*Sreedhar Reddy Konda. International Journal of Engineering Research and Applications*
*www.ijera.com*
*ISSN: 2248-9622, Vol. 14, Issue 1, January, 2024, pp 50-57*

RESEARCH ARTICLE OPEN ACCESS

# Advancements in Self-Healing Technology for Software Systems

Sreedhar Reddy Konda
*Software Engineer, Information Technology*
*Burns and McDonnell, 13737 Noel Rd Suite 700, Dallas, TX 75240*

**ABSTRACT**
This abstract reports on an intensive study of self-healing technology in software engineering as a way to shift paradigms toward the creation of self-sustaining, autonomous software systems. The main goal is to look into the fundamental concepts, approaches, and practical realizations of self-healing mechanisms, showing their relevance to improving the reliability and efficiency of systems. The presented paper considers the theoretical basis of self-healing technology and discusses the most important algorithms and models used to detect, diagnose, and correct faults in a system [19]. It further delves into different case studies proving that self-healing technology works not just in one domain but in different ones, such as healthcare, engineering, cloud computing, telecommunications, and cybersecurity [14].
Furthermore, this paper marks the challenges and limitations in this sphere, including complexity, scalability, and resource limits, and examines potential future developments, especially in the area of artificial intelligence and machine learning [9]. Overall, self-healing technology contributes to the resilience of software systems, minimizing downtime and maintenance efforts for higher performance. The paper concludes with thoughts on the future path of self-healing technology, marking it as an increasingly pressing concern in the developing panorama of software engineering [4]. Self-healing systems will open a new door for more intelligent, adaptive software solutions and improve software maintenance and reliability, as this research demonstrates.
**Keywords-**Self-Healing Technology, Software Engineering, Autonomous Systems, Fault Tolerance, Machine Learning Algorithms, Artificial Intelligence

---

Date of Submission: 08-01-2024 Date of acceptance: 20-01-2024

---

## I. INTRODUCTION

In software engineering, the concept of self-healing technology means that a software system recognizes that it is not working optimally and, without the interference of a human, rectifies itself to operate normally again. This idea stems from the biological model, where living beings repair themselves as a reaction to wounds. This involves finding, diagnosing, and fixing software issues [5]. Self-healing technology prevents downtime and eliminates human control to improve software system reliability, efficiency, and sustainability.

Since the beginning of computing, software self-healing has evolved. Software systems started as simple and manual. As systems became more complicated and big, automated self-repairing processes were needed. Self-healing software advances fault-tolerant computing, AI, and machine learning [4]. The idea of self-healing started emerging more clearly at the end of the 20th century under the influence of the growing complications of network systems and the rising costs of software maintenance [5]. Moreover, pioneering work in this area sought to grant software systems the ability to self-diagnose and recover from failures as an organism's biological immune system responds to threats to its health.

In modern software engineering, self-healing technology is ofparamount importance, given that software systems have become more and more important and critical in such domains as healthcare, finance, and telecommunications. So as such systems become a major part of life, the cost and effect of software failure start growing. A self-healing ability not only enhances the trustworthiness of a system but also contributes significantly to the security and resilience of the network in which a system is operationalized. Moreover, at a moment when software systems have to work round the clock

and deal with huge amounts of data, there is a need for self-regenerating capabilities that would guarantee the stability of operations and user satisfaction.

Advancements in technology and a fast dependence on large software systems have positioned self-healing technology as something quite popular in contemporary software engineering. This technology introduces a revolutionary change from traditional, reactive maintenance to a proactive, resilient type of maintenance, which comes with enormous advantages as far as reduced costs of maintenance, increased system uptime, and integrated user experience are concerned. Additionally, as autonomous systems get more complex, software design and architecture must incorporate self-healing methods. It helps systems adapt, evolve, and stay working in unforeseen situations, which is crucial in the ever-changing digital environment.

## II. LITERATURE REVIEW

### 1. Fundamentals of Self-Healing Software Systems

Self-healing Software design relies on software systems that can detect, diagnose, and correct their own faults [19]. It analyzes self-healing in cyber-physical systems using machine learning and discuss this unique method, which reveals how advanced algorithms affect software system evolution [13]. Due to monitoring and fixing errors by system, it enhances the automatic self-healingMachine learning is highlighted as a major pattern in this framework in their extensive examination [13]. Software anomaly detection and correction is a logical fit for machine learning algorithms because of their inherent data-learning and reliability-enhancing capabilities. These algorithms anticipate potential failures so they can come up with mechanisms to address them, thus increasing the reliability and effectiveness of systems. These algorithms are not only used in error correction but also in the adaptation of the software to changing operational environments to maintain a constant level of system performance.

Further, the incorporation of self-healing mechanisms in cyber-physical systems, as discussed in the study, reveals the usefulness of these technologies in the wider scope of interrelated and interdependent digital and physical systems [14]. In such environments, the self-healing capacity becomes even more critical because that directly determines the system's capability to continue constant and reliable operation. This is especially important in situations where manual interference is impractical or impossible.

Hence, the fundamental principles of self-healing software systems are based on the development of very smart, knowledgeable algorithms that not only detect and fix faults but also learn from these events to avoid their later repetitions. This aspect of progress and learning is what distinguishes modern self-healing systems from their more primitive predecessors.

### 2. Historical Evolution of Self-Healing Mechanisms

The evolution of self-healing mechanisms in software engineering is a story that has witnessed steady adjustment and innovation over time. The study offer an in-depth analysis of self-engineering systems from the beginning of time, explaining the process of developing self-healing technologies [3]. Initially, software systems were managed manually, a process that was effective because of their simplicity and smaller scale. However, with the increasing pace of technology and the continuously increasing level of sophistication of software systems, the manual method for self-healing became less viable, evoking the demand for automated self-healing solutions.

The transition to automation in the sphere of software engineering was not sudden but evolutionist in nature. Early work on what would come to be the domain of fault tolerance was mainly about building fault-tolerant systems, or systems that could continue to operate in the face of certain types of errors. These systems were developed to predict and cope with certain known types of faults, but their capability was constrained by the lack of more sophisticated predictive and adaptive functions. Artificial intelligence and machine learning gave software systems new flexibility and adaptability [3].

In particular, self-healing has grown. Error management has moved from static rules to dynamic, learning-driven self-healing systems. Modern self-healing systems may learn from mistakes, adapt to environmental changes, and learn

from the past thanks to more advanced algorithms. This breakthrough made software more reliable and efficient, making it smarter and more independent.

## 3. Fault Tolerance Strategies in Software Systems

Self-healing requires software fault tolerance. It enables normal operation of the system even with a failed component. The study examines methods of machine learning in the content of defect tolerance under the cyber-physical structures [13]. Therefore those precautions make the software more reliable mostly in the cases where errors can be more adverse.

Redundancy is accentuated in the research; several components perform similar functions. When one fails, the system could still depend on others. This approach is applicable to distributed systems, where the collapse of one node does not render the network inoperable. Restoring to the previous saved state minimizes data loss and latency after a failure.

Predictive analytics is another vital line of tactics in which machine learning plays a critical role. By looking at trends and other changes in the operational data of this system, machine learning algorithms can predict failure sites before they are widespread. This predictive ability allows for proactive maintenance and early interventions to help minimize the effects of failures.

In addition, research also emphasizes how critical systems of self-definition and self-repair are [13]. There are numerous examples, such as when the system can autonomously detect a fault, determine its source, and fix it. As fault tolerance transitions from a concept of persevering through failures into proactive schemes for anticipating and dealing with potential issues, the fact that these methods would be incorporated in software systems is an enormous leap forward as far as this domain goes.

The issue of redundancy is discussed in visual IoT mashups [7]. They illustrate how redundancy in equipment leads to better software maintenance. Redundancy designs duplicate critical functions or components of a system so that they can serve as backups in case the main component fails. This method is good for use in complex situations, such as Internet of Things applications, where system reliability matters a lot [1]. Redundancy can defend software against localized errors and provide operational integrity. Although modern software systems use redundancy rather pragmatically, its significance is emphasized in self-healing and robust structures of software programs [6].

## 4. Automatic Error Detection and Correction Techniques

Self-healing software systems have the capability to automatically identify and correct errors by themselves; that's why they're so significant. The study evaluates the state of the art when it comes to methodology, and they identify what needs improvement [10]. Other writers have explored an assortment of state-of-the-art algorithms and strategies for automated management of errors in software systems.

One of the main approaches considered is using anomaly detection techniques [10]. For these algorithms to work there should be a baseline for what defines typical system behavior. This baseline will be automatically reviewed by the system for any deviations, as these might indicate potential errors or issues. This response usually includes a thorough breakdown of the abnormality and its cause in order to identify it.

They also paid attention to practices for correcting errors, and such methods became far better with the introduction of artificial intelligence and machine learning. These methods use historical data and trends to be able to predict but also make corrections. For instance, machine learning algorithms can outline the habits of repeated error patterns and try to avoid such mistakes in the future. That is why such a proactive approach to the error's elimination is crucial for improving its survivability and reducing downtime.

Advocate for continuous learning error detection and repair systems is also considered [10]. Adaptability is working in environments where the software environment is changing all the time and new kinds of errors may arise. In general, the research conducted pertains to what we wish regarding automated error correction and detection [10]. The analysis reveals that these systems are always learning and trying to reinvent themselves every time the problems in software engineering evolve.

## III. ARCHITECTURE

Modern software design makes self-healing an indispensable concept, and the architecture of a set of linked programs plays a vital role in facilitating this feature. For instance, the research emphasizes the need for complex design considerations necessary to include self-healing features in software architecture successfully.

In self-healing software architectures, the principle of fundamentality relates to modularity and decentralization. Several modules are developed in a set of systems that can operate together or independently. This modular approach implies that the failure of one element does not knock out the whole system. There are self-diagnostic tools present in each module that allow it to detect anomalies for autonomous repair.

Monitoring and decision-making tools are identified as feature of these architectures based on their study. These systems endlessly monitor their state of operation and the environment around them. The choice-making process is based on artificial intelligence and machine learning algorithms such that it becomes adept of managing information from previous occurrences so as to come up with better self-healing reactions over time.

Likewise, the study highlights the significance of scalability in self-healing software architectures. The architecture must also be flexible enough to accommodate components and functionalities as systems grow bigger or become more complex, while at the same time not compromising its self-healing capability. This scalability is attained to in a way that it involves the use of flexible design patterns that enable the incorporation of new modules as well as necessary adjustments to current ones.

## IV. PERFORMANCE METRICS

It is very important to evaluate the self-healing software, even though it is very challenging in various ways. This is because evaluation helps understand whether the system is reliable and effective. The research investigated approaches to self-healing and resilience, specifically in power systems [22]. In regards to their research, it is noted that the performance indicators contribute to the ability to determine the effectiveness of how the system can detect, respond to, and resume following faults (disruptions).

Additionally, the Mean Time to Failure (MTTF) is mentioned by these authors where they indicated it as one of the major metrics and it means the average operational period required for a system before failure begins. This metric echoes the dependability of the system and is very noteworthy in situations where performance steadiness is needed. Besides, Mean Time to Repair (MTTR) is another key metric, which means the average time that it takes the system to recover from a failure. A decreased MTTR is a positive indicator, meaning the system can recover normal operations in less time.

It is crucial to determine the Recovery Point Objective (RPO) and Recovery Time Objective (RTO). The RTO endeavors to determine the appropriate duration for recovering and restoring a business process following a catastrophic event that halts desired results [20]. The Recovery Point Objective (RPO) measures the maximum acceptable level of data loss during a business incident. These measurements assist in assessing the resilience of self-healing infrastructure systems during times of adversity.

System availability is another part of the study that goes into more detail. This number measures how often a system stays up and running and can be accessed. It's important to have self-healing software because problems can happen when it's not working. The ability of a system to stay available is very important for self-healing software to work. The success rate is another important metric to think about. It shows how many times the system switched to a backup system after a loss. This measure tells us how reliable the safety features are that are being used.

## V. SCALABILITY AND EFFICIENCY

It is important to think about how scalable and successful self-healing mechanisms are in software systems when determining if they will work in different settings. The study sheds light on this topic, mainly by looking at self-healing polymers and devices that are used in harsh situations [8]. Their main focus is on materials, but the ideas and problems they talk about are similar to those in self-healing software systems, especially when it comes to speed and scalability [12].

A scalable self-healing system would indicate the ability of such a system to maintain its functionality and performance as it increases in size. When it comes to software systems, this means that the system should be able to perform efficiently with tasks, users, or data volumes. As indicated, one of the scalability challenges is that self-healing mechanisms can unnecessarily become complicated or resource-intensive as a system grows and expands in size [8]. This is very important because such complicated self-healing processes may significantly degrade efficiency and reaction time, thus contrasting with the advantages of possessing certain self-repair features in systems.

Efficiency in self-healing actions is related to the effectiveness of these actions and their low impact on system resources. Effective software self-healing involves fast problem identification and correction utilizing minimal resources without affecting system performance. In scarce or expensive situations, it is found that fault recovery and resource conservation must be balanced [8].

Their research also suggests self-adaptive mending mechanisms that can solve problems dependent on size and operation. This adaptability is crucial for self-healing to work on a large scale and as the system grows and changes environments. A robust self-healing system succeeds when it maintains efficiency and performance at all scales.

## VI. SECURITY IMPLICATIONS

It is important to think about how scalable and successful self-healing mechanisms are in software systems when determining if they will work in different settings. The study sheds light on this topic, mainly by looking at self-healing polymers and devices that are used in harsh situations [8]. Their main focus is on materials, but the ideas and problems they talk about are similar to those in self-healing software systems, especially when it comes to speed and scalability [12].

A scalable self-healing system would indicate the ability of such a system to maintain its functionality and performance as it increases in size. When it comes to software systems, this means that the system should be able to perform efficiently with tasks, users, or data volumes. As indicated, one of the scalability challenges is that self-healing mechanisms can unnecessarily become complicated or resource-intensive as a system grows and expands in size [8]. This is very important because such complicated self-healing processes may significantly degrade efficiency and reaction time, thus contrasting with the advantages of possessing certain self-repair features in systems.

Efficiency in self-healing actions is related to the effectiveness of these actions and their low impact on system resources. Effective software self-healing involves fast problem identification and correction utilizing minimal resources without affecting system performance. In scarce or expensive situations, it is found that fault recovery and resource conservation must be balanced [8].

Their research also suggests self-adaptive mending mechanisms that can solve problems dependent on size and operation. This adaptability is crucial for self-healing to work on a large scale and as the system grows and changes environments. A robust self-healing system succeeds when it maintains efficiency and performance at all scales.

## VII. CROSS-ISCIPLINARYAPPROACH

Interdisciplinary methodologies from systems engineering, biology, and artificial intelligence have helped develop software engineering's self-healing systems. The research on intelligent self-healing materials in autonomous robotics helped explain how these diverse aspects would affect self-healing technology development [2]. Software self-repair algorithms use biological healing mechanisms like how living things mend and regenerate [24]. Software systems that discover and fix problems autonomously can be modeled after these biological processes.

Artificial intelligence relies on machine learning techniques to improve self-healing software. These algorithms allow systems to learn from their failures and alter their recovery mechanisms, like living things do to new threats [20]. Artificial intelligence-powered self-healing systems increase performance and efficiency by learning from failures and adapting to new conditions.

Systems engineering is essential for self-healing software development and deployment. These approaches help develop fault-tolerant, redundant, and flexible software frameworks for self-healing systems. Systems engineering allows software engineers to build sophisticated systems

with many moving elements that can continue to operate after failure.

Interfacing these disciplines improves self-healing systems' utility value and application breadth. The study found that material science, biology, AI, and engineering are combining to create extremely sophisticated and lasting autonomous robotics systems [24]. A cross-disciplinary strategy is needed to advance self-healing technologies, which will lead to smarter, more adaptive, and longer-lasting software systems.

## VIII. THEORETICAL BACKGROUND

Self-healing software systems are based on the integration of ideas from computer science, system theory, artificial intelligence, and the fundamental foundation of technology as a basis. As a whole, self-healing technology is based on the idea that software systems are allowed to self-act and self-recognize errors and to diagnose and correct them without human intervention, which leads the systems to become much more reliable and efficient. This is based on a concept borrowed from biological systems, where self-repair is crucial for survival and adaptation.

The algorithmic and modeling bases of self-healing software systems are the algorithms and models that detect and eliminate faults automatically. One of the major algorithms is anomaly detection. This baseline serves as the normal, and any variation from it is anomalous, indicating a problem or failure. Diagnostic algorithms are employed to establish the cause of an abnormal occurrence. These algorithms include rule-based systems and advanced machine-learning models that use historical data to enhance the accuracy of disease identification.

Remedial algorithms are needed for self-healing systems. These algorithms calculate the best solution. In smaller systems, rebooting a failed service or component may solve it. Remedial solutions for intricate systems may demand code adjustments in real time, re-routing network traffic, or the dynamic allocation of resources.

Many self-healing systems use prediction models. They use past data to predict failures and to prevent large problems by taking preventive measures [11]. It is also valuable for large, critical systems, because a small failure of such a system can result in a disaster.

## IX. CASE STUDIES

There are numerous cases where self-healing technology was effectively applied to software engineering so that the systems became more reliable and effective. One notable use is cloud computing. Firms like Amazon and Google have established self-healing protocols for their cloud services [16]. These systems automatically identify and resolve issues related to server overloading or network failures so that downtime is low and services are available at all times for users.

Another important sphere of application is telecommunication. Self-healing technology is the means through which network providers manage and maintain such large networks [15]. If there is failure or disruption within a network node, the system will automatically direct the traffic to pass through other network nodes for uninterrupted service continuity. It increases the stability of the network and the user's comfort by eliminating disturbances.

Advanced driver support systems and autonomous vehicles increasingly rely on self-healing technology. These systems monitor vehicle performance using sensors, algorithms, and data analytics [18]. It automatically repairs sensor failures and software faults, thus increasing the safety and reliability of the vehicle.

Complicated IT infrastructures are being handled by self-healing enterprise software systems. Automating the process of problem diagnosis and remediation can help save a considerable amount of IT maintenance time and effort for businesses [17]. It increases operational efficiency and helps the IT department concentrate on strategic activities.

Self-healing technology is also making a lot of progress in the area of cybersecurity [21]. Lastly, self-healing systems can find holes in security and stop threats on their own. In this day and age of complex and widespread cyber dangers, standard operating procedures for cyber security are very important [14].

## X. CHALLENGES&LIMITATIONS

Self-healing technology has gotten better, but there are still some problems. It is also important to think about how hard it is to build and use these tools. It's getting harder and harder to make self-healing algorithms that work well in software settings that are getting more complicated [9]. Self-

healing is harder to do and costs more because it takes longer to grow and is more complicated.

Scalability is yet another important limit. As system numbers and levels of complexity rise, it becomes more difficult to make self-healing processes work in bigger and more complicated ones without losing their efficiency or effectiveness. Large-scale self-healing systems that are more ambitious might be a little harder to pull off and need more than just infrastructure.

Resources cause more problems. For example, this could be a problem when resources are limited or when systems are meant to make the best use of those resources. Another big problem with using this technology more often is that because of limited resources, the current self-healing abilities aren't very good.

## XI. FUTURE DIRECTIONS

With AI and ML developing rapidly in this day and age, they are considered to significantly integrate with the healing properties of software engineering. One area of improvement for future growth is AI-based predictive models. These types of models are capable of performing focused healing activities in advance so as to accurately predict system breakdown. Researchers are currently also looking to enhance self-healing systems with deeper learning to enhance adaptability to new situations and operational demands.

Improving the utilization of AI resources for the self-healing process is another promising direction. Such resource systems that, if run well, may be self-healing and consume fewer resources are good for working in environments that have a limited amount of resources. Or we may see what type of self-healing software may operate with IoT and edge computing [7]. In such instances, self-healing procedures hold interconnected, dispersed, reliable, and resilient systems.

## XII. CONCLUSION

Self-healing in software engineering improves resilience, self-healing, and the autonomy of adaptation. Studies have shown that self-healing mechanisms increase the reliability of a system, decrease maintenance costs, and function better in this industry. This is why AI and ML algorithms assist computers in automatically finding, predicting, and correcting faults. Future software systems and self-healing methods appear to be very promising. They should ensure the reliability and long-term viability of the software infrastructure because these systems involve the addition of new technology and the execution of more complicated tasks. That is why self-healing technology is not just a fashion phenomenon; it is becoming an evolutionary necessity in modern software systems as they are facing more and more demands. Overall, self-healing technology will be necessary for systems to become more resilient and independent.

## REFERENCES

[1]. B. Abdulrazak, J. Codjo and S. Paul, "Self-Healing approach for IoT Architecture: AMI Platform. In International Conference on Smart Homes and Health Telematics," pp. 3-17, 2022, June.

[2]. M. Ammar, A. Haleem, M. Javaid, S. Bahl and A. S. Verma, "Implementing Industry 4.0 technologies in self-healing materials and digitally managing the quality of manufacturing. Materials Today," vol. 52, pp. 2285-2294, 2022.

[3]. S. Brooks and R. Roy, "An overview of self-engineering systems. Journal of Engineering Design," vol. 32(8), pp. 397-447, 2021.

[4]. G. Chen, W. Tang, S. Chen, S. Wang and H. Cui, "Prediction of self-healing of engineered cementitious composite using machine learning approaches. Applied Sciences," vol. 12(7), p. 3605, 2022.

[5]. P. Dehraj and A. Sharma, "A review on architecture and models for autonomic software systems," The Journal of Supercomputing, vol. 77, pp. 388-417, 2021.

[6]. J. P. Dias, B. Lima, J. P. Faria, A. Restivo and H. S. & Ferreira, "Visual self-healing modelling for reliable internet-of-things systems.," In International Conference on Computational Science, pp. 357-370, 2020, June.

[7]. J. P. Dias, A. Restivo and H. S. & Ferreira, "Empowering visual Internet-of-Things mashups with self-healing capabilities. In 2021 IEEE/ACM 3rd International Workshop on Software Engineering Research and Practices for the IoT," pp. 44-51, 2021, June.

[8]. J. Ekeocha, C. Ellingford, M. Pan, A. M. Wemyss, C. Bowen and C. & Wan, "Challenges and opportunities of self- healing polymers and devices for extreme and hostile environments. Advanced Materials,," vol. 2008052, 2021.

[9]. C. A. Fernandez, M. Correa, M. T. Nguyen, K. A. Rod, G. L. Dai, L. Cosimbescu and V. A. Glezakou, "Progress and challenges in self-healing cementitious materials. Journal of Materials Science," vol. 56, pp. 201-230, 2021.

[10]. S. Ghahremani and H. Giese, "Evaluation of self-healing systems: An analysis of the state-of-the-art and required improvements," vol. 16, 2020.

[11]. X. Huang, M. Wasouf, J. Sresakoolchai and &. Kaewunruen, "Prediction of healing performance of autogenous healing concrete using machine learning," vol. 4068, 2021.

[12]. C. I. Idumah, "Recent advancements in self-healing polymers, polymer blends, and nanocomposites," Polymers and Polymer Composites, vol. 29(4), pp. 246-258, 2021.

[13]. O. Johnphill, A. S. Sadiq, F. Al-Obeidat, H. Al-Khateeb, M. A. Taheir, O. Kaiwartya and M. & Ali, "Self-Healing in Cyber–Physical Systems Using Machine Learning: A Critical Analysis of Theories and Tools.," Future Internet, vol. 15(7), p. 244, 2023.

[14]. S. S. Khairullah and C. R. & Elks, "Self- repairing hardware architecture for safety- critical cyber- physical- systems.," IET Cyber- Physical Systems: Theory & Applications, vol. 5(1), pp. 92-99, 2020.

[15]. D. S. Kwon and J. H. & Na, "Research status on machine learning for self-healing of mobile communication Network.," Electronics and Telecommunications Trends, vol. 35(5), pp. 30-42, 2020.

[16]. J. Lynch and D. A. & Joshi, "Towards Practical Self-Healing Distributed Databases.," In 2020 IEEE Infrastructure Conference, pp. 1-4, 2020, October.

[17]. L. McMillan and & Varga, "Towards self-healing in water infrastructure systems.," Proceedings of the Institution of Civil Engineers-Smart Infrastructure and Construction, pp. 1-9, 2022.

[18]. P. K. Murali, M. Kaboli and R. & Dahiya, "Intelligent In- Vehicle Interaction Technologies," Advanced Intelligent Systems, vol. 4(2), 2022.

[19]. J. Nikolić, N. Jubatyrov and E. & Pournaras, "Self-healing dilemmas in distributed systems: Fault correction vs. fault tolerance," IEEE Transactions on Network and Service Management, vol. 18(3), pp. 2728-2741, 2021.

[20]. P. K. Rajput and G. & Sikka, "Multi-agent architecture for fault recovery in self-healing systems.," Journal of Ambient Intelligence and Humanized Computing, vol. 12, pp. 2849-2866, 2021.

[21]. D. Sarathkumar, M. Srinivasan, A. A. Stonier, R. Samikannu, N. R. Dasari and R. A. & Raj.

[22]. E. Shittu, A. Tibrewala and S. W. X. Kalla, "Meta-analysis of the strategies for self-healing and resilience in power systems," Advances in Applied Energy, vol. 100036. , 2021.

[23]. M. Stefanidou, E. Tsampali, G. Karagiannis, S. Amanatiadis, A. Ioakim and S. Kassavetis, "Techniques for recording self- healing efficiency and characterizing the healing products in cementitious materials.," Material Design & Processing Communications, vol. 3(3), 2021.

[24]. Y. J. Tan, G. J. Susanto, A. Ali, H. P and B. C. Tee, "Progress and Roadmap for Intelligent Self- Healing Materials in Autonomous Robotics," Advanced Materials, vol. 2002800.

[25]. A. Tarinejad, H. Izadkhah, M. M. Ardakani and K. Mirzaie, "Metrics for assessing reliability of self-healing software systems," Computers & Electrical Engineering, vol. 106952., p. 90, 2021.