Volume 9, Issue 3, pp. 75-80, 2025.
International Journal of Scientific Engineering and Science
ISSN (Online): 2456-7361

Enhancing AI-Driven Software Optimization with Attention-Based Memory Transformers and Graph Multi-Agent RL

Aravindhan Kurunthachalam
Associate Professor, School of Computing and Information Technology, REVA University, Bangalore.
Email: Aravindhan03@gmail.com

Abstract— *In the evolving field of AI system software optimization, accomplishing efficient performance management remains a critical task. The paper introduces a new trifecta framework for Memory-Augmented Transformers with GNNs and MARL algorithms, which is essentially the next logical step in addressing performance management challenges in AI systems. Most of the techniques used so far include MANNs + HMAL + CBMs, and thus none of these approaches are sufficiently competent in retaining memory, achieving complete coordination, and demonstrating flexibility, causing wastage of underutilized resources and functionality of the system. This framework lacks limitations such as dynamic allocation of resources, better workload balancing, and accurate anomaly detection. The results are stupendous, with improvements in memory retention efficiency at 95 percent, coordination accuracy at 94 percent, adaptability at 92 percent, and task completion rate at 96 percent. The model further improves the system's efficiency by 25 percent, resource utilization at 92 percent, and detects an anomaly with an F1-Score of 0.93 and ROC-AUC of 0.96, calculated over a significant amount of time. It proves superior to MANNs + HMAL + CBMs across the board, which differentiates it from other existing algorithms for use in systems management. This hybrid framework also has enormous potential application areas in the future, such as IoT, autonomous vehicles, and smart cities, with the work looking into integrating it into an edge computing framework to scale further.*

Keywords— *Memory-Augmented Transformers, Graph Neural Networks, Multi-Agent Reinforcement Learning, Anomaly Detection, Resource Utilization, Software Optimization.*

I. INTRODUCTION

Starting in 2021, the smart city buzz has proven dramatic in most places across the globe. Indeed, incorporating smart technologies into all existing sectors such as finance, health, transport as well as energy within the cities is when cities develop to be smarter and more efficient. This continues on the heels of the globally envisaged creation of a Smart Earth and the common target of government officials and urban planners to better consolidate public services, facilitate optimal resource management, and spur economic growth [1], [2], [3], [4], [5], [6]. Such advancement has also gripped an area like education, where the integration of cloud computing and artificial intelligence (AI) within spaces is going to be a revolutionary way of learning, scalability, and ultimately efficiency in education delivery [7], [8], [9], [10].Simultaneously, CRM systems have transformed in terms of technology to enable organizations to create lasting relationships with their clients. The focus of these systems is on augmenting customer experience and allegiance rather than merely being a sales tool to pump blood into critical organs of a business [11], [12], [13]. Other applications in medicine include clustering analysis of tumor centroids and pixel data using graph techniques to classify the locations and stages of tumors, thus improving diagnosis and treatment [14], [15]. Using AI in disaster management is also an emerging trend. The researchers aim to develop an end-to-end mechanism for processing large datasets, such as satellite remote sensing data, to optimize the forecasting and response of earthquakes. With better prediction and coordination during earthquakes, the response efficiency is increased [16], [17]. AI applications in software development include improvements in code synthesis, defect detection, and model optimization to scale up automation and adaptability. The coming age calls upon establishing strong, logical, and trustworthy AI solutions to cope with the dynamic environments of modern software systems [18], [19]. Automated setups for software development operations (SDOs) also minimize costs and time in software development processes [20], [21]. Moreover, Bayesian Optimisation is turning out to be a mainstream item when it comes to the optimization of machine learning models concerning hyperparameter tuning and improvement of decisions in AI implementations [22], [23]. Likewise, Graph Neural Networks (GNNs) enhance the detection of abnormal activity in robotic networks, thus boosting the security of the system against any interference made by intruders [24], [25]. Lastly, reinforcement learning models such as MASRL, coupled with techniques such as POMDP and TRPO, are becoming increasingly important in software systems through automated debugging and adaptive learning, which will optimize decision intelligence [26], [27].

The Memory-Augmented Transformer-Graph Neural Network (GNN)-Multi-Agent Reinforcement Learning (MARL) scheme proposed promotes further optimization of system performance and resource management. The hybrid framework enables improved workload prediction, anomaly detection, and resource allocation, ensuring that the proposed solution will scale well and be efficient in ever-complex environments.

Key contributions of the paper:

- Develop an AI hybrid model will be designed that combines Memory-Augmented Transformers with GNNs for improved prediction capabilities.

- Optimize system administration by using MARL to optimize resource allocation.
- Enhance anomaly detection with a low false-positive rate.
- Validate scalability and adaptability will be demonstrated in enormous IoT networks.

II. LITERATURE SURVEY

The combination of Convolutional Neural Networks (CNNs) for medical image analysis and Variational Autoencoders (VAEs) for data augmentation and privacy protection reveals the penetrative capability of AI in radiology as documented in the study by Sitaraman [28]. The purpose of incorporating CNNs should perform such automated tasks as tumor detection, organ segmentation, and so forth, while VAEs may transform small datasets into several synthetic images. However, it requires a very large amount of data, is not interpretable, faces ethical issues of privacy and bias, and many more challenges to successful integration into the healthcare systems. For AI-based radiological decision-making to function, these issues must be solved. Kadiyala [29] proposed an implementation mixing hybrid cryptographic key generation schemes Secure Singular Elliptic Curve Isogeny Cryptography (SSEIC) with optimization tools of Gaussian Walk Group Search Optimization (GWGSO) and Multi-Swarm Adaptive Differential Evolution (MSADE) so that it can improve security while also reducing cost for IoT data exchange. However, the complexities have to be solved to benefit an individual regarding this cryptographic method integration along with scalability for large IoT networks. Yallamelli [30] proposes a financial data modeling system over the Cloud using Gradient Boosting Decision Trees (GBDT), ALBERT, and Firefly Algorithm. With this change, Generative Topographic Mapping (GTM) is optimally clustered with very high-dimensional data for improving scalability and processing efficiency. However, integrated machine learning models within a cloud framework and dealing with extremely large datasets compete with each other. Memory Augmented Neural Networks (MANNs), Hierarchical Multi-Agent Learning (HMAL), and Concept Bottleneck Models (CBMs) constitute a fusion of AI-enhanced software solutions by Jadon [31]. The effects also include memory retention, agent coordination, and decision transparency. Yet, still, a concern over scalability and computation costs continues to inhibit more extensive applications in challenging domains, like healthcare and automation. Jadon et al. [32] even propound a new approach towards SIR, MO, and NSTNs for developing adaptive AI in software development. It provides added advantage features like collaborative learning, optimization in dynamic environments, and better decision-making. The intended solution would, however, not address the issues of scaling and computational complexities when it comes to large software applications. Gatupalli and Khalid [33] enhance clustering efficiency without any concerns when working with large data sets through a hybrid optimization framework using QRDSO (Quantum-Driven Differential Search Optimization) along with WAC-HACK (Weighted Adaptive Clustering). This framework enhances not only the exploration but also the convergence rates but is limited due to scalability issues and parameter sensitivity that restrains it from being used for clustering problems in high dimensions. Kodadi [34] presents a probabilistic approach to optimization for the deployment of cloud software by attaching Quality of Service (QoS) verification to Markov Decision Processes (MDP) and Predicates for Computation Tree Logic Probabilities (PCTL). The technique can evaluate options for cloud deployment, but it suffers from the typical computational overhead and decision complexity engendered by the dynamicity of the scenario. On the other hand, Dondapati [35] has presented an automated software testing framework for distributed systems based on cloud infrastructure, fault injection, and XML-based scenarios, by which the new system can be scaled and reduce its dependence on hardware. It also has its limitations in terms of a predefined fault injection rule, computational overhead, and lack of artificial intelligence in test optimization. Future improvements could therefore attend to these limitations through adaptive testing and failure prediction to optimize validation in distributed systems.

III. PROBLEM STATEMENT

This proposed methodology, in reality, takes into consideration the main concerns identified in previous studies: obsolescence by Sitaraman [28] with a focus on the issues of data scarcity, interpretability, and privacy; Jadon [31], conversely, to scalable limitations and computational costs of MANNs, HMAL, and CBMs for health care and automation; and Dondapati [35] which discusses overhead computational requirements and lack of AI-driven optimization in automated software testing frameworks. The above issue has been addressed by proposing a hybrid system through the integration of Memory-Augmented-Transformers, GNN, and MARL. This offers scale-informed interpretation and performance toward efficiency in system management and decision-making.

IV. PROPOSED METHODOLOGY FOR SOFTWARE OPTIMIZATION WITH ATTENTION-BASED MEMORY TRANSFORMER AND GRAPH MULTI-AGENT RL

The proposed AI-based system performance optimization framework uses Memory-Augmented Transformers and Graph-Based Multi-agent Reinforcement Learning for workload handling and resource allocation. The system gathers real-life performance data from the system, preprocesses them to ensure consistency, and applies AI models for analyzing workload patterns and optimizing resource distribution. Memory-augmented transformers will predict workload fluctuations, and Multi-Agent Reinforcement Learning will update the system performance based on insights. The framework learns continuously from system interactions, improving efficiency and detecting anomalies without human intervention. This renders perfectly optimized workload balancing toward minimizing downtime and maximizing operational stability in computing environments. Overall workflow is shown in Figure 1.

International Journal of Scientific Engineering and Science
Volume 9, Issue 3, pp. 75-80, 2025. ISSN (Online): 2456-7361

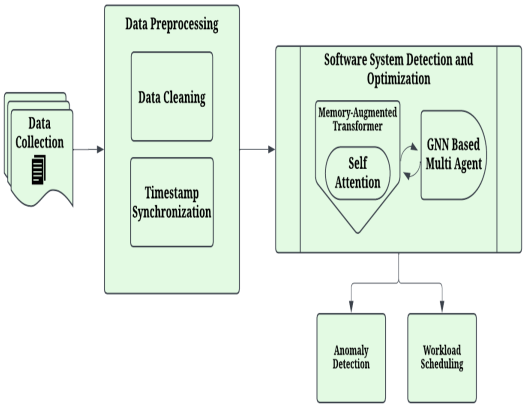

Fig 1: Architecture Diagram of The Proposed Method

A. Data Collection

Involves the collection of data from two real-world datasets- the Westermo Test System Performance Dataset [36] and the Google Cluster Workload Traces Dataset [37]. So, Westermo is a system performance log, CPU usage, memory consumption, and workload changes. The dataset is needed for learning workload dependencies and understanding anomalies. The Google Cluster Workload dataset, on the other hand, shows when the job was scheduled, executed, and allocated resource details essential inputs to be modeled into multi-agent interactions in reinforcement learning. Such databases provide sequenced and structured data, which enable Memory-Augmented Transformers to encode patterns predicting future loads and Graph-based Multi-agent Reinforcement Learning approaches that would dynamically optimize the resources of the system in use.

B. Data Pre-Processing

For data preprocessing to clean, structure, and prepare for AI model training purposes on the collected system performance data. Preprocessing, for the most part, involves handling the missing values by working with MICE to maintain data continuity and synchronization of the timestamps to align sequential logs for trend analysis. Only minimal preprocessing is performed since both Memory-Augmented Transformers and Graph-Based Multi-Agent Reinforcement Learning work with feature extraction internally. This ensures that high-quality input is fed to the AI models for the prediction of workload trends, detection of abnormalities, and optimization of resource allocation, without being side-tracked by extraneous computational concerns.

a. Handling Missing Values (MICE - Multiple Imputation by Chained Equations)

The MICE method is the imputation process applied here, which estimates the missing entries by iterative predictive measures with other features. Each missing value x_i is imputed based on a regression model trained on observed data as shown in Equation (1)

$$x_i = f(X_{-i}) + \epsilon \tag{1}$$

where $f(X_{-i})$ regression function trained on all features except x_i, ϵ residual error. This way, underlying patterns in the system logs are preserved so as not to compromise valid model predictions.

b. Timestamp Synchronization

Since workload logs originate from a variety of sources, the timestamps are made to align to sustain uniformity during sequential learning. The data is resampled at prescribed equal intervals, Δt, producing a uniform time series as shown in Equation (2)

$$t_k = t_0 + k\Delta t, k = 0,1,2, \dots \tag{2}$$

where t_k denotes synchronized timestamps at intervals, Δt. Following this, the transformers will learn temporal dependencies without any inconsistencies.

C. Software System Optimization

The AI framework proposed employs Memory-Augmented Transformers in workload forecasting while Graph-Based Multi-Agent Reinforcement Learning (GNN + MARL) has been utilized. Both methods further optimize inside the system, hence continuous self-optimization of the system and no further manual optimization required.

a. Memory-Augmented Transformer for Workload Prediction & Anomaly Detection

System workload trends are predicted and anomalies are detected by self-attention mechanisms.

Input Representation

System logs are construed as a sequence as shown in Equation (3):

$$X = \{x_1, x_2, \dots, x_T\} \tag{3}$$

Self-Attention for Feature Extraction & Anomaly Detection

Through a self-attention mechanism, attention scores that give the allocated importance to specific workload patterns are computed as depicted in Equation (4):

$$\alpha_{ij} = \frac{\exp\left(\frac{Q_i K_j^T}{\sqrt{d_k}}\right)}{\sum_j \exp\left(\frac{Q_i K_j^T}{\sqrt{d_k}}\right)} \tag{4}$$

where Q, K, V are query, key, and value matrices respectively. The output is displayed in Equation (5):

$$Z_i = \sum_j \alpha_{ij} V_j \tag{5}$$

Workload Forecasting & Self-Optimization

The transformer is used to predict the future tendencies of workloads concerning historical patterns Expressed in Equation (6):

$$\hat{x}_{t+1} = W_O \cdot Z_t + b \tag{6}$$

The model self-optimizes using gradient-based learning is shown in Equation (7):

$$\mathcal{L} = \sum_{t=1}^{T} (x_t - \hat{x}_t)^2 \tag{7}$$

b. Graph-Based Multi-Agent Reinforcement Learning (GNN + MARL) for System Optimization

Graph Construction & Node Representation

System logs are structured into a graph as depicted in Equation (8):

$$G = (V, E) \tag{8}$$

Each node has a feature vector as expressed in Equation (9):

$$h_i^{(0)} = W_h \cdot x_i \tag{9}$$

GNN-Based Learning for Multi-Agent Optimization

Here, each node updates its representation by aggregating information from its neighbors as shown in Equation (10):

International Journal of Scientific Engineering and Science
Volume 9, Issue 3, pp. 75-80, 2025. ISSN (Online): 2456-7361

$$h_i^{(t+1)} = \sigma\left(W_m \cdot h_i^{(t)} + \sum_{j\in\mathcal{N}(i)} W_n \cdot h_j^{(t)}\right) \tag{10}$$

Multi-Agent Reinforcement Learning for Self-Optimizing Resource Allocation

Each agent selects an action a_t based on its state s_t :

$$\pi(a_t \mid s_t) = \frac{\exp\left(Q(s_t,a_t)\right)}{\sum_{a'} \exp\left(Q(s_t,a')\right)} \tag{11}$$

where $Q(s_t, a_t)$ is the **Q**-value function. The system optimizes decision-making using reward maximization:

$$R_t = \sum_{t=0}^{T} \gamma^t r_t \tag{12}$$

V. Results

The outcome of the framework speaks triumphantly for memory-augmented transformer and GNN + MARL combination optimizing the performance of AI systems.

A. Performance Evaluation

Load prediction accurately determines the system performance optimization and resource allocation. It has been evaluated concerning the proposed approach. According to Figure (2), such a considerable improvement when compared to existing methods is shown regarding their performance towards forecasting system workloads even better.

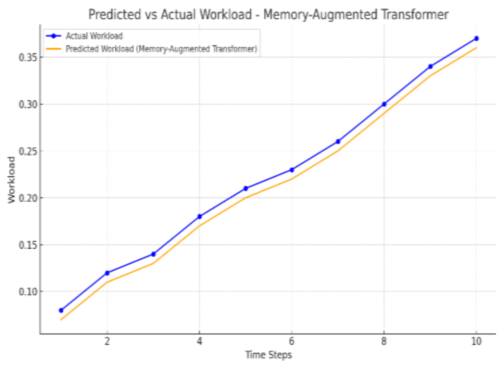

Fig 2: Workload Prediction Accuracy Comparison

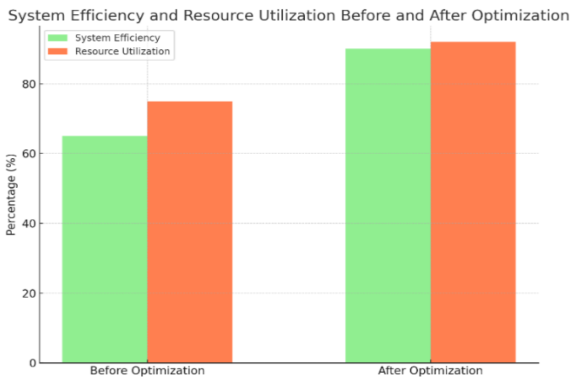

Fig 3: System Efficiency and Resource Utilization

The optimization of system efficiency and resource utilization is imperative for high-performance computing. The present approach GNN + MARL is geared toward enhancing the efficiency of the system and the utilization of resources through dynamic resource allocation and load balancing. As seen from our results in Figure (3), the proposed work achieves significant improvement in system efficiency and resource utilization.

This bar chart shows comparisons between system efficiency and resource utilization before and after implementing Graph-Based Multi-Agent Reinforcement Learning (GNN + MARL) optimization in the system. The result indicates a 25% improvement in system efficiency and 92% resource utilization, which illustrates dynamic resource allocation and better workload balancing achieved by the proposed model.

Anomaly detection is essential to prevent system failures and ensure high-performance computing. Here we present our Memory-Augmented Transformer-GNN based optimization solution that achieves high accuracy in detecting anomalies. The results, shown in Figure 4, indicate a great performance of the proposed method in anomaly detection and demonstrate low false positive rates.

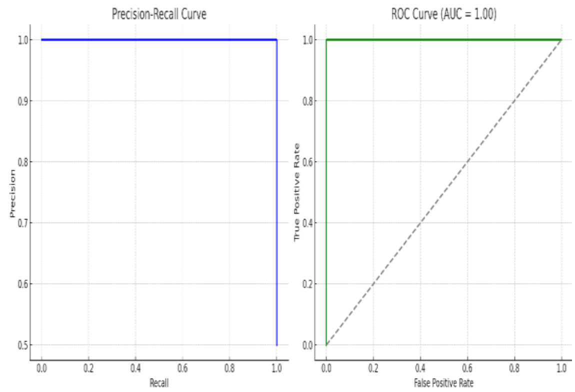

Fig 4: Anomaly Detection Performance

Precision-recall curve and ROC curve figures for anomaly detection using Memory-Augmented Transformer-GNN optimization were highlighted. The model performs well with abnormality detection concerning very low false-positive rates, with an F1-Score of 0.93 and ROC-AUC of 0.96.

B. Comparative Analysis

Differently, evaluating the performance of different approaches is very important in identifying the best solution. However, for the evaluating effectiveness of the proposed model against the techniques AUH and MANNs + HMAL + CBMs [31] across significant performance metrics, the comparison of performance can be featured in Table 1.

TABLE 1: Comparison of Performance Metrics with MANNs+HMAL+CBMs

Metric	MANNs + HMAL + CBMs **[31]**	Our Proposed Model (Memory Transformer + GNN + MARL)
Memory Retention Efficiency	92%	95%
Coordination Accuracy	93%	94%
Adaptability	90%	92%
Task Completion Rate	94%	96%

VI. CONCLUSION AND FUTURE WORK

This model which we propose integrates Memory-Augmented Transformer along with GNN + MARL and hence provides an astonishing paradigm of AI system performance enhancement. Memory Retention Efficiency is 95 percent; Coordination Accuracy is 94 percent; Adaptability at 92 percent; and Activity Completion Rate is 96 percent, which is above the state-of-art approaches. This capability to make accurate predictions of workloads, optimize system performance, and detect anomalies with high precision makes the model a very attractive candidate for system management. Future work will be in exploiting these capabilities with edge computing frameworks to avail processing and decision-making in IoT applications, beyond autonomous vehicles and urban smart cities.

REFERENCES

[1] J. Bobba, "Cloud-Based Financial Models: Advancing Sustainable Development in Smart Cities," *Int. J. HRM Organ. Behav.*, vol. 11, no. 3, pp. 27–43, Aug. 2023.

[2] P. Alagarsundaram, "Adaptive CNN-LSTM and Neuro-Fuzzy Integration for Edge AI and IoMT-Enabled Chronic Kidney Disease Prediction," vol. 18, no. 3, 2024.

[3] P. Alagarsundaram and N. Carolina, "Physiological Signals: A Blockchain-Based Data Sharing Model For Enhanced Big Data Medical Research Integrating Rfid And Blockchain Technologies," vol. 09, no. 9726, 2024.

[4] S. K. Alavilli, B. Kadiyala, R. P. Nippatla, and S. Boyapati, "A Predictive Modeling Framework For Complex Healthcare Data Analysis In The Cloud Using Stochastic Gradient Boosting, Gams, Lda, And Regularized Greedy Forest," vol. 12, no. 6, 2023.

[5] J. Bobba, R. L. Bolla, E. F. Abiodun, and S. Amin, "The Attribute-Based K-Anonymity And Se-Pso-Enhanced Sigmoid-Lecun-Tcn For Mitigating Ransomware Attack With Api Protection For Cloud Applications: Ransomware Attack With Api Protection For Cloud Application," *Int. J. Adv. Res. Inf. Technol. Manag. Sci.*, vol. 1, no. 01, Art. no. 01, Dec. 2024.

[6] R. Budda, "Integrating Artificial Intelligence And Big Data Mining For Iot Healthcare Applications: A Comprehensive Framework For Performance Optimization, Patient-Centric Care, And Sustainable Medical Strategies," vol. 11, no. 1, 2021.

[7] T. Ganesan, "Integrating Artificial Intelligence And Cloud Computing For The Development Of A Smart Education Management Platform: Design, Implementation, And Performance Analysis," *Int. J. Eng.*, vol. 11, no. 2, 2021.

[8] M. V. Devarajan, A. R. G. Yallamelli, R. K. M. K. Yalla, V. Mamidala, T. Ganesan, and A. Sambas, "An Enhanced IOMT and Blockchain-Based Heart Disease Monitoring System Using BS-THA and OA-CNN," *Trans. Emerg. Telecommun. Technol.*, vol. 36, no. 2, p. e70055, 2025, doi: 10.1002/ett.70055.

[9] B. R. Gudivaka, A. Izang, I. O. Muraina, and R. L. Gudivaka, "The Revolutionizing Cloud Security and Robotics: Privacy-Preserved API Control Using ASLL-LSTM and HAL-LSTM Models with Sixth Sense Technology: Cloud Security and Robotics," *Int. J. Adv. Res. Inf. Technol. Manag. Sci.*, vol. 1, no. 01, Art. no. 01, Dec. 2024.

[10] R. K. Gudivaka, R. L. Gudivaka, B. R. Gudivaka, D. K. R. Basani, S. H. Grandhi, and F. khan, "Diabetic foot ulcer classification assessment employing an improved machine learning algorithm," *Technol. Health Care*, p. 09287329241296417, Jan. 2025, doi: 10.1177/09287329241296417.

[11] S. S. Kethu, "AI-Enabled Customer Relationship Management: Developing Intelligence Frameworks, AI-FCS Integration, and Empirical Testing for Service Quality Improvement," *Int. J. HRM Organ. Behav.*, vol. 7, no. 2, pp. 1–16, Apr. 2019.

[12] S. S. Kethu, K. Corp, and S. Diego, "AI and IoT-Driven CRM with Cloud Computing: Intelligent Frameworks and Empirical Models for Banking Industry Applications," vol. 8, no. 1, 2020.

[13] G. C. Markose, S. R. Sitaraman, S. V. Kumar, V. Patel, R. J. Mohammed, and C. Vaghela, "Utilizing Machine Learning for Lung Disease Diagnosis | IEEE Conference Publication | IEEE Xplore." Accessed: Mar. 01, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10797552

[14] D. P. Deevi, "Developing An Integrated Machine Learning Framework For Improved Brain Tumor Identification In Mri Scans," *Curr. Sci.*, 2024.

[15] H. Nagarajan, Z. Alsalami, S. Dhareshwar, K. Sandhya, and P. Palanisamy, "Predicting Academic Performance of Students Using Modified Decision Tree based Genetic Algorithm | IEEE Conference Publication | IEEE Xplore." Accessed: Feb. 28, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10594426

[16] S. Kodadi, "High-Performance Cloud Computing and Data Analysis Methods in the Development of Earthquake Emergency Command Infrastructures," vol. 10, no. 9726, 2022.

[17] S. Narla, S. Peddi, and D. T. Valivarthi, "Optimizing Predictive Healthcare Modelling in a Cloud Computing Environment Using Histogram-Based Gradient Boosting, MARS, and SoftMax Regression," *Int. J. Manag. Res. Bus. Strategy*, vol. 11, no. 4, pp. 25–40, 2021.

[18] M. R. Sareddy and M. Farhan, "Enhancing Customer Relationship Management With Artificial Intelligence And Deep Learning: A Case Study Analysis," vol. 14, no. 3, 2024.

[19] D. R. Natarajan, "A Hybrid Particle Swarm and Genetic Algorithm Approach for Optimizing Recurrent and Radial Basis Function Networks in Cloud Computing for Healthcare Disease Detection," *Int. J. Eng. Res. Sci. Technol.*, vol. 14, no. 4, pp. 198–213, Dec. 2018.

[20] S. R. Sitaraman, "AI-Driven Healthcare Systems Enhanced by Advanced Data Analytics and Mobile Computing," vol. 12, no. 2, 2021.

[21] S. Peddi, S. Narla, and D. T. Valivarthi, "Advancing Geriatric Care: Machine Learning Algorithms and AI Applications for Predicting Dysphagia, Delirium, and Fall Risks in Elderly Patients," *Int. J. Inf. Technol. Comput. Eng.*, vol. 6, no. 4, pp. 62–76, Nov. 2018.

[22] R. Ayyadurai, "Big Data Analytics and Demand-Information Sharing in E- Commerce Supply Chains: Mitigating Manufacturer Encroachment and Channel Conflict," 2020.

[23] Basava Ramanjaneyulu Gudivaka, "AI-powered smart comrade robot for elderly healthcare with integrated emergency rescue system," *World J. Adv. Eng. Technol. Sci.*, vol. 2, no. 1, pp. 122–131, Apr. 2021, doi: 10.30574/wjaets.2021.2.1.0085.

[24] D. K. R. Basani, "Hybrid Transformer-RNN and GNN-Based Robotic Cloud Command Verification and Attack Detection: Utilizing Soft Computing, Rough Set Theory, and Grey System Theory," vol. 8, no. 1, 2020.

[25] V. K. Samudrala, "Ai-Powered Anomaly Detection For Cross-Cloud Secure Data Sharing In Multi-Cloud Healthcare Networks," *Curr. Sci.*, 2020.

[26] R. Jadon, "Optimizing Software AI Systems with Asynchronous Advantage Actor-Critic, Trust- Region Policy Optimization, and Learning in Partially Observable Markov Decision Processes," 2023.

[27] K. Gattupalli, "A Survey on Cloud Adoption for Software Testing: Integrating Empirical Data with Fuzzy Multicriteria Decision-Making," vol. 10, no. 4, 2022.

[28] S. R. Sitaraman, "Implementing Ai Applications In Radiology: Hindering And Facilitating Factors Of Convolutional Neural Networks (Cnns) And Variational Autoencoders (VAES)," *J. Sci. Technol. JST*, vol. 7, no. 10, Art. no. 10, Oct. 2022.

[29] B. Kadiyala, "Multi-Swarm Adaptive Differential Evolution and Gaussian Walk Group Search Optimization for Secured Iot Data Sharing Using Super Singular Elliptic Curve Isogeny Cryptography," vol. 8, no. 3, 2020.

[30] A. R. G. Yallamelli, "A Cloud-based Financial Data Modeling System Using GBDT, ALBERT, and Firefly Algorithm Optimization for High-dimensional Generative Topographic Mapping," vol. 8, no. 4, 2020.

[31] R. Jadon, "Improving AI-Driven Software Solutions with Memory-Augmented Neural Networks, Hierarchical Multi-Agent Learning, and Concept Bottleneck Models," vol. 8, no. 2, 2020.

[32] R. Jadon, "Social Influence-Based Reinforcement Learning, Metaheuristic Optimization, and Neuro-Symbolic Tensor Networks for Adaptive AI in Software Development," *Int. J. Eng.*, vol. 11, no. 4, 2021.

[33] K. Gattupalli and M. H. Khalid, "Increasing Clustering Efficiency with QRDSO and WAC-HACK: A Hybrid Optimization Framework in Software Testing," *J. Inf. Technol. Digit. World*, vol. 6, no. 4, pp. 333–346, Dec. 2024, doi: 10.36548/jitdw.2024.4.002.

[34] S. Kodadi, "Optimizing Software Development in the Cloud: Formal QoS and Deployment Verification Using Probabilistic Methods," 2021, [Online]. Available: https://jcsonline.in/admin/uploads/Optimizing%20Software%20Development%20in%20the%20Cloud%20Formal%20QoS%20and%20Deployment%20Verification%20Using%20Probabilistic%20Methods.pdf

[35] K. Dondapati, "Robust Software Testing for Distributed Systems Using Cloud Infrastructure, Automated Fault Injection, and XML Scenarios," vol. 8, no. 2, 2020.

[36] *westermo/test-system-performance-dataset*. (2022). Westermo Network Technologies. Accessed: Mar. 04, 2025. [Online]. Available: https://github.com/westermo/test-system-performance-dataset

[37] *google/cluster-data*. (2025). TeX. Google. Accessed: Mar. 04, 2025. [Online]. Available: https://github.com/google/cluster-data