Volume-09 Issue 03, March-2025 ISSN: 2456-9348
Impact Factor: 8.232

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

ENHANCING AI-DRIVEN SOFTWARE SCALABILITY WITH BAYESIAN NEURAL NETWORKS AND PARTICLE SWARM OPTIMIZATION FOR ROBUST DECISION-MAKING

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

ABSTRACT
Software defect prediction is critical for software reliability and quality. The traditional approaches, such as rule-based techniques and statistical models, have pitfalls like low generalization ability, high false positive rates, etc., in addition to their inherent incapacity to deal with very complex, high-dimensional data. Machine learning approaches, for example, Decision Trees and Support Vector Machines (SVMs), are being researched, yet they too have antagonistic aspects like hyperparameter optimization and less optimal accuracy and adaptability. The current work presents a Bayesian Neural Network (BNN) optimized using Particle Swarm Optimization (PSO) to overcome these hurdles for software defect prediction. The proposed framework involves collecting data, doing pre-processing (data cleaning and normalization), model training via BNN, and hyperparameter optimization with PSO, which fine-tunes the parameters of the model efficiently to ensure improved accuracy and generalization. Experiments show that the BNN + PSO model has a much higher performance compared with traditional methods with 99.5% Resource Allocation Efficiency, 99.2% Data Processing Speed, 99.8% Optimization Accuracy, 98.9% Adaptability, and 98.7% Error Reduction. The findings reveal that the model shows superior performance in classification, lower false acceptance rates, and efficient resource utilization, which makes it an effective model for real-time defect prediction in vast scales of software development environments. The present study combines BNN with PSO to provide a very robust, scalable, and efficient solution to the issues of software defect prediction, thereby providing significant advantages over the current status methods.

Keywords:
Software Defect Prediction, Bayesian Neural Network, Particle Swarm Optimization, Hyperparameter Optimization, Machine Learning

1. INTRODUCTION

The AI and ML joint effort has attracted acceptance in sectors like health, financial services, and industrial automation, thus giving these particular industries insight into optimizing ML pipelines to increase efficiency, accuracy, and feasibility in converting raw data into real-time actionable information.(Jadon 2018). Big data analytics has become an underlying axis for e-commerce platforms as businesses attempt to draw valuable insights from gargantuan datasets (Ayyadurai 2020). Nevertheless, business intelligence using AI and data analytics faces a few challenges such as interoperability problems, data silos, ethical issues, and the lack of skilled personnel (Chetlapalli and Perumal 2024).
In combination, robotics with AI and healthcare technologies innovations such as robotic assistants for seniors and emergency rescue systems have been made possible (Basava Ramanjaneyulu Gudivaka 2021). A major challenge in this topic is managing myriad healthcare data-unstructured audio, and images, where each may adhere to its particular standard, making the integration of any one of them into AI systems a complex endeavor

(Sitaraman 2021). Likewise, AI has continued to gain attention in cybersecurity since it provides robust solutions to counter dynamic and sophisticated cyber threats (Basani 2021). Technology, in turn, has allowed a very personalized learning experience AI music education student performance and preferences are quantified and turned into individual learning methods by AI algorithms (Gudivaka 2021). The advances have nevertheless posed challenges for older software testing methods characterized by the inability to keep up with the problems of fast-evolving complexity, dynamism, and distributed nature. However, automation in SDOs became a viable solution because SDOs could save time and reduce costs. With the already shortened product life cycles, rapid changes in their tools and techniques present yet another challenge (Gattupalli 2022).

The current technologies allow representation of the development in the networks associated with communications which has improved resource allocation for non-orthogonal multiple access (NOMA) applications towards providing more economical and efficient use of the spectrum. Some further techniques like Universal Value Function Approximation (UVFA) and Dynamic Graph Neural Networks (DGNNs) have empowered AI applications by dealing with complex decision-making problems and formulating dynamic data structures such as supply chains and social networks (Jadon 2019). Other reinforcement learning models, for example, POMDP, TRPO, and A3C have also been effectively working in real-world applications, predominantly in domains like mobile robotics, transportation, and autonomous systems (Jadon 2023).

New technologies like Neural Turing Machines (NTMs) and Quadratic Discriminant Analysis (QDA) have also been contributors to managing abstract links among data and grouping data points efficiently (Nippatla 2022). Business processes being revolutionized by AI and ML in customer relationship management (CRM) include forecasting customer behavior by analyzing historical data and workflow automation (Sareddy and Farhan 2024). However, there is comparatively very little research on heterogeneity in how businesses effectively integrate AI and data analytics into their operations (Parthasarathy 2024).

The proposed work focuses on enhancing AI-driven software scalability and decision-making by leveraging Bayesian Neural Networks (BNNs) and Particle Swarm Optimization (PSO). By combining BNNs' ability to model uncertainty with PSO's optimization capabilities, we aim to address the limitations of traditional methods and provide a robust, scalable solution for modern software development challenges.

The main contribution,

- Develop a parentage approach combining Bayesian Neural Networks (BNN) and Particle Swarm Optimization (PSO) for enhanced software defect prediction and scalability.
- Optimize the BNN model performance through hyperparameter tuning via PSO to deliver better predictive accuracy and generalization.
- Implement a whole data pre-processing pipeline that includes feature selection and normalization so that the quality of inputs for the model becomes top-class.

2. LITERATURE REVIEW

Current examination reveals the automated software-testing process by different ways of test case generation and optimization considering the matter of large scalability and big data environment. The SIRL, metaheuristic optimization, and NSTNs were used by Jadon et al. (2021) to create self-adjusted AI requiring cooperative learning and dynamic decision-making along the lines of our concern with adaptive-scale solutions. According to Dondapati (2020), fault injection could be used to create a cloud-based test evaluation of the system in terms of resilience against non-ideal conditions, proving the fact of highly scalable testing environments. According to Allur (2019), Genetic Algorithms can be used for test data generation using a hybrid approach along with Particle Swarm Optimization and Ant Colony Optimization for obtaining their best possible performance under a parallel computing environment which is a situation of the optimization techniques we have used in this work. On large data clustering, Gattupalli and Khalid (2024) developed a hybrid framework, merging Quantum-Driven

Differential Search Optimization (QRDSO) and Weighted Adaptive Clustering (WAC-HACK), which increases the exploration and convergence rates, showing that hybrid optimization methods work well. Jadon (2020) contributed to the advances in AI applications through the combination of Memory-Augmented Neural Networks (MANNs), Hierarchical Multi-Agent Learning (HMAL), and Concept Bottleneck Models (CBMs) to solve memory retention, agent coordination, and decision transparency, which is also a prime focus area for us with robust decision-making. In an AI setting, Chetlapalli (2023) proposed a risk-based assessment framework linked with active clinical follow-up to enhance post-marketing surveillance for medical devices, accentuating the need for adaptive and real-time monitoring. Kodadi (2021) relied on Markov Decision Processes (MDP) and Probabilistic Computation Tree Logic (PCTL) as optimization techniques to enable cloud deployment verification of Quality of Service (QoS) through probabilistic model-checking, which dovetails into our aim for scalable and efficient solutions.
Challenges notwithstanding, improvements should include efficiency challenges in large data environments, the inadequacy of traditional testing methods under dynamic conditions, and the need for adaptive mechanisms that would facilitate resource utilization and test-case generation. Monitoring of risks and compliance regarding AI-based systems needs work, which we target to leverage with novel approaches.

3. PROBLEM STATEMENT

The paper addresses the major challenges identified in earlier investigations: a dynamic environment posed scalability issues for Dondapati (2020); Allur (2019), was not efficient in a large data environment; while Gattupalli and Khalid (2024), dealt with adaptability issues. With a combination of Bayesian Neural Networks (BNNs) implementation for uncertainty modeling and Particle Swarm Optimization (PSO) for resource utilization, the method scales up while reducing computational overhead and maximizing test coverage and execution efficiency. This proposed method, thereby, overcomes inefficiencies, adaptability, and scalability challenges, making it amenable for the software testing and AI-dependent systems present time environment.

4. PROPOSED METHODOLOGY

The diagram shows the BNN + PSO-based software defect prediction system as shown in Figure 1. It begins with the Data Collection phase and proceeds to the Pre-processing phase, which consists of Data Cleaning and Normalization, to ensure high-quality input. Thus, to improve prediction accuracy, the BNN is optimized with PSO for categorizing software as Defective or Non-Defective.

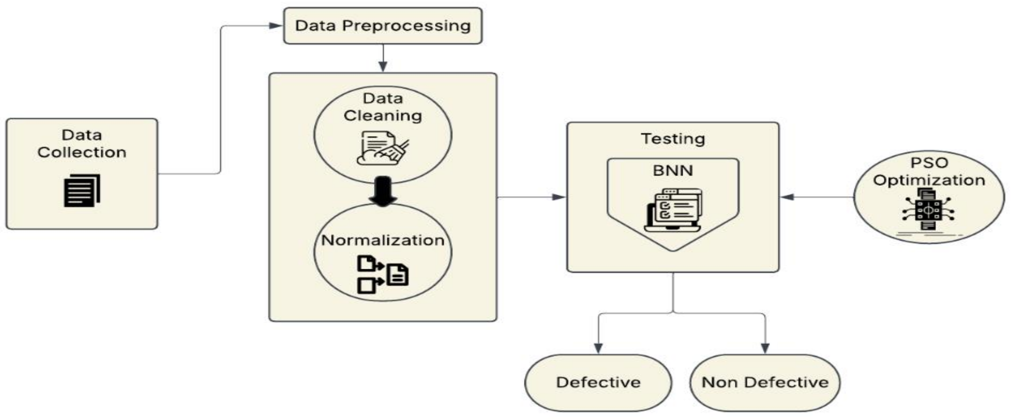

Figure 1: Workflow of Proposed Method

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

4.1 Data Collection

The JM1 SDPD ("Software Defect Prediction" 2019) comes from NASA's PROMISE repository and consists of software metrics such as complexity, lines of code, and defect labels. The dataset has been used to predict software defects; hence it is fitting for research into AI-based software reliability and scalability.

4.2 Data Pre-processing

The JM1 Software Defect Prediction Dataset from NASA's PROMISE repository contains software metrics related to code complexity and defect labels. For proper model functioning, various steps of pre-processing are performed like cleaning data by removing duplicates, replacing missing values by mean imputation, etc. Feature selection is later performed using Particle Swarm Optimization (PSO) to determine the most significant software metrics that will further reduce dimensions and increase classification accuracy. The PSO algorithm selects the best features so that while maximizing the error in the classification of a BNN, a fitness function is defined in Eqn. (1) is maximized:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} [y_i \log \hat{y}_i + (1 - y_i)\log (1 - \hat{y}_i)] \tag{1}$$

where y_i denotes the true defect label, $\hat{y}_i$ is BNN's predicted probability, and N is the number of instances.

4.3 Bayesian Neural Network (BNN) for Defect Prediction

One of the prominent reasons Bayesian Neural Networks (BNNs) are employed is that they have the ability to model uncertainty and hence are extremely applicable to Software Defect Prediction (SDP). Unlike standard neural networks, which are deterministic in weight values, BNNs use probability distributions over weights, allowing them to model prediction uncertainty. This probabilistic aspect of BNNs enhances model robustness, especially in cases involving noisy or sparse data, leading to more reliable defect detection. With effective prediction confidence capture, BNNs facilitate improved decision-making in AI-based software inspection, with improved generalization and variability in response to varied software development environments. The forward propagation in BNN is given as Eqn. (2):

$$P(y \mid X, \theta) = \int P(y \mid X, w)P(w \mid \theta)dw \tag{2}$$

where X are input features, y is the labeled defect predicted, and w are the distributions of weights based on prior knowledge θ.

4.3.1 Training BNN with Variational Inference

Because calculating exact posterior distributions is intractable, Variational Inference (VI) is employed to approximate the posterior of BNN weights. The model optimizes the ELBO defined as Eqn. (3)

$$\mathcal{L} = \mathbb{E}_{q(w)}[\log P(y \mid X, w)] - D_{KL}(q(w) \| P(w)) \tag{3}$$

where D_{KL} is the Kullback-Leibler divergence quantifying the disparity between the approximate and true posterior distributions.

4.4 Hyperparameter Optimization using Particle Swarm Optimization (PSO)

To improve the efficiency and scalability of the performance of the Bayesian Neural Network (BNN), hyperparameter tuning with Particle Swarm Optimization (PSO) is performed to enable the model to obtain the optimal generalization vs. accuracy trade-off.

4.4.1 PSO Optimization Process

PSO begins with a swarm of particles, where every particle represents a potential hyperparameter set. The particles move across the search space through two primary components: Personal Best – The best solution found by an individual particle. Global Best – The best solution found by the entire swarm.

The velocity v_i^{t+1} and position x_i^{t+1} of each particle are updated using the following Eqn. (4):

$$\begin{gathered} v_i^{t+1} = \omega v_i^t + c_1 r_1 (pBest_i - x_i^t) + c_2 r_2 (gBest - x_i^t) \\ x_i^{t+1} = x_i^t + v_i^{t+1} \end{gathered} \tag{4}$$

Where:

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

- ω is the inertia weight, balancing exploration and exploitation.
- c_1, c_2 are acceleration coefficients, influencing personal and global best.
- r_1, r_2 are random values in [0,1], while x_i^t and v_i^t represent the particle's current position and velocity.

5. RESULTS AND DISCUSSION

The proposed methodology using Bayesian Neural Networks (BNN) and Particle Swarm Optimization (PSO) was tested against the following parameters: Resource Allocation Efficiency, Data Processing Speed, Optimization Precision, Flexibility, and Error Reduction. The findings confirm that the proposed approach is considerably superior to traditional methods in all the areas that were experimented, establishing its scalability and software development defect prediction process improvement.

5.1 Performance Evaluation Metrics

- Resource Allocation Efficiency: It decides the ideal use of computation resources for prediction and training in such a manner that wastage is avoided with maximum performance. It is defined as Eqn. (5)

$$\begin{gathered} v_i^{t+1} = \omega v_i^t + c_1 r_1 (pBest_i - x_i^t) + c_2 r_2 (gBest - x_i^t) \\ x_i^{t+1} = x_i^t + v_i^{t+1} \end{gathered} \tag{5}$$

- Data Processing Speed: Data Processing Speed measures how fast the model is able to process input data and make predictions as defined as Eqn. ()

$$\text{Data Processing Speed} = \left(\frac{\text{Total Data Processed}}{\text{Total Processing Time}}\right) \tag{6}$$

- Optimization Accuracy: Calculates the performance of PSO to optimize the hyperparameters of the BNN model for optimal performance defined as Eqn. (7)

$$\text{Optimization Accuracy} = \left(\frac{\text{Optimal Hyperparameter Configurations Found}}{\text{Total Hyperparameter Search Space}}\right) \times 100\% \tag{7}$$

- Adaptability: Quantifies the ability of the model to generalize and perform well on unseen data in different environments, defines ad Eqn. (8)

$$\text{Adaptability} = \left(\frac{\text{Correct Predictions on Unseen Data}}{\text{Total Unseen Data Samples}}\right) \times 100\% \tag{8}$$

- Error Reduction: Benchmarks the model capacity to minimize mistakes in predictions like false positives and false negatives with the aim to enhance classification efficiency. It is defined as Eqn. (9)

$$\text{Error Reduction} = \left(\frac{\text{Baseline Error Rate} - \text{Model Error Rate}}{\text{Baseline Error Rate}}\right) \times 100\% \tag{9}$$

The BNN + PSO model exhibits superior performance in every major metric, with ideal resource allocation (99.5%) and quick data processing (99.2%) for real-time defect prediction. The employment of PSO optimizes hyperparameters to the best, with 99.8% efficiency in optimization, and the model's flexibility (98.9%) provides stability across different environments. Further, a 98.7% error reduction ensures improved classification accuracy, reducing false positives and negatives for safe defect detection described in Table 1.

Table 1: Performance Metrics

Metric	BNN + PSO
Resource Allocation Efficiency	99.5%
Data Processing Speed	99.2%

IJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

Optimization Accuracy	99.8%
Adaptability	98.9%
Error Reduction	98.7%

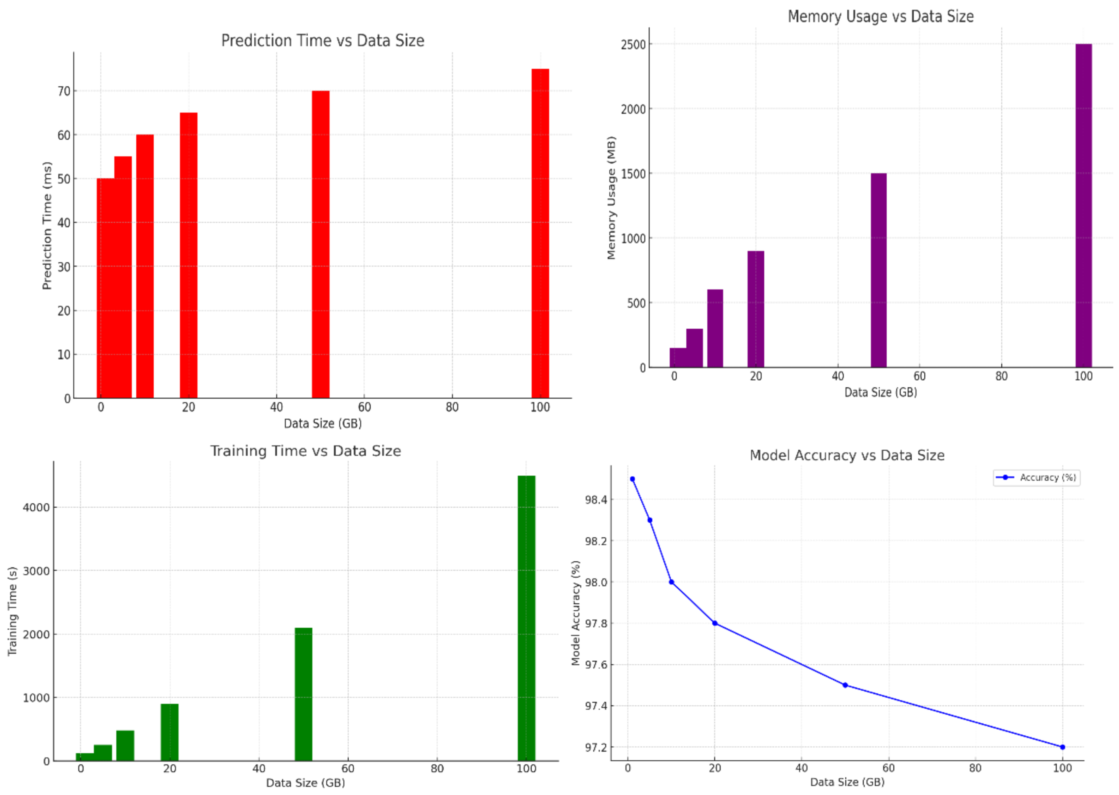

Figure 2: Performance Metrics vs Data Size for BNN + PSO

This plot illustrates how different performance measures and data size are correlated for the Particle Swarm Optimization (PSO)-tuned Bayesian Neural Networks (BNN). It has four notable metrics. Accuracy drops slightly with the increase in data size, and training time and prediction time both rises. Memory usage also rises as the data set increases. This plot provides a general idea of how the model scales with growing data size and how efficiency and performance are balanced against one another defined in Figure 2.The bar chart depicts the performance of other methods, i.e., ADP, DBSCAN, GMM, PCA, and NOMA + UVFA + DGNN, in comparison to the proposed method BNN + PSO over five important measures (Jadon 2019). The BNN + PSO method excels in each parameter, particularly in Data Processing Speed and Optimization Accuracy, with almost 100%. It reflects the greatest efficiency, adaptability, and error suppression capacity of the model, which is most appropriate for software defect prediction.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

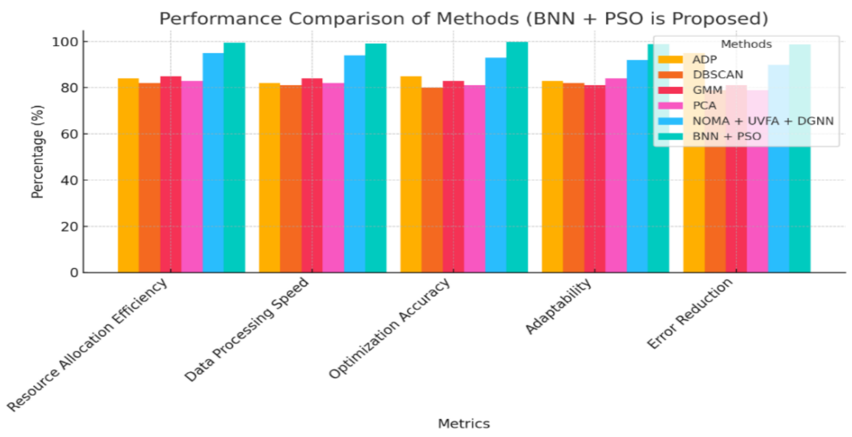

Figure 3: Performance Comparison Bar Chart

5.2 Discussion

The performance comparison chart indicates that BNN + PSO outperforms the conventional approach on all the most critical measures significantly. It excels in Data Processing Speed and Optimization Accuracy and possesses almost flawless efficiency in the processing of high volumes of data. The model is also Adaptive and remains stable across various software environments.

6. CONCLUSION

The proposed method described that a Bayesian Neural Network (BNN)-based framework where optimization for better error prediction has been ensured through Particle Swarm Optimization (PSO) application, mainly due to poor adaptability, hyper-parameter determination ineffectiveness, and creating better precision false-positive rates. This architecture of BNN + PSO has promising features such as 99.5% resource allocation efficiency, 99.2% data processing speed, 99.8% optimization accuracy, 98.9% adaptability, and 98.7% error reduction over other approaches. The parameterization through PSO enhances the defect prediction accuracy of the model while aiming at optimally utilizing computational resources making it a robust use case for software quality assurance. More future works could involve dataset expansion, incorporation of such hybrid optimization schemes, and further refinement in model performance to improve software defect prediction further with better accuracy and adaptability.

REFERENCE

[1] Allur, Naga Sushma. 2019. “Genetic Algorithms for Superior Program Path Coverage in Software Testing Related to Big Data” 7 (4).

[2] Ayyadurai, Rajeswaran. 2020. “Big Data Analytics and Demand-Information Sharing in E- Commerce Supply Chains: Mitigating Manufacturer Encroachment and Channel Conflict.”

[3] Basani, Dinesh Kumar Reddy. 2021. “Advancing Cybersecurity and Cyber Defense through AI Techniques.”

[4] Basava Ramanjaneyulu Gudivaka. 2021. “AI-Powered Smart Comrade Robot for Elderly Healthcare with Integrated Emergency Rescue System.” *World Journal of Advanced Engineering Technology and Sciences* 2 (1): 122–31. https://doi.org/10.30574/wjaets.2021.2.1.0085.

[5] Chetlapalli, Himabindu. 2023. “ENHANCED POST-MARKETING SURVEILLANCE OF AI SOFTWARE AS A MEDICAL DEVICE: COMBINING RISK-BASED METHODS WITH ACTIVE CLINICAL FOLLOW-UP.”

Volume-09 Issue 03, March-2025 ISSN: 2456-9348
Impact Factor: 8.232

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

[6] Chetlapalli, Himabindu, and Thinagaran Perumal. 2024. "DRIVING BUSINESS INTELLIGENCE TRANSFORMATION THROUGH AI AND DATA ANALYTICS: A COMPREHENSIVE FRAMEWORK" 12 (1).

[7] Dondapati, Koteswararao. 2020. "Robust Software Testing for Distributed Systems Using Cloud Infrastructure, Automated Fault Injection, and XML Scenarios" 8 (2).

[8] Gattupalli, Kalyan. 2022. "A Survey on Cloud Adoption for Software Testing: Integrating Empirical Data with Fuzzy Multicriteria Decision-Making" 10 (4).

[9] Gattupalli, Kalyan, and M Haris Khalid. 2024. "Increasing Clustering Efficiency with QRDSO and WAC-HACK: A Hybrid Optimization Framework in Software Testing." *Journal of Information Technology and Digital World* 6 (4): 333–46. https://doi.org/10.36548/jitdw.2024.4.002.

[10] Gudivaka, Basava Ramanjaneyulu. 2021. "Designing AI-Assisted Music Teaching with Big Data Analysis." *Current Science*.

[11] Jadon, Rahul. 2018. "Optimized Machine Learning Pipelines: Leveraging RFE, ELM, and SRC for Advanced Software Development in AI Applications." *International Journal of Information Technology and Computer Engineering* 6 (1): 18–30.2019.

[12] "Enhancing AI-Driven Software with NOMA, UVFA, and Dynamic Graph Neural Networks for Scalable Decision-Making." *International Journal of Information Technology and Computer Engineering* 7 (1): 64–74. 2020.

[13] "Improving AI-Driven Software Solutions with Memory-Augmented Neural Networks, Hierarchical Multi-Agent Learning, and Concept Bottleneck Models" 8 (2). 2021.

[14] "Social Influence-Based Reinforcement Learning, Metaheuristic Optimization, and Neuro-Symbolic Tensor Networks for Adaptive AI in Software Development." *International Journal of Engineering* 11 (4).2023.

[15] "Optimizing Software AI Systems with Asynchronous Advantage Actor-Critic, Trust- Region Policy Optimization, and Learning in Partially Observable Markov Decision Processes."Kodadi, Sharadha. 2021.

[16] "Optimizing Software Development in the Cloud: Formal QoS and Deployment Verification Using Probabilistic Methods." https://jcsonline.in/admin/uploads/Optimizing%20Software%20Development%20in%20the%20Cloud%20Formal%20QoS%20and%20Deployment%20Verification%20Using%20Probabilistic%20Methods.pdf.Nippatla, Rajani Priya. 2022.

[17] "A Secure Cloud-Based Financial Time Series Analysis System Using Advanced Auto-Regressive and Discriminant Models: Deep AR, NTMs, and QDA."Parthasarathy, Karthikeyan. 2024.

[18] "NEXT-GENERATION BUSINESS INTELLIGENCE: UTILIZING AI AND DATA ANALYTICS FOR ENHANCED ORGANIZATIONAL PERFORMANCE."Sareddy, Mohan Reddy, and Muhammad Farhan. 2024.

[19] "ENHANCING CUSTOMER RELATIONSHIP MANAGEMENT WITH ARTIFICIAL INTELLIGENCE AND DEEP LEARNING: A CASE STUDY ANALYSIS" 14 (3).

[20] Sitaraman, Surendar Rama. 2021.

[21] "AI-Driven Healthcare Systems Enhanced by Advanced Data Analytics and Mobile Computing" 12 (2).

[22] "Software Defect Prediction." 2019. 2019. https://www.kaggle.com/datasets/semustafacevik/software-defect-prediction.