



International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

Adaptive Neural Architectures for Real-Time Intelligent Decision Systems

Trupti V. N¹, P. Shobha Rani², M. Stanlywit³, Manoranjan Parhi⁴, T. Gomathi⁵, Anitha K⁶, Pushpalatha P⁷

¹ Assistant Professor, Department of Electrical and Electronics Engineering, Faculty of Engineering and Technology, JAIN (Deemed-to-be University), Bengaluru, Karnataka, India. Email: n.trupti@jainuniversity.ac.in, ORCID: 0000-0002-3414-100X

² Professor, Department of Artificial Intelligence and Data Science, R.M.K. Engineering College, Kavaraipettai – 601206, Tamil Nadu, India. Email: psr.ad@rmkec.ac.in

³ Associate Professor, Department of Artificial Intelligence and Data Science, R.M.K. Engineering College, Kavaraipettai – 601206, Tamil Nadu, India. Email: wst.ad@rmkec.ac.in

⁴ Professor, Centre for Data Science, Institute of Technical Education and Research, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha, India. Email: manoranjanparhi@soa.ac.in, ORCID: 0000-0002-1625-6022

⁵ Assistant Professor, Department of Electronics and Communication Engineering, Sathyabama Institute of Science and Technology, Chennai, Tamil Nadu, India. Email: gomathi.etce@sathyabama.ac.in, ORCID: 0000-0002-4371-237X

⁶ Associate Professor, Department of Management Studies, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, India. Email: anithak@maher.ac.in

⁷ Assistant Professor, Computer Science, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, India. Email: pushpalathap@maher.ac.in

Abstract

Real-time intelligent decision-making has become increasingly critical in dynamic and data-intensive environments such as finance, healthcare, and smart infrastructure. However, conventional deep learning (DL) models often struggle to adapt to rapidly changing data distributions and often fail to deliver the low-latency responses required for time-sensitive applications. To address these limitations, this research proposes an adaptive neural architecture framework designed for real-time intelligent decision systems. The proposed model integrates dynamic structural adaptation with a Quokka Swarm Optimized Adaptive Recurrent Neural Network (QSO-ARNN) to effectively capture temporal dependencies while continuously adjusting model parameters based on streaming data inputs, which contain 9,500 records from dynamic environments. Z-score normalization is utilized to normalize the dataset, while feature extraction using Linear Discriminant Analysis (LDA) improves the model's capacity to differentiate between various classes. The proposed approach is evaluated on large-scale, high-velocity data streams, simulating real-world decision-making scenarios. It enhances accuracy and responsiveness through data normalization and feature extraction for high-throughput, low-latency predictions. With a throughput of 0.1565 Gbps, a latency of 0.030 seconds, a root mean square error (RMSE) of 0.0154, a mean absolute error (MAE) of 0.0125, a mean absolute percentage error (MAPE) of 6.10%, and a high coefficient of determination that was stimulated in Python, the experimental results show excellent performance. Overall, the proposed adaptive neural architecture provides a scalable, efficient, and high-performing solution for real-time intelligent decision-making, significantly outperforming traditional static models in both accuracy and responsiveness.

Keywords: Adaptive Neural Architectures, Intelligent Decision Systems, Streaming Data Analytics, Deep Learning, Low-Latency Decision Making

1. Introduction

In sectors like smart cities, healthcare and finance, making intelligent decisions in real-time has emerged as a critical component in the dynamics. With the continued evolution of these regions, the demand for intelligent systems that could process huge amounts of data effectively has been increasing in an exponential manner. The main reason for this is the complexity and rapid pace of these fields; which necessitates the use of intelligent systems that can make accurate predictions [1]. The conventional deep learning approaches can find it hard to deal with dynamic data distribution and need low latency predictions. This leads to the creation of the need for new approaches which aid in increasing adaptability and efficiency [2]. Machine learning (ML), DL, Artificial intelligence (AI), techniques have been essential in offering automated intelligent decision-making and solving issues related to a variety of fields [3]. AI and ML have been successfully used in real-time decision-making processes, increasing decision-making efficiency, finding patterns in massive data streams, and streamlining the prediction process. In particular, DL models have revolutionized data analysis in a number of domains, including water quality monitoring, intrusion detection, and neuroimaging [4]. The application of adaptive models adds an extra layer of effectiveness for making decisions in real time due to their ability to dynamically change according to the fluidity of data, capture temporal dependencies, and be continuously adaptive [5,6]. Despite the development of AI, ML, and DL, several problems are yet to be solved to successfully apply such technologies to making intelligent decisions in real-time [7]. High-performance and prompt reaction are some of the issues that are faced by the systems where high velocity data streams are involved. The interpretation of DL algorithms is necessary for some areas, such as medicine and cybersecurity, because of ethical reasons regarding data [8].

Research aim: Developing an adaptive neural architecture framework for intelligent decision-making in real-time in dynamic, data-intensive contexts is the goal of this research. In order to efficiently capture temporal dependencies and modify model parameters depending on streaming input, the suggested method makes use of the QSO-ARNN technique.

Research organization: This research is divided into several sections. The necessity of intelligent real-time decision making in complicated, data-driven scenarios is examined in Section 1. The literature overview on feature selection techniques, optimization strategies, and adaptive neural network design is presented in Section 2. The methodology and model details of the proposed QSO-ARNN, Z-score and LDA are discussed in Section 3. The results of the analysis of the performance in terms of prediction accuracy, RMSE, MAE and MAPE have been addressed in Section 4. The findings from this research and recommendations for future research are summarized in Section 5.

2. Related work

Research in the field of intelligent systems in real time over the last two decades has been significant and is summarized in Table 1, which lists objectives, methods used, results achieved, and potential limitations. The table shows varied methods ranging from hybrid deep learning to evolutionary algorithms and includes performance indicators such as accuracy and prediction efficiency. Overfitting, dependency on data, complexity, and scaling are among the problems faced by researchers.

Table 1: Previous research on Recent Intelligent Decision-Making and Predictive Frameworks

Ref.	Research Aim	Techniques	Experimental Results	Limitations
[9]	To develop hybrid computational intelligence frameworks for real-time decision-making in dynamic and complex environments.	Reinforcement Learning (RL), Deep Neural Networks (DNNs), Fuzzy Logic	Improved decision accuracy and reduced response time; demonstrated scalability and robustness in dynamic environments across multiple application	High computational complexity, challenges in real-time deployment, and limited cross-domain generalization.

			domains (autonomous vehicles, healthcare, robotics, supply chain).	
[10]	Adaptive retirement portfolio management framework, handling market volatility, investor behavior, and regulatory changes	Evolutionary algorithms, real-time behavioral analytics, and neural architecture search (NAS) for optimal asset allocation in bonds, stocks, and alternative investments	28% increase in tax-loss harvesting efficiency, 34% decrease in portfolio volatility, and 42% quicker adaption	Computational complexity and data privacy concerns
[11]	Real-time predictive framework for underground drilling and green stormwater systems	Hybrid Long Short-Term Memory (LSTM)–Transformer architecture with attention mechanisms, online learning, and knowledge distillation	Improved predictive accuracy and faster adaptation than traditional approaches	Data drift and limited labeled data
[12]	Improve Electroencephalography (EEG)-based emotion recognition	Using a pretrained supernet and a multi-objective evolutionary algorithm, the Transformer Neural Architecture Search (TNAS) framework	Better classification accuracy, compact model size, and improved transferability for real-time Brain-Computer Interface (BCI) applications	High search time and subject-specific robustness issues
[13]	To improve sustainability, stability, and energy management by offering smart grids an adaptive real-time load forecasting and control system.	Neural Networks + Fuzzy Logic	Improved load forecasting accuracy, enhanced grid stability, and real-time responsiveness; simulations confirm effective integration with smart grid infrastructure and better energy efficiency.	Limited on scalability to large-scale real-world grids, potential computational overhead of hybrid models, and implementation complexity in practical deployments.
[14]	Examine the impact of DL and neural networks on decision-making systems	Literature review involving CNNs, fuzzy logic, and neural network approaches	DL improves decision accuracy and operational efficiency	Model interpretability and ethical concerns
[15]	Develop a real-time adaptive neural network framework for efficient and accurate AI processing in FPGA-based systems.	FPGA-based adaptive neural architecture, Dynamic Classifier Selection, k-Nearest Centroid algorithm, real-time model reconfiguration	Cut power usage by 24.5%, increased classification accuracy by 4.31%, and reduced FPGA resource utilization by 52.85% across several datasets	Limited validation on broader real-world applications, dependency on FPGA hardware, and scalability challenges for

				highly complex datasets
[16]	Improve accounting decision-making through NeuralACT	Deep Neural Network (DNN) with Bidirectional Long Short-Term Memory (Bi-LSTM) for real-time financial analytics	RMSE = 0.0181, MAE = 0.0145, MAPE = 5.14%, $R^2 = 0.99$, throughput = 0.1338 Gbps, latency = 0.039 s	Interpretability, robustness, and scalability issues

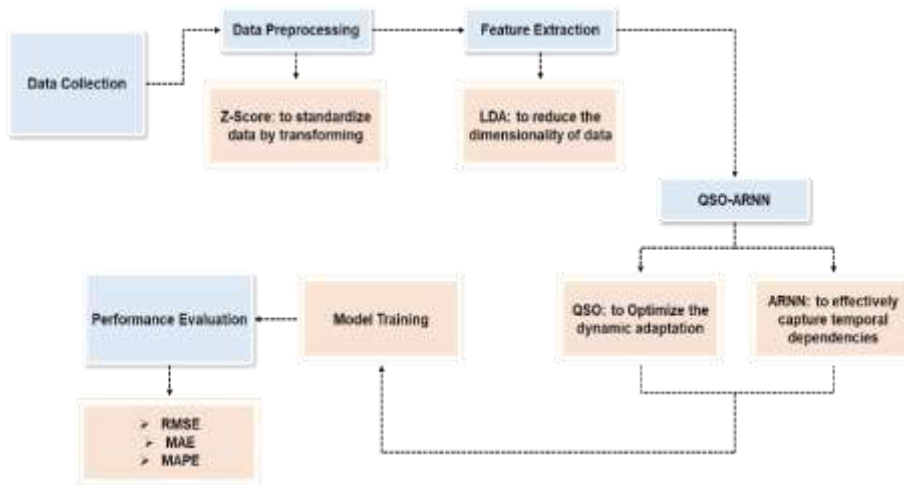
2.1 Research gap

The current deep learning frameworks applied in intelligent decision making in real-time scenarios encounter problems in adapting to varying streams of data, thus experiencing high latency, decreased accuracy, and limited scalability. Besides, the traditional frameworks fail to utilize proper pre-processing techniques that can improve the model's performance in real-time scenarios. Hence, to solve these limitations, the QSO-ARNN framework is recommended because of its ability to incorporate adaptive learning, normalization, and LDA-based feature extraction.

3. Methodology

The adaptive real-time decision support system uses adaptive neural systems and dynamic data processing techniques to effectively handle data streams that occur at high velocities. Initially, preprocessing is done using the Z-score technique. This is followed by LDA for feature extraction and dimensionality reduction. The proposed QSO-ARNN learns the temporal relationship between input variables and optimizes network parameters dynamically, as shown in Figure 1 below.

Figure 1: Methodology flow of the proposed model



3.1 Dataset description: A dataset for real-time intelligent decision-making systems was acquired from open source (<https://www.kaggle.com/datasets/colabsss/adaptive-decision-intelligence-dataset>). It consists of 9,500 records from dynamic environments. It includes measures related to system behaviors, prediction error rates, latency, throughput patterns, and adaptive response indicators. 70% of the dataset is used for training, while 30% is used for testing. Adaptive neural architecture research, real-time decision analysis, anomaly detection, sequential learning, adaptive optimization, and dynamic behavioral modeling in low-latency environments all benefit from it.

3.2 Data preprocessing using Z-Score normalization: A statistical technique called the Z-score normalization approach aids in transforming the dataset into a standardized distribution with a mean of zero and a standard

deviation of one. To achieve uniformity and improve the efficacy of models in a real-time decision system, the Z-score method is used to normalize data.

$$Y_{score} = \frac{rss(m) - \mu}{\sigma} \quad (1)$$

$$rss_{j,i} = [rss(1), rss(2), \dots, rss(M)] \quad \text{for } 1 \leq m \leq M \quad (2)$$

The Equation (1) Y_{score} defines the Z-score for the m th observation, where $rss(m)$ is the residual sum of squares for that point. Here, μ represents the mean of all RSS values, and σ is their standard deviation. The vector notation $rss_{j,i} = [rss(1), rss(2), \dots, rss(M)]$ lists all RSS values for indices j and i , where M is the total number of observations, allowing for the standardization of errors. This allows the user to compare the extent to which each residual is an outlier in the entire data set.

3.3 Feature extraction using LDA: The suggested real-time intelligent decision system uses LDA to minimize data dimensionality and extract discriminative features while maintaining class-related information. In order to improve classification accuracy and adaptive decision-making performance, LDA aims to optimize the distance between various decision classes and decrease intra-class variability. First, each class's mean vector is calculated.

$$\mu_i = 1/m_i \sum_{e_{ji} \in g_i} e_j \quad (3)$$

$$T_A = \sum_{j=1}^d m_j (\mu_j - \mu)(\mu_j - \mu)S \quad (4)$$

$$T_x = \sum_{i=1}^d \sum_{j=1}^{m_i} (e_{ji} - \mu_i)(e_{ji} - \mu_i)S \quad (5)$$

$$C = T_x^{-1} T_A \quad (6)$$

Equation (3), represents the mean of the elements e_j in the i th group g_i , where m_i is the number of elements in that group and $\sum_{e_{ji} \in g_i} e_j$ represents the sum of all elements e_j that belong to group i . In equation (4), T_A represents the between-group scatter matrix, measuring how each group's mean deviates from the overall mean, where d is the number of groups, S is the scatter matrix, and m_j is the number of samples in the j -th group. Equation (5) T_x defines the within-group scatter matrix, capturing the variance of elements e_{ji} within each group relative to their group mean μ_i . Equation (6), C represents the value used in LDA to maximize class separability.

3.4 Real-time intelligent decision system using QSO-ARNN algorithm: An online optimization algorithm for decision-making, the QSO-ARNN technique takes time dependency between sequential data into consideration by structurally evolving itself. Because it immediately solves the goals of precise, low-latency, and adaptive decision-making in dynamic, streaming data contexts, QSO-ARNN is suggested. It integrates adaptive recurrent architectures to continually learn and react in real-time with swarm intelligence for optimization.

3.4.1 ARNN for Real-Time Decision-Making: The ARNN is employed in the proposed model to perform real-time intelligent decision-making by learning temporal dependencies and dynamically adapting to changing data patterns. The primary objective of the ARNN architecture is to improve prediction accuracy, reduce decision latency, and enhance adaptive learning capability for high-velocity streaming data environments. The ARNN architecture consists of three major layers: an input layer, a hidden layer, and an output layer, as provided in Equations (7) and (8).

$$z_{mn} = x_1 g_1 + x_2 g_2 + \dots + x_M g_M = \sum_{j=1}^M x_j g_j \quad (7)$$

$$g_j = g(\sum_{i=1}^N x_{ij} w_i); j = 1 \dots M \quad (8)$$

z_{mn} represents the combined output or activation for the m -th observation and n -th neuron. The term x_j denotes the j -th input feature, while g_j is the transformed response of the j -th input after passing through a nonlinear function $g(\cdot)$. Each g_j is computed as $g_j = g(\sum_{i=1}^N x_{ij} w_i)$, where x_{ij} represents the i -th weight-adjusted input contributing to the j -th transformation, w_i is the weight associated with the i -th input, N is the total number of input features, and M is the total number of transformed responses. The function $g(\cdot)$ serves as a nonlinear activation to capture complex relationships between inputs, allowing the model to combine them effectively to produce the final output z_{mn} .

3.4.2 Adaptive Feature Selection using QSO algorithm: A quokka represents a potential solution composed on chosen dataset properties in the QSO-based optimization. In order to efficiently explore the search space and improve adaptive feature optimization for real-time decision-making, the initialization procedure attempts to generate a wide variety of solutions. Temperature, humidity, nitrogen, and drought are examples of environmental elements that are employed to mimic quokka survival behavior, enabling balanced optimization under various

circumstances. In order to improve population diversity and global search capabilities while avoiding premature convergence, random initialization entails creating a variety of feature subsets. The quokka population's initialization can be expressed mathematically as:

$$T = \begin{matrix} t_{1,1} & t_{1,f} & t_{1,m} \\ t_{a,1} & t_{a,f} & t_{a,m} \\ t_{j,1} & t_{j,e} & t_{j,m} \end{matrix} \quad (9)$$

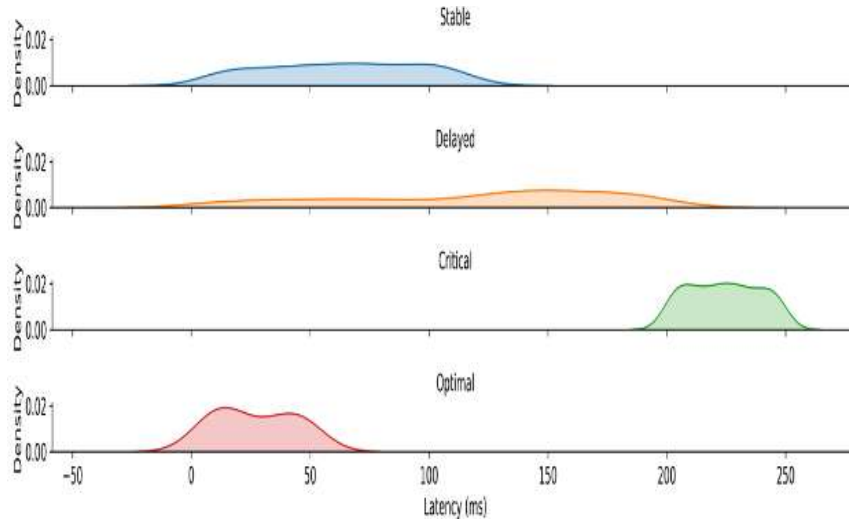
In Equation (9), T is the full data matrix and $t_{i,j}$ is the element in the i^{th} row and j -th column, and J is the total number of rows and m is the total number of columns, with some arbitrary observation in the middle, f is some intermediate feature. QSO-ARNN is a technique that supports adaptive decision-making through its process-oriented methodology. The process starts with the normalization of data through the Z-score, followed by feature extraction using the LDA approach and the initialization of a set of "quokkas". The QSO algorithm optimizes the parameters of the model, while the ARNN verifies its predictions.

4.Result and discussion

Experimental data for the proposed QSO-ARNN model are presented in this section. The evaluation includes a comparison with conventional DL models, including NeuralACT [16]. The research simulates dynamic scenarios in high-velocity, real-time streaming data environments. The experiments and performance evaluation were implemented in Python.

4.1 Data feature exploration analysis: Figure 2 presents the latency distributions for various system configurations. It depicts the working status of the system during situations of stability, delay, criticality, and optimality. The density curves display the spread and clustering of the latency values, which enable comparisons to be made among the system's various statuses.

Figure 2: Latency distribution across system states



A parallel coordinates diagram showing the correlations between several system measures, including system load, data velocity, latency, QoS index, and prediction error, is shown in Figure 3. Correlations between features that are recognized for different operational conditions can be found through the visualization. Additionally, high-dimensional correlations are considered, offering a chance to examine the influence on decision-making performance.

Figure 3: Feature interactions across system metrics

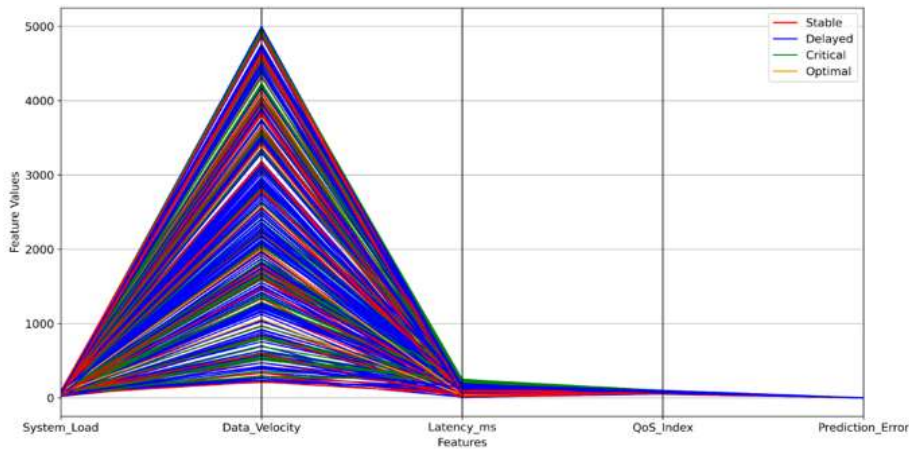
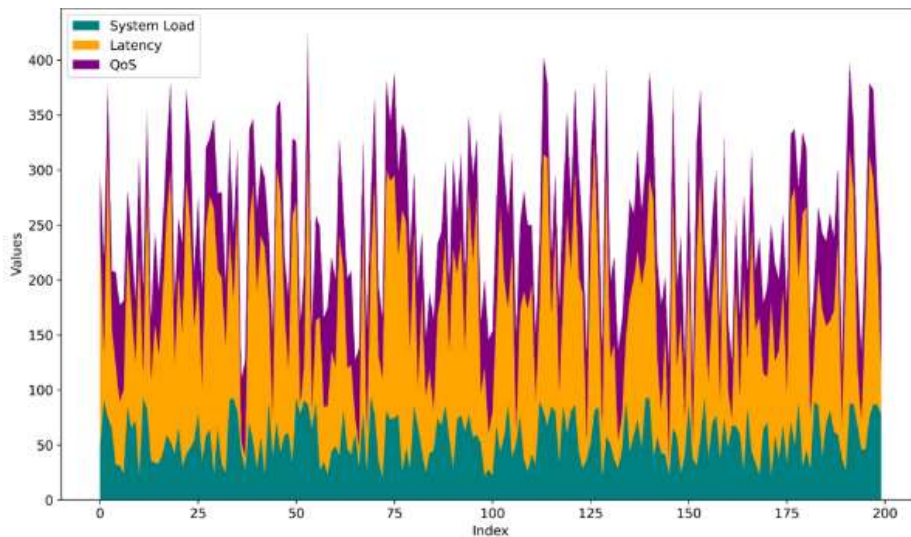


Figure 4 depicts the change in the values of the system's load, latency, and the quality-of-service index over a number of observations. The illustration helps understand the fluctuations of these factors in over-time and their interrelationship under varying conditions. The peaks and low points signify the periods of highest stress or best functioning of the system and are essential in assessing its capabilities.

Figure 4: Temporal variations in system load, latency, and QoS



4.2 Metrics explanation

Throughput: Describes how much data the system is capable of processing within a specified period of time. **Latency:** Describes how long it takes to make a prediction or make a decision. **RMSE:** Provides a measure of how far off predictions tend to be, penalizing larger differences more heavily. **MAE:** Provides a measure of the average difference between predictions and actuals. **MAPE:** Provides a measure of how far off predictions tend to be relative to actuals.

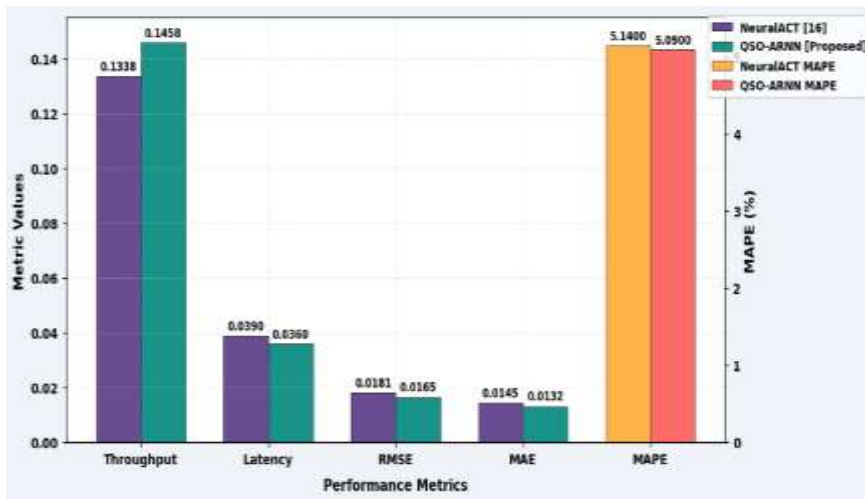
4.3 Evaluation comparison: The outcomes from the experimental analysis of the QSO-ARNN models that were trained on NeuralACT [16] data, through performance measures like throughput, latency, RMSE, MAE, and MAPE, indicate that the QSO-ARNN outperforms the NeuralACT [16]. Figures 5 and Table 2 compare the benchmark NeuralACT model with the QSO-ARNN model regarding network and prediction performance in terms of the identical dataset. The QSO-ARNN model demonstrates improved throughput from 0.1338 Gbps to 0.1458 Gbps and decreased latency from 0.039 s to 0.036 s. In terms of prediction accuracy, QSO-ARNN achieves a lower RMSE of 0.0165 and a lower MAE of 0.0132, while NeuralACT has a slight edge in MAPE at 5.14% compared to QSO-

ARNN's 5.25%. Overall, QSO-ARNN demonstrates better data throughput and execution speed with reduced absolute error rates under standard training conditions.

Table 2: Performance evaluation of the proposed model is trained using the existing dataset

Metrics	NeuralACT [16]	QSO-ARNN [Proposed]
Throughput	0.1338 Gbps	0.1458 Gbps
Latency	0.039 seconds	0.036 seconds
RMSE	0.0181	0.0165
MAE	0.0145	0.0132
MAPE	5.14%	5.09%

Figure 5: Graphical representation of the existing and proposed QSO-ARNN model trained in the existing dataset.

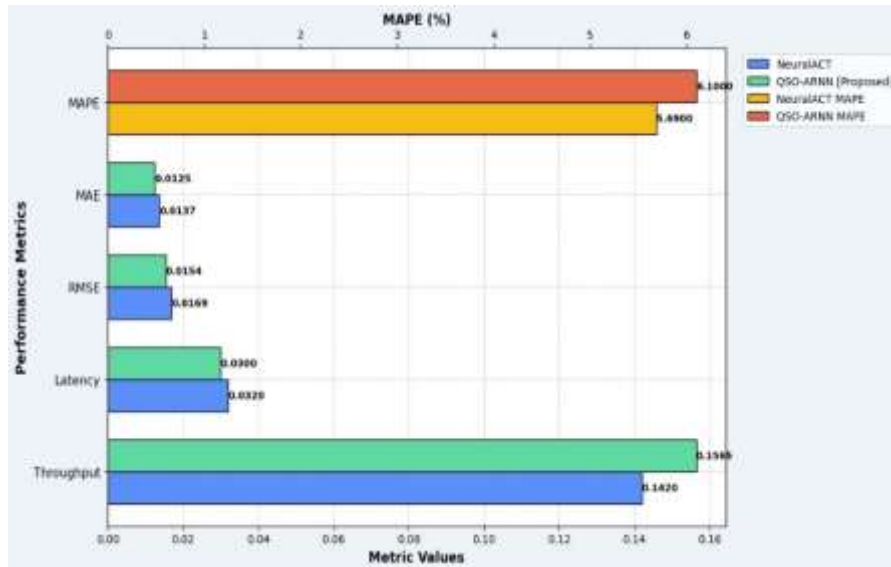


The neuralACT [16] model trained on an adaptive decision intelligence dataset, the comparison between the baseline NeuralACT model and the proposed QSO-ARNN model reveals that QSO-ARNN outperforms in predictive with a lower MAE (0.0125 vs. 0.0137) and RMSE (0.0154 vs. 0.0169). Although NeuralACT has a slight edge in MAPE (5.69% vs. 6.10%), QSO-ARNN excels in operational efficiency by reducing Latency from 0.032 to 0.030 seconds and increasing Throughput from 0.1420 Gbps to 0.1565 Gbps. In summary, the proposed model of QSO-ARNN gives better accuracy and efficiency, even though it has some percentage error, which is evident in Table 3 and Figure 6.

Table 3: Comparison of existing and proposed models trained on the adaptive decision intelligence dataset

Metrics	NeuralACT	QSO-ARNN [Proposed]
Throughput	0.1420 Gbps	0.1565Gbps
Latency	0.032 seconds	0.030 seconds
RMSE	0.0169	0.0154
MAE	0.0137	0.0125
MAPE	5.69%	6.10%

Figure 6: Presentation of NeuralACT and QSO-ARNN models on the adaptive decision intelligence dataset



There exist numerous challenges for real-time intelligent decision-making systems when working in dynamic data environments since existing deep learning models cannot cope with the problem due to their static nature and incapability of dealing with concept drift. As a result, they experience higher rates of prediction errors and become less efficient in practice, for instance, in financial and medical domains. To solve these issues, the research suggests an adaptive neural network model based on QSO-ARNN, which incorporates Quokka Swarm Optimization for structure adjustment and parameter tuning.

5. Conclusion

The proposed model of neural network with QSO-ARNN, Z-score normalization and feature extraction techniques using LDA is very effective in intelligent decision making systems in real-time. The model can learn from time-series data, process streaming data and also improves the quality of decision making with efficient preprocessing and feature selection techniques. The experiment's results demonstrate the model's high efficiency at throughput of 0.1565 Gbps, latency of 0.030 seconds, RMSE of 0.0154, MAE of 0.0125, and MAPE of 6.10, all of which indicate the model's high accuracy rate and real-time performance. But there are certain drawbacks of the technique such as high computational complexity and requirements of the quality of the dataset. Apart from the above, the model is influenced by noise in the data which can adversely impact the system's performance in very unstructured environment. While the proposed framework is challenging, some improvement in terms of adaptability and responsiveness is achieved. Future work will involve adding online learning, to further enhance the adaptability of the system without having to retrain it from the ground up. Future research allow for the system to be more robust in the presence of missing and corrupted data. Last but not least, the proposed algorithm could be expanded to be applied in real-life scenarios, such as the Internet of Things and healthcare prediction.

References

1. Thirimanne, S.P., Jayawardana, L., Yasakethu, L., Liyanaarachchi, P., and Hewage, C., 2022. Deep neural network-based real-time intrusion detection system. *SN Computer Science*, 3(2), p.145. <https://doi.org/10.1007/s42979-022-01031-1>
2. El-Shafeiy, E., Alsabaan, M., Ibrahim, M.I. and Elwahsh, H., 2023. Real-time anomaly detection for water quality sensor monitoring based on a multivariate deep learning technique. *Sensors*, 23(20), p.8613. <https://doi.org/10.3390/s23208613>
3. Huang, G.B., Zhu, Q.Y. and Siew, C.K., 2024. Real-time learning capability of neural networks. *IEEE Transactions on Neural Networks*, 17(4), pp.863-878. <https://doi.org/10.1109/TNN.2006.875974>
4. Wang, H., Han, D., Cui, M., and Chen, C., 2023. NAS-YOLOX: a SAR ship detection using neural architecture search and multi-scale attention. *Connection Science*, 35(1), pp.1-32. <https://doi.org/10.1080/09540091.2023.2257399>

5. Fathnejat, H., Ahmadi-Nedushan, B., Hosseinienejad, S., Noori, M., and Altabey, W.A., 2023. A data-driven structural damage identification approach using a deep convolutional-attention-recurrent neural architecture under temperature variations. *Engineering Structures*, 276, p.115311. <https://doi.org/10.1016/j.engstruct.2022.115311>
6. Chen, W., Milosevic, Z., Rabhi, F.A., and Berry, A., 2023. Real-time analytics: Concepts, architectures, and ML/AI considerations. *IEEE Access*, 11, pp.71634-71657. <https://doi.org/10.1109/ACCESS.2023.3295694>
7. Giannopoulos, A., Spantideas, S., Kapsalis, N., Gkonis, P., Sarakis, L., Capsalis, C., Vecchio, M., and Trakadas, P., 2022. Supporting intelligence in disaggregated open radio access networks: Architectural principles, AI/ML workflow, and use cases. *IEEE Access*, 10, pp.39580-39595. <https://doi.org/10.1109/ACCESS.2022.3166160>
8. Huang, D., Shen, D., Lu, L. and Tan, Y., 2025. Adaptive Frequency-Based Constructive Wavelet Neural Network for Nonlinear Dynamic Systems. *IEEE Transactions on Neural Networks and Learning Systems*. <https://doi.org/10.1109/TNNLS.2025.3642820>
9. Mahmoud, H.A. and Ibrahim, I.M., 2025. Adaptive Hybrid Algorithms for Real-Time Decision-Making in Autonomous Systems. *Asian Journal of Research in Computer Science*, 18(5), pp.55-64. <https://doi.org/10.9734/ajrcos/2025/v18i5638>
10. Sanepalli, U.R., 2022. Adaptive Intelligence Framework for Retirement Portfolio Management: Self-Optimizing Infrastructure for Dynamic Asset Allocation and Risk Mitigation. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)*, ISSN, pp.2456-3307. <https://doi.org/10.32628/CSEIT22557>
11. Cao, K., Zhang, T. and Huang, J., 2024. Advanced hybrid LSTM-transformer architecture for real-time multi-task prediction in engineering systems. *Scientific Reports*, 14(1), p.4890. <https://doi.org/10.1038/s41598-024-55483-x>
12. Li, C., Zhang, Z., Zhang, X., Huang, G., Liu, Y., and Chen, X., 2022. EEG-based emotion recognition via transformer neural architecture search. *IEEE Transactions on Industrial Informatics*, 19(4), pp.6016-6025. <https://doi.org/10.1109/TII.2022.3170422>
13. Wang, F. and Nishter, Z., 2024. Real-time load forecasting and adaptive control in smart grids using a hybrid neuro-fuzzy approach. *Energies*, 17(11), p.2539. <https://doi.org/10.3390/en17112539>
14. Taherdoost, H., 2023. Deep learning and neural networks: Decision-making implications. *Symmetry*, 15(9), p.1723. <https://doi.org/10.3390/sym15091723>
15. El Bouazzaoui, A., Mouhib, O. and Hadjoudja, A., 2025. NeuroAdaptiveNet: A Reconfigurable FPGA-Based Neural Network System with Dynamic Model Selection. *Chips*, 4(2), p.24. <https://doi.org/10.3390/chips4020024>
16. Theodorakopoulos, L., Theodoropoulou, A., Kampiotis, G. and Kalliampakou, I., 2025. NeuralACT: Accounting analytics using a neural network for real-time decision making from big data. *IEEE Access*, 13, pp.8621-8637. <https://doi.org/10.1109/ACCESS.2025.3526706>