a basic agile environment, the ADCC environment yields superior performance. Thanks to the improved communication between teams and users, the ADCC environment has seen a remarkable improvement. Another factor is the efficient provision of hardware and computing resources, which is achieved without any concerns about patch, update, or installation configurations.

CONCLUSION

The ADCC framework is evaluated through the use of a case study. The goals of this case study are to(1) describe the challenges of agile software development and(2) demonstrate the use of the ADCC framework to these problems. While the team in an agile software development process does have access to resources, they are not necessarily linked as they would be in a more conventional software development process. In the event that dedicated resources are arranged, the system will become expansive, necessitating the employment of additional network professionals in order to manage it. On the other hand, the ADCC framework, which is a form of cloud computing, offers a setting that is networked. The developers and other members of the team get the impression that they are working in an environment that is located on the premises. Every every system acts as if it were a single system. Differentiating the number of days needed to complete software development-related procedures shows that the ADCC framework provides benefits over a simple agile development environment.

REFERENCES

1. Dönmez, D.; Grote, G.; Brusoni, S. Routine interdependencies as a source of stability and flexibility. A study of agile software development teams. *Inf. Organ.* **2016**, *26*, 63–83. [**Google Scholar**] [**CrossRef**]
2. Ozkan, N.; Gök, M.Ş.; Köse, B.Ö. Towards a better understanding of agile mindset by using principles of agile methods. In Proceedings of the 2020 15th Conference on Computer Science and Information Systems (FedCSIS), Sofia, Bulgaria, 6–9 September 2020; pp. 721–730. [**Google Scholar**]
3. Dorairaj, S.; Noble, J.; Malik, P. Effective communication in distributed agile software development Teams. In *International Conference on Agile Software Development*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 102–116. [**Google Scholar**]
4. Sandstø, R.; Reme-Ness, C. Agile practices and impacts on project success. *J. Eng. Proj. Prod. Manag.* **2021**, *11*, 255–262. [**Google Scholar**]
5. Baham, C.; Hirschheim, R. Issues, challenges, and a proposed theoretical core of agile software development research. *Inf. Syst. J.* **2021**, *32*, 103–129
6. Pikkarainen, M.; Haikara, J.; Salo, O.; Abrahamsson, P.; Still, J. The impact of agile practices on communication in software development. *Empir. Softw. Eng.* **2008**, *13*, 303–337.
7. Arcos-Medina, G.; Mauricio, D. Identifying Factors Influencing on Agile Practices for Software Development. *J. Inf. Organ. Sci.* **2020**, *44*, 1–31.
8. Tam, C.; Moura, E.J.D.C.; Oliveira, T.; Varajão, J. The factors influencing the success of on-going agile software development projects. *Int. J. Proj. Manag.* **2020**, *38*, 165–176.
9. Taylor, R. Interpretation of the Correlation Coefficient: A Basic Review. *J. Diagn. Med. Sonogr.* **1990**, *6*, 35–39. [**Google Scholar**] [**CrossRef**]
10. Jeong, H.; Mason, S.P.; Barabasi, A.; Oltvai, Z.N. Lethality and centrality in protein networks. *Nature* **2001**, *411*, 41–42. [**Google Scholar**] [**CrossRef**] [**PubMed**] [**Green Version**]
11. Liu, D.; Zhai, Z. An Empirical Study of Agile Planning Critical Success Factors. Master's Thesis, Blekinge Institute of Technology, Karlskrona, Sweden, June 2017. [**Google Scholar**]
12. Moniruzzaman, A.; Hossain, D.S.A. Comparative Study on Agile software development methodologies. *arXiv* **2013**, arXiv:1307.3356. [**Google Scholar**]
13. Cao, L.; Ramesh, B. Agile Requirements Engineering Practices: An Empirical Study. *IEEE Softw.* **2008**, *25*, 60–67. [**Google Scholar**] [**CrossRef**]
14. Ghimire, D.; Charters, S.; Gibbs, S. Scaling agile software development approach in government organization in New Zealand. In Proceedings of the 3rd International Conference on Software Engineering and Information Management, Sydney, NSW, Australia, 12–15 January 2020. [**Google Scholar**] [**CrossRef**]
15. Salameh, H. What, when, why, and how? A comparison between agile project management and traditional project management methods. *Int. J. Bus. Manag. Rev.* **2014**, *2*, 52–74.
16. Nadella, Geeta Sandeep. Validating the Overall Impact of IS on Educators in U.S. High Schools Using IS-Impact Model – A Quantitative PLS-SEM Approach, DAI-A 85/7(E), Dissertation Abstracts International, Ann Arbor, ISBN 9798381388480, 189, (2023).

17. Gonaygunta, Hari, Factors Influencing the Adoption of Machine Learning Algorithms to Detect Cyber Threats in the Banking Industry, DAI-A 85/7(E), Dissertation Abstracts International, Ann Arbor, United States, ISBN 9798381387865, 142, 2023.
18. Ramya Manikyam, J. Todd McDonald, William R. Mahoney, Todd R. Andel, and Samuel H. Russ. 2016.Comparing the effectiveness of commercial obfuscators against MATE attacks. In Proceedings of the 6th Workshop on Software Security, Protection, and Reverse Engineering (SSPREW'16)
19. R. Manikyam. 2019.Program protection using software based hardware abstraction.Ph.D. Dissertation.University of South Alabama.
20. GPB GRADXS, N RAO, Behaviour Based Credit Card Fraud Detection Design And Analysis By Using Deep Stacked Autoencoder Based Harris Grey Wolf (Hgw) Method, Scandinavian Journal of Information Systems 35 (1), 1-8.
21. R Pulimamidi, GP Buddha, Applications of Artificial Intelligence Based Technologies in The Healthcare Industry, Tuijin Jishu/Journal of Propulsion Technology 44 (3), 4513-4519.
22. R Pulimamidi, GP Buddha, AI-Enabled Health Systems: Transforming Personalized Medicine And Wellness, Tuijin Jishu/Journal of Propulsion Technology 44 (3), 4520-4526.
23. GP Buddha, SP Kumar, CMR Reddy, Electronic system for authorization and use of cross-linked resource instruments, US Patent App. 17/203,879.
24. Hari Gonaygunta (2023) Machine Learning Algorithms for Detection of Cyber Threats using Logistic Regression, 10.47893/ijssan.2023.1229.
25. Hari Gonaygunta, Pawankumar Sharma, (2021) Role of AI in product management automation and effectiveness, https://doi.org/10.2139/ssrn.4637857
26. Sri Charan Yarlagadda, Role of Artificial Intelligence, Automation, and Machine Learning in Sustainable Plastics Packaging markets: Progress, Trends, and Directions, International Journal on Recent and Innovation Trends in Computing and Communication, Vol:11, Issue 9s, Pages: 818–828, 2023.
27. Sri Charan Yarlagadda, The Use of Artificial Intelligence and Machine Learning in Creating a Roadmap Towards a Circular Economy for Plastics, International Journal on Recent and Innovation Trends in Computing and Communication, Vol:11, Issue 9s, Pages: 829-836, 2023.
28. Amol Kulkarni, Amazon Athena Serverless Architecture and Troubleshooting, International Journal of Computer Trends and Technology, Vol, 71, issue, 5, pages 57-61, 2023.
29. Amazon Redshift Performance Tuning and Optimization,International Journal of Computer Trends and Technology, vol, 71, issue, 2, pages, 40-44, 2023.
30. B. Nagaraj, A. Kalaivani, S. B. R, S. Akila, H. K. Sachdev, and S. K. N, "The Emerging Role of Artificial intelligence in STEM Higher Education: A Critical review," International Research Journal of Multidisciplinary Technovation, pp. 1–19, Aug. 2023, doi: 10.54392/irjmt2351.
31. D. Sivabalaselvamani, K. Nanthini, Bharath Kumar Nagaraj, K. H. Gokul Kannan, K. Hariharan, M. Mallingeshwaran, Healthcare Monitoring and Analysis Using ThingSpeak IoT Platform: Capturing and Analyzing Sensor Data for Enhanced Patient Care, IGI Global eEditorial Discovery, Pages: 25, 2024. DOI: 10.4018/979-8-3693-1694-8.ch008.