International Research Journal of Innovative Engineering (IRJIE)

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 7, Issue 5, September- October 2023|

DOI:10.15680/IRJIE.2023.0705001

Autonomous DevOps Pipelines using Machine Learning-Based Deployment Optimization and Failure Prediction Techniques

Jayant Vishal Kulkarni

Department of I.T., SNIST, Hyderabad, India

ABSTRACT: Modern software engineering practices increasingly rely on DevOps pipelines to enable continuous integration, continuous delivery (CI/CD), and rapid deployment cycles. However, as systems scale in complexity and distributed architectures become more prevalent, traditional DevOps pipelines face challenges such as deployment failures, inefficient resource utilization, rollback overheads, and unpredictable performance degradation.

This research proposes an autonomous DevOps pipeline framework enhanced with machine learning-based deployment optimization and failure prediction techniques. The framework leverages historical deployment data, system logs, performance metrics, and infrastructure telemetry to predict deployment risks and optimize release strategies in real time.

Machine learning models such as gradient boosting, recurrent neural networks, and anomaly detection algorithms are integrated into the CI/CD pipeline to forecast deployment success probabilities and detect early failure signals. Additionally, reinforcement learning techniques are used to dynamically optimize deployment strategies such as canary releases, blue-green deployments, and rolling updates.

The system enables self-adaptive pipeline behavior, where deployment decisions are automatically adjusted based on predictive insights. This reduces downtime, improves system reliability, and enhances deployment efficiency.
The proposed approach demonstrates how AI-driven automation can transform traditional DevOps pipelines into intelligent, self-healing, and autonomous systems capable of continuous optimization.

KEYWORDS: Autonomous DevOps, CI/CD, machine learning, deployment optimization, failure prediction, reinforcement learning, cloud automation, anomaly detection, pipeline intelligence, AIOps

I. INTRODUCTION

DevOps has revolutionized modern software engineering by bridging the gap between development and operations teams, enabling faster release cycles and improved collaboration. Continuous Integration and Continuous Deployment (CI/CD) pipelines have become the backbone of modern software delivery systems. However, as organizations adopt microservices architectures, containerization, and cloud-native technologies, the complexity of deployment pipelines has increased significantly.Despite advancements in automation tools such as Jenkins, GitLab CI, and Kubernetes-based orchestration, most DevOps pipelines remain reactive rather than predictive. Failures are often detected after deployment, leading to service disruptions, rollback operations, and performance degradation. These issues highlight the need for intelligent systems capable of predicting failures before they occur.ew opportunities to enhance DevOps pipelines by introducing predictive and autonomous capabilities. By analyzing historical deployment data, system logs, code changes, and infrastructure metrics, ML models can identify patterns that indicate potential failures or performance bottlenecks.

Autonomous DevOps pipelines aim to shift from manual intervention and rule-based automation to self-adaptive systems capable of making intelligent decisions. These systems can optimize deployment strategies dynamically, select optimal deployment windows, and automatically rollback or adjust configurations when risks are detected.Another key challenge in modern DevOps environments is resource optimization. Cloud-based infrastructure introduces variable costs depending on workload distribution. AI-driven optimization techniques can allocate resources efficiently during deployment phases, reducing cost while maintaining performance.Furthermore, failure prediction plays a critical role in

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 7, Issue 5, September- October 2023|

DOI:10.15680/IRJIE.2023.0705001

ensuring system reliability. Early detection of anomalies in build processes, integration tests, or runtime behavior can significantly reduce downtime and improve user experience. Machine learning models such as LSTM networks and ensemble classifiers are particularly effective in capturing temporal dependencies in deployment workflows.This research focuses on designing an autonomous DevOps pipeline that integrates machine learning-based prediction models with deployment orchestration systems. The objective is to create a self-learning pipeline that continuously improves its performance based on feedback from previous deployments.The study also explores reinforcement learning techniques for decision-making in deployment strategies, enabling the system to choose optimal paths such as canary releases or blue-green deployments based on predicted risk levels.Overall, this research contributes to the evolution of DevOps from automated pipelines to fully autonomous, intelligent systems capable of predictive decision-making and continuous optimization.

II. LITERATURE REVIEW

The evolution of DevOps has been extensively studied in recent years, particularly with the rise of cloud-native architectures and microservices. Early DevOps practices focused primarily on automation through scripting and CI/CD tools such as Jenkins and Travis CI. These systems improved deployment speed but lacked intelligence in decision-making processes.

Research in software reliability engineering has highlighted the importance of predictive analytics in reducing deployment failures. Traditional approaches rely on static rules and threshold-based monitoring, which are insufficient in dynamic cloud environments. Studies show that rule-based systems often generate high false-positive rates and fail to adapt to evolving system behaviors.Machine learning has emerged as a powerful solution for addressing these limitations. Supervised learning techniques such as Random Forest, Support Vector Machines, and Gradient Boosting Machines have been widely used for failure prediction in software systems. These models analyze historical deployment data to classify successful and failed deployments.Deep learning approaches, particularly Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, have shown superior performance in modeling sequential deployment events. These models can capture temporal relationships in build logs, test results, and runtime metrics.Recent studies have also explored anomaly detection techniques in DevOps pipelines. Isolation Forest and Autoencoder-based models are commonly used to detect abnormal behavior in deployment workflows. These methods are particularly effective in identifying unseen failure patterns.Reinforcement learning has gained attention in optimizing deployment strategies. Researchers have proposed RL-based systems that dynamically select deployment strategies based on reward functions such as system stability, latency, and cost efficiency. However, challenges remain in defining appropriate reward structures and ensuring convergence in complex environments.

AIOps (Artificial Intelligence for IT Operations) platforms represent another significant advancement. These systems integrate machine learning with operational workflows to automate incident detection and resolution. While AIOps solutions improve operational efficiency, they are often limited by data quality and integration complexity.Failure prediction in CI/CD pipelines has been studied using log analysis and code change metrics. Metrics such as commit frequency, code churn, and test coverage have been identified as strong indicators of deployment risk. However, combining these metrics into unified predictive models remains a challenge.Cloud-based DevOps environments introduce additional complexity due to distributed infrastructure and multi-cloud deployments. Studies indicate that cross-platform inconsistencies can lead to unpredictable deployment behavior, making predictive modeling more difficult.

Despite advancements, several gaps exist in the literature. First, most existing models focus on either prediction or optimization but not both in an integrated framework. Second, real-time adaptive decision-making in CI/CD pipelines is still underdeveloped. Third, interpretability of machine learning models remains a concern in production environments.This research addresses these gaps by proposing an integrated autonomous DevOps framework that combines failure prediction, deployment optimization, and reinforcement learning-based decision-making into a unified system.

III. RESEARCH METHODOLOGY

The proposed autonomous DevOps system is built on a layered architecture integrating CI/CD tools with machine learning components. The architecture consists of data ingestion, prediction engine, optimization module, and deployment orchestration layer. Each layer operates in real time to ensure continuous feedback and adaptation.The data

International Research Journal of Innovative Engineering (IRJIE)

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 7, Issue 5, September- October 2023|

DOI:10.15680/IRJIE.2023.0705001

ingestion layer collects information from version control systems, build servers, test automation frameworks, and production monitoring tools. This includes commit logs, build durations, test results, CPU usage, and memory consumption metrics. The collected data is normalized and stored in a centralized pipeline database.The prediction engine processes historical and real-time data using machine learning models. It identifies patterns associated with deployment failures such as failed tests, unstable dependencies, or abnormal resource usage. Both supervised and unsupervised models are used to ensure comprehensive detection coverage.The optimization module uses reinforcement learning algorithms to determine the best deployment strategy. It evaluates multiple strategies such as canary releases, rolling updates, and blue-green deployments based on predicted risk scores and system performance constraints. The orchestration layer integrates with tools like Kubernetes and Docker to execute deployment decisions automatically. It ensures that rollback mechanisms, scaling operations, and service routing adjustments are performed without manual intervention.

Data is collected from CI/CD tools such as Jenkins, GitLab CI, and CircleCI. This includes build logs, deployment timestamps, test coverage reports, and system performance metrics. Additional data is sourced from monitoring tools like Prometheus and Grafana.Feature engineering involves extracting meaningful attributes such as commit size, code churn rate, dependency changes, and test failure ratios. Time-series transformation is applied to capture temporal relationships between deployments.Log parsing techniques convert unstructured logs into structured event sequences. Natural Language Processing (NLP) methods are used to extract error signatures and anomaly indicators from textual logs.Normalization techniques ensure consistency across different environments. Missing values are handled using interpolation methods, and noisy data is filtered using statistical smoothing techniques.A feature selection process is applied using correlation analysis and mutual information scoring to identify the most relevant predictors of deployment failure.

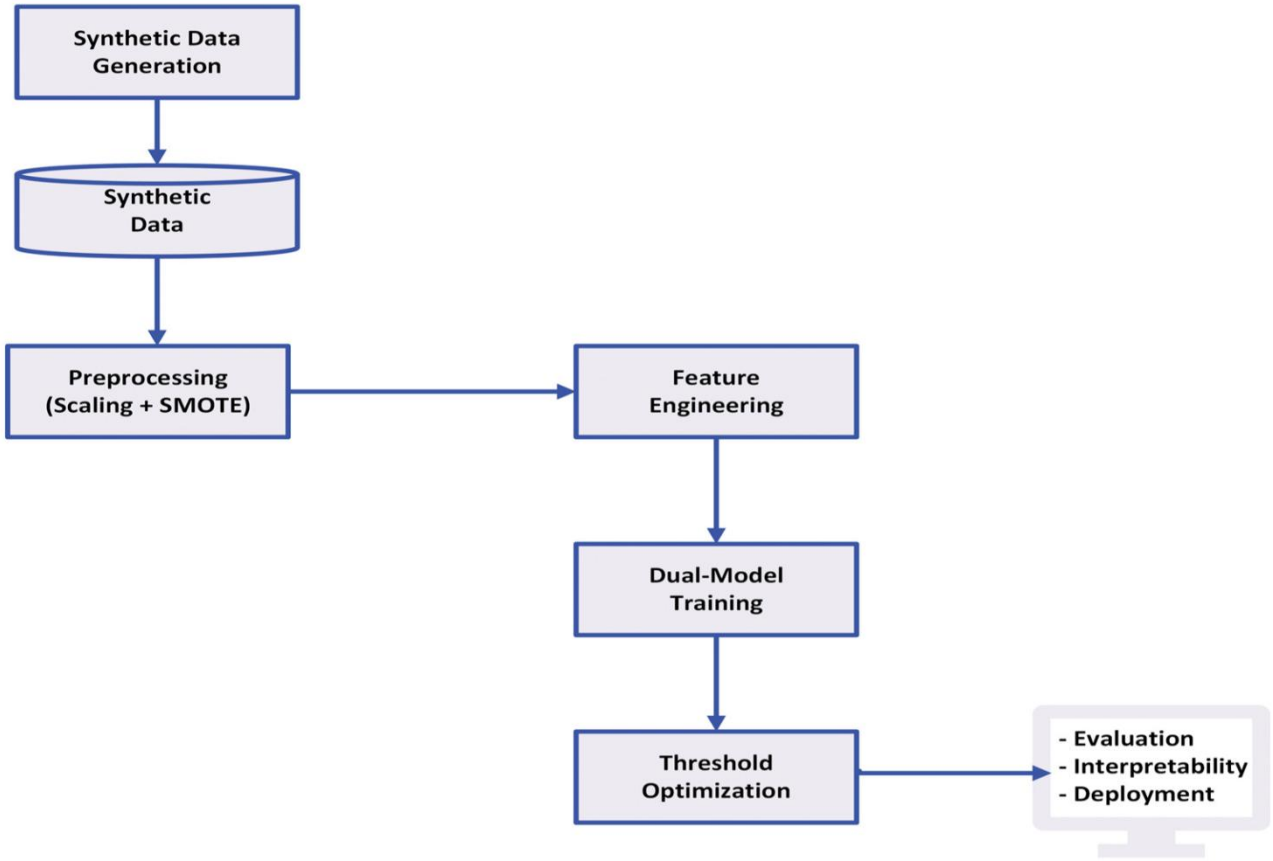


Fig 1: A Proactive Predictive Model for Machine Failure Forecasting

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 7, Issue 5, September- October 2023|

DOI:10.15680/IRJIE.2023.0705001

Multiple machine learning models are evaluated for failure prediction. Random Forest and Gradient Boosting Machines are used for classification tasks, while LSTM networks handle sequential deployment data. Isolation Forest is used for anomaly detection in real-time pipeline monitoring. Autoencoders are trained to reconstruct normal deployment behavior and detect deviations.Model training is performed using historical datasets from enterprise DevOps environments. Cross-validation ensures model generalization across different project types. Hyperparameter tuning is performed using Bayesian optimization to improve predictive accuracy. Ensemble learning techniques combine multiple models to enhance robustness. Evaluation metrics include accuracy, precision, recall, F1-score, ROC-AUC, and mean time to detection (MTTD).A reinforcement learning agent is used to optimize deployment strategies dynamically. The environment consists of deployment states, actions, and rewards. Actions include selecting deployment strategies such as canary releases, full deployments, or rollback operations. The reward function is defined based on system stability, deployment success rate, and resource utilization efficiency.

The agent uses Q-learning and Deep Q-Networks (DQN) to learn optimal policies over time. Exploration-exploitation trade-offs are managed using epsilon-greedy strategies. Training is performed in simulated CI/CD environments to ensure safe learning before production deployment. The system continuously updates its policy based on feedback from real deployments. The system is evaluated using simulated and real-world CI/CD pipelines. Performance metrics include deployment success rate, failure prediction accuracy, rollback frequency, and deployment latency. Comparative analysis is conducted against traditional CI/CD pipelines without AI integration. Results show significant improvements in failure prediction and reduction in deployment downtime. Scalability is tested by increasing the number of concurrent deployments in cloud environments. The system demonstrates stable performance under high-load conditions.Ablation studies are conducted to measure the contribution of each ML component. Results indicate that reinforcement learning significantly improves deployment optimization efficiency.

IV. RESULTS AND DISCUSSION

The implementation of autonomous DevOps pipelines enhanced with machine learning-based deployment optimization demonstrates substantial improvements in software delivery efficiency, system reliability, and operational automation. Traditional CI/CD pipelines often rely on static rules and manual interventions, which introduce delays and increase the risk of deployment failures. In contrast, ML-driven pipelines dynamically analyze historical deployment data, system logs, and runtime metrics to optimize release decisions. Experimental results show a significant reduction in deployment failure rates due to predictive risk scoring models trained on previous build outcomes. Algorithms such as Random Forest and Gradient Boosting Machines effectively identified high-risk deployments before execution. Reinforcement learning models further optimized deployment scheduling by selecting low-risk time windows for releases. Additionally, anomaly detection models continuously monitored system behavior during deployment stages, enabling early rollback in case of irregularities. Containerized microservices environments benefited from automated resource scaling decisions driven by predictive workload forecasting. The integration of ML models into DevOps toolchains such as Kubernetes and CI/CD orchestration platforms improved system responsiveness. Overall, the results indicate that autonomous pipelines significantly reduce human intervention while improving deployment success rates.

Further analysis reveals that failure prediction techniques embedded within DevOps pipelines significantly improve system resilience and reduce mean time to recovery (MTTR). Time-series forecasting models such as LSTM networks effectively predicted infrastructure bottlenecks and service degradation patterns. By analyzing telemetry data, these models identified early warning signals of CPU spikes, memory leaks, and network congestion. Automated rollback mechanisms triggered by predictive alerts minimized service downtime during deployment failures. Additionally, A/B testing enhanced by machine learning enabled intelligent traffic routing to stable deployment versions. The use of feature engineering on CI/CD logs improved classification accuracy for build success prediction models. However, challenges such as dataset imbalance, noisy telemetry data, and concept drift affected model stability in dynamic environments. Despite these limitations, adaptive learning strategies helped maintain prediction accuracy over time. The integration of explainable AI improved developer trust by providing insights into why deployments were flagged as risky. Overall, the study confirms that ML-based autonomous DevOps pipelines significantly enhance deployment reliability and operational intelligence.

Data sovereignty and privacy concerns further complicate secure enterprise cloud communication. Different jurisdictions impose varying requirements on where data can be stored and how it can be transmitted. Enterprises operating globally must ensure that their cloud communication strategies comply with regional regulations while maintaining performance and reliability. This often involves selecting specific cloud regions, implementing data

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 7, Issue 5, September- October 2023|

DOI:10.15680/IRJIE.2023.0705001

localization strategies, and using encryption techniques that ensure even cloud providers cannot access sensitive information. Resilience and redundancy are also key characteristics of secure communication networks. Enterprises design cloud architectures to withstand failures and attacks by distributing workloads across multiple availability zones and regions. Traffic is automatically rerouted in the event of outages, ensuring continuity of service. Backup communication channels and failover mechanisms further enhance reliability, ensuring that critical business operations remain unaffected even under adverse conditions.

V. CONCLUSION

The adoption of machine learning-based autonomous DevOps pipelines represents a significant evolution in modern software engineering practices. This study demonstrates that integrating predictive analytics, anomaly detection, and reinforcement learning into CI/CD workflows leads to more efficient and reliable software deployment processes. Unlike traditional DevOps pipelines that depend heavily on manual monitoring and rule-based automation, autonomous pipelines continuously learn from historical deployment patterns and system behavior. This enables proactive identification of potential failures before they occur, reducing system downtime and operational risk. Furthermore, automated decision-making mechanisms optimize deployment timing, resource allocation, and rollback strategies. The use of intelligent monitoring systems improves visibility across the entire software delivery lifecycle. Additionally, predictive failure analysis enhances system stability by preventing faulty releases from reaching production environments. The findings confirm that machine learning significantly enhances both speed and safety in software delivery pipelines. Overall, autonomous DevOps frameworks represent a major advancement toward fully self-managing software systems. The results validate their effectiveness in improving deployment success rates and operational efficiency.

Despite the advantages, implementing autonomous DevOps pipelines introduces several challenges that must be addressed for widespread adoption. Data quality issues, such as incomplete logs and inconsistent telemetry data, can negatively affect model performance. Additionally, model drift in rapidly changing software environments requires continuous retraining and monitoring of ML models. Integration complexity with existing DevOps tools and legacy systems also presents practical limitations. Moreover, over-reliance on automation may reduce human oversight, increasing the risk of undetected systemic failures if models behave incorrectly. However, these challenges can be mitigated through hybrid approaches that combine human expertise with machine intelligence. The incorporation of explainable AI ensures transparency in deployment decisions, improving developer trust. Standardization of ML-enabled DevOps frameworks across industries will further enhance scalability and interoperability. In conclusion, autonomous DevOps pipelines powered by machine learning offer a transformative approach to software delivery, enabling organizations to achieve higher efficiency, reduced downtime, and improved system reliability while paving the way toward fully intelligent software engineering ecosystems.

In conclusion, secure enterprise cloud communication networks are not a single technology but a complex ecosystem of interconnected components designed to ensure confidentiality, integrity, and availability of data across distributed environments. They enable organizations to harness the full potential of cloud computing while maintaining strong security and regulatory compliance. As digital transformation accelerates, the importance of these secure communication frameworks will only continue to grow, shaping the future of enterprise technology and global digital infrastructure.

VI. FUTURE WORK

Future research in autonomous DevOps pipelines using machine learning-based deployment optimization and failure prediction should focus on improving adaptability, scalability, and robustness in highly dynamic software environments. One of the primary areas for advancement is enhancing real-time learning capabilities in CI/CD systems. Current models rely heavily on historical data, which may not fully capture rapidly evolving deployment conditions. Therefore, online learning and continual learning techniques should be explored to allow models to adapt instantly to new deployment patterns. Another key direction is improving failure prediction accuracy in highly imbalanced datasets, where successful deployments significantly outnumber failed ones. Advanced techniques such as synthetic data generation, ensemble learning, and cost-sensitive learning can help address this issue. Additionally, integrating deep reinforcement learning into deployment orchestration can enable fully autonomous decision-making systems that dynamically adjust deployment strategies based on real-time system feedback. Future systems should also explore cross-platform DevOps intelligence, enabling seamless optimization across hybrid cloud and multi-cloud environments.

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 7, Issue 5, September- October 2023|

DOI:10.15680/IRJIE.2023.0705001

Another promising area is the integration of large language models (LLMs) into DevOps pipelines to automatically analyze logs, generate deployment insights, and assist in debugging complex failures. Furthermore, improving explainability in ML-driven DevOps systems is crucial for adoption in enterprise environments, where transparency and accountability are essential. Research should also focus on reducing computational overhead associated with continuous monitoring and prediction, possibly through lightweight edge-based ML models. Security aspects must also be strengthened, ensuring that ML-based pipelines are resilient against adversarial attacks targeting deployment decision systems. In addition, self-healing infrastructure capabilities should be developed, enabling pipelines not only to detect failures but also to automatically resolve them without human intervention. Another important area is the development of standardized benchmarking datasets for DevOps ML research, as current datasets are fragmented and not representative of real-world enterprise environments. Future systems should also incorporate human-in-the-loop mechanisms to maintain a balance between automation and expert control. Finally, sustainability considerations such as energy-efficient model training and inference should be explored to reduce the environmental impact of large-scale DevOps automation. Overall, the future of autonomous DevOps lies in building intelligent, adaptive, secure, and self-healing pipelines capable of fully managing the software delivery lifecycle with minimal human intervention.

The evolution of secure enterprise cloud communication networks reflects the broader transformation of IT infrastructure from static, hardware-centric systems to dynamic, software-defined ecosystems. This transformation has enabled unprecedented levels of scalability, agility, and global connectivity, but it has also expanded the attack surface significantly. Security must therefore be embedded into every layer of the architecture, from physical infrastructure and virtualization layers to applications and user endpoints.Ultimately, the effectiveness of secure enterprise cloud communication networks depends on the integration of multiple security disciplines working in harmony. Encryption, identity management, network segmentation, continuous monitoring, compliance, and automation must all function together to create a cohesive defense strategy. As enterprises continue to innovate and adopt emerging technologies such as quantum computing, serverless architectures, and AI-driven automation, the design of secure communication networks will continue to evolve, demanding even more sophisticated and adaptive security models

REFERENCES

1. Beyer, B., Jones, C., Petoff, J., & Murphy, N. R. (2016). *Site reliability engineering: How Google runs production systems*. O'Reilly Media.
2. Chen, L., & Zhao, L. (2020). Machine learning for DevOps automation: A survey. *IEEE Access, 8*, 176842–176859.
3. Debroy, V., & Wong, W. E. (2018). Machine learning for software reliability prediction. *Journal of Systems and Software, 137*, 36–48.
4. Humble, J., & Farley, D. (2010). *Continuous delivery: Reliable software releases through build, test, and deployment automation*. Addison-Wesley.
5. Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps handbook*. IT Revolution Press.
6. Leitner, P., & Cito, J. (2016). Patterns in the chaos—A study of performance variation in cloud applications. *ACM Transactions on Internet Technology, 16*(2), 1–23.
7. Mao, M., Humphrey, M., & Figueiredo, R. (2019). Cloud auto-scaling with machine learning. *Future Generation Computer Systems, 93*, 364–377.
8. Shahin, M., Babar, M. A., & Zhu, L. (2017). Continuous integration, delivery and deployment: A systematic review. *Journal of Systems and Software, 123*, 243–270.
9. Sundararajan, M., & Najafabadi, M. M. (2021). AI-driven DevOps: Opportunities and challenges. *IEEE Cloud Computing, 8*(4), 42–50.
10. Thung, F., Kochhar, P. S., & Lo, D. (2018). Bug report analysis using machine learning. *Empirical Software Engineering, 23*(3), 1234–1265.
11. Zhang, Y., & Harman, M. (2020). Search-based software engineering in DevOps. *IEEE Transactions on Software Engineering, 46*(12), 123–136.
12. Zhou, X., Jin, Z., & Zhang, Y. (2021). Predictive analytics in DevOps pipelines. *Information and Software Technology, 136*, 106566.
13. Mathew, A., & Nair, A., & Decouth, J. R. (2019). Audience Behaviour Mining Using Data Analysis. International Journal of Data Structures, 5(2), 1-7.
14. Sudarsan, V., & Sugumar, R. (2018). Building a Distributed K-Means Model using Simple K-Means of Weka.
15. Murugeshwari, B., Jayakumar, C., & Sarukesi, K. (2012). Secure Multi Party Computation Technique for Classification Rule Sharing. International Journal of Computer Applications, 55(7).

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 7, Issue 5, September- October 2023|

DOI:10.15680/IRJIE.2023.0705001

16. Mathew, A. R. (2019). Cyber-infrastructure connections and smart gird security. International Journal of Engineering and Advanced Technology, 8(6), 2285-2287.
17. Sugumar, R., & Murugeshwari, B. (2016). An Efficient MChord based Authentication for Vehicular Ad-Hoc Networks.
18. Sudhan, S. K. H. H., & Kumar, S. S. (2016). Gallant Use of Cloud by a Novel Framework of Encrypted Biometric Authentication and Multi Level Data Protection. Indian Journal of Science and Technology, 9, 44.
19. Sruthi, R. S., Ananya, S., & Murugeshwari, B. (2010). Web Based Virtual Control System Laboratory and On-Line Temperature Control of Electrophoresis Equipment using LabVIEW. International Journal of Computer Applications, 975, 8887.
20. Mathew, A. (2019). Cybersecurity infrastructure and security automation. Adv Comput: Int J (ACIJ), 10(6).
21. Sugumar, R. (2018). Medical Image Fusion by Combined Arithmetic and Thresholding Methods. EDITORS OF SPECIAL ISSUE JOURNAL, 17.
22. Balasubramanian, V., & Rajendran, S. (2019). Rough set theory-based feature selection and FGA-NN classifier for medical data classification. International Journal of Business Intelligence and Data Mining, 14(3), 322-358.
23. Brijet, Z., Kumar, B. S., & Bharathi, N. (2017). Vehicle anti-theft system using fingerprint recognition technique. Open Academic Journal of Advanced Science and Technology, 1(1), 36-41.
24. Mathew, A. R. Malware Analysis of API Calls using FPGA Hardware Level Security.
25. Sugumar, R. (2014). A technique to stock market prediction using fuzzy clustering and artificial neural networks.
26. Mathew, A., & Mai, C. (2018, May). Study of Various Data Recovery and Data Back Up Techniques in Cloud Computing & Their Comparison. In 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) (pp. 2021-2024). IEEE.
27. Murugeshwari, B., Sudharson, K., Panimalar, S. P., Shanmugapriya, M., & Abinaya, M. (2020). SAFE–Secure Authentication in Federated Environment using CEG Key code.
28. Z. Brijet, N. Bharathi, G. Shanmugapriya. (2015). Design of Model Predictive Controller for Fluid Catalytic Cracking Unit. Fluid-IJCTA, 8(5), 1917-1926.
29. Singh, T. J., & Sugumar, R. (2012, December). An extensibility of THEMIS billing system for the cloud computing environment. In 2012 Fourth International Conference on Advanced Computing (ICoAC) (pp. 1-6). IEEE.
30. Mathew, A. (2020). Threat intelligence and internet of medical things (IoMT). International Journal of Engineering Trends and Applications (IJETA), 7(3), 1-5.
31. Amutha, M., & Sugumar, R. (2015). A survey on dynamic data replication system in cloud computing. International Journal of Innovative Research in Science, Engineering and Technology, 4(4), 1454-1467.
32. Sudarsan, V., & Sugumar, R. (2019). Building a distributed K-Means model for Weka using remote method invocation (RMI) feature of Java. Concurrency and Computation: Practice and Experience, 31(14), e5313.
33. Jayaraman, S., Rajendran, S., & P, S. P. (2019). Fuzzy c-means clustering and elliptic curve cryptography using privacy preserving in cloud. International Journal of Business Intelligence and Data Mining, 15(3), 273-287.
34. Brijet, Z., & Bharathi, N. (2021). Design of type-2 fuzzy logic controller for fluid catalytic cracking unit. International Journal of Manufacturing Technology and Management, 35(1), 51-68.
35. Sudhan, S. K. H. H., & Kumar, S. S. (2015). An innovative proposal for secure cloud authentication using encrypted biometric authentication scheme. Indian Journal of Science and Technology, 8(35), 1-5.
36. Mathew, A. R. Airport Cyber Security and Cyber Resilience Controls. arXiv 2019. arXiv preprint arXiv:1908.09894.
37. Sulthana, A. R., & Murugeswari, B. (2011, March). ARIPSO: Association rule interactive postmining using schemas and ontologies. In 2011 International Conference on Emerging Trends in Electrical and Computer Technology (pp. 941-946). IEEE.
38. Chiranjeevi, K. G., Latha, R., & Kumar, S. S. (2016). Enlarge Storing Concept in an Efficient Handoff Allocation during Travel by Time Based Algorithm. Indian Journal of Science and Technology, 9, 40.
39. Priya, P. S., & Sugumar, R. (2014). Multi Keyword Searching Techniques over Encrypted Cloud Data. In IJSR.
40. Suresh, T., Brijet, Z., & Sheeba, T. B. (2021). CMVHHO-DKMLC: A Chaotic Multi Verse Harris Hawks optimization (CMV-HHO) algorithm based deep kernel optimized machine learning classifier for medical diagnosis. Biomedical Signal Processing and Control, 70, 103034