

# AI-Driven Self-Healing Cloud Systems: Enhancing Reliability and Reducing Downtime through Event-Driven Automation

Rajeev Kumar Arora<sup>1</sup>, Anoop Kumar<sup>2</sup>, Arpita Soni<sup>3</sup>, Aniruddh Tiwari<sup>4</sup>

MIAMI, FL, USA<sup>1</sup>

San Francisco, CA, USA<sup>2</sup>

Concord, NC, USA<sup>3</sup>

Chantilly, VA, USA<sup>4</sup>

Corresponding author: Rajeev Kumar Arora, Email: rajeev04.study@gmail.com

The goal of this study is to create and carry out a self-healing cloud system by combining an event-driven automation framework depending on the if-this-then-that principle for managing incidents and recovery. A recovery engine with Artificial Intelligence (AI)-based decision-making approaches is presented—which chooses the best remedial actions from a pre-established catalogue in order to maximise system reliability and minimise downtime. The system is tested on an OpenStack-based video on demand service—where multiple issues are replicated in order to assess the efficaciousness of various recovery actions and workflows. The decision-making module of the recovery engine examines data from these experiments to determine the most effective remedial actions, taking into account their impact on the quality of service and other factors. The recovery engine is only meant to need human input when it comes to parameterizing and optimising decision models at particular points in time. In order to show how these AI-driven decision-making techniques can enhance mean time to repair and overall service quality in cloud environments—the study presents and assesses their results. This novel strategy represents a change towards cloud systems that are more sturdy autonomous, and able to effectively manage anomalies and recover from failure.

**Keywords:** Artificial Intelligence, Cloud System, Engine, OpenStack.

## **1 Introduction**

Using an OpenStack video streaming service as the testbed—this study examines the creation and application of an event-driven automation system coupled with a decision-making mechanism at the community cloud. The execution and decision-making components of the system's design are the main emphasis of the research. This study is driven by the quickly changing demands on cloud systems, which are becoming more difficult to manually operate because of the number of users and complexity of duties involved. In the past, system dependability and Service Level Agreements (SLAs) were the responsibilities of IT operations. But given the size of modern cloud operations, this is a challenging task. Because of this change, cloud service providers are now using automation to handle processes like orchestration, scaling, and service provisioning. The objective is to meet Service Level Agreements (SLAs), improve customer satisfaction, and lessen the need for ongoing human supervision, which is often ineffective and prone to mistakes.

Therefore, this study presents a system in which the experiment controller interacts with Stackstorm and OpenStack to manage self-recovery operations while simulating anomalies. In order to maximise recovery actions, this system creates Artificial Intelligence (AI) [1–8] based decision models and logs results. The careful selection of automated remedial procedures is necessary to preserve the overall stability and performance of the system, emphasising the critical balance that must be maintained between automation and system integrity. Putting in place an event-driven automation system and a decision-making process that chooses the best recovery sequence with the least amount of overhead and disturbance is the major goal. The decision-making system assesses several recovery workflows according to success rate, Mean Time To Repair (MTTR), and Quality of Service (QoS). With this method, the system is able to determine which recovery workflow is most suitable for executing according to the particular circumstances. The study delves deeper into the process of making the decision models further applicable. It studies approaches where human interaction is minimised and takes into account instances where human specialists can parameterize and prioritise decision inputs. Because of its dual approach, the system is strong and flexible in addressing a variety of operational conditions and anomalous circumstances. This study advances the domains of automated systems management and cloud computing by highlighting how integrated decision-making frameworks might improve the effectiveness and responsiveness of cloud services. The study presents a model for future advancements in cloud system administration, where automation and wise decision-making are essential for preserving system health and assuring service continuity. This model is carefully designed and implemented.

The study is as follows; similar papers are shown in the following section. The system's implementation is described in Section 3. The experimental analysis is carried out in Section 4. The outcome analysis is given in Section 5, and we wrap up the investigation with some conclusions and ideas for future work in Section 6.

## **2 Related Works**

[23] The rapidly developing subject of AI for IT operations, or AIOps, combines AI and Machine Learning (ML) with IT operations to optimise and automate key processes related to resource management. [22] An increasing number of Cloud Service Providers (CSPs) and tech giants such as Microsoft and Baidu are leading the charge in developing comprehensive AIOps solutions—as cloud computing continues to dominate the IT landscape. With its Azure-based AIOps solution, Microsoft has developed a complete strategy that addresses all four essential areas—execution, analysis, decision-making, and monitoring. With minimal downtime and better system response, our all-encompassing approach guarantees a smooth and continuous IT operation flow. Likewise, the features that Baidu'sAIOps solution concentrates on include anomaly detection, traffic planning, forecasting patterns, and root cause analysis. Additional frameworks and techniques created to improve fault tolerance in cloud systems are explored in the literature. Fault-tolerance techniques are thoroughly reviewed by [9] who emphasise that cloud systems must be able to adapt and learn from errors.

Reactive, proactive, and resilient fault detection and recovery techniques are the three categories into which their research divides methods. Reactive techniques, such as restarting the system, are often used; however, the most important way to improve fault tolerance is through resilient techniques that anticipate errors and communicate with the cloud system in an intelligent manner. [10] who were among the first to recognise the significance of self-diagnosis and self-healing in cloud systems. [11] has made several significant contributions to the subject—their suggested hybrid tool efficiently manages and recovers from system anomalies by combining multivariate decision-making with the Naive Bayes classifier. In the study on the effects of virtualization in cloud systems, [12] highlighted how important it is for facilitating efficient load balancing. To assure data availability and integrity, their research uses data replication strategies. [13] added to the AIOps landscape with the proposal of a self-healing framework that dynamically assesses cloud service performance and creates recovery plans accordingly. In order to minimise interruption, this preventive approach focuses on implementing recovery plans right away or preserving them for later use. A self-healing framework that dynamically assesses cloud service performance and creates recovery plans in accordance with the findings of [14] further improved the AIOps landscape. Using recovery plans either immediately or by storing them for later use is the main goal of this preventive strategy to guarantee the least amount of disturbance. [15] presented a framework designed specifically for cloud-based web applications. With the use of this framework, users should be able to identify workload and performance anomalies and report them to CSPs so that appropriate modifications can be made. The contributions to AIOps from academia and industry are numerous and diverse. For instance, [16] proposed an autonomic cloud resource management technique—places a strong emphasis on reliable and reasonably priced cloud services. By identifying defects and implementing the necessary corrective measures, it prioritises self-healing. It also continuously assesses service quality in accordance with predetermined SLA parameters. Additionally, AIOps' practical use is demonstrated by solutions like Kubernetes and Winston from Netflix. Winston is an automation platform that operates based on events and allows for automated issue diagnosis and correction. The container self-healing feature is leveraged by Kubernetes, which is well-known for managing clusters of containers, to preserve system health without compromising the overall state.

### **3 Pre-Experimental Analysis**

The main goal of this study is to use OpenStac to integrate an advanced decision-making framework into an event-driven automation system as shown in Figure 1. The main emphasis is on the decision-making and execution components of the system architecture, specifically focusing on the regions. Figure 2 shows the experiment controller—responsible for organising the experimental setup, controlling events related to the system state, keeping track of recovery actions, and serving as the central component of the recovery engine. This engine generates and runs AI-based decision models, logs results, simulates anomalies, and determines how effective these actions were in the end. The recovery engine operates under the presumption that it is informed accurately about the state of the system—whether it is normal or abnormal—and that it is mindful of any anomalies that may be occurring. The self-healing pipeline's data evaluation and tracking sections are the source of this information. The deployment of Virtual Network Functions (VNFs) and edge services on OpenStack is not covered in the study—but it is stated to provide context for the kinds of services that could be implemented on the cloud infrastructure. The idea of closed loop automation, which is essential to the automation of cloud services, orchestration, and IT operations, is covered in great detail throughout the study. From fundamental case-specific automation scripts to all-inclusive technologies that automate several software delivery lifecycle stages, this idea has developed over time. One cycle of monitoring, analysing, and planning responses to operational warnings is what defines closed-loop automation. By reducing the requirement for human involvement, this cycle improves process efficiency and lowers mistake rates. A closed-loop solution in this networking setting combines orchestration, analysis, and monitoring to produce a smooth operational flow. Operational warnings are gathered and monitored by the system, which then analyses them to classify or retrieve pertinent data. Based on the analysis,

the system determines the best course of action for auto-remediation or self-recovery and implements the recovery plan.

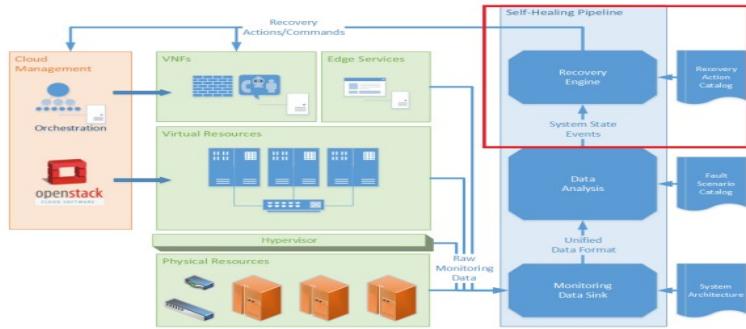


Figure 1. Project scope

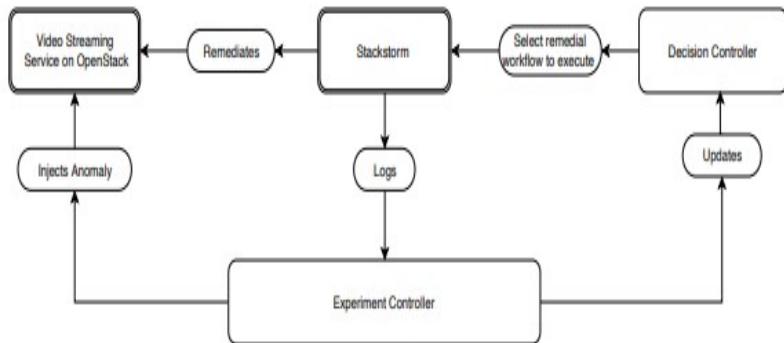


Figure 2. Component flow diagram

Now being modified for network and cloud-based contexts—this concept is consistent with past uses of closed-loop control systems in a variety of engineering disciplines. The study also examines multi-criteria decision making, also known as Multi-Criteria Decision Analysis, or Multi-Criteria Decision Making (MCDA/MCDM)—which is a technique for choosing the optimal recovery workflow from a range of options. System statistics, past performance information, or values that have been assigned by hand could be some examples of these criteria [17–23]. Many MCDM models, such as the Weighted Sum Model (WSM), TOPSIS, and an entropy-weighted model, are used in the system's decision-making process. By multiplying scores corresponding to several criteria by their associated weights, the WSM technique uses an easy additive weighting scheme. Furthermore, to minimise subjective bias in decision-making, the entropy-weighted model—which has its roots in information theory—is utilised to objectively determine criterion weights. This approach is especially helpful in situations where decision-makers cannot agree on how to weigh the criteria. The study contributes to the theoretical knowledge of automated system recovery and offers practical insights into its use in a real-world cloud context by putting these decision-making frameworks into practice. The ultimate objective is to create a self-healing pipeline that provides increased agility and speed, easier troubleshooting, fewer human errors, less downtime for services, and a shift in emphasis towards system optimisation. The examination and assessment of these approaches highlight the possibilities and efficacy of incorporating automation and advanced decision-making technology into modern IT operations.

## 4 Post-Experimental Analysis

This study's experimental configuration makes use of an OpenStack testbed that is providing a video-on-demand service—along with Stackstorm—an event-driven automation system built on the if-this-then-that principle. This all-inclusive framework manages and assesses the system's reaction to operational anomalies by integrating multiple components—including an experiment controller and a decision controller. A diverse range of equipment and technologies are utilised to establish a resilient testing setting. With the help of integration modules like Ansible 0.5.2 and Openstack 0.7.2, which enable automated workflows and infrastructure management, Stackstorm version 2.10.4 operates on Python. To provide isolation and expansion, the system also uses Docker to set up services in containers. [22] Data processing and analysis tasks that are essential for decision-making processes are supported by essential Python libraries like Numpy, Pandas, and Scikit-learn. The video streaming service is designed with video servers running NGINX-RTMP to manage real-time media streaming—an NGINX-based load balancer to disperse incoming traffic, and client simulations that produce Real-Time Messaging Protocol (RTMP) streams to put the video servers under stress. The system's fault tolerance and scalability are evaluated under settings that closely resemble heavy traffic in the real world. The automation of this experimental setup relies heavily on Stackstorm. Set up inside a Docker container, it coordinates the workflow by invoking operations according to predefined rules that are described in Yet Another Markup Language (YAML) and activated via Hypertext Transfer Protocol (HTTP) webhooks. Across the cloud infrastructure, these rules enable real-time response and recovery activities. In order to test the system's resilience, a variety of abnormalities, including CPU strain and memory leaks, are simulated during the experimental procedure. These abnormalities trigger recovery actions, such as server reboots, service restarts, and dynamic changes like server scaling. The efficacy of these actions in preserving and restoring the operational integrity of the system is carefully recorded and examined. The impact of these recovery actions is measured by the calculation of QoS indicators. Through the monitoring of performance metrics like bytes per second and the quantity of open streams, the resilience of the system to unfavourable circumstances is evaluated. This assessment aids in improving fault tolerance and service availability by optimising recovery procedures. In order to assess and enhance system resilience, this study uses modern automation and monitoring approaches to provide a thorough analysis of cloud-based video services under pressure. The intelligent usage of OpenStack and Docker in conjunction of Stackstorm for real-time automation represents a progressive method for overseeing and streamlining cloud operations. This study highlights the potential of automated systems in improving cloud service efficiency and accuracy, and it provides important insights into the deployment of AI-driven operational methods in cloud environments.

## 5 Result Analysis

Applying AI-based decision-making models to the problem of selecting and carrying out self-recovery activities or auto-remediation processes in response to system anomalies is the goal of this research. The experimental controller is the core component of this study. It controls events related to the system state, maintains a catalogue of recovery actions, and serves as the primary engine for recovery. The controller can create AI-based decision models, log results, simulate anomalies, run corrective actions, and assess outcomes with this configuration. In order to choose the best recovery activities, the decision-making process is based on the analysis of the input data and the use of complex approaches. Nine test scenarios were looked at during the study—and AI modelling approaches were used to identify the best recovery strategies. These experiments' results shed light on the effects and operational effectiveness of various recovery workflows. The "Stop VM" anomaly was simulated in the trials against six different recovery methods, and the outcomes showed varying effects on the overall system performance as well as the QoS. For example, as anticipated, the soft reboot recovery strategy was ineffective since a soft reboot cannot be started on a stopped virtual machine as shown in Figure 3—which would negatively affect QoS. The hard reboot and virtual machine migration processes as shown in Figure 4, on the other hand, proved effective and had a favourable effect on QoS—indicating

that these recovery measures are effective under the evaluated settings. This study highlights the value of automation in modern IT operations by extensively utilising closed loop automation principles to inform system design. This strategy improves system efficiency by reducing the requirement for human intervention and increasing recovery process speed and reliability as shown in Figure 5-8. Apart from periodic modifications to AI-based models for optimisation, the closed-loop system gathers and evaluates operational alerts, determines appropriate responds, and carries out recovery plans without human input.

action_time	6.000000
service_restart_time	0.000000
get_ip_time	0.000003
total_time	25.581866
qos_difference	-45443.327281
qos_average	235093.446125
qos_error_percentage	0.373711
normal_qos	228421.963163

**Figure 3.** Outcome of a soft reboot

action_time	17.000000
service_restart_time	31.000000
get_ip_time	33.141676
total_time	84.728058
qos_difference	16934.010517
qos_average	237047.206678
qos_error_percentage	0.001219
normal_qos	226538.779028

**Figure 4.** Outcome of a hard reboot

action_time	18.000000
action_confirm_time	6.000000
total_time	47.179675
ansbie_run	0.000000
qos_difference	-53051.536285
qos_average	236976.184171
qos_error_percentage	1.945856
normal_qos	230238.380912

**Figure 5.** Outcome of a resize

total_time	2.664301
qos_difference	-62294.450846
qos_average	255066.324790
qos_error_percentage	1.619087

**Figure 6.** Outcome of a restart service

total_time	154.247029
create_new_snapshot	24.000000
create_new_vm	8.000000
assign_floating_ip	54.000000
ansible_run	34.431835
qos_difference	51609.558954
qos_average	236801.445267
qos_error_percentage	0.001889
normal_qos	251949.852912

**Figure 7.** Outcome of a scale up

action	84.853236
total_time	23.000000
create_new_vm	6.000000
assign_floating_ip	6.000000
ansible_run	1.227992
qos_difference	4122.361648
qos_average	237110.188650
qos_error_percentage	0.001177
normal_qos	228519.870643

**Figure 8.** Outcome of a migrate

The study also examined MCDM strategies to enhance the decision-making process. Models like the WSM as shown in Figure 9 and Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) as shown in Figure 10 were used in this. Based on several factors, such as MTTR, influence on QoS, and success likelihood, these models assisted with evaluating the possible recovery pathways. A methodical approach for choosing recovery procedures was made possible by the implementation of these decision-making frameworks. In particular, the WSM and TOPSIS models were helpful in evaluating the combined outcomes of the test scenarios since they provide a methodical framework for contrasting the effectiveness of every recovery strategy. The models took into account a number of variables, including the amount of time needed to complete recovery operations, the QoS before and after the workflow, and the error rate during recovery. The use of AI-based models, like TOPSIS and WSM, has highlighted how crucial it is to have strong, data-driven frameworks for cloud system decision-making. By weighing the pros and cons of various recovery approaches, these models help determine which course of action is best for reducing anomalies and boosting system resilience. The way that this study shows how integrated AI-driven decision-making frameworks can enhance the automation and self-recovery capabilities of cloud services makes a substantial contribution to the field of cloud computing. The study employs rigorous testing and analysis to demonstrate how advanced decision-making strategies can improve operational efficiency and dependability in cloud systems.

ALT./CRIT.	MTTR (min) W.0.1	Number of Steps (min) W.0.1	Impact (max) W.0.1	Error Percentage (min) W.0.1	Success Probability (max) W.0.5	Rank
soft reboot	25.5819	3	16851.1	0.373711	0	5
hard reboot	84.7281	3	78228.5	0.0012185	100	2
resize	47.1797	4	9242.91	1.94586	0	6
migrate	23	6	66416.8	0.00117718	100	1
scaleUp	154.247	5	113904	0.00188861	100	3
restart_service	2.6643	1	0	1.61909	0	4

**Figure 9.** Outcome of a WSM

ALT./CRIT.	MTTR (min) W.0.1	Number of Steps (min) W.0.1	Impact (max) W.0.1	Error Percentage (min) W.0.1	Success Probability (max) W.0.5	Rank
soft reboot	25.5819	3	16851.1	0.373711	0	4
hard reboot	84.7281	3	79228.5	0.0012185	100	1
resize	47.1797	4	9242.91	1.94586	0	6
migrate	23	6	66416.8	0.00117716	100	2
scaleUp	154.247	5	113904	0.00188861	100	3
restart_service	2.6643	1	0	1.61909	0	5

**Figure 10.** Outcome of a TOPSIS

## 6 Conclusion and Future Directions

Using Stackstorm and OpenStack, this study investigates the creation and application of a self-healing recovery engine coupled with an event-driven automation system. Based on the if-this-then-that premise, Stackstorm, operating within a Docker container, coordinates recovery operations for a video-on-demand service. These REST call and webhook-triggered procedures are intended to handle irregularities in the service environment. During the experimental phase, six recovery workflows were put into place and three distinct types of anomalies were injected into the system to simulate test scenarios. In order to provide quantifiable indicators of workflow effectiveness, these situations were analysed using metrics such as MTTR, success potential, and the influence on QoS. During the last stage, recovery tactics were optimised using AI-based decision-making techniques such as TOPSIS, and WSM. To lessen human bias, these techniques made use of weight vectors derived from presumptive expert knowledge and an entropy-based computation. Multi-criteria analysis was used to evaluate the recovery outcomes in order to determine the success of the decision-making process. This innovative method demonstrated how automation and AI may improve the reliability and efficacy of cloud services.

## References

- [1] Habib, H., Kashyap, G.S., Tabassum, N., Nafis, T.: Stock Price Prediction Using Artificial Intelligence Based on LSTM– Deep Learning Model. In: Artificial Intelligence & Blockchain in Cyber Physical Systems: Technologies & Applications. pp. 93–99. CRC Press (2023). <https://doi.org/10.1201/9781003190301-6>.
- [2] Kashyap, G.S., Mahajan, D., Phukan, O.C., Kumar, A., Brownlee, A.E.I., Gao, J.: From Simulations to Reality: Enhancing Multi-Robot Exploration for Urban Search and Rescue. (2023).
- [3] Wazir, S., Kashyap, G.S., Malik, K., Brownlee, A.E.I.: Predicting the Infection Level of COVID-19 Virus Using Normal Distribution-Based Approximation Model and PSO. Presented at the (2023). [https://doi.org/10.1007/978-3-031-33183-1\\_5](https://doi.org/10.1007/978-3-031-33183-1_5).
- [4] Wazir, S., Kashyap, G.S., Saxena, P.: MLOps: A Review. (2023).
- [5] Kashyap, G.S., Malik, K., Wazir, S., Khan, R.: Using Machine Learning to Quantify the Multimedia Risk Due to Fuzzing. Multimedia Tools and Applications. 81, 36685–36698 (2022). <https://doi.org/10.1007/s11042-021-11558-9>.
- [6] Naz, S., Kashyap, G.S.: Enhancing the predictive capability of a mathematical model for pseudomonas aeruginosa through artificial neural networks. International Journal of Information Technology 2024. 1–10 (2024). <https://doi.org/10.1007/S41870-023-01721-W>.
- [7] Kashyap, G.S., Wazir, S., Hamdard, J., Delhi, N., Malik, I.K., Sehgal, V.K., Dhakar, R.: Revolutionizing Agriculture: A Comprehensive Review of Artificial Intelligence Techniques in Farming. (2024). <https://doi.org/10.21203/RS.3.RS-3984385/V1>.
- [8] Kaur, P., Kashyap, G.S., Kumar, A., Nafis, M.T., Kumar, S., Shokeen, V.: From Text to Transformation: A Comprehensive Review of Large Language Models' Versatility. (2024).

- [9] Khder, M.A.: Web scraping or web crawling: State of art, techniques, approaches and application. International Journal of Advances in Soft Computing and its Applications. 13, 144–168 (2021). <https://doi.org/10.15849/ijasca.211128.11>.
- [10] Erway, C.C., Küpcü, A., Papamanthou, C., Tamassia, R.: Dynamic provable data possession. In: ACM Transactions on Information and System Security. ACM PUB27 New York, NY, USA (2015). <https://doi.org/10.1145/2699909>.
- [11] Chen, M., Mao, S., Liu, Y.: Big data: A survey. In: Mobile Networks and Applications. pp. 171–209. Springer (2014). <https://doi.org/10.1007/s11036-013-0489-0>.
- [12] Cravero, A., Sepúlveda, S.: Use and adaptations of machine learning in big data—applications in real cases in agriculture, <https://www.mdpi.com/2079-9292/10/5/552/htm>, (2021). <https://doi.org/10.3390/electronics10050552>.
- [13] Lu, Y.: Artificial intelligence: a survey on evolution, models, applications and future trends, <https://www.tandfonline.com/doi/abs/10.1080/23270012.2019.1570365>, (2019). <https://doi.org/10.1080/23270012.2019.1570365>.
- [14] Popa, R.A., Lorch, J.R., Molnar, D., Wang, H.J., Zhuang, L.: Enabling security in cloud storage SLAs with CloudProof. In: Proceedings of the 2011 USENIX Annual Technical Conference, USENIX ATC 2011. pp. 355–368 (2019).
- [15] Hew, T.S., Kadir, S.L.S.A.: Predicting the acceptance of cloud-based virtual learning environment: The roles of Self Determination and Channel Expansion Theory. Telematics and Informatics. 33, 990–1013 (2016). <https://doi.org/10.1016/j.tele.2016.01.004>.
- [16] Hummer, W., Muthusamy, V., Rausch, T., Dube, P., El Maghraoui, K., Murthi, A., Oum, P.: ModelOps: Cloud-based lifecycle management for reliable and trusted AI. In: Proceedings - 2019 IEEE International Conference on Cloud Engineering, IC2E 2019. pp. 113–120. Institute of Electrical and Electronics Engineers Inc. (2019). <https://doi.org/10.1109/IC2E.2019.00025>.
- [17] Kashyap, G.S., Siddiqui, A., Siddiqui, R., Malik, K., Wazir, S., Brownlee, A.E.I.: Prediction of Suicidal Risk Using Machine Learning Models, <https://papers.ssrn.com/abstract=4709789>, (2021).
- [18] Kashyap, G.S., Brownlee, A.E.I., Phukan, O.C., Malik, K., Wazir, S.: Roulette-Wheel Selection-Based PSO Algorithm for Solving the Vehicle Routing Problem with Time Windows. (2023).
- [19] Marwah, N., Singh, V.K., Kashyap, G.S., Wazir, S.: An analysis of the robustness of UAV agriculture field coverage using multi-agent reinforcement learning. International Journal of Information Technology (Singapore). 15, 2317–2327 (2023). <https://doi.org/10.1007/s41870-023-01264-0>.
- [20] Kanojia, M., Kamani, P., Kashyap, G.S., Naz, S., Wazir, S., Chauhan, A.: Alternative Agriculture Land-Use Transformation Pathways by Partial-Equilibrium Agricultural Sector Model: A Mathematical Approach. (2023).
- [21] Kashyap, G.S., Sohlot, J., Siddiqui, A., Siddiqui, R., Malik, K., Wazir, S., Brownlee, A.E.I.: Detection of a facemask in real-time using deep learning methods: Prevention of Covid 19. (2024).
- [22] R. Arora, S. Gera, M. Saxena “Mitigating Security Risks on Privacy of Sensitive Data used in Cloud-based ERP Applications”; Proceedings of the 15th INDIACOM; INDIACOM-2021; IEEE Conference ID: 51348. 2021 8th International Conference on “Computing for Sustainable Global Development”, 17th - 19th March, 2021. pp 458-463 Electronic ISBN:978-93-80544-43-4. PoD ISBN:978-1-7281-9546-9.
- [23] A. Soni, S. Alla, S. Doddha, H. Volikatla “Advancing Household Robotics: Deep Interactive Reinforcement Learning for Efficient Training and Enhanced Performance,” in Journal of Electrical Systems (JES), vol. 20, no. 3s(2024), pp. 1349-1355 May 2024, doi: <https://doi.org/10.52783/jes.1510>.