International Journal of Research Radicals in Multidisciplinary Fields (IJRRMF), ISSN: 2960-043X
Volume 1, Issue 2, July-December, 2022, Available online at: www.researchradicals.com

Advanced Techniques in End-to-End Testing of Core Banking Solutions

Ashwini Shivarudra

Independent Researcher, USA

ABSTRACT

There has always been difficulty with integration testing, particularly when the system being tested is big and includes plenty of interfaces and subsystems. A strategy to designing End-to-End (E2E) integration testing, which includes tool support, test case creation, and test scenario specification, is presented in this work. For a lot of individuals these days, banking is essentially about using technology to do their financial chores. While many tasks still need a visit to a bank office, supporting mobile usage is quickly becoming the core goal of many bank initiatives. Large quantities of activity must be supported by the fundamental banking systems that handle financial transactions, and they must continue to be highly accessible. Many applications in the fields of medicine, civil engineering, marine engineering, military, and household use have resulted from this growth. While the demands and difficulties in each of these fields vary, data security is a common factor. In this paper, we share our experiences in modernizing and transforming banks and provide an overview of the approaches that will be used to implement these significant initiatives. From the perspective of the end user, test scenarios are given as thin threads, each of which represents a single function. It is possible to arrange thin threads in a hierarchical tree structure, where each branch represents a group of connected capabilities that are represented by a set of linked thin threads. Test engineers may do additional related jobs like risk analysis and assignment, regression testing, and ripple impact analysis, among others, by using thin-thread trees to construct test cases in a methodical manner. For the J2EE platform, a prototype tool has been created to facilitate E2E testing in a distributed setting.

Keywords: - End-To-End (E2E), Representing, E2E Testing, Domains, Tool Support, Financial Tasks, Large Programs, Financial Tasks, Threads, Integration Testing, Regression Testing.

INTRODUCTION

One type of alternative banking system is the agency banking model, wherein banks use non-bank agents to deliver financial services. Local supermarkets, grocers, booths, pharmacies, and kiosks may fall under this category [1]. The channel enables banks to provide their current services to locations where they lack the space to open a fully functional branch, particularly in rural and isolated places where a sizable portion of the population lacks access to banking. In the lack of bank branches, agency banking—especially in rural areas—has taken over as the main method of obtaining financial services [1, 2]. When banks are not present in a certain location, agency banking guarantees prompt and easy access to financial services. A bank or other financial institution with a license provides its customers with financial services through a retail agent. These customer-friendly goods and services are often provided by the bank at its branches, although an agent is also given charge of providing the services. Since its introduction to the banking sector back in 2010, agency banking has performed incredibly well, attracting more customers than banking halls and offering a wider variety of banking services that were previously exclusive to banking halls [1, 2]. Brazil is widely seen as a trailblazer in this domain, having been among the first countries to implement the agency banking model. Over time, the country has expanded and established a network of agent banks that serves over 99% of its towns [2].

After public block chains like Bitcoin and Ethereum gained popularity, researchers and practitioners discovered that these systems' lack of efficiency, privacy, and governance made them unsuitable for corporate users' demands [2, 3]. As a result, they created systems with authorization such Hyper Ledger Fabric, Corda, and Quorum. Block chains may enforce a permission notion to keep data secret, run on bespoke hardware, be modified as needed, and offer various designs based on a particular use case. As there are so many alternatives accessible, it is difficult to forecast results. Vendors frequently brag about their systems' exceptional performance, but reliable comparisons of similar workloads and circumstances are lacking [2, 3]. It is challenging to assess permissioned block chain systems because of their very different designs. Although there are currently a number of publications available for benchmarking multiple block chains, they fall short of meeting our requirements in order to provide a thorough performance review [3].

Today, the main method for ensuring quality is still testing. While there have been various testing approaches offered in the past, the majority of them centre on module testing [3, 4]. In actuality, integration testing is frequently the most costly and time-consuming aspect of testing [3]. Software development projects frequently allocate 50% to 70% of their resources to testing, and an additional 50% to 70% to integration testing. According to a survey of the literature on integration testing, the majority of integration testing methods either investigate the programming or design structure of the program or are based on concepts like gradual, top-down, and bottom-up integration [3, 5]. Though helpful, the approaches that examine programming or design structure are limited to software created with such techniques in mind.

For instance, evaluating a legacy software written in COBOL may not be possible using an integration testing approach designed for an object-oriented Java application [4, 5]. It's even feasible that the testing method won't work with a C++ application because C++ contains pointers whereas Java doesn't. The Department of Defence (DoD) started an End-to-End (E2E) integration testing initiative as a result of these factors. This testing procedure confirms the proper operation of a specified group of linked systems, which are now subsystems of the integrated system [5].

The term "core banking" has historically been used to describe a number of services such as loans, deposits, payments processing, merchant assistance, and card-related operations [6, 7]. This will apply to wealth management, commercial lending, and monetary exchange markets for institutional banking [6, 7]. These tasks are widely acknowledged as belonging to the conventional core banking areas, even if the exact breadth will vary throughout financial organizations [6, 7]. These fundamental duties comprise the mission-critical, day-to-day activities that banks must support. The highest availability requirements apply to the information technology and communication systems needed to provide these services, since any disruptions to business continuity would have a substantial effect on banking operations and revenue [7, 8]. For example, the average cost of downtime for brokerage and financial organizations is reportedly between USD$1.5 million and USD$6.5 million per hour [8].

Furthermore, because of the enormous volume of transactions that need to be supported, these systems primarily depend on high-end platforms or massive clusters of commodity hardware. The selection of platforms and technologies for constructing core banking systems has progressed to the extent that current packages include an extensive array of functionalities to cater to different functional areas. But in reality, these financial packages usually need a lot of customisation in order to be deployed within the intended institution [8, 9], which frequently results in major delays and higher project development costs. Our goal is to present a method for updating and/or replacing key banking systems that draws from the knowledge gained from significant transformation initiatives [9].

Core Banking's Operational Scope and Motivation

In this part, we go over some of the main reasons for replacing core banks and start out by outlining the essential components and functional range of core banking. This offers a starting point for supported functions. Furthermore, an explanation is provided for the expanded range of tasks that are optionally regarded as a component of fundamental banking operations [9, 10]. While each institution will have a different core banking program, the financial sector seems to embrace the capabilities that are offered. We also provide an overview of the operational banking and channel support systems [10, 11] for completeness. The relationship between core bank systems and various banking systems is depicted in the following figure, Fig. 1. We first describe some of the main reasons for pursuing such a program, and then go into more depth about each of the components that are shown in the following sections [11, 12].

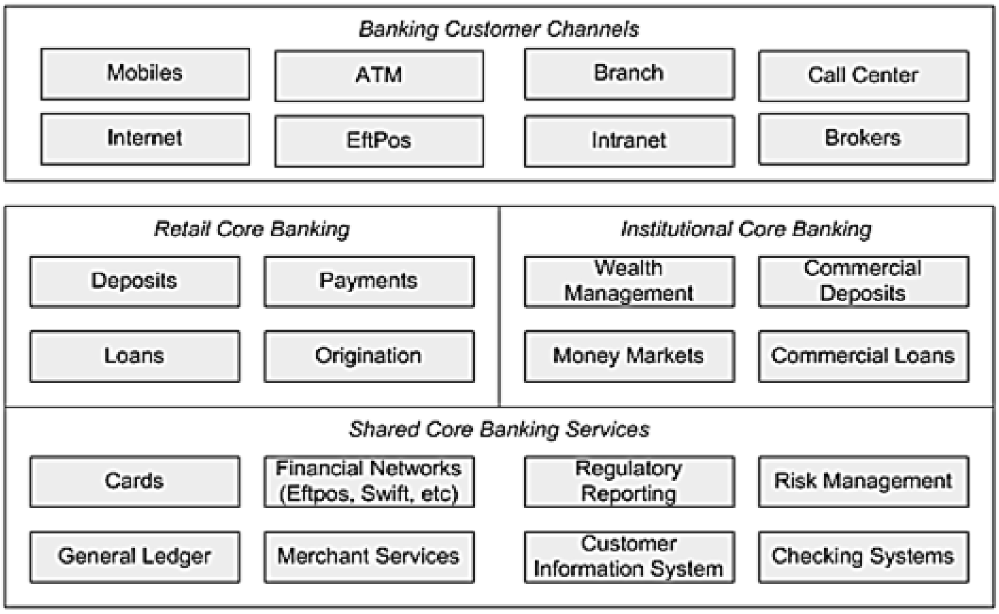

Fig. 1 System View Functionality for Core and Support Banks

Motivation

The bank is frequently motivated to modernize by a number of variables that finally reach a critical mass and provide the momentum to go forward. Existing methods and procedures that prevent the company from quickly introducing new financial products to the market are typically where the momentum starts [12, 13]. They could be seen as rigid and a hindrance to the market agility of businesses. Then, other driving forces that frequently surface include:

- ✓ Since older technologies predate the more recent virtual channels, they limit the business's ability to handle numerous emerging financial channels, such mobile and internet [13].
- ✓ An important reason to start modernizing the bank is that having too many financial products adds to the complexity of the market. As a result, product reduction and consolidation are often necessary.
- ✓ The requirement to rationalize and reduce the number of active banking systems in order to control expenses.
- ✓ A new core banking system should be introduced or existing systems should be upgraded due to the pressure that rising transaction volume is putting on them [14].
- ✓ Rival banks in the area declare their intention to update, and as a result, the market perceives them as more forward-thinking since they are prepared to take risks rather than staying the same.

Lastly, the IT environment is also disrupted by the ongoing change brought about by mergers, acquisitions, and business unit disposals [14, 15].

The Systems of Retail Core Banking

While the exact definition of "core banking" varies somewhat according on the organization and the area, the functional areas that are commonly included in this classification include payment cards, loans, deposits, and origination. These services are often connected to both retail customers and business entities in the context of core banking; we will now provide a brief description of each of these core banking services [15, 16]. The cash management tasks that underpin bank deposits made by retail customers are represented by deposits. While a bank deposit used to be a simple account with a variable interest rate yield, there are now other products that need assistance, such as term deposits, check accounts, trust accounts, loan offset accounts, and classic savings accounts. For both retail and business clients, secured and unsecured loans make up the fundamental banking services [16].

Shared Core Financial Systems, Institutions and Other

The range of financial services provided to small and medium-sized enterprises as well as major organizations is represented by institutional banking. This covers dealing with the money markets (trading and investing), managing significant financial portfolios, and conducting business internationally [17]. As a result, several of the fundamental functions—such as deposits, loans, and payments—are comparable to those of retail core banking; yet, they are focused on assisting company operations on a commercial scale. These systems also include expanded interfaces for global clearing networks, merchant support services for processing payments across financial-related networks, and wealth management (including superannuation) [17, 18].

E2E testing is predicated on the completion and approval of both module and integration testing, which may involve several stages of integration testing. Numerous facets of end-to-end testing are covered by the project, such as risk assessment and assignment, ripple impact analysis, test design, test execution, test result analysis, and regression testing. The test design, which forms the basis of the E2E project, is presented in this publication [18, 19].

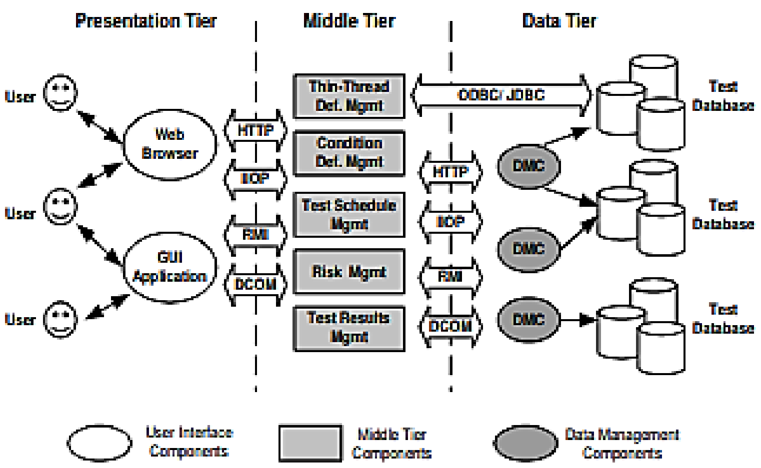

Fig. 2 The E2E Test Management Tool's Design. [18]

E2E TEST SPECIFICATIONS

Slender threads A thin thread is defined as follows in the DOD Management Plan: A minimally representative number of external information that is input is converted through an interconnected network of systems (architecture) to create a minimally represented sampling of external output data, forming a full trail (E2E) of data/messages. A thin thread's execution serves as an example of how to carry out a certain task. Therefore, the integrated system's minimal usage scenario is a thin thread. From the perspective of the user, a thin thread is essentially an entire scenario in which the system receives input data, processes it, and outputs the results [18, 19]. Only one operation is described by the thin thread, which also describes the entire scenario. Examples of thin threads in a financial application are successful local bank withdrawals and unsuccessful remote bank withdrawals because of inadequate funds [19]. A thin-thread group is made up of thin threads that have certain things in commonality.

Identify Thin Threads

An in-depth comprehension of the system being tested, including its operation and design, is necessary to identify thin threads [19, 20]. System functionality-identified thin threads are black-box thin thread because they are detected from outside perspectives of the system, whereas system structure-identified thin threads are white-box thin threads because inside perspectives are taken into account. These suggestions might help you spot thin threads by using the banking industry as an example:

- Analyse thin-thread groups by focusing on the core business operations. In other words, black-box thin threads are found first. For instance, the primary functions of the banking system are as follows: check-balance, deposit, withdrawal, and [19, 20].
- Analyse the inputs of thin-thread groups to break them down. The group check-balance transactions, for instance, may be divided into two subgroups: one with proper inputs and the other with incorrect input.
- Analyse the outputs of thin-thread groups to break them down. Group withdrawal operations, for instance, may result in two different outcomes: a successful withdrawal and an error message indicating insufficient funds. This thin-thread group may thus be divided into two subgroups: successful withdrawal transactions and unsuccessful withdrawal transactions because of insufficient funds [20].
- Analyse a thin-thread group's potential execution pathways to break it down. For instance, there are two ways that group withdrawal transactions could be carried out: locally and remotely. As a result, it may be divided into two smaller categories: withdrawals from nearby banks and withdrawals from distant banks [18, 20].

Connections between thin threads

The execution routes of thin threads are related to one another:

- **Contained in:** The execution path of a thin thread can be part of the path of another thin-thread. • Identical: Two thin threads have the same paths. In this case, they may share a certain set of attributes, such as conditions [20].
- **Independent:** Thin-threads have completely different paths.

Construct Thin-Thread Trees

The root of a thin-thread tree represents the overall integrated system under test, a branch node represents a collection of related thin threads (or thin-thread group) and a leaf represents a concrete thin thread. In most cases, the thin threads within a thin-thread group are related by their common functionality [20, 21].

Thus, a thin-thread tree can be viewed as a functional decomposition of the system under test. Thin thread tree can be constructed in many ways:

- **Top-down:** find the main thin-thread groupings and build the tree by breaking down the groups of thin threads at each level; [22],
- **Bottom-up:** find the atomic use situations and build the tree by combining the thin threads (or groups) at each level;
- **Combined approach:** Combine the method of top-down and bottom-up methods when working on the project. A prime instance thin-thread tree for an application for banking is shown in Figure 3 [22, 23].

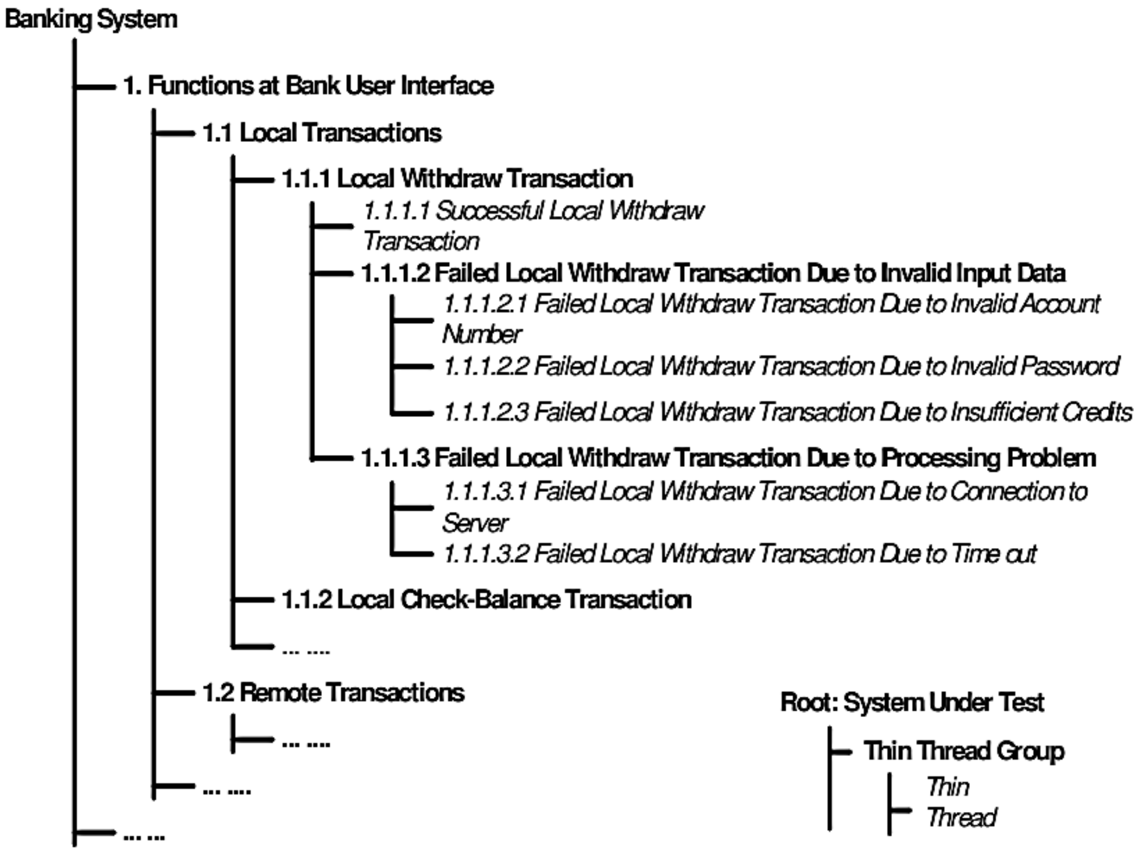

Fig. 3 A Thin-Thread Tree Examples [22]

Link Information to a Thin Thread

It is required to identify both the input and the output information as well as the altered data for each thin-thread/thin-thread-group [22, 23]. In a banking scenario, for instance, a withdrawal transaction may include the following information:

- **a.** Bank ID;
- **b.** Account information;
- **c.** Account password;
- **d.** Transaction information.

These files ought to be kept in the test database and linked to the relevant thin thread. They are helpful in creating test cases.

Conditions

Predicates known as conditions have an impact on how thin threads are executed [22, 23]. When all of a thin thread's requirements are met, it becomes active. Among the scenarios that might occur are:

- **Conditions for communication:** Delays in communication within time-out, recovery, security, and protocols procedures are a few samples.
- **Timing and sequence parameters:** Order sequences, the maximum and minimum permitted delays, daily, monthly, and quarterly updates, and cooperation amongst collaborating parties are a few examples.
- **Data conditions:** Data items are involved in these circumstances. Consider a user account.

Identify Conditions

Like thin threads, circumstances can be determined by the functioning and structure of the system [23]. Recognizing the condition's advantages and disadvantages is crucial. Several pointers for identifying conditions include:

Determine the data conditions using a thin thread's input and output. For instance, account number, password, withdrawal amount, and account balance may be submitted for a thin thread successful withdrawal transaction [22, 24]. Thus, the requirements for this short thread are as follows:

1. A legitimate account number;
2. A matching, valid password;
3. Amount of valid withdrawal; and
4. Adequate amount in the account.

a. Determine the external user interface's data circumstances. Examples of data elements that might be present in an ATM user interface are the account ID, password, [24, 25], and payment type. As a result, criteria like a valid and invalid account identification number and username may be deduced.
b. Determine the data conditions based on the information shared between software components. A transaction is encrypted, for instance, by an ATM before it is sent to a bank. Therefore, a valid transaction message and effective encryption are required for a transaction to be successful.
c. Determine the criteria for communication based on the logical and physical connections that exist between components. For instance, an ATM's ability to communicate with its financial system identifies the need for proper communication protocol.
d. Determine network conditions based on the physical linkage among dispersed subsystems. For instance, a point-to-point connection and a maximum delay of one second may be conditions found in the connections between an ATM and its banking system [24].

Connections among the Conditions
Testing every possible combination of circumstances might not be possible. In order for a tester to create an appropriate set of test circumstances to test the application, it is crucial to analyse the links between the criteria. Common connections between conditions are:

- ✓ **Independent:** If one of two situations may occur without the other, then they are independent.
- ✓ **Trigger/trigger-by:** One situation might cause the other to come into play.
- ✓ **Mutually exclusive:** If two conditions cannot both occur at the same moment, then they are mutually exclusive. For instance, having a legitimate bank account is one need that conflicts with another prerequisite [21, 22]. Bank account not in good standing.
- ✓ **Related:** If two criteria appear in the same thread or are mutually exclusive, they are connected to one another [22[.

Construct Condition Trees
To make reuse and administration easier, conditions can also be arranged into a tree structure [21, 22]. Like a thin thread tree, a test tree has the system being tested at the root, a set of related conditions at the branch node, and a concrete situation at the leaf nodes.

Thin Threads Tree and Conditions Tree Relationships
There is an interdependence between the thin thread tree and condition tree constructions. First, conditions may be identified using the thin thread tree [22, 23]. Certain thin threads, such successfully local withdrawal transactions and successful distant withdrawal transactions, could have the identical input data condition, for instance.

Certain thin threads could be subject to exclusive requirements; for example, a successful local withdrawal transaction could also fail if the user password is incorrect. Conditions can be found by analysing the thin thread tree [24, 25].

Checking for Completeness and Consistent (C&C)
Slender threads C&C indicates that every E2E system function can be tracked back to the system description and is recognized without leakage or conflict. Moreover, each thin-thread and thin-thread grouping has had all the requirements fully and consistently linked to it [22, 23].

CREATING TEST CASES AND SCENARIOS

Creating Test Scenarios
The system under test's atomic E2E functions are identified by thin threads. Testing is not, however, restricted to the performance of atomic operations. In addition, a series or a blend of atomic operations may be included [22]. The following operators are used to build composite thin threads, which are specified by a sophisticated test case:

- **Sequencing:** A different approach thin thread can follow a thin thread immediately;
- **Looping:** You may repeat a thin thread several times; [24], and
- **Conditioned execution:** This creates a complicated test case by include the control decision.

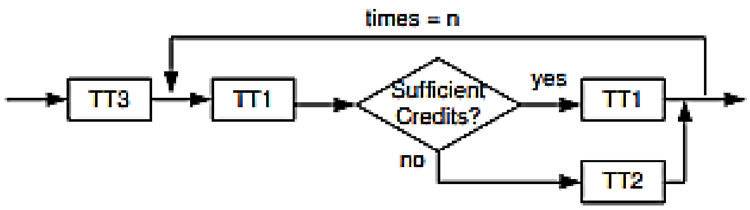

Fig. 4 Example of Complex Test Scenario [26]

E2E TEST CASE GENERATION

The following methods can be used to create test cases from a thin thread or complicated scenario:

(1) Using various testing strategies, determine whether input data fulfil the thread's criteria; and
(2) Use the thread description to ascertain the anticipated outcomes [26, 27].

Assistance for Tools E2E testing is difficult

An online communication instrument has been created to facilitate the examination. As seen in Figure 4 [29], the tool has a three-tier design and is implemented on the J2EE platform via Enterprise JavaBean (EJB) [27, 28].

1. Test data is stored on file/database servers located at the backend;
2. The fundamental tasks of the E2E testing are carried out by the intermediate layer; [30],
3. The presentation layer is the top tier. It offers customers a variety of graphical representations of the test data and analytical findings.

CONCLUSION

The implementation of a bank modernization program needs strong backing from business sponsors and stakeholders. As with any significant change, experience has shown that these initiatives will be delivered under a great deal of strain and will benefit from good technical and project management methods.

A methodical approach to E2E testing design, comprising test definition, test case creation, and tool support, is shown in this work. The following features of the strategy are present:

(1) It employs combined white-box and black-box evaluation methodologies, which yields adequate data for functional test case development, coverage analysis, result analysis, defect detection, and software assessment;
(2) Low-level source code implementation and high-level use case description are connected by thin threads;
(3) Because it's semi-formal, it's simple to employ;
(4) It may be used with contemporary approaches like OO to apply it to both old systems and applications;
(5) It makes test case and scenario reuse easier;
(6) It facilitates change management, which enables accurate and effective execution of regression testing and ripple impact analysis;
(7) It facilitates dispersed collaboration and remote project management, enabling engineers and project managers to collaborate online.

REFERENCES

[1]. R. Liua, F. Wua, Y. Patnaikb, S. Kumaranb, “Business Entities: An SOA Approach to Progressive Core Banking Renovation,” in 2009 IEEE International Conference on Services Computing, pp.466-473.
[2]. Beimborn, J. Franke, H-T. Wagner, T. Weitzel, “The Influence of Alignment on the Post-Implementation Success of a Core Banking Information System: An Embedded Case Study,” in Proceedings of the 40th Hawaii International Conference on System Sciences, 2007.
[3]. Metwally, “Implementation of IT-based Delivery Systems in Union Trust Bank of Egypt,” in 2010 International Conference on Education and Management Technology (ICEMT 2010), pp.645-649.
[4]. A. Malan, L Pretorius, JHC Pretorius, “Managing the Implementation of Banking Systems for Repeatable Success,” in PICMET 2008 Proceedings, 27-31 July, Cape Town, South Africa (c) 2008 PICMET, pp.2366-2375
[5]. A.P. Noble, R. Kopaee, A. Melek, and N. Nandy, “Data Leak Prevention,” ISACA White Paper, 2010.

[6]. L. Zhang and Y. Wang, “The Banking System Reference Implementation Based on SOA,” in the Fourth International Joint Conference on Computational Sciences and Optimization (CSO), April 2011, pp. 374-377.
[7]. T. Rojahn, R. Lebsack, C. Pavlovski, “Towards a Computing Workload Classifier,” IBM J. Research and Development, Vol. 58, November 2014, To Appear.
[8]. J. Kilimnik and C. Pavlovski, “Core Banking Modernization,” in the 9th International Conference on Information Technology and Applications (ICITA), July, 2014, Sydney, Australia.
[9]. Fischer K. End-to-End security for DTLS-SRTP. IETF DRAFT, 2008. Fischl J, Tschofenig H, Rescorla E. Framework for establishing an SRTP security context using DTLS. IETF DRAFT, 2008.
[10]. Galvin J, Murphy S, Crocker S, Freed N. Security multipart for mime: multipart/signed and multipart/encrypted. IETF RFC 1847, 1995.
[11]. Guo JI, Yen JC, Pai HF, New voice over internet protocol technique with hierarchical data security protection. In: IEE Proceedings vision, image & signal processing, vol. 149, no. 4; 2002. p. 237–43.
[12]. D. C. Kung, P. Hsia, and J. Gao, Testing Object-Oriented Software, IEEE Computer Society Press, Los Alamitos, CA, 1999.
[13]. P. Loo, W. T. Tsai, W. K. Tsai, “Random Testing Revisited”, Information and Software Technology, Vol. 30, No. 7, Sept. 1988, pp. 402-417.
[14]. J. Myers, The Art of Software Testing, New York, Wiley Inter-science, New York, New York, 1979.
[15]. R.S. Pressman, Software Engineering: A Practitioner’s Approach, McGraw Hill, 5 th edition, 2000.
[16]. E. Roman, Mastering Enterprise JavaBeans TM and the Java TM 2 Platform, Enterprise Edition, John Wiley & Sons, Inc. New York, New York, 1999.
[17]. W. T. Tsai, R. Paul, W. Shao, S. Rayadurgam, and J. Li, “Assurance-Based Y2K Testing”, 1999 Proc. 4 th IEEE International Symposium on HighAssurance Systems Engineering, pp. 27-34.
[18]. W. T. Tsai, Y. Tu, W. Shao and E. Ebner, “Testing Extensible Design Patterns in Object-Oriented Frameworks through Hierarchical Scenario Templates”, Proc. COMPSAC, 1999, pp. 166-171.
[19]. W. T. Tsai, V. Agarwal, B. Huang, R. Paul, “Augmenting Sequence Constraints in Z and its Application to Testing”, Proc. 3 rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology, 2000, pp. 41-48.
[20]. W. T. Tsai, X. Bai, R. Paul, G. Devaraj, and V. Agarwal, “An Approach to Modify and Test Expired Window Logic”, Proc. of IEEE Asia-Pacific Conference on Quality Software, 2000, pp. 99-108.
[21]. W.T. Tsai, X. Bai, R. Paul, and L. Yu, “Scenario-Based Functional Regression Testing”, to appear in IEEE Proceedings of COMPSAC, 2001.
[22]. Lowis, L. and R. Accorsi, 2011. Vulnerability analysis in SOA-based business processes. IEEE Trans. Services Compute. 4: 230-42.
[23]. Lux, K.D., M.J. May, N.L. Bhattad and C.A. Gunter, 2005. WSEmail: Secure internet messaging based on web services. Proceedings of the IEEE International Conference on Web Services, Jul. 11-15, IEEE Xplore Press,
[24]. Orlando, FL, USA, pp: 75-82. DOI: 10.1109/ICWS.2005.138 Mainka, C., J. Somorovsky and J. Schwenk, 2012. Penetration testing tool for web services security. Proceedings of the IEEE 8th World Congress on Services, Jun. 24-29, IEEE Xplore Press.
[25]. Honolulu, HI, USA, pp: 163-170. DOI: 10.1109/SERVICES.2012.7 Morris, E.J., W.B. Anderson, S. Balasubramanian, D.J. Carney and J. Morley, 2010. Testing in service oriented environments.
[26]. A. Premchand and A. Choudhry, "Open Banking & APIs for Transformation in Banking," 2018 International Conference on Communication, Computing and Internet of Things (IC3IoT), 2018, pp. 25-29,
[27]. E. Ünsal, B. Öztekin, M. Çavuş and S. Özdemir, "Building a Fintech Ecosystem: Design and Development of a Fintech API Gateway," 2020 International Symposium on Networks, Computers and Communications (ISNCC), 2020, pp. 1-5,
[28]. B. Nokovic, N. Djosic and W. O. Li, "API Security Risk Assessment Based on Dynamic ML Models," 2020 14th International Conference on Innovations in Information Technology (IIT), 2020, pp. 247-252.
[29]. G. Baye, F. Hussain, A. Oracevic, R. Hussain and S. M. Ahsan Kazmi, “API Security in Large Enterprises: Leveraging Machine Learning for Anomaly Detection," 2021 International Symposium on Networks, Computers and Communications (ISNCC), 2021, pp. 1-6.
[30]. M. A. K. Azad, S. Islam, A. K. M. M. Islam and M. S. H. Mukta, " Comparative Study of QoS Parameters for Different Web Services," International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM), 2021, pp. 267-272.
[31]. Saoji, R., Nuguri, S., Shiva, K., Etikani, P., & Bhaskar, V. V. S. R. (2019). Secure federated learning framework for distributed AI model training in cloud environments. International Journal of Open Publication and Exploration (IJOPE), 7(1), 31. Available online at https://ijope.com.
[32]. Savita Nuguri, Rahul Saoji, Krishnateja Shiva, Pradeep Etikani, & Vijaya Venkata Sri Rama Bhaskar. (2021). OPTIMIZING AI MODEL DEPLOYMENT IN CLOUD ENVIRONMENTS: CHALLENGES AND

SOLUTIONS. International Journal for Research Publication and Seminar, 12(2), 159–168. https://doi.org/10.36676/jrps.v12.i2.1461

[33]. Kaur, J., Choppadandi, A., Chenchala, P. K., Nuguri, S., & Saoji, R. (2022). Machine learning-driven IoT systems for precision agriculture: Enhancing decision-making and efficiency. Webology, 19(6), 2158. Retrieved from http://www.webology.org.

[34]. Saoji, R., Nuguri, S., Shiva, K., Etikani, P., & Bhaskar, V. V. S. R. (2021). Adaptive AI-based deep learning models for dynamic control in software-defined networks. International Journal of Electrical and Electronics Engineering (IJEEE), 10(1), 89–100. ISSN (P): 2278–9944; ISSN (E): 2278–9952

[35]. Kavuri, S., & Narne, S. (2020). Implementing effective SLO monitoring in high-volume data processing systems. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 6(2), 558. http://ijsrcseit.com

[36]. Kavuri, S., & Narne, S. (2021). Improving performance of data extracts using window-based refresh strategies. International Journal of Scientific Research in Science, Engineering and Technology, 8(5), 359-377. https://doi.org/10.32628/IJSRSET

[37]. Narne, S. (2022). AI-driven drug discovery: Accelerating the development of novel therapeutics. International Journal on Recent and Innovation Trends in Computing and Communication, 10(9), 196. http://www.ijritcc.org

[38]. Kavuri, S., & Narne, S. (2020). Implementing effective SLO monitoring in high-volume data processing systems. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 6(2), 558. http://ijsrcseit.com

[39]. Kavuri, S., & Narne, S. (2021). Improving performance of data extracts using window-based refresh strategies. International Journal of Scientific Research in Science, Engineering and Technology, 8(5), 359-377. https://doi.org/10.32628/IJSRSET

[40]. Narne, S. (2022). AI-driven drug discovery: Accelerating the development of novel therapeutics. International Journal on Recent and Innovation Trends in Computing and Communication, 10(9), 196. http://www.ijritcc.org

[41]. Rinkesh Gajera , "Leveraging Procore for Improved Collaboration and Communication in Multi-Stakeholder Construction Projects", International Journal of Scientific Research in Civil Engineering (IJSRCE), ISSN : 2456-6667, Volume 3, Issue 3, pp.47-51, May-June.2019

[42]. Balachandar, P. (2020). Title of the article. International Journal of Scientific Research in Science, Engineering and Technology, 7(5), 401-410. https://doi.org/10.32628/IJSRSET23103132

[43]. Paulraj, B. (2022). Building Resilient Data Ingestion Pipelines for Third-Party Vendor Data Integration. Journal for Research in Applied Sciences and Biotechnology, 1(1), 97–104. https://doi.org/10.55544/jrasb.1.1.14

[44]. Paulraj, B. (2022). The Role of Data Engineering in Facilitating Ps5 Launch Success: A Case Study. International Journal on Recent and Innovation Trends in Computing and Communication, 10(11), 219–225. https://doi.org/10.17762/ijritcc.v10i11.11145

[45]. Paulraj, B. (2019). Automating resource management in big data environments to reduce operational costs. Tuijin Jishu/Journal of Propulsion Technology, 40(1). https://doi.org/10.52783/tjjpt.v40.i1.7905

[46]. Balachandar Paulraj. (2021). Implementing Feature and Metric Stores for Machine Learning Models in the Gaming Industry. European Economic Letters (EEL), 11(1). Retrieved from https://www.eelet.org.uk/index.php/journal/article/view/1924

[47]. Bhatt, S. (2020). Leveraging AWS tools for high availability and disaster recovery in SAP applications. International Journal of Scientific Research in Science, Engineering and Technology, 7(2), 482. https://doi.org/10.32628/IJSRSET2072122

[48]. Kavuri, S., & Narne, S. (2020). Implementing effective SLO monitoring in high-volume data processing systems. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 5(6), 558. https://doi.org/10.32628/CSEIT206479

[49]. Swethasri Kavuri, " Advanced Debugging Techniques for Multi-Processor Communication in 5G Systems, IInternational Journal of Scientific Research in Computer Science, Engineering and Information Technology(IJSRCSEIT), ISSN : 2456-3307, Volume 9, Issue 5, pp.360-384, September-October-2023. Available at doi : https://doi.org/10.32628/CSEIT239071

[50]. Mehra, A. (2021). The impact of public-private partnerships on global educational platforms. Journal of Informatics Education and Research, 1(3), 9–28. http://jier.org

[51]. Ankur Mehra. (2019). Driving Growth in the Creator Economy through Strategic Content Partnerships. International Journal for Research Publication and Seminar, 10(2), 118–135. https://doi.org/10.36676/jrps.v10.i2.1519

[52]. Ankur Mehra. (2022). Effective Team Management Strategies in Global Organizations. Universal Research Reports, 9(4), 409–425. https://doi.org/10.36676/urr.v9.i4.1363

[53]. Ankur Mehra. (2022). Effective Team Management Strategies in Global Organizations. Universal Research Reports, 9(4), 409–425. https://doi.org/10.36676/urr.v9.i4.1363

[54]. Ankur Mehra. (2022). Effective Team Management Strategies in Global Organizations. Universal Research Reports, 9(4), 409–425. https://doi.org/10.36676/urr.v9.i4.1363

[55]. Ankur Mehra. (2022). The Role of Strategic Alliances in the Growth of the Creator Economy. European Economic Letters (EEL), 12(1). Retrieved from https://www.eelet.org.uk/index.php/journal/article/view/1925

[56]. Bizel, G., Parmar, C., Singh, K., Teegala, S., & Voddi, V. K. R. (2021). Cultural health moments: A search analysis during times of heightened awareness to identify potential interception points with digital health consumers. Journal of Economics and Management Sciences, 4(4), 35. https://doi.org/10.30560/jems.v4n4p35

[57]. Reddy, V. V. K., & Reddy, K. K. (2021). COVID-19 case predictions: Anticipating future outbreaks through data. NeuroQuantology, 19(7), 461–466. https://www.neuroquantology.com/open-access/COVID-19+Case+Predictions%253A+Anticipating+Future+Outbreaks+Through+Data_14333/?download=true

[58]. Santhosh Palavesh. (2019). The Role of Open Innovation and Crowdsourcing in Generating New Business Ideas and Concepts. International Journal for Research Publication and Seminar, 10(4), 137–147. https://doi.org/10.36676/jrps.v10.i4.1456

[59]. Santosh Palavesh. (2021). Developing Business Concepts for Underserved Markets: Identifying and Addressing Unmet Needs in Niche or Emerging Markets. Innovative Research Thoughts, 7(3), 76–89. https://doi.org/10.36676/irt.v7.i3.1437

[60]. Palavesh, S. (2021). Co-Creating Business Concepts with Customers: Approaches to the Use of Customers in New Product/Service Development. Integrated Journal for Research in Arts and Humanities, 1(1), 54–66. https://doi.org/10.55544/ijrah.1.1.9

[61]. Santhosh Palavesh. (2022). Entrepreneurial Opportunities in the Circular Economy: Defining Business Concepts for Closed-Loop Systems and Resource Efficiency. European Economic Letters (EEL), 12(2), 189–204. https://doi.org/10.52783/eel.v12i2.1785

[62]. Santhosh Palavesh. (2022). The Impact of Emerging Technologies (e.g., AI, Blockchain, IoT) On Conceptualizing and Delivering new Business Offerings. International Journal on Recent and Innovation Trends in Computing and Communication, 10(9), 160–173. Retrieved from https://www.ijritcc.org/index.php/ijritcc/article/view/10955

[63]. Santhosh Palavesh. (2021). Business Model Innovation: Strategies for Creating and Capturing Value Through Novel Business Concepts. European Economic Letters (EEL), 11(1). https://doi.org/10.52783/eel.v11i1.1784

[64]. Vijaya Venkata Sri Rama Bhaskar, Akhil Mittal, Santosh Palavesh, Krishnateja Shiva, Pradeep Etikani. (2020). Regulating AI in Fintech: Balancing Innovation with Consumer Protection. European Economic Letters (EEL), 10(1). https://doi.org/10.52783/eel.v10i1.1810

[65]. Challa, S. S. S. (2020). Assessing the regulatory implications of personalized medicine and the use of biomarkers in drug development and approval. European Chemical Bulletin, 9(4), 134-146.

[66]. D.O.I10.53555/ecb.v9:i4.17671

[67]. EVALUATING THE EFFECTIVENESS OF RISK-BASED APPROACHES IN STREAMLINING THE REGULATORY APPROVAL PROCESS FOR NOVEL THERAPIES. (2021). Journal of Population Therapeutics and Clinical Pharmacology, 28(2), 436-448. https://doi.org/10.53555/jptcp.v28i2.7421

[68]. Challa, S. S. S., Tilala, M., Chawda, A. D., & Benke, A. P. (2019). Investigating the use of natural language processing (NLP) techniques in automating the extraction of regulatory requirements from unstructured data sources. Annals of Pharma Research, 7(5), 380-387.

[69]. Ashok Choppadandi. (2022). Exploring the Potential of Blockchain Technology in Enhancing Supply Chain Transparency and Compliance with Good Distribution Practices (GDP). International Journal on Recent and Innovation Trends in Computing and Communication, 10(12), 336–343. Retrieved from https://www.ijritcc.org/index.php/ijritcc/article/view/10981

[70]. Challa, S. S. S., Chawda, A. D., Benke, A. P., & Tilala, M. (2020). Evaluating the use of machine learning algorithms in predicting drug-drug interactions and adverse events during the drug development process. NeuroQuantology, 18(12), 176-186. https://doi.org/10.48047/nq.2020.18.12.NQ20252

[71]. Challa, S. S. S., Tilala, M., Chawda, A. D., & Benke, A. P. (2022). Quality Management Systems in Regulatory Affairs: Implementation Challenges and Solutions. Journal for Research in Applied Sciences and Biotechnology, 1(3), 278–284. https://doi.org/10.55544/jrasb.1.3.36

[72]. Ranjit Kumar Gupta, Sagar Shukla, Anaswara Thekkan Rajan, & Sneha Aravind. (2022). Strategies for Effective Product Roadmap Development and Execution in Data Analytics Platforms. International Journal for Research Publication and Seminar, 13(1), 328–342. Retrieved from https://jrps.shodhsagar.com/index.php/j/article/view/1515

[73]. Ranjit Kumar Gupta, Sagar Shukla, Anaswara Thekkan Rajan, & Sneha Aravind. (2022). Leveraging Data Analytics to Improve User Satisfaction for Key Personas: The Impact of Feedback Loops. International Journal for Research Publication and Seminar, 11(4), 242–252. https://doi.org/10.36676/jrps.v11.i4.1489

[74]. Ranjit Kumar Gupta, Sagar Shukla, Anaswara Thekkan Rajan, Sneha Aravind, 2021. "Utilizing Splunk for Proactive Issue Resolution in Full Stack Development Projects" ESP Journal of Engineering & Technology Advancements 1(1): 57-64.

[75]. Sagar Shukla. (2021). Integrating Data Analytics Platforms with Machine Learning Workflows: Enhancing Predictive Capability and Revenue Growth. International Journal on Recent and Innovation Trends in Computing and Communication, 9(12), 63–74. Retrieved from https://ijritcc.org/index.php/ijritcc/article/view/11119

[76]. Sneha Aravind. (2021). Integrating REST APIs in Single Page Applications using Angular and TypeScript. International Journal of Intelligent Systems and Applications in Engineering, 9(2), 81 –. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/6829

[77]. Aravind, S., Cherukuri, H., Gupta, R. K., Shukla, S., & Rajan, A. T. (2022). The role of HTML5 and CSS3 in creating optimized graphic prototype websites and application interfaces. NeuroQuantology, 20(12), 4522-4536. https://doi.org/10.48047/NQ.2022.20.12.NQ77775

[78]. Rishabh Rajesh Shanbhag, Rajkumar Balasubramanian, Ugandhar Dasi, Nikhil Singla, & Siddhant Benadikar. (2022). Case Studies and Best Practices in Cloud-Based Big Data Analytics for Process Control. International Journal for Research Publication and Seminar, 13(5), 292–311. https://doi.org/10.36676/jrps.v13.i5.1462

[79]. Siddhant Benadikar. (2021). Developing a Scalable and Efficient Cloud-Based Framework for Distributed Machine Learning. International Journal of Intelligent Systems and Applications in Engineering, 9(4), 288 –. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/6761

[80]. Siddhant Benadikar. (2021). Evaluating the Effectiveness of Cloud-Based AI and ML Techniques for Personalized Healthcare and Remote Patient Monitoring. International Journal on Recent and Innovation Trends in Computing and Communication, 9(10), 03–16. Retrieved from https://www.ijritcc.org/index.php/ijritcc/article/view/11036

[81]. Challa, S. S., Tilala, M., Chawda, A. D., & Benke, A. P. (2019). Investigating the use of natural language processing (NLP) techniques in automating the extraction of regulatory requirements from unstructured data sources. Annals of PharmaResearch, 7(5), 380-387.

[82]. Chaturvedi, R., & Sharma, S. (2022). Assessing the Long-Term Benefits of Automated Remittance in Large Healthcare Networks. Journal for Research in Applied Sciences and Biotechnology, 1(5), 219–224. https://doi.org/10.55544/jrasb.1.5.25

[83]. Chaturvedi, R., & Sharma, S. (2022). Enhancing healthcare staffing efficiency with AI-powered demand management tools. Eurasian Chemical Bulletin, 11(Regular Issue 1), 675-681. https://doi.org/10.5281/zenodo.13268360

[84]. Neha Yadav,Vivek Singh, "Probabilistic Modeling of Workload Patterns for Capacity Planning in Data Center Environments" (2022). International Journal of Business Management and Visuals, ISSN: 3006-2705, 5(1), 42-48. https://ijbmv.com/index.php/home/article/view/73

[85]. Dr. Saloni Sharma, & Ritesh Chaturvedi. (2017). Blockchain Technology in Healthcare Billing: Enhancing Transparency and Security. International Journal for Research Publication and Seminar, 10(2), 106–117. Retrieved from https://jrps.shodhsagar.com/index.php/j/article/view/1475

[86]. Dr. Saloni Sharma, & Ritesh Chaturvedi. (2017). Blockchain Technology in Healthcare Billing: Enhancing Transparency and Security. International Journal for Research Publication and Seminar, 10(2), 106–117. Retrieved from https://jrps.shodhsagar.com/index.php/j/article/view/1475

[87]. Saloni Sharma. (2020). AI-Driven Predictive Modelling for Early Disease Detection and Prevention. International Journal on Recent and Innovation Trends in Computing and Communication, 8(12), 27–36. Retrieved from https://www.ijritcc.org/index.php/ijritcc/article/view/11046

[88]. Chaturvedi, R., & Sharma, S. (2022). Assessing the Long-Term Benefits of Automated Remittance in Large Healthcare Networks. Journal for Research in Applied Sciences and Biotechnology, 1(5), 219–224. https://doi.org/10.55544/jrasb.1.5.25

[89]. Hitali Shah.(2017). Built-in Testing for Component-Based Software Development. International Journal of New Media Studies: International Peer Reviewed Scholarly Indexed Journal, 4(2), 104–107. Retrieved from https://ijnms.com/index.php/ijnms/article/view/259

[90]. Pavan Ogeti, Narendra Sharad Fadnavis, Gireesh Bhaulal Patil, Uday Krishna Padyana, Hitesh Premshankar Rai. (2022). Blockchain Technology for Secure and Transparent Financial Transactions. European Economic Letters (EEL), 12(2), 180–188. Retrieved from https://www.eelet.org.uk/index.php/journal/article/view/1283

[91]. Fadnavis, N. S., Patil, G. B., Padyana, U. K., Rai, H. P., & Ogeti, P. (2020). Machine learning applications in climate modeling and weather forecasting. NeuroQuantology, 18(6), 135-145. https://doi.org/10.48047/nq.2020.18.6.NQ20194

[92]. Narendra Sharad Fadnavis. (2021). Optimizing Scalability and Performance in Cloud Services: Strategies and Solutions. International Journal on Recent and Innovation Trends in Computing and Communication, 9(2), 14–21. Retrieved from https://www.ijritcc.org/index.php/ijritcc/article/view/10889

[93]. Gireesh Bhaulal Patil. (2022). AI-Driven Cloud Services: Enhancing Efficiency and Scalability in Modern Enterprises. International Journal of Intelligent Systems and Applications in Engineering, 10(1), 153–162. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/6728

[94]. Patil, G. B., Padyana, U. K., Rai, H. P., Ogeti, P., & Fadnavis, N. S. (2021). Personalized marketing strategies through machine learning: Enhancing customer engagement. Journal of Informatics Education and Research, 1(1), 9. http://jier.org

[95]. Krishnateja Shiva. (2022). Leveraging Cloud Resource for Hyperparameter Tuning in Deep Learning Models. International Journal on Recent and Innovation Trends in Computing and Communication, 10(2), 30–35. Retrieved from https://www.ijritcc.org/index.php/ijritcc/article/view/10980

[96]. Raina, Palak, and Hitali Shah."Security in Networks." International Journal of Business Management and Visuals, ISSN: 3006-2705 1.2 (2018): 30-48.

[97]. Shiva, K., Etikani, P., Bhaskar, V. V. S. R., Palavesh, S., & Dave, A. (2022). The rise of robo-advisors: AI-powered investment management for everyone. Journal of Namibian Studies, 31, 201-214.

[98]. Bhaskar, V. V. S. R., Etikani, P., Shiva, K., Choppadandi, A., & Dave, A. (2019). Building explainable AI systems with federated learning on the cloud. Journal of Cloud Computing and Artificial Intelligence, 16(1), 1–14.

[99]. Ogeti, P., Fadnavis, N. S., Patil, G. B., Padyana, U. K., & Rai, H. P. (2022). Blockchain technology for secure and transparent financial transactions. European Economic Letters, 12(2), 180-192. http://eelet.org.uk

[100]. Hitali Shah."Millimeter-Wave Mobile Communication for 5G". International Journal of Transcontinental Discoveries, ISSN: 3006-628X, vol. 5, no. 1, July 2018, pp. 68-74, https://internationaljournals.org/index.php/ijtd/article/view/102.

[101]. Vijaya Venkata Sri Rama Bhaskar, Akhil Mittal, Santosh Palavesh, Krishnateja Shiva, Pradeep Etikani. (2020). Regulating AI in Fintech: Balancing Innovation with Consumer Protection. European Economic Letters (EEL), 10(1). https://doi.org/10.52783/eel.v10i1.1810

[102]. Dave, A., Shiva, K., Etikani, P., Bhaskar, V. V. S. R., & Choppadandi, A. (2022). Serverless AI: Democratizing machine learning with cloud functions. Journal of Informatics Education and Research, 2(1), 22-35. http://jier.org

[103]. Dave, A., Etikani, P., Bhaskar, V. V. S. R., & Shiva, K. (2020). Biometric authentication for secure mobile payments. Journal of Mobile Technology and Security, 41(3), 245-259.

[104]. Saoji, R., Nuguri, S., Shiva, K., Etikani, P., & Bhaskar, V. V. S. R. (2021). Adaptive AI-based deep learning models for dynamic control in software-defined networks. International Journal of Electrical and Electronics Engineering (IJEEE), 10(1), 89–100. ISSN (P): 2278–9944; ISSN (E): 2278–9952

[105]. Narendra Sharad Fadnavis. (2021). Optimizing Scalability and Performance in Cloud Services: Strategies and Solutions. International Journal on Recent and Innovation Trends in Computing and Communication, 9(2), 14–21. Retrieved from https://www.ijritcc.org/index.php/ijritcc/article/view/10889

[106]. Nitin Prasad. (2022). Security Challenges and Solutions in Cloud-Based Artificial Intelligence and Machine Learning Systems. International Journal on Recent and Innovation Trends in Computing and Communication, 10(12), 286–292. Retrieved from https://www.ijritcc.org/index.php/ijritcc/article/view/10750

[107]. Prasad, N., Narukulla, N., Hajari, V. R., Paripati, L., & Shah, J. (2020). AI-driven data governance framework for cloud-based data analytics. Volume 17, (2), 1551-1561.

[108]. Pillai, Sanjaikanth E. VadakkethilSomanathan, et al. "MENTAL HEALTH IN THE TECH INDUSTRY: INSIGHTS FROM SURVEYS AND NLP ANALYSIS." JOURNAL OF RECENT TRENDS IN COMPUTER SCIENCE AND ENGINEERING (JRTCSE) 10.2 (2022): 23-34.

[109]. Big Data Analytics using Machine Learning Techniques on Cloud Platforms. (2019). International Journal of Business Management and Visuals, ISSN: 3006-2705, 2(2), 54-58. https://ijbmv.com/index.php/home/article/view/76

[110]. Shah, J., Narukulla, N., Hajari, V. R., Paripati, L., & Prasad, N. (2021). Scalable machine learning infrastructure on cloud for large-scale data processing. Tuijin Jishu/Journal of Propulsion Technology, 42(2), 45-53.

[111]. Narukulla, N., Lopes, J., Hajari, V. R., Prasad, N., & Swamy, H. (2021). Real-time data processing and predictive analytics using cloud-based machine learning. Tuijin Jishu/Journal of Propulsion Technology, 42(4), 91-102

[112]. Secure Federated Learning Framework for Distributed Ai Model Training in Cloud Environments. (2019). International Journal of Open Publication and Exploration, ISSN: 3006-2853, 7(1), 31-39. https://ijope.com/index.php/home/article/view/145

[113]. Paripati, L., Prasad, N., Shah, J., Narukulla, N., & Hajari, V. R. (2021). Blockchain-enabled data analytics for ensuring data integrity and trust in AI systems. International Journal of Computer Science and Engineering (IJCSE), 10(2), 27–38. ISSN (P): 2278–9960; ISSN (E): 2278–9979.

[114]. DOI: 10.36227/techrxiv.171340711.17793838/v1

[115]. Challa, S. S. S., Tilala, M., Chawda, A. D., & Benke, A. P. (2019). Investigating the use of natural language processing (NLP) techniques in automating the extraction of regulatory requirements from unstructured data sources. Annals of Pharma Research, 7(5),

[116]. Challa, S. S. S., Tilala, M., Chawda, A. D., & Benke, A. P. (2021). Navigating regulatory requirements for complex dosage forms: Insights from topical, parenteral, and ophthalmic products. NeuroQuantology, 19(12), 15.

[117]. Challa, S. S. S., Tilala, M., Chawda, A. D., & Benke, A. P. (2022). Quality management systems in regulatory affairs: Implementation challenges and solutions. Journal for Research in Applied Sciences

[118]. Vijay Kumar Reddy Voddi, Komali Reddy Konda(2022), “Success and Struggle: Countries that Minimized COVID-19 Cases and the Factors Behind Their Outcomes,”ResMilitaris,Volume -12, Issue -5 (2022) Keywords: COVID-19, Pandemic Response, Public Health Strategies, Case Minimization, GlobalHealth,Epidemiology,https://resmilitaris.net/issue-content/success-and-struggle-countries-that-minimized-covid-19-cases-and-the-factors-behind-their-outcomes-4043

[119]. Vijay Kumar Reddy, Komali Reddy Konda(2021),“Unveiling Patterns: Seasonality Analysis of COVID-19 Data in the USA”, Keywords: COVID-19, Seasonality, SARS-CoV-2, Time Series Analysis, Environmental Factors, USA, Neuroquantology | October 2021 | Volume 19 | Issue 10 | Page 682-686|Doi: 10.48047/nq.2021.19.10.NQ21219

[120]. Vijay Kumar Reddy, Komali Reddy Konda(2021), “COVID-19 Case Predictions: Anticipating Future Outbreaks Through Data” Keywords: COVID-19, Case Predictions, Machine Learning, Time Series Forecasting, Pandemic Response, Epidemiological Modeling, NeuroQuantology | July 2021 | Volume 19 | Issue 7 | Page 461-466| doi: 10.48047/nq.2021.19.7.NQ21136

[121]. Vijay Kumar Reddy Voddi, Komali Reddy Konda(2021),“Spatial Distribution And Dynamics Of Retail Stores In New York City,” Pages: 9941-9948 Keywords: Retail Distribution, Urban Planning, Economic Disparities, Gentrification, Online Shopping Trends.https://www.webology.org/abstract.php?id=5248

[122]. T Jashwanth Reddy, Voddi Vijay Kumar Reddy, T Akshay Kumar (2018),” Population Diagnosis System,” Published in International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Keywords: Apache Hadoop 1.2.1,Apache hive-0.12.0,Population Diagnosis System, My SQL. https://ijarcce.com/upload/2018/february-18/IJARCCE%2038.pdf