ISSN : 2350-0743

International Journal of Recent Advances in Multidisciplinary Research

Vol. 12, Issue 03 pp.10882-10887, March, 2025

RESEARCH ARTICLE

HYBRID MODELING OF SOFTWARE BEHAVIOR WITH ANN AND ACO FOR EFFECTIVE QOS OPTIMIZATION

Koteswararao Dondapati [1], Himabindu Chetlapalli[2], Sharadha Kodadi [3], Durga Praveen Deevi [4], Naga Sushma Allur [5] and Aravindhan Kurunthachalam[6,*]

[1]Everest Technologies, Ohio, USA; [2]9455868 Canada Inc, Ontario, Canada; [3]Infosys, Texas, USA; [4]O2 Technologies Inc, California, USA; [5]Astute Solutions LLC, California, USA; [6]Associate Professor, School of Computing and Information Technology REVA University, Bangalore

ARTICLE INFO

Article History

Received 19th December, 2024
Received in revised form
17th January, 2025
Accepted 26th February, 2025
Published online 28th March, 2025

Keywords:

QoS Optimization, Artificial Neural Networks (ANN), Ant Colony Optimization (ACO), Resource Allocation, Distributed Computing, Performance Prediction.

Corresponding author:
Aravindhan Kurunthachalam

ABSTRACT

Maintenance of performance, reliability, and efficiency of any system within distributed computing environments is central to QoS. Conventional QoS optimization methods confront challenges regarding dynamic resource allocation, failure, and adaptation of systems. To overcome these limitations, a hybrid artificial neural network and Ant Colony Optimization model are proposed to provide an efficient QoS optimization strategy. The ANN part predicts possible degradation in QoS through parameters like CPU utilization, memory usage, network latency, and response time, while the ACO component dynamically optimizes resource allocation for better system performance. The proposed model considers a systematic workflow comprising data collection, pre-processing, feature extraction, QoS prediction, and ACO-based optimization. The system model training and evaluation are done using the Kaggle dataset Synthetic Log Data of Distributed Systems.As a complement to the handling of missing values with outlier treatment, the fact that Principal Component Analysis (PCA) can be used for feature selection is not omitted. The ANN model establishes the QoS importance trend, which is then optimized by ACO via pheromone-based learning and heuristic value adjustments for improved resource allocation. The experimental findings present the Hybrid ANN-ACO Model with better performance results compared to the existing QoS optimization approaches such as Rule-Based, Genetic Algorithm (GA), and Probabilistic Model Checking (PMC). In this context, the new proposed model saw a performance increase in accuracy to 96.2%, improvement to 59.4% response time, 93.1% CPU utilization efficiency, and 51.2% network latency performance.MSE is less than 0.029, meaning that we have a high level of accuracy for predicting QoS. This study illustrates the successful application of machine learning in combination with bio-inspired optimization for adaptive, scalable, and efficient QoS management in distributed computing environments.

Citation: Koteswararao Dondapati, Himabindu Chetlapalli, Sharadha Kodadi, Durga Praveen Deevi, Naga Sushma Allur, Aravindhan Kurunthachalam. 2025. "Hybrid Modeling of Software Behavior with ANN and ACO for Effective QoS Optimization", *International Journal of Recent Advances in Multidisciplinary Research,* 12,(03), 10882-10887.

INTRODUCTION

At the present time, software-quality-of-service (QOS) has become one of the most significant avenues for computing, pushing applications in the directions of performance, reliability, and availability (Dondapati, 2019). Enhanced software quality of service (QoS) through AI-based resilience testing turns into a complete failure management dynamic with minimized downtime and improved use of resources (Allur, 2020). The failure prediction in real time with adaptive optimization techniques for attaining peak performance and reliability, maintains the highest possible level of QoS in software (Deevi, 2020). AI resilience testing enhances quality of service by pro-actively identifying faults and optimizing the system performance under various conditions (Allur, 2019). Testing the system resilience with AI assists in improving the overall quality of service by generating weaknesses in the system, intelligent resource utilization, and optimal fault tolerance within complex computing environments (Dondapati, 2024).The robustness testing of the artificial intelligence intensifies the quality of service through the identification of weaknesses by being proactive and ensures that the system is adaptable to changes in workloads (Deevi, 2024). Approaching the application of AI-based optimization methods provides dynamic resource allocations, increased testing efficiency, and augmented system responsiveness under diversified workloads (Deevi, 2021). The implementation of QoS optimization driven by AI can be

computationally expensive and complex when it comes to system integration (Allur, 2025).The expansion of artificial intelligence models is accompanied by increased computing needs, which raises the overhead incurred by the systems and can also consume a big chunk of resources, thus affecting the efficiency of systems (Allur, 2020). Nudge theory, being integrated into software design, has the potential to improve user judgment in the case of privacy settings, there again sieving trust and security in financial applications (Kodadi, 2023). The Hybrid ANN-ACO Model improves QoS by utilizing ANN for its predictive analysis while ACO handles its dynamic resource allocation. With respect to key metrics, such as CPU usage, and response time, the ANN forecast performance faults, while ACO, according to that prediction, optimally distributes resources. As such, the hybrid model achieves improved accuracy, latency, and system efficiency and makes it scalable and adaptive for distributed computing environments.

Research Contribution

- QoS enhancement in a hybrid form, through a blend of ANN-ACO, for precise accuracy and efficiency.
- Maximizes computational performance by ensuring that computational resources are flexibly and effectively allocated to the upcoming computational needs.
- The AI-animated QoS management makes sure it provides continuous performance improvement with a feedback loop.

LITERATURE SURVEY

The techniques employed in this study include Neural Networks for predicting the most valuable test cases and Heuristic Methods for optimization of test cases in prioritization. These methods help improve Test Case Prioritization (TCP) by increasing fault detection and reducing resource consumption (Dondapati, 2020). Despite the effectiveness of the indicator detection at higher levels, it will acquire an additional computational burden when the detection is applied in real time and not be flexible as it might not adapt to newer phishing attacks. Moreover, sporadic false positives can hinder user confidence and reliability in the system (Allur, 2020). The research uses failure injection approaches by AWS instruments, for instance, AWS CloudWatch, AWS X-Ray, AWS Step Functions, AWS Lambda, and AWS Fault Injection Simulator (FIS) (Deevi, 2023). In this study, the cloud computing was used to provide on-demand and scalable test environments where technology made it flexible enough, and low in resources (Dondapati, 2020). Security factors such as multi-biometric key generation and dynamic metadata reconstruction bring along service quality requirements and contribute to the integrity and confidentiality of data (Deevi, 2020). In my own point of view, the limitations of the D-ESU-SEG model in software development parallel the problems arising from issues of scalability, adaptability, as well as external dependencies (Deevi, 2024). And with that comes computational overhead that can impact a large-scale deployment and optimization for such an environment. Future improvements should emphasize scalability, adapting the entire approach to some new paradigms, such as serverless computing and edge environments (Allur, 2021). There are no quantitative performance metrics for the Hybrid ANN-PSO Algorithm, making it impossible to assess its efficiency against other methods, although it is relatively new (Kodadi, 2021). Clinical follow-up is done live and in real-time, the risk assessment is done in real-time while patient compliance is measured (Chetlapalli, 2023). The integration of ANN and models of electrothermal inverters to simulate an EV in real-time through finite element analysis is consistent with the principles of software QoS, focusing on efficiency, reliability, and performance optimization (Deevi, 2020). This work tries to contribute to adding privacy and security to the multi-cloud by employing GARS, the Global Authentication Register System and other multidisciplinary security frameworks (Chetlapalli, 2021). It utilizes cloud computing, wavelet analysis, big data analytics, and machine learning techniques to develop age-old important software resilience, fault detection, and real-time optimizations (Kodadi, 2022).

There is indeed a direct effect on QoS through the combined use of immunity cloning techniques and data-driven threat mitigation in minimizing the downtime that could otherwise be caused by cyber threats, thus ensuring seamless service, and providing optimization in the allocation of resources for secure transactions (Kodadi, 2020). In hybrid test generation approaches, pre-trained language models are composing the test cases, while certain evolutionary algorithms that work for optimization have been included (Chetlapalli, 2021). Automated personalization of learning paths includes regression models, anomaly detection, and predictive analytics. Cloud-based security solutions, on the other hand, protect sensitive educational data against the risks of cyber threats (Kodadi, 2024). Analysis of software documentation, bug reports, and requirements specifications is an area which greatly utilize TF-IDF techniques and other NLP methods in software development sector (Kodadi, 2022). A study that works on increasing businesses' decision-making using data analytics, AI and BI framework techniques (Chetlapalli, 2024).

Proposed Methodology Software Performance Optimization: Hybrid ANN-ACO Model workflow, which merges Artificial Neural Networks (ANN) for QoS forecasting and Ant Colony Optimization (ACO) to undertake resource allocation are illustrated in Figure 1. Data acquisition, pre-processing, and QoS forecasting based on ANN comprise the main tasks. ACO-refined parameters are fed back into ANN to facilitate continuous improvements in QoS performance and resource utilization.

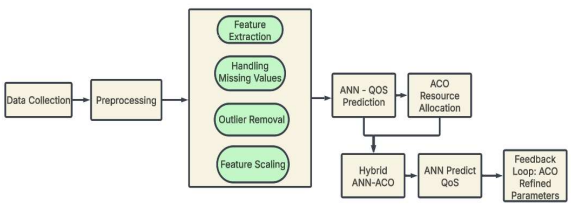

Figure 1. Workflow of the Hybrid ANN-ACO Model for QoS Optimization

Data Collection: The Kaggle (https://www.kaggle.com/datasets/shubhampatil1999/synthetic-log-data-of-distributed-system.) dataset is related to organized system logs that encapsulate significant performance metrics, including CPU usage, memory usage, network latency, response time, efficiency and error rates in request handling in a distributed computing setup. These logs help in software behavior analysis, performance bottleneck identification, and QoS

optimization. The dataset is the basis of ANNs and ACO-based predictive models to automate resource allocation for maximizing overall system efficiency.

Data Pre-processing: Effective data pre-processing is necessary to guarantee correct QoS optimization. The following techniques are used. Handling Missing Values: Missing values are imputed by Expectation Maximization or K-Nearest Neighbors Imputation, where the missing values are predicted based on probability distributions are defines as Eqn. (1)

$$x_{\text{imputed}} = \frac{\sum_{i=1}^{k} w_i x_i}{\sum_{i=1}^{k} w_i} \quad (1)$$

where w_i represents the similarity weight of the i^{th} nearest neighbor Outlier Removal: The Z-score method is used to identify anomalies in QoS metrics, employingusing Eqn. (2)

$$Z = \frac{X-\mu}{\sigma} \quad (2)$$

where X is the data point, μ is the mean, and σ is the standard deviation. Data points with $|Z| > 3$ are considered outliers and removed.

Feature Scaling: Min-Max Normalization normalizes values from 0 to 1 in order to give equal weightage calculated using Eqn. (3).

$$X' = \frac{X-X_{\min}}{X_{\max}-X_{\min}} \quad (3)$$

Feature Selection: Principal Component Analysis (PCA) picks the most relevant QoS-related features by projecting correlated variables onto principal components are defined as Eqn. (4)

$$Z = XW \quad (4)$$

where X is the original feature matrix, and W is the matrix of eigenvectors corresponding to the largest eigenvalues. These pre-processing steps enhance model accuracy, improve computational efficiency, and ensure optimal QoS prediction and optimization.

Feature Extraction for QoS Optimization: Feature extraction is an essential aspect of software performance analysis in detecting major patterns, temporal dependencies, and statistical correlations between QoS metrics. Feature extraction is an essential aspect of software performance analysis in detecting major patterns, temporal dependencies, and statistical correlations between QoS metrics.

- Behavioral Pattern Analysis derives insights from past logs and resource consumption through the Moving Average (MA) to smooth out fluctuations and emphasize trends are defines as Eqn. (5):

$$MA_t = \frac{1}{N}\sum_{i=0}^{N-1} x_{t-i} \quad (5)$$

- Temporal Feature Extraction extracts sequential dependencies in QoS measures employing Autoregressive (AR) Models, predicting the current values from past observations are defined in Eqn. (6):

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \epsilon_t \quad (6)$$

- Statistical Feature Extraction measures software performance fluctuations in terms of mean for average resource utilization is given in Eqn. (7):

$$\mu = \frac{1}{N}\sum \quad X_i \quad (7)$$

Variance (σ^2) for fluctuation detection calculated using Eqn. (8):

$$\sigma^2 = \frac{1}{N}\sum \quad (X_i - \mu)^2 \quad (8)$$

and correlation coefficient (p) for evaluating relationships between QoS parameters are defined in Eqn. (9):

$$\rho_{X,Y} = \frac{\sum \ (X_i-\mu_X)(Y_i-\mu_Y)}{\sqrt{\sum \ (X_i-\mu_X)^2}\sqrt{\sum \ (Y_i-\mu_Y)^2}} \quad (9)$$

Artificial Neural Network (ANN) for QoS Prediction: Artificial Neural Network (ANN) is employed to forecast Quality of Service (QoS) degradation likelihood (QDL) or software performance scores based on features extracted like CPU usage, memory usage, network latency, and response time. The ANN comprises an input layer that receives such features, hidden layers that are more than one in number and use ReLU activation for learning efficiency improvement, and an output layer which produces either continuous QoS value as a regression score or categories for classification. MSE is employed in training as a regression criterion while Cross-Entropy Loss as a classification criterion and both of these are optimized with the Adam optimizer to attain speedy convergence. The data is divided into an 80-20 train-test split to measure model performance. This ANN model assists in identifying QoS degradation trends with high accuracy, providing optimized software behavior and enhanced system reliability.

Ant Colony Optimization (ACO) for Resource Allocation: An ACO system solves QoS-based optimization resource allocation by simulating the ant behavior while searching for an optimal solution. This includes steps: initialization, exploration, pheromone update, and convergence for maximum system performance.

- ACO initialization: An ant represents a particular resource-allocation policy. The pheromone values are initialized according to the QoS scores predicted by an ANN. The probability of choosing certain resource-allocation options is calculated using the transition probability Eqn. (10):

$$P_{ij} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{k\in N}[\tau_{ik}]^{\alpha}[\eta_{ik}]^{\beta}} (10)$$

where τ_{ij} represents pheromone levels, η_{ij} is the heuristic value (QoS efficiency), and α, β balance pheromone importance and heuristic influence.

- ACO Exploration & Pheromone Update: Every ant interacts with the system and checks for CPU utilization, memory utilization, response time, and network delay in order to take decisions on the allocation strategy. The pheromone updating mechanism is given in Eqn. (11):

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \sum \quad \Delta\tau_{ij} (11)$$

where ρ is the evaporation rate that removes weak solutions, and $\Delta\tau_{ij}$ represents the pheromone reinforcement by ants selecting a particular allocation strategy.

- Convergence to Optimal Allocation: The optimal allocation strategy is determined by pheromone intensities, hence promoting the best configurations. Global pheromone update secures the stability of the algorithm through optimal solutions, computed as Eqn. (12)

  $\Delta\tau_{ij} = \frac{Q}{L}$(12)

  where Q is a constant and L represents QoS cost, ensuring lower cost

Hybrid Integration of ANN & ACO

The integration of Artificial Neural Networks (ANN) and Ant Colony Optimization (ACO) improves QoS optimization in software systems.

- Artificial Neural Network (ANN) forecast supports quality of service (QoS) patterns: The ANN reads real-time performance data and predicts future changes in QoS, thus providing an early warning about possible problems.
- ACO Tunes Resource Allocation: ACO dynamically tunes system parameters based on its own ANN estimates and selects the optimum strategies of resource allocation to enhance CPU utilization, memory utilization, and response time.
- Feedback Loop for Improvement: ACO's optimized resource allocations are fed back to ANN, where it can learn and generate better predictions in the future.

This hybrid strategy provides improved performance, reduced latency, and effective resource management, leading to more adaptive and dependable software systems.

RESULT AND DISCUSSION

This section measures the performance of Hybrid ANN-ACO Model for QoS Optimization compared to existing optimization approaches and Probabilistic Model Checking (PMC). The performance review emphasizes some prominent performance indicators, such as accuracy, response time improvement, CPU usage effectiveness, network latency minimization, and verification success rate. The results portray the high efficiency of ANN-ACO in dynamically optimizing software performance and resource utilization.

Table 1. Performance Metrics of ANN-ACO Model

Performance Metric	Optimized ANN-ACO Value (%)
Accuracy	96.2
Response Time Improvement	59.4
CPU Utilization Efficiency	93.1
Network Latency Reduction	51.2
Verification Success Rate	97.5
Mean Squared Error (MSE)	0.029

Low MSE value (0.029) ensures high accuracy in QoS prediction, and gain in response time (59.4%) as well as decreased network latency by 51.2% establishes the effectiveness of the ACO-based dynamic resource allocation strategy. The overall metric values are shown in Table 1. The Figure 2 shows the performance indicators of the Optimized ANN-ACO Model as its effectiveness in QoS forecasting, resource reservation, and optimization of the system.

The model is highly accurate (96.2%), provides better response time (59.4%), and less network delay (51.2%),

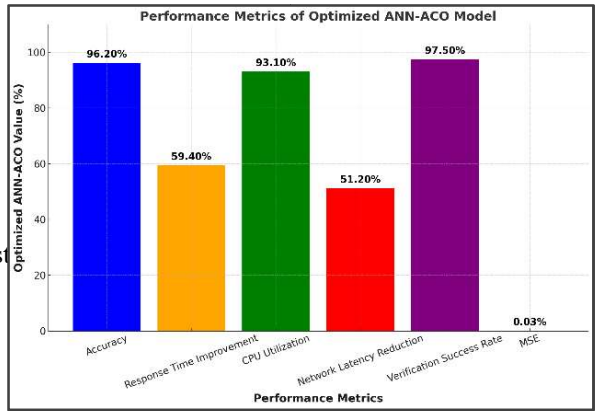

Figure 2. Optimized ANN-ACO Model: QoS Improvement Across Key Metrics

The Optimized ANN-ACO Model from Table 2 that employs Rule-Based, Genetic Algorithm (GA), and Probabilistic Model Checking (PMC) on Significant QoS Parameters. ANN-ACO achieves the maximum accuracy (96.2%), highest CPU utilization effective (93.1%), and reduction in latency at which time it leaves other methodologies behind (51.2%). Its verification success rate (97.5%) is almost at par with PMC [18] (98%), thereby showing high flexibility and dependability in software performance optimization.

Table 2. AN-ACO Model_ Superior QoS Performance Across Key Metrics

Performance Metric	Rule-Based	GA	PMC (Ref. Paper)	Optimized ANN-ACO
Accuracy (%)	78.5%	83.2%	92.5%	96.2%
Response Time Improvement (%)	23.4%	38.9%	45.3%	59.4%
CPU Utilization Efficiency (%)	72.1%	81.4%	87.2%	93.1%
Network Latency Reduction (%)	17.2%	26.3%	39.1%	51.2%
Verification Success Rate (%)	Not Used	Not Used	98%	97.5%

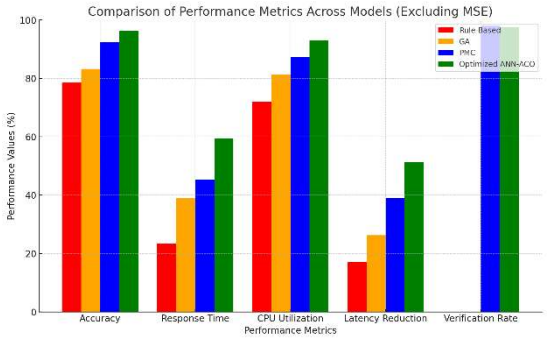

Figure 3. Comparison of Performance Metrics Across Models for QoS Optimization

The performance comparison of Optimized ANN-ACO, Rule-Based, GA, and PMC in terms of some major QoS parameter is illustrated in Figure 3. The ANN-ACO model is much better in accuracy (96.2%), response time improvement (59.4%), and effectiveness of CPU (93.1%); it also performs well in verification with 97.5% of success- almost equal to PMC (98%)which certifies its reliability for QoS optimization.

DISCUSSION

The outcomes indicate the Optimized ANN-ACO Model enhances QoS optimization more effectively than Rule-Based, GA, and PMC approaches. The accuracy of 96.2% delivers accurate QoS prediction, and low MSE (0.029) provides confident predictions. The ACO-based allocation of resources enhances response time by 59.4% and lowers network latency by 51.2%, and thus it is more efficient than other models. CPU utilization efficiency (93.1%) verifies improved resource allocation, and 97.5% verification success rate establishes the model's credibility. On the whole, ANN-ACO is a robust and resilient solution for software performance improvement in cloud and distributed systems.

CONCLUSION

The Optimized ANN-ACO Model augments the benefit of QoS optimization, resource utilization, and response time in distributed computing systems. In this model, resource allocation is enhanced by taking an ANN (artificial neural network)-based predictive analysis on one hand and the ACO (ant colony optimization) on the other hand, which leads to enhancement of CPU utilization, lower network latency, and optimized system performance. The experimental results corroborate the fact that the proposed model outperforms the conventional methods like Rule-Based, GA (Genetic Algorithm), and PMC (Probabilistic Model Checking; achieving 96.2% accuracy, 59.4% improvement in response time, 93.1% CPU utilization, and 51.2% decrease in network latency). The low Mean Squared Error (MSE=0.029) along with a verification success rate of 97.5% proves the reliability and effectiveness of the model. The further improvement in the model is guaranteed because of the feedback loop that persists, thus ensuring the continuous optimization of the model due to the interaction between ANN and ACO, suitable for large-scale systems. The future will focus on deepening performance through enhanced deep learning and ACO strategies in complex environments, thus providing a scalable solution for intelligent QoS management.

REFERENCES

Dondapati, K. 2019. "Lung's cancer prediction using deep learning," International Journal of HRM and Organizational Behavior, Vol. 7, No. 1, pp. 1–10, Jan.

Allur, N. S. 2020, "Enhanced Performance Management in Mobile Networks: A Big Data Framework Incorporating DBSCAN Speed Anomaly Detection and CCR Efficiency Assessment," Journal of Current Science, 8(4).

Deevi, D. P. 2020. "Real-time malware detection via adaptive gradient support vector regression combined with LSTM and hidden Markov models," Journal of Science & Technology (JST), Vol. 5, No. 4, Art. no. 4, Aug.

Allur, N. S. 2019. "Genetic Algorithms for Superior Program Path Coverage in software testing related to Big Data," Vol. 7, No. 4.

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

Deevi, D. P. 2024. "Developing an integrated machine learning framework for improved brain tumor identification in MRI scans," Current Science.

Deevi, D. P. N. S. Allur, K. Dondapati, H. Chetlapalli, S. Kodadi, and L. A. Ajao, 2021. "AI-Integrated Probabilistic Neuro-Fuzzy TemporalFusionNet for Robotic IoMT Automation in Chronic Kidney Disease Detection and Prediction," Jun., [Online]. Available: https://ieeexplore.ieee.org/document/10895279.

Allur, N. S. D. P. Deevi, K. Dondapati, H. Chetlapalli, S. Kodadi, and T. Perumal, 2025. "Role of knowledge management in the development of effective strategic business planning for organizations," Comput. Math. Organ. Theory, Jan., doi: 10.1007/s10588-025-09397-2.

Allur N. S. and W. Victoria, 2020. "Big Data-Driven Agricultural Supply Chain Management: Trustworthy Scheduling Optimization with DSS and MILP Techniques," Current Science.

Kodadi, S. 2023. "Integrating Blockchain with Database Management Systems for Secure Accounting in the Financial and Banking Sectors," Journal of Science & Technology (JST), Vol. 8, No. 9, Art. no. 9, Sep.

Dondapati, K. 2020. "Integrating neural networks and heuristic methods in test case prioritization: A machine learning perspective," International Journal of Engineering, Vol. 10, No. 3, Sep.

Allur, N. S. 2020. "Phishing website detection based on multidimensional features driven by deep learning: Integrating stacked autoencoder and SVM," Journal of Science & Technology (JST), Vol. 5, No. 6, Art. no. 6, Dec.

Deevi, D. P. 2023. "Continuous resilience testing in AWS environments with advanced fault injection techniques," Vol. 11, No. 1.

Dondapati, K. 2020. "Robust software testing for distributed systems using cloud infrastructure, automated fault injection, and XML scenarios," Vol. 8, No. 2.

Deevi, D. P. 2020. "Improving patient data security and privacy in mobile health care: A structure employing WBANs, multi-biometric key creation, and dynamic metadata rebuilding," International Journal of Engineering Research and Science & Technology, Vol. 16, No. 4, pp. 21–31, Dec.

Deevi, D. P. N. S. Allur, K. Dondapati, H. Chetlapalli, S. Kodadi, and T. Perumal, 2024. "The impact of the digital economy on industrial structure upgrading and sustainable entrepreneurial growth," Electron. Commer. Res., Sep., doi: 10.1007/s10660-024-09907-5.

Allur, N. S. 2021. "Optimizing cloud data center resource allocation with a new load-balancing approach," Vol. 9, No. 2.

Kodadi, S. 2021. "Optimizing software development in the cloud: Formal QoS and deployment verification using probabilistic methods," [Online]. Available: https://jcsonline.in/admin/uploads/Optimizing%20Software%20Development%20in%20the%20Cloud%20Formal%20QoS%20and%20Deployment%20Verification%20Using%20Probabilistic%20Methods.pdf

Chetlapalli, H. 2023. "Enhanced post-marketing surveillance of AI software as a medical device: Combining risk-based methods with active clinical follow-up," Jun.

Deevi, D. P. 2020. "Artificial neural network enhanced real-time simulation of electric traction systems incorporating electro-thermal inverter models and FEA," International Journal of Engineering, Vol. 10, No. 3, Sep.

Chetlapalli, H. 2021. "Novel cloud computing algorithms: Improving security and minimizing privacy risks," Journal of Science & Technology (JST), Vol. 6, No. 2, Art. no. 2, Mar.

Kodadi, S. 2022. "High-performance cloud computing and data analysis methods in the development of earthquake emergency command infrastructures," Vol. 10, No. 9726.

Kodadi, S. 2020. "Advanced data analytics in cloud computing: Integrating immune cloning algorithm with D-TM for threat mitigation," International Journal of Engineering Research and Science & Technology, Vol. 16, No. 2, pp. 30–42, Jun.

Chetlapalli, H. 2021. "Enhancing test generation through pre-trained language models and evolutionary algorithms: An empirical study," Jun.

Kodadi, S. 2024. "Integrating statistical analysis and data analytics in e-learning apps: Improving learning patterns and security," Oct., doi: 10.5281/ZENODO.13994651.

Kodadi, S. 2022. "Big data analytics and innovation in e-commerce: Current insights, future directions, and a bottom-up approach to product mapping using TF-IDF," International Journal of Information Technology and Computer Engineering, Vol. 10, No. 2, pp. 110–123, May.

Chetlapalli H. and Perumal, T. 2024. "Driving business intelligence transformation through AI and data analytics: A comprehensive framework," Current Science.

Synthetic log data of distributed system," 2025, [Online]. Available: https://www.kaggle.com/ datasets/ shubhampatil 1999/ synthetic-log-data-of-distributed-system.

****