



A Survey on the Role of Digital Twin Technology in Software Engineering and Development

Mr. Deepak Mehta

Assistant Professor

Department of Computer Sciences and Assistant Professor: Applications

Mandsaur University, Mandsaur

deepak.mehta@meu.edu.in

Abstract—Software engineering is a crucial field for the creation of software systems that are reliable, efficient, and maintainable. But complexities, changing needs, tests, and maintenance issues remain as hurdles along the software development journey. The development of digital twins (DT) has allowed for the creation of a solution that can be utilised for optimisation, analysis, simulation, and real-time monitoring of software or physical systems. This survey paper introduces some software engineering concepts and the concept of DT in the modern software development. The study elaborates on software development processes, major engineering challenges, Digital Twin types, enabling technologies and DT integration levels. Additionally, it looks at the use of DT throughout the Software Development Life Cycle, such as system monitoring, security management, process modeling, and systems engineering. Also taken up in the paper is a review of recent research and industrial applications of DT technology. The results indicate that DT can contribute to better software quality, decision-making, predictive maintenance and continuous optimization of the system. DT present a great opportunity for improving software engineering practices and software lifecycle management as software systems become more complex.

Keywords—Software Engineering, Digital Twin, Software Development Lifecycle, Software Digital Twin, Real-Time Monitoring, Predictive Maintenance, Systems Engineering.

I. INTRODUCTION

In today's technology-driven society, software engineering is an essential component in the creation of software systems. This discipline is seeing fast growth. Software engineering is a multi-faceted discipline that aims to provide organisations and individuals with dependable software solutions by planning, designing, developing, testing, and maintaining software systems. The SDLC is a popular model for software development that describes the phases of creating, releasing, and maintaining software. There are a lot of SDLC models that have been suggested to solve for various development requirements and methodologies [1]. Despite this, software development is a complicated process that requires proper documentation of system requirements, implementation strategies, testing, deployment and maintenance throughout the software lifecycle [2].

The cutting-edge digital twin (DT) technology makes a perfect copy of anything, either digitally or physically, and then updates it with all the most recent data [3]. This technology has completely transformed several sectors by enhancing simulation accuracy, real-time monitoring, predictive analysis, and performance optimisation. The broad use of digital twin technology in the present period has several

benefits, such as improved product design and construction, faster problem identification and resolution, more operational efficiency, and better firm performance [4][5]. Among its benefits are the following: risk assessment, real-time monitoring and control, enhanced cooperation, predictive maintenance, and data-driven decision-making [6][7]. Software engineers and developers have been quite interested in DT in recent years. Software Digital Twins (SDTs) are computer models of software systems that are constantly updated with execution data for the purpose of testing, optimising, and monitoring program behaviour [8][9]. While the traditional static testing methods [10] operate under pre-defined test conditions, the SDTs simulate real-world operational environments and rely on AI analytics for defect prediction, performance testing, security testing and predictive maintenance [11]. These features make Digital Twin a potential paradigm for enhancing software quality, reliability, and lifecycle management. Hence, this survey seeks to examine the impact of the DT technology on software engineering and software development with applications, merits, challenges, and future research directions.

A. Structure of Paper

The paper is organized as follows: An introduction to software engineering is given in Section II. The Digital Twin concepts and enabling technologies are covered in Section III. In Section IV, Digital Twin applications in software engineering are discussed. The review of industrial applications is provided in Section V. The literature review and the research gap analysis are in section VI. Last, Section VII brings the paper to an end and provides some ideas for future research.

II. SOFTWARE ENGINEERING PROCESSES AND CHALLENGES

A fast-expanding area of software in which software systems are systematically designed, developed, tested, and maintained. Software engineers need to adhere to a set of basic principles that guide their work to make a high quality software system. In this article, examine the key principles of software engineering and discuss their significance in software development.

- **Modularity:** Modularity refers to the principle of decomposition of software system into smaller, independent modules which can be developed and tested separately [12]. This principle promotes the reusability and maintainability of software systems by allowing developers to make changes to one module without affecting the other modules.

- **Maintainability:** Maintainability is the ability to modify, update, and maintain a software system over time. In software development, this principle is crucial because it allows software engineers to adapt or modify the software system when necessary, without suffering huge costs and time restraints [13].
- **Reliability:** The principle that software systems are trustworthy, reliable and can perform their intended functions without failure. In software development, this principle is relevant in terms of the system quality and user confidence in the quality of the system's performance of its intended functions.
- **Usability:** Usability is the design principle of user friendly and easy to use software systems. It is very important to software development because it has a significant impact on users' adoption and success using the software system.
- **Efficiency:** Efficiency refers to designing software systems that are optimized for performance and use resources efficiently [14]. In software development, this principle can influence the scalability and performance of the application, particularly in large-scale deployments.

A. Software Development Process

Software engineers design, create, and provide software systems through a sequence of activities and procedures known as the software development process. There are several steps to the process, including collecting requirements, designing, implementing, testing, and maintaining. Systematic software development entails a lifecycle (SDLC) that includes the steps shown in Fig. 1. The following procedures must be followed: preparation, investigation, design, execution, evaluation, incorporation, and upkeep.



Fig. 1. Software Development Life Cycle (SDLC) Phases

- **Planning:** The first step in developing software is gathering requirements. During this phase, stakeholders and software developers work together to determine the requirements of the software system and its intended usage. This phase guarantees that the end product satisfies the requirements of all stakeholders and lays the groundwork for the development process.
- **Analysis:** The analysis phase involves examining the gathered requirements to assess feasibility, identify system functionalities, and define detailed specifications for developing an efficient and reliable software system.
- **Design:** Use the needs gathered in the requirements collecting phase to inform the design step, which entails creating a detailed plan of the software system's architecture, data structures, algorithms, and user interfaces. This step sets the stage for execution.
- **Implementation:** Software developers create the actual software system based on the requirements laid out in the design specifications. The process involves coding, fixing bugs, and testing each part to make the end result.
- **Testing:** Software engineers test the software system to find and correct bugs or errors in this phase. It encompasses both unit and integration testing and system testing to guarantee quality and reliability of software.
- **Maintenance:** The last step is to carry out the continuous updates, bug fixes, and improvements to the software system. Maintenance is crucial to the long-term success and dependability of software.

B. Challenges Faced by a Software Engineer

Software engineering is a field that faces numerous challenges that can affect the quality and success of software systems. Software programmers encounter a number of significant obstacles, including:

- **Complexity:** Software systems are becoming increasingly complex, with a large number of components, functions, and interactions. Software developers may face challenges in comprehending and controlling the system due to its complexity, which in turn increases the likelihood of errors and other difficulties.
- **Requirements gathering:** Software developers must guarantee that the software system satisfies the demands of stakeholders by correctly gathering their requirements. Nevertheless, stakeholders' needs could alter or even clash while development is underway. As a result, stakeholders may end up with software solutions that fall short of their expectations.
- **Testing:** In software development, testing is essential but sometimes a painstaking and time-consuming procedure. Software engineers need to ensure they have adequate test coverage while balancing the need to deliver the software system within a specified timeline.
- **Maintenance:** Software systems need to be maintained over time to fix bugs, add new features, and keep up with changing technology. This can be a significant challenge for software engineers, as it requires them to understand and modify complex systems.
- **Quality:** Software engineers need to ensure that the software systems they develop are of high quality, including being reliable, secure, and easy to use. This can be difficult, as it requires balancing multiple trade-offs, such as speed and security.
- **Skill shortages:** Software engineering is a highly skilled field, and there is a shortage of skilled software engineers. This can lead to a lack of resources for software development projects and can result in projects being delayed or not being completed.

III. DIGITAL TWIN CONCEPTS AND TECHNOLOGIES

The idea of DT is relatively new, but it has become more popular as the Fourth Industrial Revolution has progressed. Constantly evolving digital technology is reshaping industries in what is being called the Fourth Industrial Revolution. Next, take a look at the software industry's existing expectations and compare them to those in manufacturing. The goal is to understand how DTs may improve software development, deployment, and maintenance. present an array of digital twin types applicable to various physical systems.

- **Prototype:** The prototype digital twin enables designers and engineers to assess the system's functionality, find problems, and implement fixes prior to building the real system [15]. Additionally, it enables the testing of many situations and operational settings that might be challenging, if not impossible, to reproduce in the real world.
- **Monitoring:** Use it to view and comprehend the system's operation, locate and resolve issues, and predict future inefficiencies or difficulties [16]. It is possible to construct a monitoring digital twin by adding data from sensors, operational data in real-time, and other pertinent information from the physical system to the digital replicate. As a result, the digital twin can simulate the system's actions and results almost instantly.
- **Predictive:** The most up-to-date methods for predicting the system's future performance using data collected from the physical system are data analytics and machine learning.
- **Prescriptive:** Integrating the predictive DT's ability to forecast future operations with the system's sophisticated analytics and decision-making capabilities, the system suggests specific actions to improve the system's performance.
- A system, process, or physical object's "imaginary digital twin" allows for analysis, optimisation, and simulation of that object's or system's behaviour without actually testing it. By manipulating this "imaginary" twin, may learn a lot about the object's efficiency, dependability, and performance under various situations. It is more of a theoretical model than an exact replica of an actual thing, as the "imaginary" part implies.
- A physical system, process, or product's digital twin can be autonomous if it gets data from the real thing and uses that data to make choices and perform actions on its own and may use it to operate the system, make choices in real-time, and imitate the system's behavior—all without involving a person [17].

A. Digital twin technologies

Data gathering in real-time, data acquisition to provide insightful insights, data duplication of physical items, and data storage are the four cornerstones of digital twin technology. Cloud computing, AI, the IoT, and XR are all parts of this category of technology [18]. Furthermore, Digital Twin makes varying degrees of use of a certain technology depending on the nature of the application.

Cloud Computing: "Cloud computing" is the act of offering managed services over the Internet. Data storage and retrieval over the Internet are both made easier by this technology [19]. Cloud computing enables DT to access data

processing and storage capabilities. Despite working with enormous volumes of data, Digital Twin may use cloud computing to store information on a virtual cloud and access it easily from any location [20][21]. DT can avoid the problems of storing enormous amounts of data and significantly reduce the computation time of complex systems by using cloud computing.

Artificial Intelligence (AI): The field of CS known as AI studies how the brain functions with the ultimate goal of creating robots with intelligence on par with that of humans. AI encompasses a wide range of disciplines, some of which include robotics, picture recognition, and language recognition [22][23]. DT can gain a significant analytical advantage with the use of AI technologies like deep learning, expert systems, neural networks, and machine learning. These tools can automatically analyse data, make conclusions, predict outcomes, and provide solutions to potential problems [24].

Extended Reality (XR): A generic term for all forms of immersive technology. The merging of the real and virtual worlds is one way these developments could push reality's thresholds farther. The advent of XR has made it feasible to model physical objects digitally, allowing them to live and interact in real time. To facilitate user interaction with digital material, DT employs AR capabilities to digitally recreate real-world items.

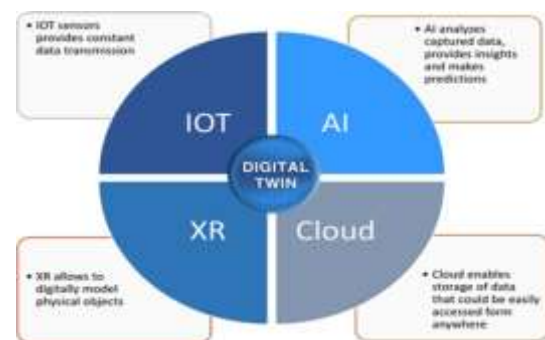


Fig. 2. Technologies of Digital Twins

Figure 2: The IoT, AI, VR, and cloud computing are all components of digital twin technology. This enables real-time analysis, monitoring, and smart decision-making.

B. Levels of Digital Twin Integration

Three tiers of digital twin integration exist: digital model, digital shadow, and digital twin in its most basic form. These levels indicate the evolution from a basic digital representation to a fully connected real-time system.

- **Digital Model:** A digital paradigm in which data is not exchanged between the physical system and the DT.
- **Digital Shadow:** A digital shadow occurs when the DS receives just one-way communication on changes made to the physical system.
- **Digital Twin:** A digital twin is a two-way communication system in which a modification to one component influences the other [25].

IV. APPLICATIONS OF DIGITAL TWINS IN SOFTWARE ENGINEERING

A digital twin's lifespan is intricately linked to that of the system it is simulating. It is not necessary for the digital twin's

life cycle to coincide with that of its real system. The life cycle of a digital twin used for simulation-based prediction during system design, for instance, would end when the actual system was built and put into operation; after that, the digital twin would no longer be relevant. Like engineered items, software systems have a life cycle. Beginning with system analysis and design, developers go on to implementation, generation, operation, re-engineering, stopping, or replacement in the end-of-life phase of waterfall-like development processes. Such a software system might have a digital twin at its side during its whole life cycle.

A. Real-Time Monitoring, Security, and System Control

Software systems are monitored and controlled, enabling iterative improvements and adaptations based on real data. The real-time monitoring and assessment of risks in software systems, especially in cyber-physical systems, enabled by the deployment of DTs, can enhance software system security. Integrated a secure software development life cycle with DTs to minimize risks associated with cybersecurity. This means DTs can proactively manage risks and even develop a contingency plan. The implementation of DT in software development enables the continuous virtualization of physical entities that require data analysis and control capabilities[15]. This, in turn, tracks and controls functionality, resulting in better architectural design and promoting real-time assessment and optimization of system performance. DTs help monitor Complex Systems. This is facilitated by the management of intricate software architectures and by ensuring that all components function harmoniously.

B. Models of a Software Engineering Process

- Software engineering process models encompass various aspects. These include process models, which detail activity graphs and their relationships to stakeholders, data models that pertain to development information (such as tickets and developer properties), constraint models that the DTs' services rely on (such as those regarding available resources), and AI models that inform process decisions.
- Software engineering models should also be taken into account, such as architectural design models, goal models that describe user-centric system requirements, sequence models that detail the system's behaviour, and test models that detail test cases and data.

C. Digital Twins as Enablers of System Engineering

An intelligent DT is one that studies the needs of a system and uses it to direct the engineering of its target entity. A DT may create or refine models, which in turn helps a system evolve and adapt during its lifetime. To validate its claims, the DT might direct its engineering by assuming it can replicate a certain system. This can lead to features like virtual testing. By facilitating system modelling and allowing for the testing of system attributes in simulation, the DT can even be a help to the design process [26]. Digital prototyping aids engineering DT in the shape of a DT Prototype [27].

V. DIGITAL TWINS IN THE SOFTWARE INDUSTRY

A lot of ideas from other fields have been brought into the software business to make it more efficient and productive. However, there are significant distinctions between the two camps due to the software market's incessantly changing

nature and the tremendous flexibility of its participants. Following are a few examples of how the software business has effectively incorporated the practices and procedures of more conventional sectors: An essential idea modified using various methods and structures grounded in lean ideas from the Japanese business world. The Kanban system used by Toyota assembly lines is one possible example.

- **Quality assurance and quality control:** The software industry uses Six Sigma and other disciplined approaches despite the fact that software quality is not the same as manufacturing quality.
- **Software engineering:** The software industry has benefited from this field's use of conventional engineering concepts and practices. Software engineering has adopted lean manufacturing ideas.
- **Operations:** A number of ways in which modern manufacturing has impacted the software business are substantial. Among these are reliability, automated testing, process automation, and supply chain management. Recent developments in software development have given birth to the DevOps paradigm, which emphasises integration.
- **Security:** Modern software packages must incorporate enterprise-level IT security measures from the start to protect sensitive data from cybercriminals and other online threats [25]. Technologies that continuously assess potential vulnerabilities at any stage of the software development pipeline and development processes that encourage security and compliance have a direct impact on software product security.

VI. LITERATURE REVIEW

This section provides that Digital Twin technology is increasingly adopted across software and industrial domains, offering enhanced monitoring, interoperability, and lifecycle management while facing challenges related to standardization, scalability, real-time integration, and practical deployment.

M. Luber, L. Wittmann, and J. Schilp (2026) offers a framework for software development whereby modular, prefabricated components appropriate platforms host the digital twin components, which are then linked through data linkages. The infrastructure utilizes the Asset Administration Shell to ensure interoperability with production management applications and employs a shop floor integration platform for data connection and transformation. In a controlled laboratory setting, its efficacy is proven through the use of assembly robot and production machine DT coupled with ML models for capacity prediction and discrete event simulation [28].

Y. Yan et al., (2025) Data perception and virtual model development have received the most of the research attention, while real-time data processing and interactions between physical and virtual models have received surprisingly little attention. In road engineering, Digital Twin applications are mainly concentrated in operation and maintenance, with limited focus on construction and demolition phases. Future research should emphasize standardization, advanced data interaction techniques, cost optimization, and lifecycle-wide applications to accelerate digital transformation in road engineering[29].

Z. Lei, J. Wan, and L. Fang (2025) Introducing Factory Twin, an innovative open-source software solution for smart

workshops that includes dual-mode scalable digital twin capabilities. Through the use of this software, which mimics smart workshops, it is possible to remotely monitor and manage physical space equipment in real time. There is also assurance that the software model can be scaled to meet different needs. Used the smart manufacturing line to test how well the software worked [30].

M. A. Guinea-Cabrera and J. A. Holgado-Terriza (2024) ALM is the focus of this research, which looks at how the software industry is currently using DT. After sorting through 251 papers, a comprehensive literature search using Scopus, Web of Science, and ScienceDirect uncovered 93 papers that met the inclusion criteria. The findings indicate that DT are increasingly influencing the software industry, with applications across software development, deployment, and maintenance, and are expected to significantly transform the software development lifecycle[25].

J. Liu et al., (2023) presents an ALMS based on Digital Twin technology. The system integrates lifecycle

management and monitoring of airfield lights, signs, and cables, and employs advanced Digital Twin technologies to develop a realistic 3D platform. It enables centralized monitoring of maintenance, performance, hazards, and spare parts while improving airport management efficiency and reliability [31].

D. Sun (2023) discusses the potential of using Digital Twin technology for civil airport safety management and suggests a framework for its implementation. The results show that Digital Twin enables spatial visualization, monitoring, and management of safety information, while supporting emergency rescue simulation, command, and dispatch through integration with rescue plans and expert systems[32].

Table I summarizes recent studies on Digital Twin technology in software engineering, highlighting their approaches, advantages, challenges, and future research recommendations.

TABLE I. TABLE I. SUMMARY OF THE STUDY ON DIGITAL TWIN TECHNOLOGY IN SOFTWARE ENGINEERING

Author's	Approach	Advantages	Challenges	Recommendations
M. Luber, L. Wittmann, and J. Schilp (2026)	Configurable DT software infrastructure with ML and simulation	Interoperability, scalability, predictive capability	Limited validation in laboratory settings	Deploy and evaluate in large-scale industrial environments
Y. Yan et al., (2025)	DT applications in road engineering lifecycle	Improved monitoring and lifecycle management	Limited real-time interaction and lifecycle coverage	Develop standards and extend DT to all lifecycle phases
Z. Lei, J. Wan, and L. Fang (2025)	Factory Twin for smart workshops using open-source technologies	Real-time monitoring, online control, scalability	Limited evidence of large-scale industrial adoption	Investigate deployment in diverse manufacturing environments
M. A. Guinea-Cabrera and J. A. Holgado-Terriza (2024)	Systematic review of DT adoption in software industry	Comprehensive overview of DT applications in SDLC	Lack of unified frameworks and implementation methods	Develop standardized DT frameworks for software engineering
J. Liu et al., (2023)	DT-based Airfield Lighting Management System	Centralized monitoring and enhanced operational efficiency	Domain-specific implementation	Extend DT solutions to broader airport management systems
D. Sun (2023)	DT framework for civil airport safety management	Improved safety monitoring and emergency response	Limited real-world deployment and validation	Evaluate scalability and integration in operational airports

Research Gap: There are still some Research gaps in the field of Digital Twin. The primary research efforts are centered on domain-specific implementations and on virtual model development, whereas problems such as real-time synchronization, interoperability and integration across the model's lifecycle are under-researched. Furthermore, there are no common standards and only limited validation in large-scale industrial applications, which limits the spread of use. So, to make the most of Digital Twin technology in software development and engineering, future studies should focus on standardised techniques, improved physical-virtual interaction, scalable designs, and thorough assessments.

VII. CONCLUSION AND FUTURE WORK

Software engineering and development are seeing a surge in interest in Digital Twin (DT) technology as a means to model software systems virtually and provide continual interaction between the virtual and physical worlds. Understanding DTs in software engineering is essential, and this review covers the fundamentals as well as enabling technologies, applications, industry usage, and current research trends in the field. Results show that DTs are beneficial in many ways, such as improving lifecycle management, security, intelligence, decision-making, and real-time monitoring. Even though these benefits, there are

still open research questions, including interoperability, scalability, standardization, and real-time synchronization. In conclusion, the implementation of DT technology can revolutionize software engineering practices, offering the promise of more reliable, efficient, secure, and intelligent software systems.

Further studies are needed to create standard Digital Twin methods for software engineering, enhance interoperability and real-time synchronization methods, and introduce scalable architectures for large-scale systems. Furthermore, the use of Artificial Intelligence (AI) with DT for autonomous decision-making and adaptive software lifecycle management is still a promising path for future research

REFERENCES

[1] M. A. Akbar et al., "Improving the quality of software development process by introducing a new methodology-Az-model," *IEEE Access*, 2017, doi: 10.1109/ACCESS.2017.2787981.

[2] I. Jacobson, P. W. Ng, P. E. McMahon, I. Spence, and S. Lidman, "The essence of software engineering: The SEMAT kernel," *Queue*, 2012, doi: 10.1145/2381996.2389616.

[3] Y. Fu, G. Zhu, M. Zhu, and F. Xuan, "Digital Twin for Integration of Design-Manufacturing-Maintenance: An Overview," 2022. doi: 10.1186/s10033-022-00760-x.

- [4] M. Attaran and B. G. Celik, "Digital Twin: Benefits, use cases, challenges, and opportunities," *Decis. Anal. J.*, 2023, doi: 10.1016/j.dajour.2023.100165.
- [5] M. H. Iqbal, "A Review on the Role of IoT in Modern Electrical Engineering Education," *J. Eng. Comput. Intell. Rev.*, 2023.
- [6] A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," *IEEE Access*, 2020, doi: 10.1109/ACCESS.2020.2970143.
- [7] C. Bae, E. Choi, and S. Lee, "Technologies, Applications, and Challenges of Digital Twin Across Industries: A Systematic Review of the State-of-the-Art Literature," *IEEE Access*, 2025, doi: 10.1109/ACCESS.2025.3601615.
- [8] D. Jones *et al.*, "Towards integrated version control of virtual and physical artefacts in new product development: Inspirations from software engineering and the digital twin paradigm," in *Procedia CIRP*, 2021. doi: 10.1016/j.procir.2021.05.121.
- [9] A. K. R. Seella, "Deductive Modeling of User Satisfaction: A Synthetic-User Digital Twin Framework for Early Software Adoption Analysis," in *SoutheastCon 2026*, Huntsville, AL, USA: IEEE, 2026, pp. 1–6, April. doi: 10.1109/SoutheastCon63549.2026.11476440.
- [10] P. C. Shekhar, "From Traditional to Transformational: Leveraging Digital Twins for Advanced Testing in Life Insurance," *Int. Sci. J. Eng. Manag.*, vol. 2, no. 8, pp. 1–8, AUG, 2023, doi: 10.55041/ISJEM0419.
- [11] A. Barbie, W. Hasselbring, and M. Hansen, "Digital Twin Prototypes for Supporting Automated Integration Testing of Smart Farming Applications †," *Symmetry (Basel)*, 2024, doi: 10.3390/sym16020221.
- [12] R. S. Pressman, *Software Engineering A Practitioner's Approach 7th Ed - Roger S. Pressman*. 2009. doi: 10.1017/CBO9781107415324.004.
- [13] M. Chanda, "A Low-Cost System for Acquiring Login/Logout Data for On-Ground Racks of in-Flight Entertainment Systems," California State University, 2016.
- [14] O. E. Worlasi, "SOFTWARE ENGINEERING : An in-depth look into the fundamentals , process , and importance PROCESS , AND IMPORTANCE Written by: Osborn Edudzi Worlasi," no. January, 2023.
- [15] D. M. Botín-Sanabria, S. Mihaita, R. E. Peimbert-García, M. A. Ramírez-Moreno, R. A. Ramírez-Mendoza, and J. de J. Lozoya-Santos, "Digital Twin Technology Challenges and Applications: A Comprehensive Review," 2022. doi: 10.3390/rs14061335.
- [16] A. Hasan, Z. Hu, A. Haghshenas, A. Karlsen, S. Alaliyat, and U. Cali, "Aninteractive digital twin platform for offshore wind farms' development," in *Digital Twin Driven Intelligent Systems and Emerging Metaverse*, 2023. doi: 10.1007/978-981-99-0252-1_13.
- [17] A. Hasan and M. Aghaei, "Digital twin technology: fundamental aspects and advances," in *Digital Twin Technology for the Energy Sector: Fundamentals, Advances, Challenges, and Applications*, 2024. doi: 10.1016/B978-0-443-14070-9.00002-0.
- [18] R. K. Gadiraju, "Artificial Intelligence for Resource Optimization in Cloud Computing Environments," *J. Electr. Syst.*, vol. 20, no. 6s, pp. 3164–3174, Apr. 2024, doi: 10.52783/jes.9485.
- [19] V. K. Sharma, "Cloud Computing IoT: 5G Focused IoT with Cloud Solutions," *Int. J. AI, BigData, Comput. Manag. Stud.*, vol. 6, no. 3, 2025, doi: 10.63282/3050-9416.IJAIBDCMS-V6I3P103.
- [20] Z. Shu, J. Wan, D. Zhang, and D. Li, "Cloud-Integrated Cyber-Physical Systems for Complex Industrial Applications," *Mob. Networks Appl.*, 2016, doi: 10.1007/s11036-015-0664-6.
- [21] J. B. Mehta, "AN AUTONOMOUS DEPENDENCY AND FAILURE-PROPAGATION MODELING SYSTEM FOR SECURITY-CRITICAL DISTRIBUTED ENTERPRISE CLOUD COMPUTING ENVIRONMENTS," 202641001713, 2026
- [22] L. Hou, S. Wu, G. K. Zhang, Y. Tan, and X. Wang, "Literature review of digital twins applications in constructionworkforce safety," 2021. doi: 10.3390/app11010339.
- [23] N. Bhandarwar, "Integrating Generative AI and Model Context Protocol (MCP) with Applied Machine Learning for Advanced Agentic AI Systems," *Int. J. Comput. Trends Technol.*, vol. 73, no. 8, pp. 7–14, August, Aug. 2025, doi: 10.14445/22312803/IJCTT-V73I8P102.
- [24] A. K. R. Seella, "Data-Driven Prediction of Advertising Digital Campaign Effectiveness Using Artificial Intelligence," in *2026 14th International Symposium on Digital Forensics and Security (ISDFS)*, IEEE, Mar. 2026, pp. 1–6. doi: 10.1109/ISDFS69419.2026.11459096.
- [25] M. A. Guinea-Cabrera and J. A. Holgado-Terriza, "Digital Twins in Software Engineering—A Systematic Literature Review and Vision," 2024. doi: 10.3390/app14030977.
- [26] A. M. Madni, C. C. Madni, and S. D. Lucero, "Leveraging digital twin technology in model-based systems engineering," *Systems*, 2019, doi: 10.3390/systems7010007.
- [27] C. Lehner, A. Padovano, C. Zehetner, and G. Hackenberg, "Digital twin and digital thread within the product lifecycle management," in *Procedia Computer Science*, 2024. doi: 10.1016/j.procs.2024.02.104.
- [28] M. Luber, L. Wittmann, and J. Schilp, "Software infrastructure for ready-to-use, data analytics-based Digital Twins utilising the Asset Administration Shell," *Procedia Comput. Sci.*, vol. 277, pp. 3113–3122, 2026, doi: 10.1016/j.procs.2026.02.347.
- [29] Y. Yan, L. Ni, L. Sun, Y. Wang, and J. Zhou, "Digital Twin Enabling Technologies for Advancing Road Engineering and Lifecycle Applications," 2025. doi: 10.1016/j.eng.2024.12.017.
- [30] Z. Lei, J. Wan, and L. Fang, "FactoryTwin: Dual-Mode Scalable Digital Twin Software for Smart Workshop," in *2025 International Wireless Communications and Mobile Computing (IWCMC)*, 2025, pp. 467–472. doi: 10.1109/IWCMC65282.2025.11059458.
- [31] J. Liu, M. Guan, H. Yang, W. Chen, H. Niu, and Q. Fu, "Airfield Lighting Management System Development Based on Digital Twin Technology," in *2023 IEEE 5th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, 2023, pp. 533–539. doi: 10.1109/ICCASIT58768.2023.10351572.
- [32] D. Sun, "Research on the construction of civil airport safety management system based on digital twin technology," *China Saf. Sci. J.*, 2023, doi: 10.16265/j.cnki.issn1003-3033.2023.S1.0257.