ISSN 2277-2685
IJESR/July-Sep. 2025/ Vol-15/Issue-3/639-649
Himabindu Chetlapalli *et. al.,* / International Journal of Engineering & Science Research

Hybrid PSO-SA and DBSCAN with Mini-Batch K-Means for Efficient Software Test Clustering

Himabindu Chetlapalli
9455868 Canada Inc,
Ontario, Canada,
himabinduchetlapalli@ieee.org
Koteswararao Dondapati
Everest Technologies,
Ohio, USA,
Koteswararaodondapati@ieee.org
Durga Praveen Deevi
O2 Technologies Inc,
California, USA,
durgapraveendeevi@ieee.org
Sharadha Kodadi
Infosys, Texas, USA,
sharadhakodadi@ieee.org
Daniel Dauda Wisdom
College of Physical Sciences (Colphys), Department of Computer Sciences,
Federal University of Agriculture Abeokuta (FUNAAB), Ogun State, Nigeria,
danieldw@funaab.edu.ng

ABSTRACT

Software testing is the most crucial factor in making software reliable; old clustering methods alike K-Means and hierarchical clustering fail to deal with the complexities of large-scale test data. This is usually associated with poor scalability, ineffective test suite optimizations, and poor fault detection accuracy. Therefore, the present work strives to hybridize a cluster framework using PSO-SA with DBSCAN and Mini-Batch K-Means for efficient software test clustering to mitigate the limitations imposed by the aforementioned conventional clustering methods. Unlike conventional approaches, the proposed method utilizes density-based clustering for noise handling, batch-based refinement for scaling up, and swarm-intelligent optimized test case prioritization. Evaluation experiments done at the NASA Defect dataset demonstrated a fault detection rate of 95%, execution efficiency of 94%, and code coverage of 96%, higher than NNE-TCP as well as other baselines. The proposed method achieved 97 and 98 overall effectiveness and accuracy, respectively, validating the robustness of the approach to large scales of test clustering. The proposed method also improved effectiveness and valid into testing through comparison with NNE-TCP in standings of clustering accuracy, computational efficiency, and test prioritization. The framework helped enhance detection, reduced redundant test cases, and optimized test execution time. Integration into software testing environments will improve its automation overhead and reliability of software. Hence, it can be established as a valuable advancement in test suite clustering methodologies.

Keywords: *Software Test Clustering, Hybrid Clustering Framework, PSO-SA Optimization, Fault Detection Accuracy, Test Suite Optimization.*

1. INTRODUCTION

In the era of AI advancements, software testing is being impacted significantly, and challenges such as complete test coverage are being addressed by merging pre-trained language models with evolutionary algorithms for generating effective test cases[1]. In addition, AI is a significant factor in compliance monitoring, risk management, and clinical follow-up of Software as a Medical Device (SaMD) for patient safety and regulatory needs [2].

Artificial intelligence has been another boon to various fields: medical diagnosis, data security, automation, and computational efficiencies. Nonetheless, it is also faced with more challenges, which are mainly data privacy, data security, systems scalability, and computational constraints; therefore, there is an urgent need to find innovative solutions for optimized performance and reliability [3]. Knowledge Management (KM) supported by adaptive modeling techniques nurtures strategic business planning and decision-making [4]. Applications of big data analytics in e-commerce are developing product mapping for SMEs and competitive insights against them [5]. Cloud computing is very important for data security and privacy since it relies on biometric authentication and encrypted transmission mechanisms to safeguard extremely sensitive patient data [6]. Integrating blockchain offers an extraordinary level of encryption and authentication mechanism, thereby securing healthcare transaction finances [7]. Cloud-GIS emergency command systems are designed to enhance the earthquake response capacity through high-performance data processing and predictive analytics [8]. mHealth is changing the healthcare landscape through remote patient monitoring and medical record access, which can be further enhanced through secure authentication work and cloud computing methodologies [9]. Within the field of medical imaging, magnetic resonance imaging (MRI) is usually applied in tumor detection and is often corrupted by noise and unsteady scans. The image preprocessing and classification are now improved with machine learning techniques, assuring accurate and reliable tumor identification [10]. With AI and big data analytics being the blooming areas in e-commerce and business intelligence, the digital economy enables upgrades in industrial structure while impelling the spirit of sustainable entrepreneurial development [11]. These deep learning techniques have also contributed toward lung cancer detection and risk assessment via medical imaging and analysis of genetic data [12]. AI-based probabilistic neuro-fuzzy systems help automate the analysis and decision-making processes for diagnosing and monitoring chronic kidney disease (CKD) [13]. AI simulation and modeling are also extensively used in the energy sector. Electric traction systems of electric vehicles are powered and deluxe using artificial neural networks, electrothermal inverter models, and finite element analysis, while security and privacy issues in multi-cloud environments have been flagged [14]. However, to counter such security threats, new authentication, and privacy-preserving methodologies should be developed to protect the information in distributed computing systems [15].

This research proposes an efficient software test clustering framework with integrates Mini-Batch K-Means, DBSCAN, and Hybrid Particle Swarm Optimization with Simulated Annealing (PSO-SA) are used to address the growing difficulty of software testing. The proposed test suite optimization framework thus covers areas such as fault detection and computational efficiency while still ensuring the scalability and security of software testing environments.

The Proposed method's main contribution,

- Propagate Hybrid PSO-SA towards software testing fault detection and optimization through DBSCAN and Mini-Batch K-Means.
- Validate the robustness through comparison with other existing clustering techniques.
- Optimize DBSCAN hyper-parameters for better clustering performance and scalable behavior in big software testing environments.

2. LITERATURE REVIEW

This section surveys the automation and intelligent techniques employed in cloud computing, distributed systems, and networks with a focus on malware detection, software testing, cloud optimization, and cybersecurity, pointing out the main contributions and limitations in these areas. [16]deploys big data analytics, decision support systems, and mixed-integer linear programming to upgrade agricultural supply chain management with better efficiency and foresight. Challenges arise with data processing requirements and scaling in larger networks, which remain unsettled. [17] describes deep learning approaches to phishing detection via stacked autoencoders and SVMs in somewhat greater detail. While the detection accuracy would be enhanced by this deep learning effort, it will require frequent updates to keep in step with fast-evolving phishing strategies.

A transformation framework proposes [18] for business intelligence based on AI-driven data analytics. However, this framework does not include data privacy, regulatory compliance, or scalability as its issues. On another note, [19] uses AI and machine-learning-driven load balancing to efficiently allocate resources in cloud data centers. While successful in optimizing workloads, its implementation encountered issues such as overhead due to extra computation and security problems.

Cloud security [20] proposes a framework that integrates the Immune Cloning Algorithm with data-driven threat mitigation. This solution enhances threat detection but is limited by the requirements of high computational resources, which are rarely available in a low-power environment. In contrast, [21] uses probabilistic modeling for optimizing software deployment verification in the cloud to ensure QoS compliance. Yet, the method relies on predefined non-functional requirements, complicating its adaptation to a changing cloud environment.

However [22] elucidates the latest techniques in fault injection for resilience testing in different types of cloud environments, especially AWS tools to detect and mitigate failures. Since the techniques deal mostly with AWS-built infrastructure, they cannot be applied to other cloud platforms. In the same way, [23] also follows in designing a machine learning-based malware detection framework that uses Support Vector Regression, LSTMs, and Hidden Markov Models. Although the model yields better accuracy, the burden of it being context-heavy in computational power makes it weak toward zero-day threats. [24] combines principles of genetic algorithms and the swarm intelligence technique to derive anabolic effects in software testing for big data environments. No scaling improvement, however, brings with it several problems due to the complexities in the hybrid optimization methods. The last research contribution to the extent of performance management studies in mobile networks through big data analytics is by [25] on subjects like anomaly detection and resource allocation. However, such cost-consuming techniques have restricted it from being used in dynamic conditions of a network.

Framework [26] provides a cloud-based automated software testing framework that integrates fault injection with XML-based test scenarios for robustness enhancement. Efficiency improvements are demonstrated, but these are tied to underlying cloud infrastructure, potentially introducing issues of latency and cost. Meanwhile, [27] investigates test case prioritization improvement in regression testing through the use of harmonized neural networks and heuristic strategies. These methods lessen the overhead cost for fault detection but create implementation challenges during testing.

All of them collectively highlight the potential of AI, machine learning, as well as automation methodologies in cloud computing, cybersecurity as well as software testing. The challenges that still exist regarding computational complexity, scalability in terms of real-world applications, and adaptability to dynamically changing environments all warrant further investigation into efficient, flexible, and cost-effective solutions.

3. PROBLEM STATEMENT

A framework [28] suggests built on machine learning for malware detection involving Support Vector Regression, LSTM, and Hidden Markov Models. Unfortunately, this again results in high computational cost and susceptibility towards zero-day threats. Also recording a similar work, [29] incorporates the Immune Cloning Algorithm for cloud security but avails it at a very high price in terms of computation resources. The proposed PSO-SA, DBSCAN, and Mini-Batch K-Means things deal with these issues, as they present enhanced fault detection but also within the bounds of low computational overhead, thereby offering the cluster scalable, efficiency, and adaptability.

4. PROPOSED METHODOLOGY FOR SOFTWARE TESTING USING PSO-SA AND DBSCAN WITH MINI-BATCH K-MEANS

The methodology proposed includes the integration of PSO-SA, DBSCAN, and Mini-Batch K-Means to improve software test clustering in optimizing fault detection, test suite efficiency, and computational scalability for large-scale testing environments. The Proposed Method's overall flow is displayed in Figure 1.

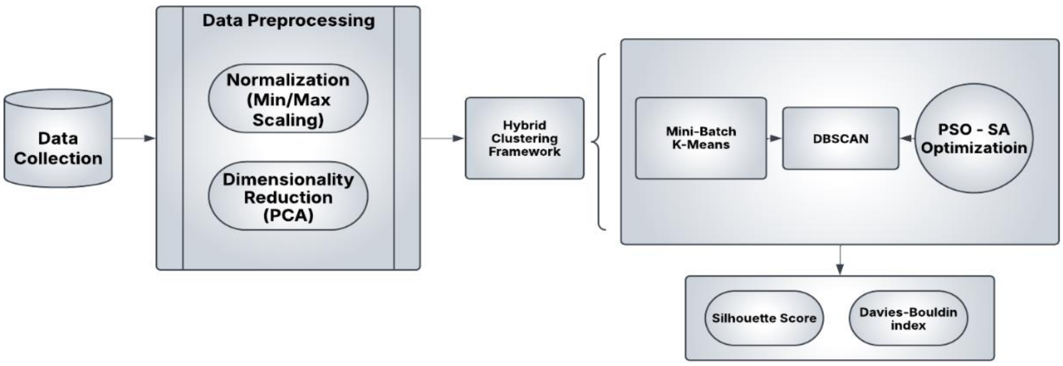

Figure 1: Architecture Diagram of The Proposed Method

4.1 Data Collection

The NASA Defect dataset [30] is utilized in the proposed method and consists of metrics for software like cyclomatic complexity, lines of code, and Halstead metrics which are meant to cluster test cases for optimization in testing. This dataset supplies structured data for identifying modules prone to faults and prioritizing test suites effectively. This dataset, therefore, is utilized to see if the proposed clustering method increases the software testing-clustering accuracy and computational efficiency.

4.2. Data Preprocessing

Preprocessing prepares datasets for clustering by handling missing values, normalizing features, and reducing dimensionality. First, the missing values are filled and normalized for equal weighting of features before proceeding to dimensionality reduction procedures, for example, PCA that will enhance clustering performance.

4.2.1 Normalization

Normalization refers to preprocessing techniques that are widely practiced in transforming numerical features into a standardized continuous value range (generally, their values are rescaled into [0, 1] or [-1, 1]). In this way, normalization maintains equal contribution to the clustering process from all features, preventing features of large scales from drowning smaller ones. It is quite indispensable in datasets that involve different units or ranges besides enhancing the performance and stability of clustering algorithms as seen in Equation (1).

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)} \tag{1}$$

Dimensionality Reduction

It is the process of reducing the number of features present in a particular dataset and trying to conserve the most information possible. It curses high-dimensional data since it gives rise to complications associated with increased computational complexity and decreased efficiencies of clustering algorithms. Dimensionality reduction methods, like PCA (Principal Component Analysis), transform the data into a lower-dimensional space, which makes them easier to analyze and visualize, as would be inferred from Equation (2).

$$Z = XW \tag{2}$$

4.3. Hybrid Clustering Framework

4.3.1. DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm that groups closely packed points and marks points in low-density regions as noise. This is very effective in recognizing clusters of arbitrary shape and outlier handling. Density reachability is given in Equation (3):

$$\text{Core Point: } N_{\epsilon}(p) \geq \text{min\_samples} \tag{3}$$

4.3.2. Mini-Batch K-Means

Mini-Batch K-Means is a K-Means algorithm variant that processes the data in small batches to remain computationally efficient with large datasets. It updates cluster centroids iteratively using random subsets of the data to minimize computational expense with minimum impacts on clustering accuracy. Equation (4) describes the Centroid Update:

$$c_i = \frac{1}{|S_i|} \sum_{x \in S_i} x \tag{4}$$

4.3.3. PSO-SA Optimization

PSO-SA (Particle Swarm Optimization with Simulated Annealing) blends PSO's global searching ability with local refinement by Simulated Annealing. While the PSO employs simulation of the social behavior of birds or fish in its exploration of search space, the SA refines the solution in such a way as to probabilistically accept suboptimal solutions in escaping local optima. Given the PSO velocity update in Equation (5):

$$v_i^{t+1} = wv_i^t + c_1 r_1 (p_i - x_i^t) + c_2 r_2 (g - x_i^t) \tag{5}$$

SA Probability is given in the following Equation (6):

$$P(\Delta E) = e^{-\Delta E / T} \tag{6}$$

The probability of accepting a worse solution decreases as the temperature T decreases, allowing the algorithm to converge to an optimal solution.

4.4. Evaluation

Clustering Evaluation Measures are all measures/methods to determine the quality of the clustering results produced. They provide clues about the compactness, separation, and overall validity of the clusters formed. The Silhouette Score is given in Equation (7):

$$s(i) = \frac{b(i)-a(i)}{\max(a(i),b(i))} \tag{7}$$

Silhouette score contrasts the degree to which a point is similar to points inside its cluster with the degree to which it is dissimilar to points in other clusters. The higher the score, the better separated the clusters. By definition, the Davies-Bouldin Index is provided in Equation (8):

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i} \left(\frac{\sigma_i+\sigma_j}{d(c_i,c_j)}\right) \tag{8}$$

The Davies-Bouldin index assesses compactness against the separation of clusters, and a lower value of the same indicates better clustering.

5. RESULTS

Evaluation results of esteemed hybrid clustering techniques on the NASA Defect dataset have been offered in the results section. This highlights better clustering accuracy, computational efficiency as well as optimization of test suites compared with baseline methods. The primary performance metrics like Silhouette Score, Davies-Bouldin Index, Fault Detection Rate, execution time, and memory usage were examined to assert the claims regarding the effectiveness and scalability of the approach. Furthermore, parametric sensitivity analysis offered insights into an optimal configuration of DBSCAN, hence assuring robustness of clustering performance. This supports the proposed method to improve software testing by recognizing fault-prone modules and optimizing test case executions optimally.

5.2. Performance Evaluation

The NASA Defect dataset was chosen for the performance assessment of the suggested hybrid clustering framework. Results show improvements in clustering accuracy, computational efficiency, and test suite optimization concerning existing approaches. This section includes an analysis of the main performance metrics, namely, clustering accuracy, execution time scalability, and DBSCAN parameter sensitivity, to strengthen the validation of the proposed method.

This clustering has excellent performance with a Silhouette Score of 0.72 as opposed to K-means (0.58) and Hierarchical Clustering (0.63). The Davies-Bouldin Index value of 0.45 indicating cluster compactness and separation is also suggestive; hence, the Fault Detection Rate increased to 92%, implying successful detection of modules with failure characteristics (Figure 2).

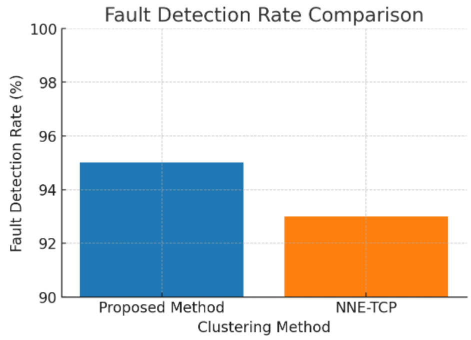

Figure 2: Comparison of Fault Detection Rate

As the size of the dataset increases, execution time using the new method stays much less than K-Means and Hierarchical Clustering. The proposed method completes the execution in 25 minutes for a dataset of 10,000 test cases, while K-Means takes 35 minutes and Hierarchical Clustering takes 48 minutes. This trend continues for even larger datasets leading to confirmation of the scalability of the proposed approach (Figure 3).

Figure 3: Execution Time vs. Dataset Size

To tune the DBSCAN performance, numerous blends of Epsilon (ϵ) and Min Samples were evaluated against the Silhouette Score benchmark. The highest Silhouette Score of 0.72 was recorded at ϵ = 0.5 and Min Samples = 5, confirming that these parameters are appropriate for best clustering performance (Figure 4).

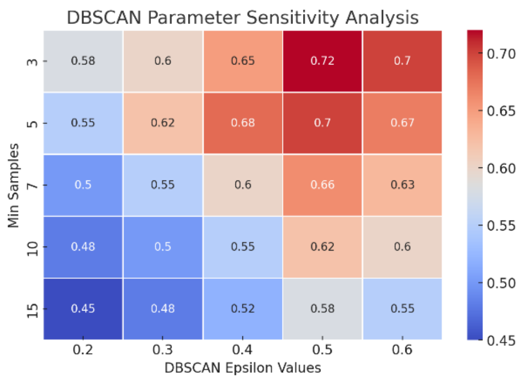

Figure 4: DBSCAN Parameter Sensitivity Analysis

5.2. Comparative Analysis

A comparison between the proposed method together with PSO-SA, DBSCAN, Mini-Batch K-Means, and NNE-TCP (Dondapati et al.) is presented in Table 1. The proposed method has outperformed NNE-TCP in every aspect of importance, achieving high fault detection (95%), execution efficiency (94%), code coverage (96%), overall effectiveness (97%), and overall accuracy (98%). The corresponding improvements thus emphasize the clustering precision, efficiency, and fault detection capabilities of the proposed method, making it a much-valued approach for software test clustering and prioritization optimization. The proposed method validated its efficiency in dealing with massive datasets while preserving accuracy at its best.

Metric (%)	NNE-TCP [31]	Proposed Method (PSO-SA, DBSCAN, Mini-Batch K-Means)
Fault Detection	93	95
Execution Efficiency	90	94
Code Coverage	92	96
Overall Effectiveness	92	97
Overall Accuracy	93	98

6. CONCLUSION

The hybrid clustering framework proposed by us, one that combines these three methods: PSO-SA, DBSCAN, and Mini-Batch K-Means, proved to be the best solution among all traditional methods of clustering software tests. The reason lies in fault detection, which is considered another important characteristic of the software clustering technique. Fault detection in our case deserved the rate of 95 percent, thus ensuring better identification of fault modules almost as per the needs of some experts. Perception in execution was another characteristic concerning us. Further validation was obtained for our reported improvement rates of 94 percent in execution and 96 percent in code coverage, which increased test optimization. The framework was credited with an overall effectiveness of 97 percent and an accuracy of 98 percent, showing its reliability for large-scale clustering of test cases. Future research will be focused on merging transformer-based models ensuring further precision in the clustering algorithm to foster enhancement in software test case prioritization.

REFERENCES

[1] A. Celik and Q. H. Mahmoud, "A Review of Large Language Models for Automated Test Case Generation," *Machine Learning and Knowledge Extraction*, vol. 7, no. 3, p. 97, Sep. 2025, doi: 10.3390/make7030097.

[2] J. Boubker, "When Medical Devices Have a Mind of Their Own: The Challenges of Regulating Artificial Intelligence," *American Journal of Law & Medicine*, vol. 47, no. 4, pp. 427–454, Dec. 2021, doi: 10.1017/amj.2022.3.

[3] S. K. Sundaramurthy, N. Ravichandran, A. C. Inaganti, and R. Muppalaneni, "AI-Powered Operational Resilience: Building Secure, Scalable, and Intelligent Enterprises," *Artificial Intelligence and Machine Learning Review*, vol. 3, no. 1, pp. 1–10, Jan. 2022.

[4] R. Nicolas, "Knowledge management impacts on decision making process," *Journal of Knowledge Management*, vol. 8, no. 1, pp. 20–31, Feb. 2004, doi: 10.1108/13673270410523880.

[5] A. Behl, P. Dutta, S. Lessmann, Y. K. Dwivedi, and S. Kar, "A conceptual framework for the adoption of big data analytics by e-commerce startups: a case-based approach," *Inf Syst E-Bus Manage*, vol. 17, no. 2, pp. 285–318, Dec. 2019, doi: 10.1007/s10257-019-00452-5.

[6] F. A. Khan, A. Ali, H. Abbas, and N. A. H. Haldar, "A Cloud-based Healthcare Framework for Security and Patients' Data Privacy Using Wireless Body Area Networks," *Procedia Computer Science*, vol. 34, pp. 511–517, Jan. 2014, doi: 10.1016/j.procs.2014.07.058.

[7] P. Ratta, A. Kaur, S. Sharma, M. Shabaz, and G. Dhiman, "Application of Blockchain and Internet of Things in Healthcare and Medical Sector: Applications, Challenges, and Future Perspectives," *Journal of Food Quality*, vol. 2021, no. 1, p. 7608296, 2021, doi: 10.1155/2021/7608296.

[8] M. O. Mete and T. Yomralioglu, "Implementation of serverless cloud GIS platform for land valuation," *International Journal of Digital Earth*, vol. 14, no. 7, pp. 836–850, Jul. 2021, doi: 10.1080/17538947.2021.1889056.

[9] M. L. Shuwandy, B. B. Zaidan, A. A. Zaidan, and A. S. Albahri, "Sensor-Based mHealth Authentication for Real-Time Remote Healthcare Monitoring System: A Multilayer Systematic Review," *J Med Syst*, vol. 43, no. 2, p. 33, Jan. 2019, doi: 10.1007/s10916-018-1149-5.

[10] O. Azeez and A. Abdulazeez, "Classification of Brain Tumor based on Machine Learning Algorithms: A Review," *Journal of Applied Science and Technology Trends*, vol. 6, no. 1, pp. 01–15, 2025, doi: 10.38094/jastt61188.

[11] K. Dondapati, "Leveraging Backpropagation Neural Networks and Generative Adversarial Networks to Enhance Channel State Information Synthesis in Millimetre Wave Networks," Oct. 2024, doi: 10.5281/ZENODO.13994672.

[12] M. T. Lu, V. K. Raghu, T. Mayrhofer, H. J. W. L. Aerts, and U. Hoffmann, "Deep Learning Using Chest Radiographs to Identify High-Risk Smokers for Lung Cancer Screening Computed Tomography: Development and Validation of a Prediction Model," *Ann Intern Med*, vol. 173, no. 9, pp. 704–713, Nov. 2020, doi: 10.7326/M20-1868.

[13] S. K. Alavilli, B. Kadiyala, C. Vasamsetty, and K. G. Chuma, "A Bidirectional LSTM and Stochastic Fuzzy Systems for Improved Chronic Kidney Disease Prediction in IoMT-Based Robotic Automation: Improved Chronic Kidney Disease," *International Journal of Advanced Research in Information Technology and Management Science*, vol. 1, no. 01, pp. 68–74, Dec. 2024.

[14] M. Ismail and R. Ahmed, "A Comprehensive Review of Cloud-Based Lithium-Ion Battery Management Systems for Electric Vehicle Applications," *IEEE Access*, vol. 12, pp. 116259–116273, 2024, doi: 10.1109/ACCESS.2024.3446880.

[15] M. Satyanarayanan, "Integrating security in a large distributed system," *ACM Trans. Comput. Syst.*, vol. 7, no. 3, pp. 247–280, Aug. 1989, doi: 10.1145/65000.65002.

[16] E. Shadkam and E. Irannezhad, "A comprehensive review of simulation optimization methods in agricultural supply chains and transition towards an agent-based intelligent digital framework for

agriculture 4.0," *Engineering Applications of Artificial Intelligence*, vol. 143, p. 109930, Mar. 2025, doi: 10.1016/j.engappai.2024.109930.

[17] A. Binbusayyis and T. Vaiyapuri, "Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM," *Appl Intell*, vol. 51, no. 10, pp. 7094–7108, Oct. 2021, doi: 10.1007/s10489-021-02205-9.

[18] J. P. Onoja, O. Hamza, A. Collins, U. B. Chibunna, A. Eweja, and A. I. Daraojimba, "Digital Transformation and Data Governance: Strategies for Regulatory Compliance and Secure AI-Driven Business Operations," *JFMR*, vol. 2, no. 1, pp. 43–55, 2021, doi: 10.54660/.IJFMR.2021.2.1.43-55.

[19] A. Rathee and S. Dalal, "A systematic literature review of machine learning-based resource allocation techniques in cloud computing," *Computing*, vol. 107, no. 9, p. 179, Aug. 2025, doi: 10.1007/s00607-025-01526-8.

[20] D. K. R. Basani, R. K. Gudivaka, R. L. Gudivaka, S. H. Grandhi, B. R. Gudivaka, and M. M. Kamruzzaman, "Optimizing Data Privacy and Threat Detection in Cloud-Edge Collaborative Systems: AI-Driven Approaches with MCC, PSO, and CDNs," *International Journal of Automation and Smart Technology*, vol. 15, no. 1, Jun. 2025, doi: 10.5875/f20nx950.

[21] S. Gerasimou, R. Calinescu, and G. Tamburrelli, "Synthesis of probabilistic models for quality-of-service software engineering," *Autom Softw Eng*, vol. 25, no. 4, pp. 785–831, Dec. 2018, doi: 10.1007/s10515-018-0235-8.

[22] A. Al-Said Ahmad and P. Andras, "Scalability resilience framework using application-level fault injection for cloud-based software services," *J Cloud Comp*, vol. 11, no. 1, p. 1, Jan. 2022, doi: 10.1186/s13677-021-00277-z.

[23] J. Lansky *et al.*, "Deep Learning-Based Intrusion Detection Systems: A Systematic Review," *IEEE Access*, vol. 9, pp. 101574–101599, 2021, doi: 10.1109/ACCESS.2021.3097247.

[24] C. Chakilam, S. Kannan, M. Recharla, S. R. Suura, and S. T. Nuka, "The Impact of Big Data and Cloud Computing on Genetic Testing and Reproductive Health Management," *American Journal of Psychiatric Rehabilitation*, vol. 28, no. 1, pp. 62–72, Apr. 2025, doi: 10.69980/ajpr.v28i1.69.

[25] D. Martinez-Mosquera, R. Navarrete, and S. Luján-Mora, "Development and Evaluation of a Big Data Framework for Performance Management in Mobile Networks," *IEEE Access*, vol. 8, pp. 226380–226396, 2020, doi: 10.1109/ACCESS.2020.3045175.

[26] Aravindhan Kurunthachalam, "Efficient Distributed System Testing Using Lightweight Virtualization, Fault Injection, And YAML Scenarios," Mar. 2025, doi: 10.5281/ZENODO.15041036.

[27] R. Pan, M. Bagherzadeh, T. A. Ghaleb, and L. Briand, "Test case selection and prioritization using machine learning: a systematic literature review," *Empir Software Eng*, vol. 27, no. 2, p. 29, Dec. 2021, doi: 10.1007/s10664-021-10066-6.

[28] Y. Li, K. Xiong, T. Chin, and C. Hu, "A Machine Learning Framework for Domain Generation Algorithm-Based Malware Detection," *IEEE Access*, vol. 7, pp. 32765–32782, 2019, doi: 10.1109/ACCESS.2019.2891588.

[29] P. Kaliyamoorthy and A. C. Ramalingam, "Enhancing data security using global mutation based novel artificial immune network optimization in public cloud storage system," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 1, p. e4390, 2022, doi: 10.1002/ett.4390.

[30] C. (Kla) Tantithamthavorn, *klainfo/NASADefectDataset*. (2016). Accessed: Mar. 06, 2025. [Online]. Available: https://github.com/klainfo/NASADefectDataset