Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

A Reliable Network on Chip based using Whale Firefly Optimization and Deep Neural Network

Smriti Srivastava [1], Veena R S [2], Madhura J [3], Shalini K B [4], Bharath BC [5]
[1] *Associate Professor, R V College of Engineering, Bengaluru. Email: smritis@rvce.edu.in*
[2] *Associate Professor, Department of Information Science and Engineering, DayanandaSagar Academy of Technology and Management, Bengaluru. Email: veena-ise@dsatm.edu.in*
[3] *Assistant Professor, Department of Information science and engineering, Dayananda Sagar College of engineering, Bengaluru. Email: madhu-ise@dayanandasagar.edu*
[4] *Assistant Professor, Department of Information science and engineering, Dayananda Sagar College of engineering, Bengaluru. Email: shalini-ise@dayanandasagar.edu*
[5] *Assistant Professor, Department of Information science and engineering, Dayananda Sagar College of engineering, Bengaluru. Email: bharath-ise@dayanandasagar.edu*

ARTICLE INFO

Received: 17 Dec 2024

Revised: 18 Feb 2025

Accepted: 26 Feb 2025

ABSTRACT

Network on Chip (NoC) provides the technology for forming interconnect pattern to perform a task. NoC suffers from high power consumption and low reliability to perform computation. Prediction of network congestion helps to effectively handle the network parameters and increases the network performance. Existing reinforcement learning and decision tree methods have the limitation of overfitting problem and unstable performance. In this research, Whale Firefly Optimization (WFO) of hybrid optimization method with Deep Neural Network (DNN) is proposed to improve the reliability and reduces the power consumption in NoC. The WFO-DNN method tends to provides the optimized reward value based on the latency, power and traffic congestion. Traffic congestion is measured from the simulated network and applied for the WFO-DNN method to provide optimal reward update. The hybrid optimization method is applied for parameter settings for the NoC to maintain the trade-off for latency, power and reliability. The hybrid method of WFO have advantages of good convergence and improve the exploration of the method. The DNN model provides the efficient performance in the prediction of faults in the network to improve the reliability. The DNN model is tested with various number of hidden layer to find the suitable number of layer. The proposed WFO-DNN model has 1.34 improve of speed, 1.94 energy efficiency and existing reinforcement method has 1.24 improvement of speed, and 1.77 energy efficiency.

Keywords: Deep Neural Network, Energy Efficiency, Fault tolerance method, Network on Chip, and Whale Firefly Optimization.

INTRODUCTION

Network on Chip (NoC) is developing technology that form interconnect patterns of multiprocessor state. A variety of multiprocessor requirement is adapted in NoC technology and existing NoC technology doesn't support user applications. Several problems are present in power consumption and traffic congestion related to low efficiency of network due to complex routing connections [1]. Wireless NoC design consists of two important aspects such as power consumption and traffic congestion. This is preferred to avoid the topologies that results in die stacking [2]. The Wireless NoC is introduced to handle chip design characteristics such as energy efficiency, performance and flexibility. Some of the researches involves in apply graphical approach to effectively map the multi-core in Network. Different vertices are mapped to different cores in an NoC and if more than one vertex is assigned to a single core, then on priority basis vertices are scheduled [3]. Fault prediction is challenging task in mixed critical requirement for host on fault prone environment. The problem complexity is further increased in dynamic scenarios for real time applications to enter or leave multicore platform at any time instant [4]. The input buffer length of traditional static

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

router is fixed and each port is allocated with buffer resources that are independent and unique. The traditional static router has bad performance due application-oriented network has lower buffer utilization [5].

The design expansion of NoC is modified to provide energy efficient NoC based on various models. Major requirement is to have continuous monitoring of engineering process and policies to apply machine learning for energy efficient NoCs [6]. Distributed intelligent scheduling using reinforcement learning enhances performance in real-time and enables dynamic event management, including cloud computing, scheduling, and traffic congestion interconnection [7, 8]. Permanent faults are easily predictable and need to dealt at time of manufacturing using redundancy/spares. Transient faults affect the performance of application that are difficult to predict. Handling transient faults is a daunting task in real-time systems, particularly for programs running under rigid timing constraints [9, 10]. The objectives of the model are discussed as below

1. This research involves in develop Intelligent and adaptive Wireless NoC framework to predict the congestion in the network and adaptively transmit the data. This helps to improves the network performance and decreases in latency.
2. The proposed method is involving in reduces the power consumption for multi-core NoC based on the prediction process. The proposed method prediction helps to improve the network throughput based on congestion control.

This paper is organized as review of recent methods to improve NoC reliability is given in section 2, the explanation of WFO-DNN model is given in section 3, the results and discussion is given in section 4 and conclusion is in section 5.

LITERATURE REVIEW

Network On Chips (NOC) provides standard connection for multiple cores, memory modules and other hardware components. Various researches have been going on NOC and machine learning to improve performance and reliability with lower power usage of the process.

Wang and Louri [11] used a deep reinforcement learning approach named CURE to optimize network efficiency while minimizing latency and errors. Reversible Multi-function Adaptive Channels (RMC) were used to reduce network power consumption and latency. Fault-secure adaptive error correction was included in each router to enhance model reliability. Moreover, a bypass design and router power gating strategy were used to increase the chip lifespan and decrease power consumption. Reinforcement learning was used to improve performance, fault tolerance, and energy efficiency. The CURE approach effectively minimized end-to-end delay and enhanced energy efficiency. Nevertheless, reinforcement learning also raised the danger of overfitting, adversely affecting model performance.

Chen and Louri [12] proposed a quality control and data approximation system that is aimed at minimizing packet size, hence network latency and power usage. An error tolerance system was proposed to further minimize power usage and latency. The quality control technique detects error-resilient variables while in transit and computes errors based on quality. A light-weight lossy compression algorithm is used to minimize the packet size for the error-resilient variables. Analysis shows that the suggested technique performs better than other techniques. Though it reduces latency efficiently, it also raises packet loss in the network

Zheng and Louri, [13] combined Dynamic Voltage and Frequency Scaling (DVFS) and power gating to improve the performance and NoC power saving. The Agile-designed solution uses a reinforcement-based control policy and design architecture to avoid adverse network effects. A bypass mechanism that is straightforward guarantees network connectivity with powered-off routers. The pipeline stages are bypassed, allowing network latency to be effectively minimized with an optimized pipeline. A reversible link channel buffer adapts throughput dynamically. An ANN-based model of reinforcement learning is used to forecast NoC traffic, facilitating optimal power gating and runtime configuration adjustments. But the ANN-based model suffers from overfitting and imbalance of data within the network.

Vashist, *et al.* [14] applied machine learning method to secure against Eavesdropping and Denial of Services attack on Network on chip. The developed model protects against both internal and external attacks on NoC. Machine

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

learning based error correction method is applied for burst error detection and correction. The lightweight data scrambling attack and low-latency method is applied to secure for the DoS attack on NoC. The developed method has effective performance on the attack detection and prevention on NoC. The model is characterized by excessive power consumption, and the decision tree model also shows inconsistent behavior in the system.

Lee and Han [15] designed a Q-learning-based adaptive routing protocol for NoC, taking traffic and thermal into account. The approach provides deadlock-free routing based on topology knowledge and Runtime Thermal Management (RTM) for better throughput and traffic balance. The Q-learning method is plagued by overfitting and imbalanced data during classification.

Choi *et al.* [16] employed CNN models in energy-aware wireless NoC in heterogeneous many-core architectures such as CPU-GPU architecture and on-chip communication. They employed two deep CNN models, CDBNet and LeNet, with enhanced performance over optimized wireless mesh NoC. But the methodology suffers from the disadvantage of limited performance caused by overfitting.

Farahnakian, *et al.* [17] applied adaptive routing method Q-routing method for learning method in distributed traffic in wireless NoC network. The learning method selects the minimum latency path based on local and global traffic networks. The clustering method is applied to reduce the area overhead due to area consumption is important problem in Q-routing algorithm. The traffic condition is observed in the developed method to improve the performance of model. The power consumption and latency of the developed model is high for wireless NoC.

The suggested method for controlling traffic congestion made use of a machine learning-based system. Without explicit programming, machine learning allows computers to learn and grow on their own [21], which helps to alleviate congestion in the current communication network. The algorithm regularly recalculated the shortest route based on multiple parameters to prevent users from being stranded in traffic. Traffic congestion assessment has changed dramatically with the advent of big data, sophisticated sensors, vehicle tracking data, and the broad use of machine learning technology. Congestion prediction is made easier by machine learning, and the suggested study examined estimates of traffic congestion in real time.

The results reviewed previous research that used machine learning models and a variety of artificial intelligence techniques to create the best possible solutions for communication network traffic congestion [22]. The study also looked at ML-based strategies for congestion management. Conventional Transmission Control Protocol (TCP) might not work well when full utilisation of bottleneck bandwidth is required without appreciably raising end-to-end delay. The suggested solution included more recent TCP variations, such CUBIC. Instead of depending just on newer versions of TCP, there is now increased interest in using machine learning to rethink congestion management protocols. By combining and analysing a large number of current studies, the suggested approach uses machine learning to reconsider congestion reduction tactics and provide guidance for future study in the area [23].

A decentralized method was created for multi-hop networks to improve congestion control [24], taking into account various quality-of-service demands for various data flows. Gradient boosting with decision trees, an ML-inspired technique, was used to forecast the arrival times of data packets. Accurate forecasting was made possible by the evaluation of network load circumstances based on buffer occupancy and channel utilization at network nodes. To guarantee informed decision-making, a dissemination process was created to distribute these parameters among all nodes. On the basis of anticipated network demands, a transmission prioritization technique was also put into place.

The system's efficacy was validated by simulations, which showed gains in traffic differentiation, network transit time, and packet delivery ratio.

Problem Definition: Wireless router is a key component of wireless NoC that handles the large number of data packets in network. This tends to cause network congestion that increases the power consumption and affects the network performance. The proposed method involves in developing the Intelligent framework for high throughput, low power consumption and low latency for many core processor architectures. This involves in improves the wireless nodes alleviate and ensure stable performance.

PROPOSED METHOD

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

This study proposes the WFO-DNN model to improve NoC reliability and decrease power usage. The WFO-DNN model has a benefit of strong convergence and also easily escaping from local optima. A network model is implemented to test performance and efficiency with WFO-DNN. WFO is adopted for parameter setup, while the DNN is used for the prediction of fault in the network. A description of the WFO-DNN model within NoC is given in Figure 1.

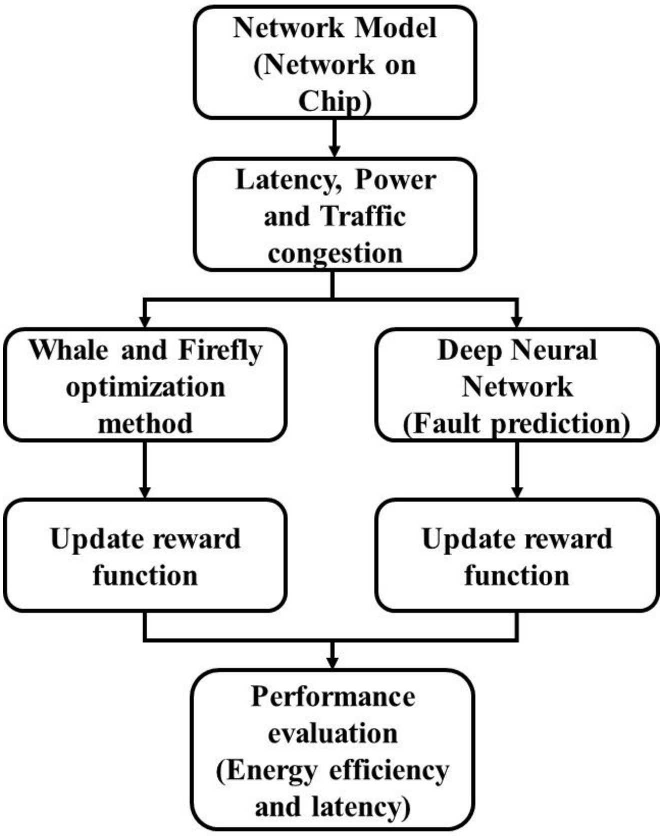


Figure 1. The overview of the WFO-DNN model

Network Model

Reinforcement learning method uses the agents to learn the environment and optimize the behaviour [11]. Agents and environment are dynamically interactive at runtime to learn the system. In network model, each agent act as a learner and discretely interact with environment at time step sequence $t = 0,1,2,3$ and so on. Agent observe current time step t from extraction of attributes in system runtime like buffer utilization, temperature and traffic congestion etc., the proposed reinforcement learning is applied to take action and applied in next time step $t_0 = t + 1$. Action of previous step is considered at time step of $t + 1$ that changes NoC attributes and provides a new state s_0. New state provide feedback to agent and incremented time step and reward r is received in agent from new state observation. The states to actions is mapped by policy π to select an action related to environmental state for increases the cumulative reward. Agent receives the rewards for actions based on traffic congestion, performance and efficiency of entire sequence [11]. The policy is evolving by reinforcement method based on agent interaction with NoC system environment.

Action-Value and Design Space Function: In network model, a vector of several NoC system features are used to denote state space S. These features are local operation temperature (attribute 20), RMC related metrics (attribute 16 to 19), output-related metrics (attributes 11 to 15), and input-related metrics (attributes 1 to 10). The agent takes action based on monitored state at each time step. The action space $A = \{a_0, a_1, a_2, \dots, a_9\}$ contains ten operation modes from which routers selects the features.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

The agent aims to optimize long-term return to represented discounted sum of future rewards. The time step t return is given in equation (1).

$$\mathbb{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} \quad (1)$$

The discount rate is defined as variable γ (where $0 \leq \gamma \leq 1$) that determines the future rewards impact to determine on the total return: as γ is near to 1, the more weight is given to future rewards to make agent less near-sighted.

The proposed method aims to improve network performance based on traffic congestion, performance and energy, reward function is designed for router i at time step t as equation (2).

$$r_{i,t} = -\log_\alpha(Latency_{i,t}) - \log_\beta(Power_{i,t}) - \log_\lambda(Congestion_{i,t}) \quad (2)$$

The Latency denotes the average end-to-end latency of specific router i, both static and dynamic power consumption is considered for power. The network traffic congestion is measured in simulation. Each individual goal importance is denoted as α, β, and λ. In this paper, three parameters of α, β, and λ is set to 1.

Whale optimization method

Whale Optimization Algorithm (WOA) is a probabilistic optimization algorithm used in this study for feature selection [16, 17]. It utilizes a population of search agents to find the global optimum for addressing optimization problems. The search is initiated with an initial set of random solutions that are iteratively improved according to the optimization conditions. With each iteration, WOA updates candidate solutions by mimicking humpback whales' hunting patterns, such as their bubble-net feeding behavior. The algorithm wisely mimics humpback whales' foraging and encircling activities, with its most fundamental mathematical expression given in Equation (3).

$$X(t+1) = \begin{cases} X^*(t) - AD & p < 0.5 \\ D' e^{bl}\cos(2\pi t) + X^*(t) & p \geq 0.5 \end{cases} \quad (3)$$

The optimal solution is the whale, and prey and distance are represented accordingly. A logarithmic spiral form is characterized by a constant, and a random number within the range [0,1] is chosen, and another random number within the interval [-1,1]. The current iteration is denoted as, and during iterations, a linear parameter comes down from 2 to 0. A random vector is also selected within the range [0,1].

The encircling process is based on the first component's equation, and the bubble-net method is used in the second. A variable alternates between these two components with equal probability.

The optimization process starts with a random set of solutions initially. In each step of optimization, search agents update their positions based on either a randomly chosen search agent or the best solution obtained so far. Other search agents update the position when $|X| > 1$ based on pivot point to maintain convergence and exploration. The pivot point plays the role in updating the position in the other situation (when $|X| < 1$).

Firefly optimization

The Firefly Algorithm is based on the collective movement of fireflies and is modeled for optimization. It is mostly controlled by two main factors: attractiveness and light intensity [18]. The process of optimization is described as follows:

Firefly light intensity is measure in equation (4).

$$I = I_0 \times e^{-\gamma r_{ij}} \quad (4)$$

Where initial light intensity is represented by I_0, the distance between firefly i and j is denoted by r_{ij}, and light absorption coefficient is γ.

A firefly attractiveness is expressed by equation (5).

$$\beta = \beta_0 \times e^{-\gamma r_{ij}^2} \quad (5)$$

where, β_0 is the attractiveness at $r_{ij} = 0$.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Equation (6) is the movement of firefly i movement to firefly j based on attraction.

$$x_i = x_i + \beta(x_j - x_i) + \alpha \times (rand - 1/2) \tag{6}$$

The positions of fireflies i and j are indicated as x_i and x_j.A random number that is uniformly distributed in the interval [0,1] is chosen, whereas the step factor is represented as α.

Deep Neural Network

A simple Autoencoder is a deep learning model that is used to reconstruct the input signal at the output by feeding it through an intermediate layer with fewer hidden nodes [19, 20]. With fewer hidden nodes, the Autoencoder learns abstract and deep features to enable accurate reconstruction. The input data is represented as $p \in \Re^N$ reduced to F features in high abstraction, so original signal is reconstructed into $z \in \Re^N$.

The Autoencoder training is involves in reproducing input signals at model output and internal units provide the original information. The layer value is applied as new reduced features to denotes the original signal p to ensure a proper reconstruction. The Autoencoder satisfies equation (7 & 8).

$$y = f(w_y p + b_y) \tag{7}$$

$$z = f(w_z y + b_z) \tag{8}$$

Where $y \in \Re^F$ is internal variable measured from p by the weights w_y, common bias b_y, the reconstruction signal $z \in \Re^N$, supposed to match p that is measured directly from output layer y by w_z and b_z. The activation function is denoted as $f(\cdot)$ that introduces the non-linearity in the network.

The error between p and z is need to be reduced to determine the optimized parameters and train the Autoencoder, as shown in equation (9).

$$\arg \min_{w_y, w_z, b_y, b_z} [error(p, z)] \tag{9}$$

Several layers are introduced in between the input and output to expand this concept as Soft Auto Encoder (SAE). Abstraction levels are used to obtain final features and SAE with two layers are usually $F < L$.

Unsupervised feature reduction is achieved through the use of Stacked Autoencoders (SAEs). Every pixel is converted into output values that match the deepest layer during training. Multiple internal coefficients are iteratively updated during the training phase of SAEs in order to reduce the error between the input pixel and its reconstructed form at the network's output. Until it drops below a predetermined level, this mistake is gradually decreased. Equation (8) states that little reconstruction error is the outcome of successful training, guaranteeing the extraction of significant internal features.

After appropriate training, there is a high degree of similarity between the reconstructed profile and the original spectral data, indicating that the SAE network successfully reconstructs the input pixel with less internal layer nodes. This indicates that the reduced features F are appropriate for feature extraction since they capture crucial information through high-level abstraction.

EXPERIMENTAL RESULTS

Network on chip requires the method for effective fault tolerance and energy consumption to improve the performance. This study proposes WFO-DNN model to improve the reliability and reduces the power consumption in NoC. Simulation has been carried out in system consists of Intel i9 processor, 128 GB of RAM, 22 GB GPU and Windows 10 OS. The proposed method is evaluated with latency, energy efficiency, and accuracy.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

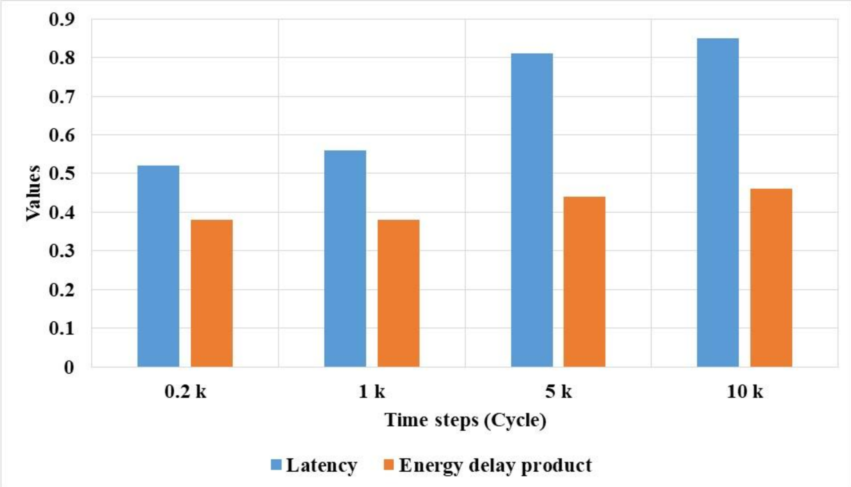


Figure 2. The proposed WFO-DNN model latency and energy delay product (Normalized)

The proposed WFO-DNN model latency and energy delay product is measured for various time steps (cycle), as shown in Figure 2. The time steps of 0.2 k to 10 k is varied to test the performance of the proposed WFO-DNN model. The proposed WFO-DNN model has advantage of maintain the efficiency, reliability and power consumption. The energy delay product is low for various time steps and latency is considerable in the network. The proposed WFO-DNN model has 0.56 latency and 0.38 energy delay product in 1k time steps.

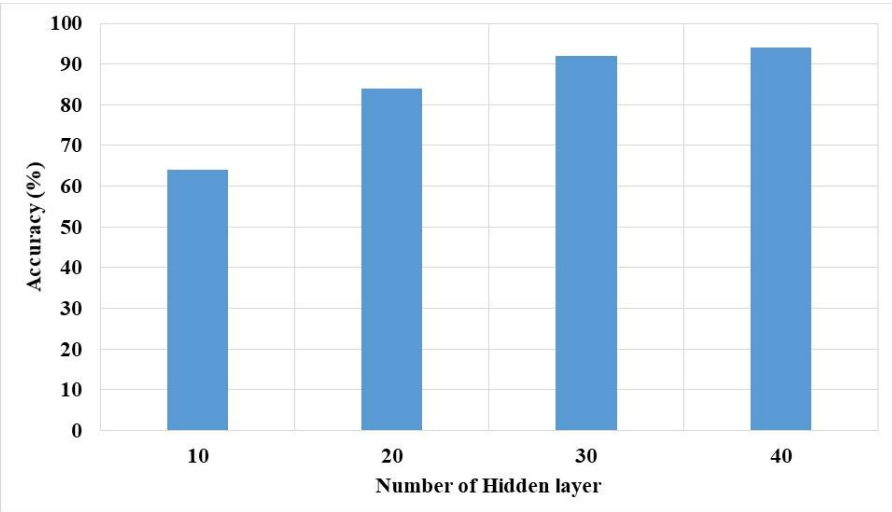


Figure 3. Accuracy of the model for various hidden layer

The DNN model performance is evaluated for various hidden layer with accuracy, as shown in Figure 3. The increases in the hidden layer increases the feature generation and analysis the data. The hidden layer is varied for 10 to 40 layer to test the performance of DNN model in fault prediction. The DNN model has higher accuracy in 30 hidden layer for fault prediction on NoC.

Table 1: Speed up execution time of proposed WFO-DNN model

Methods	Speed up of execution time
CURE [11]	1.27
Error tolerance framework [12]	1.06
Agile [13]	1.12
Decision tree [14]	1.02

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Methods	Speed up of execution time
Reinforcement Learning [15]	1.24
WFO - DNN	1.34

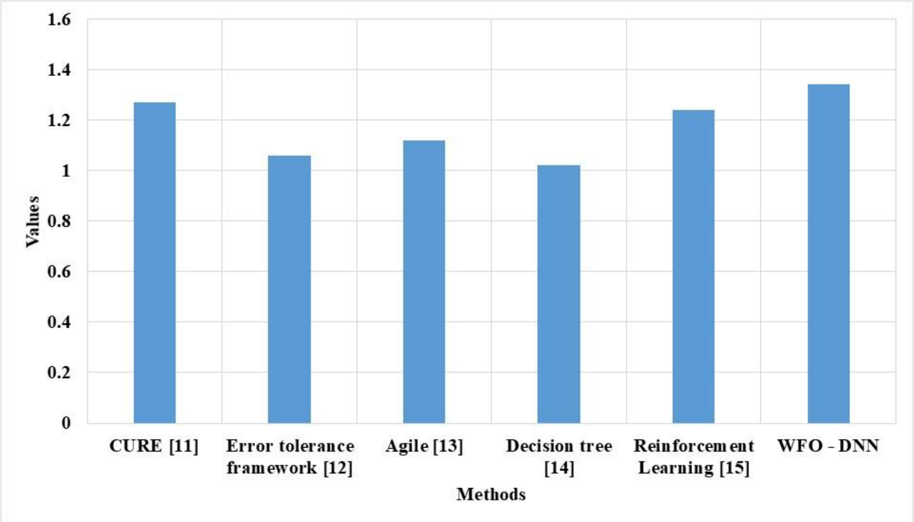


Figure 4. Proposed WFO-DNN model in speed up execution (Normalized)

The proposed WFO-DNN model is tested for fault tolerance in speed up execution and compared with existing methods, as shown in Figure 4 and Table 1. The proposed WFO-DNN model has higher performance in speed up execution compared to existing methods. The proposed WFO-DNN model has the good convergence and provides efficient classification in fault prediction. The decision tree model has the instable performance and reinforcement learning method suffers from low performance. The reinforcement learning method has overfitting problem in CURE that affects the performance of the model. The ANN model suffer from the imbalance data problem in agile method.

Table 2: Average end to end delay of proposed WFO-DNN model

Methods	Average end to end latency (ms)
CURE [11]	194
Error tolerance framework [12]	246
Agile [13]	233
Decision tree [14]	251
Reinforcement Learning [15]	204
WFO - DNN	176

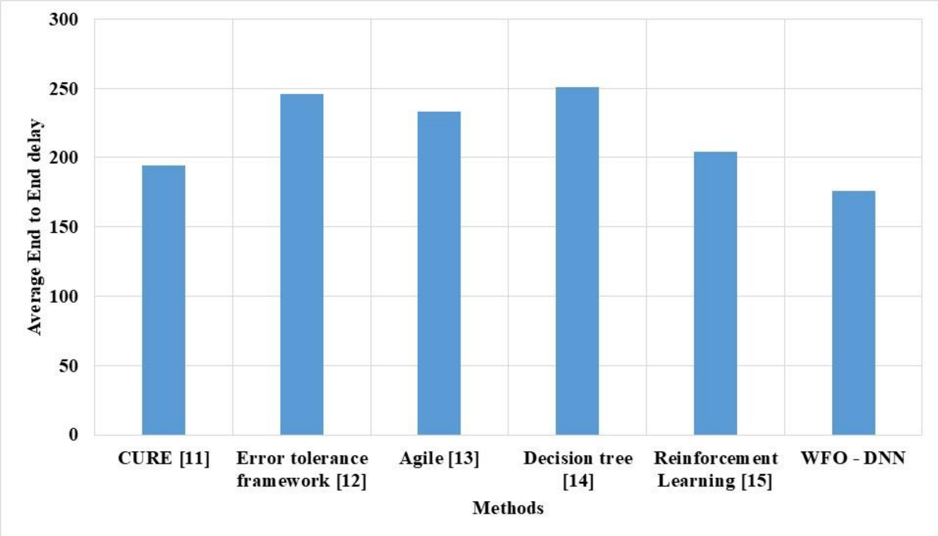
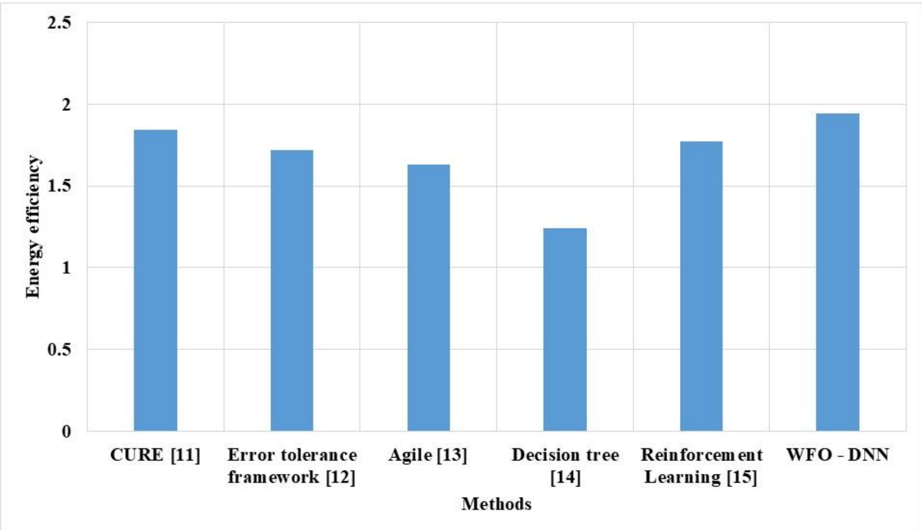


Figure 5. Average end to end delay in NoC

The proposed WFO-DNN model average end to end delay is measured and compared with existing methods, as shown in Figure 5 and Table 2. The proposed WFO-DNN model has lower average end to end delay than existing methods. The proposed WFO-DNN model has advantage of high convergence and higher performance classification in DNN model. The CURE method has limitation of overfitting problem in reinforcement learning. The decision tree method has instable performance in the fault tolerance.

Table 3: Energy efficiency of proposed WFO-DNN model

Methods	Energy efficiency
CURE [11]	1.84
Error tolerance framework [12]	1.72
Agile [13]	1.63
Decision tree [14]	1.24
Reinforcement Learning [15]	1.77
WFO - DNN	1.94



Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Figure 6. Energy efficiency in network model (Normalized)

CONCLUSION

Existing fault tolerance methods in NoC involves in applying the machine learning based models to improve the reliability. Reinforcement learning and decision tree models in fault tolerance have the limitations of overfitting problem and instable performance. In this research, WFO-DNN model is proposed to improve the reliability and reduces the power consumption of network. The WFO-DNN model have advantages of good convergence and easily escape from local optima. The DNN model provides the high accuracy in the prediction of faults in the network. The DNN model is tested with various number of hidden layer to find suitable number of hidden layer. The DNN model has higher performance for 30 number of hidden layers. The proposed WFO-DNN model has higher speed up execution, lower latency, and lower energy consumption in the network. The proposed WFO-DNN model have average end to end delay of 176 ms, 1.94 energy efficiency and existing reinforcement learning method has latency of 204 ms, and 1.77 energy efficiency. The future direction of this method involves in applying the deep feature learning method to improve the performance.

REFERENCES

[1] Arulananth, T.S., Baskar, M., SM, U.S., Thiagarajan, R., Rajeshwari, P.R., Kumar, A.S. and Suresh, A., 2021. Evaluation of low power consumption network on chip routing architecture. Microprocessors and Microsystems, 82, p.103809.

[2] Wang, X., Choi, T.M., Yue, X., Zhang, M. and Du, W., 2019. An effective optimization algorithm for application mapping in network-on-chip designs. IEEE Transactions on Industrial Electronics, 67(7), pp.5798-5809.

[3] Beechu, N.K.R., Harishchandra, V.M. and Balachandra, N.K.Y., 2018. An energy-efficient fault-aware core mapping in mesh-based network on chip systems. Journal of Network and Computer Applications, 105, pp.79-87.

[4] Paul, S., Chatterjee, N. and Ghosal, P., 2019. A permanent fault tolerant dynamic task allocation approach for Network-on-Chip based multicore systems. Journal of Systems Architecture, 97, pp.287-303.

[5] Su, N., Gu, H., Wang, K., Yu, X. and Zhang, B., 2018. A highly efficient dynamic router for application-oriented network on chip. The Journal of Supercomputing, 74(7), pp.2905-2915.

[6] Gupta, N., Vaisla, K.S. and Kumar, R., 2021. Design of a Structured Hypercube Network Chip Topology Model for Energy Efficiency in Wireless Sensor Network Using Machine Learning. SN Computer Science, 2(5), pp.1-13.

[7] Xiao, Y., Nazarian, S. and Bogdan, P., 2019. Self-optimizing and self-programming computing systems: A combined compiler, complex networks, and machine learning approach. IEEE transactions on very large scale integration (VLSI) systems, 27(6), pp.1416-1427.

[8] Beechu, N.K.R., Harishchandra, V.M. and Balachandra, N.K.Y., 2017. High-performance and energy-efficient fault-tolerance core mapping in NoC. Sustainable Computing: Informatics and Systems, 16, pp.1-10.

[9] Chatterjee, N., Paul, S. and Chattopadhyay, S., 2018. Task mapping and scheduling for network-on-chip based multi-core platform with transient faults. Journal of Systems Architecture, 83, pp.34-56.

[10] Mohammed, H.J., Flayyih, W.N. and Rokhani, F.Z., 2019. Tolerating permanent faults in the input port of the network on chip router. Journal of Low Power Electronics and Applications, 9(1), p.11.

[11] Wang, K. and Louri, A., 2020. Cure: A high-performance, low-power, and reliable network-on-chip design using reinforcement learning. IEEE Transactions on Parallel and Distributed Systems, 31(9), pp.2125-2138.

[12] Chen, Y. and Louri, A., 2020. An approximate communication framework for network-on-chips. IEEE Transactions on Parallel and Distributed Systems, 31(6), pp.1434-1446.

[13] Zheng, H. and Louri, A., 2020. Agile: A learning-enabled power and performance-efficient network-on-chip design. IEEE Transactions on Emerging Topics in Computing.

[14] Vashist, A., Keats, A., Dinakarrao, S.M.P. and Ganguly, A., 2019. Securing a wireless network-on-chip against jamming-based denial-of-service and eavesdropping attacks. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 27(12), pp.2781-2791.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

[15] Lee, S.C. and Han, T.H., 2020. Q-function-based traffic-and thermal-aware adaptive routing for 3D network-on-chip. Electronics, 9(3), p.392.

[16] Cao, Y., Li, Y., Zhang, G., Jermsittiparsert, K. and Nasseri, M., 2020. An efficient terminal voltage control for PEMFC based on an improved version of whale optimization algorithm. Energy Reports, 6, pp.530-542.

[17] Chen, H., Xu, Y., Wang, M. and Zhao, X., 2019. A balanced whale optimization algorithm for constrained engineering design problems. Applied Mathematical Modelling, 71, pp.45-59.

[18] Hassan, B.A., 2021. CSCF: a chaotic sine cosine firefly algorithm for practical application problems. Neural Computing and Applications, 33(12), pp.7011-7030.

[19] Soleymani, F. and Paquet, E., 2020. Financial portfolio optimization with online deep reinforcement learning and restricted stacked autoencoder—DeepBreath. Expert Systems with Applications, 156, p.113456.

[20] Liu, G., Bao, H. and Han, B., 2018. A stacked autoencoder-based deep neural network for achieving gearbox fault diagnosis. Mathematical Problems in Engineering, 2018.

[21] Rohilla, C., Kumar, K., & Rohilla, N. (2022). Congestion Control Using ML. 2022 COM-IT-CON. https://doi.org/10.1109/com-it-con54601.2022.9850731

[22] Kaushik, P. (2023). Congestion Articulation Control Using Machine Learning Technique. Amity Journal of Professional Practices, 3(01). https://doi.org/10.55054/ajpp.v3i01.631

[23] Wei, W., Gu, H., & Li, B. (2021). Congestion Control: A Renaissance with Machine Learning. IEEE Network, 35(4), 262–269. https://doi.org/10.1109/mnet.011.2000603.

[24] Astudillo León, J. P., de la Cruz Llopis, L. J., & Rico-Novella, F. J. (2023). A machine learning based Distributed Congestion Control Protocol for multi-hop wireless networks. Computer Networks, 231, 109813. https://doi.org/10.1016/j.comnet.2023.109813