Journal for Research in Applied Sciences and Biotechnology
ISSN: 2583-4053
Volume-3 Issue-5 || October 2024 || PP. 107-118
www.jrasb.com
https://doi.org/10.55544/jrasb.3.5.13

Challenges and Solutions in Testing Mainframe Applications in Modern Banking

Ashwini Shivarudra
Independent Researcher, USA.

www.jrasb.com || Vol. 3 No. 5 (2024): October Issue

Received: 13-10-2024	**Revised:** 16-10-2024	**Accepted:** 24-10-2024

ABSTRACT

Because the foundation of their operations is still housed in legacy systems, banks continue to use them. Banks must modify their systems to remain competitive in light of the swift changes in technology and customer needs. Mainframe systems have been an integral part of corporate computing for many years, enabling critical applications across a range of industries, including banking, retail, and healthcare. There is rising interest in leveraging Application Programming Interfaces (APIs) to expose these old apps' data and features, hence accelerating the construction of new applications and maximising their capability and enabling their reuse. Nonetheless, there are several obstacles to overcome in the process of locating and making available APIs for various business use cases. In this study, we examine the difficulties involved in developing APIs for mainframe systems that are no longer in use and suggest a unique architecture to enable communication for a range of use cases. We performed a qualitative poll with nine mainframe developers, whose average experience was fifteen years, to evaluate the efficacy of our methodology. We were able to determine which APIs were candidates and how long it would take to implement them for two industry mainframe applications and the public mainframe application GENAPP thanks to the poll. We create a list of artefacts, such as screens, transactions, business rules, inter-micro service calls, control flow blocks, and data accesses, in order to identify APIs. IBM Watsonx Code Assistant for Z Refactoring Assistant includes an implementation for computing API signatures. We proved the practicality of our technique by running the discovered APIs on an IBM Z mainframe system to assess their accuracy.

Keywords- Banks, Mainframe System, GENAPP, Application Programming Interfaces (APIs), Legacy Mainframe, Control Flow Blocks, Z Mainframe System, Banking, Retail, and Healthcare.

I. INTRODUCTION

Mainframe systems continue to be an essential part of corporate computing because of their outstanding performance, stability, and dependability in sectors including banking, retail, insurance, and healthcare. However, there is a growing tendency towards the "APIfication" of mainframe systems due to the growing popularity of cloud computing and micro services. Over 90% of financial institutions either use or intend to utilise APIs to increase income from current clients, according to McKinsey & Co in 2021 [1, 2]. Using solutions like zOS Connect, mainframe APIfication entails modernising systems by providing data and functions through APIs (application programming interfaces) [2]. This change has several advantages:

1. ***Increased Efficiency:*** Faster development cycles and lower operating expenses result from developers' increased ease and efficiency when interacting with mainframe systems [2, 3].
2. ***Integration with Modern Technologies:*** APIfication makes it easier to integrate older technology with more modern ones, such as cloud computing, AI, and the Internet of Things (IoT).
3. ***Access Control Flexibility:*** Fine-grained security controls may be implemented and improved access management is made possible by opening up mainframe systems through APIs.
4. ***Business Agility:*** APIfication promotes agility by enabling quick reactions to shifting business requirements and empowering enhanced analytics [3]. Additionally, developing APIs for fresh applications creates new commercial prospects.

Adopting z-Cloud integration patterns, however, may be hampered by the need for skilled mainframe developers to invest a great deal of time and energy in

ISSN: 2583-4053
Volume-3 Issue-5 || October 2024 || PP. 107-118
https://doi.org/10.55544/jrasb.3.5.13

creating the proper APIs. Innovation is necessary to automate API generation, increasing its efficiency and lowering the cognitive burden and effort in order to handle issues.

Organisations now rely more and more on information systems, or "systems," since the advent of computing [3, 4]. Because of this, some organisations have amassed huge portfolios over time that include hundreds or even thousands of distinct systems. Many of these systems would have been created from the ground up to specifically handle the needs in a certain functional area, such stock control, people, and accounting. But this functional orientation [5] has a tendency to strengthen departmental silos inside the company, creating disjointed "islands of applications" where different systems are isolated from one another. Notably, modern enterprise business solutions like e-business, supply chain management, and customer relationship management have not often benefited from such an approach since they necessitate close information and process integration across several organisational divisions [5, 6].

This conundrum is further limited by the fact that recent entrants into the banking industry have already opened up new platforms that allow applications for round-the-clock online, internet, mobile, and other banking services. Because replacing old systems would be expensive and risky, and because a proprietary system would go bankrupt if a change were to occur, legacy systems are still in use today [8]. There's yet another explanation that has to do with business logic. The development of the apps began in the 1970s, and they were gradually modified to meet commercial needs. As a result, the systems' business logic is highly complicated, which increases the likelihood that it will fail if they are replaced [8, 9].

An analyst at Tower Group stated in 2008 that pressure was coming from both the commercial and technological perspectives. Rivals are employing new open architectures with data mining capabilities, smart report generation, and integration capabilities with tablets, laptops, and smartphones, among other things [9, 10]. This facilitates faster company closing and provides an improved mechanism for decision-making. Since the business and technical communities brought up these concerns [11, 12], big businesses have created services-oriented architecture, or SOA.

Furthermore, cloud computing-based solutions are provided by these suppliers. The dilemma at hand is whether banks need to upgrade their current legacy systems, develop new ones from the ground up, or purchase solutions from other suppliers [12]. We must be aware of the limitations imposed by the continued usage of the legacy systems in order to properly answer this question:

— Legacy systems were not intended to stay in operation for very long—more than 15 years—and were instead created to meet immediate demands;

— These systems were constructed with certain limitations (e.g., low cost, resource availability, etc.);

— Aspects of current systems include commercial components, corporate objectives, complexity, and [11, 12].

Rewriting the program is hazardous (many problems may surface during acceptance testing, and more importantly, the business may change during development) and costly (time consuming). The benefit is that the software program may be customised by the bank to match its unique needs [12]. Purchasing ready-to-use commercial off-the-shelf (COTS) systems that require customisation for the bank's operations is the second alternative. The risks are associated with maintaining such items since they often come with greater modification expenses [12]. The benefits include the software's instant use and the ability to create Service Level Agreements (SLAs) for support services.

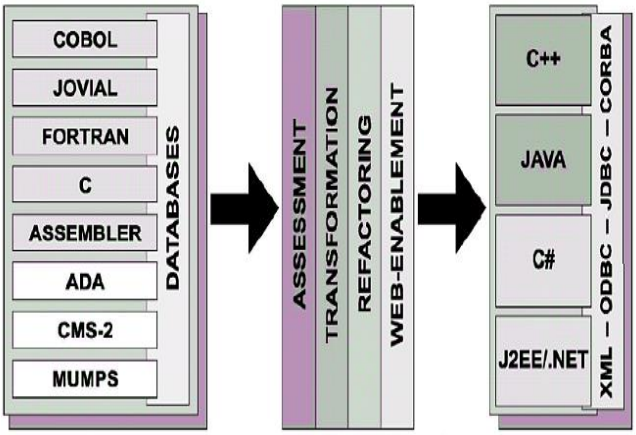

Figure. 1 Actions to transform legacy systems. [12, 13]

1.1 Mainframe

The mainframe computer has long been a myth. They have been there since the beginning of computers and are still around today in improved form. But in the face of cloud computing, mainframes appear destined to completely fade into the very specialised market in which they were once dominant. Right now, having a mainframe already is its greatest benefit. If you don't already have one, there's really no incentive to get one [13, 14], as cloud computing offers options that are nearly always far more affordable. The total control over their own data that mainframes provide big businesses is one of its advantages [13].

The advantages of moving to the cloud would likely be outweighed by the expense of transferring hundreds of thousands of lines of code [13, 14]. Furthermore, because the hardware is user-controlled, mainframes may be more specialised and customised than cloud services. The fact that mainframe computers are unaffected by your internet connection is advantageous as it conserves bandwidth and makes using them convenient even when the internet is unavailable [14].

ISSN: 2583-4053
Volume-3 Issue-5 || October 2024 || PP. 107-118
https://doi.org/10.55544/jrasb.3.5.13

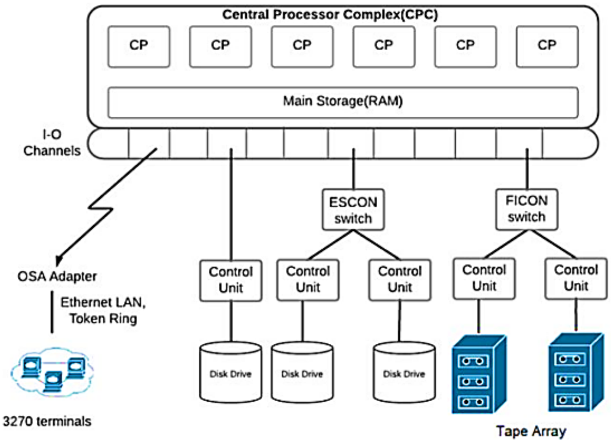

Figure 2: The CPC system. [14]

1.2 Early technological innovations and Mainframes

1.2.1 Doing Business with Information Technology

As of the beginning of the twenty-first century, almost every major corporation in the world has an information technology department, and very few still run their operations without using technology. Nevertheless, a lot of industries, particularly the banking sector, have extremely outdated applications that they still rely on [13]. This industry was one of the first to use computers in their business models, and even though the first practical use of computers was more than 30 years ago, some of these corporations still rely on this outdated code for mission-critical tasks [13, 14].

1.2.2 The Mainframe as the preferred early platform

These early programs were almost often written in PL/I or COBOL and were executed on the mainframe platform. These computers might supply a central server from which several terminals could run applications in a dependable and safe manner [14, 15]. In those early days, the mainframe was more than capable of meeting the needs of the banking industry since it could deliver the several applications required to carry out the institution's core operations in a secure and dependable manner.

1.2.3 Currently using these outdated apps

The needs of modern business, which has drastically altered from these early days of information technology, are something that many banking institutions that employed the mainframe and its applications for must now confront [15]. These organisations must choose how to redesign or reengineer the "legacy applications" in order to give the software solutions that modern consumer’s want [16].

II. THE BANKING SECTOR'S NEED FOR TECHNOLOGY

2.1 The Banking Manufacturing's Nature

In the modern global economy, a bank's function is constantly evolving. A bank's traditional functions included protecting funds and valuables, granting loans, and providing credit and payment services like money orders, cashier's checks, and checking accounts [15]. However, as the banking sector gradually becomes more deregulated, banks are also providing additional investments and insurance products, which they were previously not allowed to sell [16].

2.1.1 Industrial Use of Technological

The banking industry has been conducting business mostly through information technology for a some now [16]. Automated Teller Machines (ATMs) have made it possible for customers to access their accounts around-the-clock and perform many of the ordinary banking activities that were formerly performed in person by tellers, such as making deposits and withdrawals [16, 17]. Businesses and governments can electronically move funds into other accounts by using direct deposit options. When a debit card or "smart card" is swiped via a machine at a retail cash register, money is immediately taken out of the account [17]. Ultimately, bill payment and money transfers between accounts are made possible via phone and, increasingly, online electronic banking [17, 18].

III. NEW TECHNOLOGY-RELATED DIFFICULTIES

Systems that are more functional, accessible, and offer all of the aforementioned services with improved security and performance are necessary in the ever-demanding global world of today. Furthermore, each new program must include Graphical User Interfaces (GUIs), which demand additional processing power [18]. The present high rate of process reengineering combined with these additional expectations is pushing the limitations of existing systems. With the introduction of the Internet, e-commerce, online banking, Wireless Application Protocol (WAP), GUIs, Object Orientated Analysis and Design (OOAD), and Wireless Application Protocol (WAP), there are further pressures to shift away from previous technologies [18, 19]. In addition, information technology (IT) systems must frequently be integrated as part of business takeovers and mergers. When these factors come together, older systems find themselves in a situation where they are unable to advance and make the necessary adjustments. List a few of the dangers that businesses face as a result of the rapid advancements in technology. Among them are:

- The need for constant updating of information arises from the quick changes in markets [19], which raises the price of market intelligence.
- Selling novel technological items carries a higher chance of failure in the marketplace
- Investing in information technology entails significant fixed costs. This cost is further increased by

ISSN: 2583-4053
Volume-3 Issue-5 || October 2024 || PP. 107-118
https://doi.org/10.55544/jrasb.3.5.13

[19], security, performance, interoperability, and equipment depreciation. Lastly, the financial foundation needed for business IT solutions might be quite large.

- The interaction between technology and the human user introduces a human restriction.

The bank most likely obtained consumer data from a third-party ad tech aggregator system, which it then used to target the client. This ad tech company is scuffing data from a variety of sources, including the internet, users' browser histories, and, if turned on, their geolocation data. Subsequently, the bank is blasting out emails to every prospect who could wind up becoming a customer via a marketing system. A web application is being used by Thank Bank to create a website where customers may apply for accounts. The bank then uses the identification and income verification system of a government agency to confirm the customer's identity [19, 20]. The client's profile is being created by the bank in the CRM, or the client relationship management system. To create the account number while opening the bank account, the bank uses a Core banking system [20]. It employs a dispatch system for statements and letters to provide clients with regulatory data, such as notices and statement. Thus, in this simple case study of the customer journey, the data is passing via nine different distributed systems. previously a single monolithic program could have handled and simplified them. However, a monolithic application is limited in its ability to pull data from other systems only if those systems were not developed using the same technology stack or, as in this example, from the same particular vendor [19].

3.1 Distributed vs. Monolithic Organisations

With the introduction of mainframe computers, computer systems for widespread commercial usage began to evolve. Even though other businesses began producing similar systems, IBM's research and technological advancements allowed them to dominate the market. Early in the 1990s, PC use changed the landscape with the commercialisation of the operating system Windows and the widespread adoption of the Unix/Linux operational system [20, 21]. The distributed software systems were developed using code written in languages like C, C++, Java7, etc. [21] and were based on standard PC hardware. They were able to create software solutions that met a variety of reasonably priced corporate demands. Distributed computing gained popularity as a result of the democratisation of PCs. There was a great need to connect these systems for the exchange of data that could improve the user experience in many ways, even though distributed software systems met the needs of a wide range of businesses and essentially generated a more instinctive and user-friendly experiences that is not just limited to computers and scientists [20, 21].

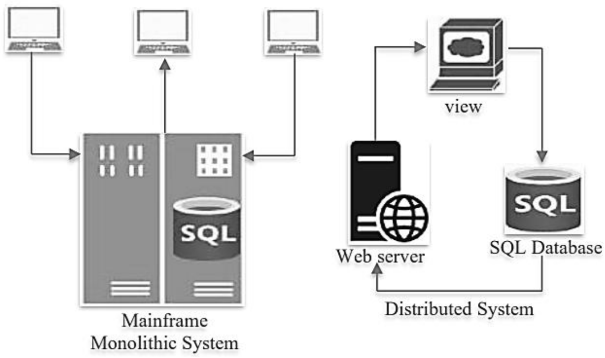

Figure. 3 A logical diagram showing the differences between distributed and monolithic systems. [20]

3.1.1 Identifying Potential APIs Automatically

Our approach entails creating an exhaustive inventory of all artefacts found in the program, including as screens, transactions, control flow blocks, inter-micro service calls, company regulations, data accesses, and others, in order to locate APIs in mainframe systems [20, 21].

1. **Transactions and Jobs:** Transactions and jobs are the usual places where mainframe programs begin their functions, and these may be utilised to find potential APIs [21, 22]. SSC1 for customer management, for instance, and SSP1, SSP2, SSP3, and SSP4 for housing, motor, [22, 23], endowment, and commercial policy management, respectively, are the five primary transactions of GENAPP. Program LGTESTP1, the initial program that SSP1 calls, is where we start the API identification process [23].
2. **Control Flow Blocks:** Since crucial capabilities linked to initiating transactions are frequently found in essential control flow blocks, we take into consideration APIfying them [23, 24].
3. **Data Access Points:** Code that accesses databases or files, loads variables, resolves errors, and any code that comes before or after may all be made available as APIs. We provide Data APIs that support both dynamic and fixed queries, allowing users to pass in any SQL query as an argument to the API. In doing so, a micro service layer for data access is created [24].
4. **Procedures as APIs:** Typically, procedures carry out essential corporate functions. The best candidates to be converted into APIs are stand-alone processes that represent individual business capabilities and do not call other procedures [24, 25]. For instance, operation GET-MOTOR-DB2-INFO in program LGIPDB01 of GENAPP accesses Policy and Motor tables and may be considered an API candidate [25, 26].
5. **Screen Modernisation and Screen APIs:** In mainframe programs, screens are terminals that are used to view the menu. Through the parsing of text files including screen coordinates, we are able to deduce the fields needed for each capability. Each screen's functionalities may then be turned into an API [26, 27],

Journal for Research in Applied Sciences and Biotechnology

www.jrasb.com

ISSN: 2583-4053

Volume-3 Issue-5 || October 2024 || PP. 107-118

https://doi.org/10.55544/jrasb.3.5.13

where input fields serve as requests and output fields serve as answers. Although APIfication makes screen scraping automated, we suggest isolating pertinent code and optimising it to cut down on pointless procedures [28].

6. Rules and Business Features as APIs: We suggest utilising technologies like to detect programs containing business rules. The required code will be extracted by our framework and hosted in decision managers like RedHat Decision Manager.

7. APIs for Communication between Micro Services: In a monolithic design, crossmicroservice function calls must be replaced by APIs to allow connectivity across candidate's micro services. The potential micro services' cross-micro services dependencies will be found using our framework as APIs [25, 26].

8. Application-based APIs from outside and customer demands: The needs of the client or the external method used to call the mainframe program might be used to identify APIs. Our methodology's automated approach to API identification will decrease the work needed to find the best API insertion places within mainframe programs and speed up interface discovery [26].

3.1.2 Locating API Signatures Automatically

COBOL applications frequently use copybooks, which specify Cobol's data structures, to call other programs. Nevertheless, a copybook may have a large number of fields, making it difficult to manually determine which elements would be requests and answers for API purposes. Here, we outline several techniques that are intended to mechanically recognise the API signature, which includes the request and response data [27, 28].

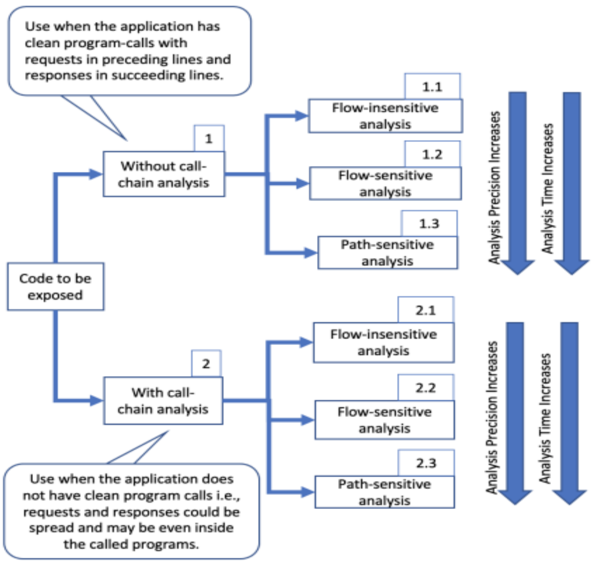

Figure. 4 Variants to use in calculating requests and answers. Combinations of flow sensitive, flow insensitive, path sensitive, and call chain analysis with or without. [28]

IV. EXPERIMENTAL AND QUALITATIVE STUDY

The IBM Watsonx Coding Assistant for Z Refactoring, also Assistant includes our API signature computation implementation. We have used flow-insensitive and flow-sensitive static evaluations in combination with call chain analyses both with and without. Since it can be very time- and resource-intensive, if not impossible, to analyse every potential control flow path for big, complicated software, we have not implemented the path-sensitive technique. In order to maintain secrecy, we used the GENAPP application, a private business application, and a public industry application called CBSA for our tests [26, 28]. Despite the industry's pervasive reliance on mainframes, there is a shortage of COBOL/mainframe developers, which made it difficult to conduct tests on mainframe computer systems and implement COBOL parsing in Python.

4.1 Evaluation of Automated vs. Manual API Identification

Below is a comparison of the manual and our suggested automated methods for GENAPP's APIfication.

- **Identification of Manual APIs:** We worked with five SMEs (subject matter experts) who had 10 years of experience on average [28, 29]. They were knowledgeable in APIfication, mainframe modernisation, and had a solid grasp of the GENAPP code. In the GENAPP usage, the SMEs manually identified 10 data-accessing APIs and 15 transaction-based APIs.
- **Identification of APIs Automatically:** Leveraging our automated methodology, our system identified 15 transaction-based APIs, 21 procedure-based APIs that do not involve data access, and 21 data accessing APIs [25].

Table 1: Analysis of the four variations of Calculating Request and Response Times in Seconds.

API	Without call chain		With call chain	
	FI	FS	FI	FS
TI	19.65	25.64	49.86	125.69
TA	25.65	41.56	120.99	485.22
TU	48.96	27.96	168.92	541.65
DI	29.68	29.89	16.59	29.65
DA	28.96	23.59	14.95	21.56
DU	26.98	24.96	19.95	21.69

Table 2 Number of Fields for Requests and Respond Using the Four Computational Variants for Working Storage + Linkage Sections.

API	Without call chain		With call chain	
	FI	FS	FI	FS
TI	14.59	21.65	21.59	15.54
TA	21.59	22.54	22.26	21.65
TU	19.85	26.54	29.65	26.55

ISSN: 2583-4053
Volume-3 Issue-5 || October 2024 || PP. 107-118
https://doi.org/10.55544/jrasb.3.5.13

DI	11.69	21.54	22.69	21.65
DA	14.52	21.59	25.59	21.25
DU	11.54	26.59	21.69	19.65

Table 3 Count of Request and Reply Fields Using the Four Computational Variants.

API	Without call chain		With call chain	
	FI	**FS**	**FI**	**FS**
API 1	9	1	5	4
API 2	6	5	4	5
API 3	15	16	16	6
API 4	28	24	122	14
API 5	30	39	219	119

4.1.1 Describing the Requirement for Automatic APIfication

Because of the complexity and scale of the data-stores involved, it is imperative to automate the process of recognising requests and answers in mainframe systems [26, 28]. Large data-stores that are frequently defined by mainframe applications and shared amongst programs make it difficult and error-prone to manually identify pertinent queries and answers.

Table 4 Execution Duration (seconds) for client-side calls to GENAPP API exposing on an IBM mainframe.

API	TIME
Find more about a customer claim's automobile policy.	0.58
A customer's motor policy is added.	0.96
Update a customer's automobile insurance.	0.41
Find out the business policy of a client.	0.66
Find out about a customer's claim's housing policy.	0.28
Get information about a customer's endowment policy.	0.61
Ask customers if they have any claims.	0.98
Find out a customer's insurance number.	0.84

4.1.2 Qualitative Study

Nine practitioners with an average of almost fifteen years of work experience participated in our qualitative research. These professionals have a great foundation in APIfication and Mainframe application development, and they are well-versed in the difficulties involved in these operations [29, 30]. A few of them have also worked directly with clients in a variety of sectors, including banking, retail, insurance, and travel. We gave the practitioner access to both the private industry application and GENAPP during the research. Then, we posed certain queries to them concerning the process of manually identifying API endpoints and calculating their signatures. We specifically asked [29] about any obstacles they faced and the amount of time needed to finish these chores by hand [30].

V. CONCLUSION

Owing to the quick advancements in hardware and software, the banking industry strives to keep up with emerging trends. Banks concentrate on replacing outdated software with more recent versions of their programs. In this article, we suggest an all-inclusive architecture for mainframe application APIfication automation. Our method computes API signatures and extracts APIs properly by using approaches from static program analysis. Through building, we verify that our procedures are proper. Using tests and a qualitative research on two industrial mainframe applications and a public mainframe application (GENAPP), we show the scalability and efficacy of our methodology. The findings demonstrate that, in comparison to manual approaches, our automated methodology greatly shortens the time needed for API identification and API signature computation. We also successfully run a portion of our candidate APIs on an IBM Z mainframe system, demonstrating the usefulness of our methodology in real-world situations. All things considered, our suggested framework offers a scalable and effective way to deal with the difficulties associated with mainframe APIfication and makes it easier to convert mainframe programs into contemporary, micro services-based architectures.

REFERENCES

[1] R. C. Seacord, D. Plakosh and G. A. Lewis, “Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices,” Carnegie Mellon Software Engineering Institute, ISBN 0-321-11884-7, February 2003.

[2] A. A. Almonaies, J. R. Cordy, T. R. Dean, Legacy System Evolution towards Service-Oriented Architecture.

[3] S. C. Dorda, K.C. Wallnau, R.C. Seacord and J.E. Robert, “A Survey of Black-Box Modernization Approaches for Information Systems,” Proceeding ICSM '00 Proceedings of the International Conference on Software Maintenance (ICSM'00), IEEE Computer Society Washington, DC, USA ©2000, ISBN: 0-7695-0753-0.

[4] P. Newcomb and R. A. Doblar, “Automated Transformation of Legacy Systems,” CrossTalk: The Journal of Defense Software Engineering, December 2001.

[5] Hafiyyan Putra Pratama, Ary Setijadi Prihatmanto, and Agus Sukoco, “Implementation Messaging Broker Middleware for Architecture of Public Transportation Monitoring System,” 2020 6th International

ISSN: 2583-4053
Volume-3 Issue-5 || October 2024 || PP. 107-118
https://doi.org/10.55544/jrasb.3.5.13

Conference on Interactive Digital Media (ICIDM), Bandung, Indonesia, pp. 1-5, 2020.

[6] Min Luo, and Liang-Jie Zhang, “Practical SOA: Service Modeling, Enterprise Service Bus and Governance,” 2008 IEEE Congress on Services Part II (Services-2 2008), Beijing, China, pp. 13-14, 2008.

[7] Tobias Simon et al., “A Lightweight Message-Based Inter-Component Communication Infrastructure,” 2013 Fifth International Conference on Computational Intelligence, Communication Systems and Networks, Madrid, Spain, pp. 145-152, 2013.

[8] Ridhima Mishra et al., “Transition from Monolithic to Micro services Architecture: Need and Proposed Pipeline,” 2022 International Conference on Futuristic Technologies (INCOFT), Belgaum, India, pp. 1-6, 2022.

[9] Mehdi Bahrami, and Wei-Peng Chen, “Composing Web API Specification from API Documentations through an Intelligent and Interactive Annotation Tool,” 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, pp. 4573-4578, 2019.

[10] Yongxin Feng, and Qin Li, “The Distributed UDDI System Model Based on Service-Oriented Architecture,” 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, pp. 585-589, 2016.

[11] Ian Thomas New combe, “Mainframe Relevance in Modern IT: How a 50+ Year Old Computing Platform Can Still Play a Key Role in Today’s Businesses,” University of New Hampshire, Durham, pp. 1-44, 2016.

[12] Campbell-Kelly Martin, and Daniel D. Garcia-Swartz, From Mainframes to Smartphones: A History of the International Computer Industry, Harvard University Press, pp. 1-220, 2015.

[13] Ross, J.W. and Vitale, M. (2000), The ERP Revolution: Surviving vs. Thriving, Information Systems Frontiers 2(2), pp.233-241.

[14] Sarkis, J. and Sundarraj, R.P. (2003), Managing large-scale global enterprise resource planning systems: a case study at Texas Instruments, International Journal of Information Management, 23, pp.43 1-442, 2003.

[15] Sawhney, M. (2001), ‘Don’t Homogenize, Synchronize’, Harvard Business Review, July-August 2001.

[16] Sharif, A .M. Elliman, T., Love, P.E.D. and Badii, A. (2004), Integrating the IS with the Enterprise: Key EAI research challenges, The Journal of Enterprise Information Management, 17(2), pp.l64-170.

[17] Soh, C., Kien, S .S. Boh, W. F., and Tang, M. (2003), Misalignments in ERP Implementation: A Dialectic Perspective, International Journal of Human-Computer Interaction, 16(1), pp. 81-100.

[18] Manel Abdellatif, Anas Shatnawi, Hafedh Mili, Naouel Moha, Ghizlane El Boussaidi, Geoffrey Hecht, Jean Privat, and Yann-Gaël Guéhéneuc. 2021. A taxonomy of service identification approaches for legacy software systems modernization. Journal of Systems and Software 173 (2021), 110868.

[19] Seza Adjoyan, Abdelhak-Djamel Seriai, and Anas Shatnawi. 2014. Service Identification Based on Quality Metrics Object-Oriented Legacy System Migration Towards SOA. In SEKE: Software Engineering and Knowledge Engineering. Vancouver, Canada, 1–6.

[20] G. Canfora, A.R. Fasolino, G. Frattolillo, and P. Tramontana. 2006. Migrating interactive legacy systems to Web services. In Conference on Software Maintenance and Reengineering (CSMR’06). 10 pp.–36.

[21] Paulius Danenas, Tomas Skersys, and Rimantas Butleris. 2020. Natural language processing-enhanced extraction of SBVR business vocabularies and business rules from UML use case diagrams. Data and Knowledge Engineering 128 (2020), 101822.

[22] Gargi B Dasgupta. 2021. AI and its Applications in the Cloud strategy. In 14th Innovations in Software Engineering Conference (formerly known as India Software Engineering Conference). 1–1.

[23] Utkarsh Desai, Sambaran Bandyopadhyay, and Srikanth Tamilselvam. 2021. Graph neural network to dilute outliers for refactoring monolith application. In Proceedings of 35th AAAI Conference on Artificial Intelligence (AAAI’21).

[24] Alex Mathai, Sambaran Bandyopadhyay, Utkarsh Desai, and Srikanth Tamilselvam. 2021. Monolith to Microservices: Representing Application Software through Heterogeneous GNN. arXiv preprint arXiv:2112.01317 (2021).

[25] Andrew J. McAllister. 2011. The Case for Teaching Legacy Systems Modernization. In 2011 Eighth International Conference on Information Technology: New Generations. 251–256.

[26] Juan Manuel Rodriguez, Marco Crasso, Cristian Mateos, Alejandro Zunino, and Marcelo Campo. 2013. Bottom-Up and Top-Down COBOL System Migration to Web Services. IEEE Internet Computing 17, 2 (2013), 44–51.

[27] R. Rodríguez-Echeverría, F. Maclas, V.M. Pavón, J.M. Conejero, and F. SánchezFigueroa. 2014. Generating a REST Service Layer from a Legacy System. In Information System Development. 433–444.

ISSN: 2583-4053
Volume-3 Issue-5 || October 2024 || PP. 107-118
https://doi.org/10.55544/jrasb.3.5.13

[28] Tomas Skersys, Paulius Danenas, Rimantas Butleris, Armantas Ostreika, and Jonas Ceponis. 2021. Extracting SBVR Business Vocabularies from UML Use Case Models Using M2M Transformations Based on Drag-and-Drop Actions. Applied Sciences 11, 14 (2021).

[29] Mili, H., El Boussaidi, G., Shatnawi, A., Guéhéneuc, Y. G., Moha, N., Privat, J., & Valtchev, P. (2017). Service-oriented re-engineering of legacy JEE applications: Issues and research directions.

[30] Jamshidi, P., Pahl, C., & Mendonça, N. C. (2017). Pattern-based multi-cloud architecture migration. Software: Practice and Experience, 47(9), 1159-1184.

[31] Kulkarni, A. (2024). Digital transformation with SAP Hana. International Journal on Recent and Innovation Trends in Computing and Communication, 12(1), 338–344. Retrieved from //ijritcc.org/index.php/ijritcc/article/view/10849

[32] Kulkarni, A. (2024). Generative AI-driven for SAP Hana analytics. International Journal on Recent and Innovation Trends in Computing and Communication, 12(2), 438–444.

[33] Enhancing Customer Experience with AI-Powered Recommendations in SAP HANA. (2024). International Journal of Business Management and Visuals, ISSN: 3006-2705, 7(1), 1-8. https://ijbmv.com/index.php/home/article/view/84

[34] Kulkarni, Amol. "Digital Transformation with SAP Hana.‖, International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169, Volume 12, Issue 1, Pages 338-344, 2024.

[35] Kulkarni, Amol. "Enhancing Customer Experience with AI-Powered Recommendations in SAP HANA." International Journal of Business Management and Visuals, ISSN: 3006-2705 7.1 (2024): 1-8.

[36] Amol Kulkarni. (2024). Natural Language Processing for Text Analytics in SAP HANA. International Journal of Multidisciplinary Innovation and Research Methodology, ISSN: 2960-2068, 3(2), 135–144. Retrieved from https://ijmirm.com/index.php/ijmirm/article/view/93

[37] 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.

[38] 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

[39] 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.

[40] Lohith Paripati, Varun Nakra, Pandi Kirupa Gopalakrishna Pandian, Rahul Saoji, Bhanu Devaguptapu. (2023). Exploring the Potential of Learning in Credit Scoring Models for Alternative Lending Platforms. European Economic Letters (EEL), 13(4), 1331–1241. https://doi.org/10.52783/eel.v13i4.1799

[41] Etikani, P., Bhaskar, V. V. S. R., Nuguri, S., Saoji, R., & Shiva, K. (2023). Automating machine learning workflows with cloud-based pipelines. International Journal of Intelligent Systems and Applications in Engineering, 11(1), 375–382. https://doi.org/10.48047/ijisae.2023.11.1.37

[42] Etikani, P., Bhaskar, V. V. S. R., Palavesh, S., Saoji, R., & Shiva, K. (2023). AI-powered algorithmic trading strategies in the stock market. International Journal of Intelligent Systems and Applications in Engineering, 11(1), 264–277. https://doi.org/10.1234/ijsdip.org_2023-Volume-11-Issue-1_Page_264-277.

[43] 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

[44] Varun Nakra, Arth Dave, Savitha Nuguri, Pradeep Kumar Chenchala, Akshay Agarwal. (2023). Robo-Advisors in Wealth Management: Exploring the Role of AI and ML in Financial Planning. European Economic Letters (EEL), 13(5), 2028–2039. Retrieved from https://www.eelet.org.uk/index.php/journal/article/view/1514

[45] Pradeep Kumar Chenchala. (2023). Social Media Sentiment Analysis for Enhancing Demand Forecasting Models Using Machine Learning Models. International Journal on Recent and Innovation Trends in Computing and Communication, 11(6), 595–601. Retrieved from

ISSN: 2583-4053
Volume-3 Issue-5 || October 2024 || PP. 107-118
https://doi.org/10.55544/jrasb.3.5.13

https://www.ijritcc.org/index.php/ijritcc/article/view/10762

[46] Varun Nakra. (2023). Enhancing Software Project Management and Task Allocation with AI and Machine Learning. International Journal on Recent and Innovation Trends in Computing and Communication, 11(11), 1171–1178. Retrieved from https://www.ijritcc.org/index.php/ijritcc/article/view/10684

[47] Lindiawati, Indrianawati, Astuti, S. W., Nuguri, S., Saoji, R., Devaguptapu, B., & Prasad, N. (2023). The Information Quality of Corporate Social Responsibility in Leveraging Banks CSR Reputation: A Study of Indonesian Banks. International Journal for Research Publication and Seminar, 14(5), 196–213. https://doi.org/10.36676/jrps.v14.i5.144

[48] Krishnateja Shiva, Pradeep Etikani, Vijaya Venkata Sri Rama Bhaskar, Savitha Nuguri, Arth Dave. (2024). Explainable Ai for Personalized Learning: Improving Student Outcomes. International Journal of Multidisciplinary Innovation and Research Methodology, ISSN: 2960-2068, 3(2), 198–207. Retrieved from https://ijmirm.com/index.php/ijmirm/article/view/100

[49] Varun Nakra. (2024). AI-Driven Predictive Analytics for Business Forecasting and Decision Making. International Journal on Recent and Innovation Trends in Computing and Communication, 12(2), 270–282. Retrieved from https://ijritcc.org/index.php/ijritcc/article/view/10619

[50] Agarwal, A., Devaguptapu, B., Saoji, R., Naguri, S., & Avacharmal, R. (2024). Implementing artificial intelligence in salon management: Revolutionizing customer relationship management at PK Salon. Journal Name, 45(2), 1700.

[51] Avacharmal, R., Agarwal, A., Devaguptapu, B., Saoji, R., & Naguri, S. (2024). Implementing artificial intelligence in salon management: Revolutionizing customer relationship management at PK Salon. Journal of Propulsion Technology, 45(2), 1700-1712.

[52] Harishbhai Tilala M, Kumar Chenchala P, Choppadandi A, Kaur J, Naguri S, Saoji R, Devaguptapu B. Ethical Considerations in the Use of Artificial Intelligence and Machine Learning in Health Care: A Comprehensive Review. Cureus.16(6):e62443. doi: 10.7759/cureus.62443. PMID: 39011215; PMCID: PMC11249277.Jun 15, 2024.

[53] 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

[54] 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

[55] 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.

[56] Rinkesh Gajera. (2024). Comparative Analysis of Primavera P6 and Microsoft Project: Optimizing Schedule Management in Large-Scale Construction Projects. International Journal on Recent and Innovation Trends in Computing and Communication, 12(2), 961–972. Retrieved from https://www.ijritcc.org/index.php/ijritcc/article/view/11164

[57] 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

[58] Rinkesh Gajera , "Integrating Power Bi with Project Control Systems: Enhancing Real-Time Cost Tracking and Visualization in Construction", International Journal of Scientific Research in Civil Engineering (IJSRCE), ISSN : 2456-6667, Volume 7, Issue 5, pp.154-160, September-October.2023. URL : https://ijsrce.com/IJSRCE123761

[59] Rinkesh Gajera, "The Impact of Smartpm's Ai-Driven Analytics on Predicting and Mitigating Schedule Delays in Complex Infrastructure Projects", Int J Sci Res Sci Eng Technol, vol. 11, no. 5, pp. 116–122, Sep. 2024, Accessed: Oct. 02, 2024. [Online]. Available: https://ijsrset.com/index.php/home/article/view/IJSRSET24115101

[60] Rinkesh Gajera. (2024). IMPROVING RESOURCE ALLOCATION AND LEVELING IN CONSTRUCTION PROJECTS: A COMPARATIVE STUDY OF AUTOMATED TOOLS IN PRIMAVERA P6 AND MICROSOFT PROJECT. International Journal of Communication Networks and Information Security (IJCNIS), 14(3), 409–414. Retrieved from https://ijcnis.org/index.php/ijcnis/article/view/7255

ISSN: 2583-4053

Volume-3 Issue-5 || October 2024 || PP. 107-118

https://doi.org/10.55544/jrasb.3.5.13

[61] Gajera, R. (2024). Enhancing risk management in construction projects: Integrating Monte Carlo simulation with Primavera risk analysis and PowerBI dashboards. Bulletin of Pure and Applied Sciences-Zoology, 43B(2s).

[62] Gajera, R. (2024). The role of machine learning in enhancing cost estimation accuracy: A study using historical data from project control software. Letters in High Energy Physics, 2024, 495-500.

[63] Rinkesh Gajera. (2024). The Impact of Cloud-Based Project Control Systems on Remote Team Collaboration and Project Performance in the Post-Covid Era. International Journal of Research and Review Techniques, 3(2), 57–69. Retrieved from https://ijrrt.com/index.php/ijrrt/article/view/204

[64] Rinkesh Gajera, 2023. Developing a Hybrid Approach: Combining Traditional and Agile Project Management Methodologies in Construction Using Modern Software Tools, ESP Journal of Engineering & Technology Advancements 3(3): 78-83.

[65] Paulraj, B. (2023). Enhancing Data Engineering Frameworks for Scalable Real-Time Marketing Solutions. Integrated Journal for Research in Arts and Humanities, 3(5), 309–315. https://doi.org/10.55544/ijrah.3.5.34

[66] 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

[67] Balachandar Paulraj. (2024). LEVERAGING MACHINE LEARNING FOR IMPROVED SPAM DETECTION IN ONLINE NETWORKS. Universal Research Reports, 11(4), 258–273. https://doi.org/10.36676/urr.v11.i4.1364

[68] 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

[69] 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

[70] 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

[71] 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

[72] Balachandar Paulraj. (2024). SCALABLE ETL PIPELINES FOR TELECOM BILLING SYSTEMS: A COMPARATIVE STUDY. Darpan International Research Analysis, 12(3), 555–573. https://doi.org/10.36676/dira.v12.i3.107

[73] Ankur Mehra, Sachin Bhatt, Ashwini Shivarudra, Swethasri Kavuri, Balachandar Paulraj. (2024). Leveraging Machine Learning and Data Engineering for Enhanced Decision-Making in Enterprise Solutions. International Journal of Communication Networks and Information Security (IJCNIS), 16(2), 135–150. Retrieved from https://www.ijcnis.org/index.php/ijcnis/article/view/6989

[74] Bhatt, S., Shivarudra, A., Kavuri, S., Mehra, A., & Paulraj, B. (2024). Building scalable and secure data ecosystems for multi-cloud architectures. Letters in High Energy Physics, 2024(212).

[75] Balachandar Paulraj. (2024). Innovative Strategies for Optimizing Operational Efficiency in Tech-Driven Organizations. International Journal of Intelligent Systems and Applications in Engineering, 12(20s), 962 –. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/6879

[76] 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

[77] Bhatt, S. (2023). A comprehensive guide to SAP data center migrations: Techniques and case studies. International Journal of Scientific Research in Science, Engineering and Technology, 10(6), 346. https://doi.org/10.32628/IJSRSET2310630

[78] 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

[79] Kavuri, S., & Narne, S. (2023). Improving performance of data extracts using window-based refresh strategies. International Journal of Scientific Research in Science, Engineering and Technology, 10(6), 359. https://doi.org/10.32628/IJSRSET2310631

[80] Kavuri, S. (2024). Automation in distributed shared memory testing for multi-processor systems. International Journal of Scientific

Research in Science, Engineering and Technology, 12(4), 508. https://doi.org/10.32628/IJSRSET12411594

[81] Swethasri Kavuri, "Integrating Kubernetes Autoscaling for Cost Efficiency in Cloud Services", Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol, vol. 10, no. 5, pp. 480–502, Oct. 2024, doi: 10.32628/CSEIT241051038.

[82] Swethasri Kavuri. (2024). Leveraging Data Pipelines for Operational Insights in Enterprise Software. International Journal of Intelligent Systems and Applications in Engineering, 12(10s), 661–682. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/6981

[83] 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

[84] Mehra, A. (2023). Strategies for scaling EdTech startups in emerging markets. International Journal of Communication Networks and Information Security, 15(1), 259–274. https://ijcnis.org

[85] 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

[86] 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

[87] Mehra, A. (2023). Leveraging Data-Driven Insights to Enhance Market Share in the Media Industry. Journal for Research in Applied Sciences and Biotechnology, 2(3), 291–304. https://doi.org/10.55544/jrasb.2.3.37

[88] 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

[89] Mehra, A. (2023). Innovation in brand collaborations for digital media platforms. IJFANS International Journal of Food and Nutritional Sciences, 12(6), 231. https://doi.org/10.XXXX/xxxxx

[90] 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

[91] Mehra, A. (2023). Leveraging Data-Driven Insights to Enhance Market Share in the Media Industry. Journal for Research in Applied Sciences and Biotechnology, 2(3), 291–304. https://doi.org/10.55544/jrasb.2.3.37

[92] 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

[93] 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

[94] 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

[95] 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

[96] Narne, S. (2023). Predictive analytics in early disease detection: Applying deep learning to electronic health records. African Journal of Biological Sciences, 5(1), 70–101. https://doi.org/10.48047/AFJBS.5.1.2023.7

[97] Swethasri Kavuri. (2024). Leveraging Data Pipelines for Operational Insights in Enterprise Software. International Journal of Intelligent Systems and Applications in Engineering, 12(10s), 661–682. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/6981

[98] Narne, S. (2024). The impact of telemedicine adoption on patient satisfaction in major hospital chains. Bulletin of Pure and Applied Sciences-Zoology, 43B(2s).

[99] 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

[100] Vijay Kumar Reddy Voddi, Komali Reddy Konda. (2024),"Electric Cars Meet AI: Machine Learning Revolutionizing the Future of Transportation," International Journal of Communication Networks and Information Security (IJCNIS), 16(2), 157–160. Keywords: Electric Vehicles, Artificial Intelligence, Machine Learning, Autonomous Driving,

Battery Management, Predictive Maintenance, Sustainable Transportation. Retrieved from https://ijcnis.org/index.php/ijcnis/article/view/7367

[101] K. R. Voddi, "Bike Sharing: An In-Depth Analysis on the Citi Bike Sharing System of Jersey City, NJ," 2023 6th International Conference on Recent Trends in Advance Computing (ICRTAC), Chennai, India, 2023, pp. 796-804, doi: 10.1109/ICRTAC59277.2023.10480792. keywords: {Costs;Shared transport;Urban areas;Sociology;Bicycles;Predictive models;Market research;component;formatting;style;styling;insert} https://ieeexplore.ieee.org/document/10480792

[102] Reddy Voddi, V. K. (2023)," The Road to Sustainability: Insights from Electric Cars Project," International Journal on Recent and Innovation Trends in Computing and Communication, 11(11), 680–684. Keywords: Electric Vehicles, Sustainability, Environmental Impact, Battery Technology, Charging Infrastructure, Policy, Renewable Energy https://doi.org/10.17762/ijritcc.v11i11.10071

[103] 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

[104] 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

[105] 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

[106] 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

[107] 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