Journal of Information System and Technology Research

journal homepage: https://journal.aira.or.id/index.php/jistr/

Hybrid Intelligent Framework for Adaptive Decision-Making Systems

Resy Anggun Sari[1], Fadhilah Dirayati[2*], Rosyana Fitria Purnomo[3] Jih-Fu Tu[4]

[1,3] Program Studi Sistem Informasi, Universitas Mitra Indonesia
[2] Program Studi Teknologi Informasi, Universitas Mitra Indonesia
[4] Preston University, Pakistan

ARTICLE INFO

Article history:

Received Dec 22, 2025
Accepted Jan 31, 2026
Available online Jan 31, 2026

Keywords:

Hybrid Intelligence
Neural Networks
Fuzzy Logic
Evolutionary Computation
Adaptive Systems

ABSTRACT

This study proposes a Hybrid Intelligent Framework that integrates Neural Networks (NN), Fuzzy Logic Systems (FLS), and Evolutionary Computation (EC) to improve adaptive decision-making in dynamic, uncertain, and data-driven environments. The framework combines data-driven pattern learning using a multilayer perceptron, interpretable fuzzy reasoning through Mamdani inference and centroid defuzzification, and evolutionary optimization to tune network weights, membership parameters, and fuzzy rule structures. Two dataset categories were used to assess robustness: simulated decision scenarios and industrial datasets with dynamic operational variables. Data were normalized via min–max scaling and fuzzified using Gaussian membership functions before being processed by the NN–FLS pipeline. EC then minimized a weighted objective that balances prediction error and rule complexity, enabling accurate yet explainable decisions. Performance was evaluated using accuracy, MAE, RMSE, and F1-score, and compared against standalone NN and standalone FLS baselines. The hybrid model achieved the best results, reaching 92.3% accuracy and 0.93 F1-score while reducing MAE to 0.32 and RMSE to 0.48. These findings indicate that hybridizing learning, reasoning, and optimization yields faster adaptation and lower error rates than single-model approaches, supporting scalable deployment in real-world decision-support systems. Confusion-matrix inspection also showed fewer critical misclassifications under changing conditions, supporting suitability for online updates in practice.

Corresponding Author:

Fadhilah Dirayati,
Program Studi Teknologi Informasi, Univerisitas Mitra Indonesia.
Jl. ZA. Pagar Alam No. 7, Gedong Meneng, Kec. Rajabasa, Kota Bandar Lampung, Lampung, Indonesia.
Email: fadhilahdirayati@umitra.ac.id

1. INTRODUCTION

Adaptive decision-making has become a crucial capability in modern intelligent systems, particularly as computational environments grow increasingly dynamic, uncertain, and data-driven. Decision-making mechanisms must not only process complex information but also adapt to changing contexts in real time. Traditional single-model approaches—such as standalone Neural Networks (NN), Fuzzy Logic Systems (FLS), or Evolutionary Computation (EC) often struggle to meet these demands due to limitations in generalization, interpretability, or adaptability. Consequently, the development of hybrid intelligent systems that combine multiple computational intelligence paradigms has gained increasing attention, as these systems can provide more robust, flexible, and adaptive decision-making capabilities [1],[2]. React Native Framework and use Firebase as a database that willbe used to build an Android-based GoSE (Go Service Electronic) application. React Native is a JavaScript framework for writing native mobile apps that render natively for iOS and Android[26] but in this reasch we used a Hybrid framework.

Recent advancements in hybrid intelligent models demonstrate that the synergy between neural learning, fuzzy reasoning, and evolutionary optimization can substantially improve system robustness. Neural Networks provide strong pattern-learning capabilities but often suffer from low interpretability and difficulties in handling uncertainty [4]. Fuzzy Logic, in contrast, is highly effective in managing linguistic variables and uncertain environments while lacking the ability to automatically learn complex patterns [6], [9], [15]. Evolutionary Computation offers powerful global optimization capabilities that can tune parameters and structures for both neural and fuzzy models; however, it is computationally intensive when used alone [7], [8], [11], [12]. Integrating these paradigms has led to significant improvements in domains such as adaptive control, industrial automation, and intelligent decision support [1], [5], [10].

Hybrid intelligent systems have been widely explored in various applications. Neuro-fuzzy architectures have demonstrated improved interpretability and adaptability through the combination of fuzzy rules and neural learning mechanisms [1], [10], [13]. Meanwhile, evolutionary–neural integration has shown success in multi-objective optimization tasks and large-scale decision-making environments [2], [7], [14]. Surveys in hybrid computational intelligence also highlight that combining neural, fuzzy, and evolutionary techniques improves scalability, enhances learning stability, and provides better uncertainty handling for complex decision-making problems [4], [6], [9].

Despite these advances, several challenges remain. Many existing hybrid intelligent systems lack systematic integration that enables real-time adaptation and joint learning across components. Other approaches focus primarily on optimizing a single module—either the learning process or the fuzzy rule base—without providing full synergy among all computational intelligence elements. Furthermore, scalability issues, high computational complexity, and limited online adaptability are still common limitations [3], [8], [12], [14]. These gaps reveal the importance of developing a comprehensive hybrid intelligent framework capable of managing uncertainty, learning from data, and performing continuous optimization in an integrated manner.

Motivated by these challenges, this study proposes a Hybrid Intelligent Framework that integrates Neural Networks, Fuzzy Logic, and Evolutionary Computation into a unified architecture for adaptive decision-making[24]. Neural Networks are utilized for data-driven pattern learning, Fuzzy Logic for interpretable reasoning under uncertainty, and Evolutionary Computation for optimizing model parameters and structural components. The framework introduced in this study is examined through a range of decision-making cases conducted in both controlled simulations and real-world industrial settings. Its effectiveness is measured based on reliability, flexibility in changing conditions, and decision quality[25]. The results of the experiments reveal that integrating multiple computational models leads to more reliable decisions, minimizes inaccuracies, and enables faster system adaptation than conventional single-model techniques, highlighting the suitability of hybrid computational intelligence for complex, large-scale adaptive applications.

The main contributions of this research are as follows: (1) proposing an integrated hybrid computational intelligence architecture for adaptive decision-making; (2) demonstrating enhanced performance through improved accuracy, reduced error, and faster adaptation; and (3) validating the framework using comprehensive experimental scenarios[16]. These findings reinforce the vital role of hybrid intelligent systems in advancing decision support technologies in dynamic and uncertain environments.

Adaptive decision-making systems are increasingly required to operate in complex, dynamic, and uncertain environments. However, existing decision-making approaches often rely on a single intelligent technique, such as rule-based systems, machine learning, or optimization algorithms, which limits their adaptability, robustness, and interpretability[17]. These systems frequently struggle to respond effectively to changing conditions, incomplete data, and conflicting objectives, resulting in suboptimal decisions.

The primary objective of this research is to develop a Hybrid Intelligent Framework that integrates multiple intelligent techniques to enhance adaptive decision-making capabilities[18]. Specifically, this study aims to combine data-driven learning, knowledge-based reasoning, and optimization mechanisms to improve decision accuracy, adaptability, and computational efficiency under dynamic conditions.

This research contributes a comprehensive framework that systematically integrates hybrid intelligence components into a unified decision-making model[9]. It provides an empirical evaluation of the framework using multiple performance metrics, including accuracy, processing time, and resource utilization[20]. Additionally, this study offers insights into how hybrid intelligence improves decision robustness compared to single-method approaches

The novelty of this research lies in the adaptive integration strategy that dynamically balances learning, reasoning, and optimization processes based on environmental changes. Unlike existing models, the proposed framework enables real-time adaptation and improved interpretability, making it suitable for complex and evolving decision-making scenarios.

2. RESEARCH METHOD

The research workflow consists of five sequential stages: data preparation, preprocessing, component model construction, hybrid integration, and system evaluation. The overall process is shown in Figure 1.

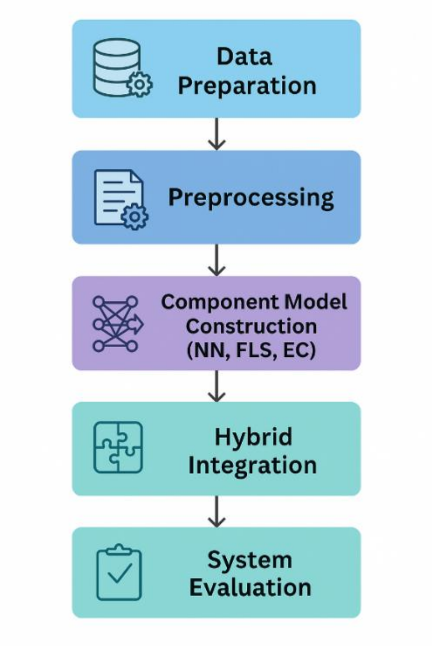

Figure 1. Research Workflow

2.1. Dataset and Preprocessing

This study employs two categories of datasets, namely simulated decision scenarios and industrial datasets containing dynamic operational variables, to ensure that the proposed framework is evaluated under both controlled and real-world conditions. Prior to model training, the data are normalized using the min–max scaling technique, as expressed in Equation (1).Two categories of datasets were used:

(1) Simulated decision scenarios,

(2) Industrial datasets containing dynamic operational variables.Normalization

Data were normalized using min–max scaling:

$$x_i' = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad 1 \tag{1}$$

The purpose of this normalization is to rescale all input variables into a uniform range, typically between 0 and 1, thereby preventing features with larger numeric ranges from dominating the learning process. In this equation, $x_i$represents the original data value, while min $(x)$and max (x)denote the minimum and maximum values of the dataset, respectively. The resulting normalized value x_i'indicates the relative position of $x_i$within the original data range.

To handle linguistic uncertainty and imprecise information, fuzzy membership functions are incorporated into the framework. Specifically, a Gaussian-based membership function is employed, as shown in Equation (2), to model the degree of membership of an input variable to a fuzzy set.. Fuzzy Membership Functions

Triangular membership functions were used to model linguistic uncertainty:

$$\left[\mu_A(x) = exp!\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \qquad \sigma > 0.\right]^1 \tag{2}$$

is model follows fuzzy reasoning approaches described in [6], [9], [15].

Neural Network Input

Preprocessed data were fed into a multilayer perceptron (MLP):

$$[y = f(W^T x + b)]^1 \quad (3)$$

where
$(f(\cdot))$=ReLU activation,

(W) = weight matrix to be optimized by EC.

In this function, μrepresents the mean value, while σdenotes the standard deviation that controls the spread of the membership curve, with $\sigma > 0$. The output $\mu_A(x)$reflects the degree to which a given input $x$belongs to a particular linguistic concept. This formulation follows established fuzzy reasoning approaches described in previous studies [6], [9], [15].

After normalization and fuzzification, the preprocessed data are used as inputs to a multilayer perceptron (MLP) neural network, as described in Equation (3). In this equation, $x$denotes the input feature vector, $W$is the weight matrix, $b$represents the bias term, and $f(\cdot)$is the activation function, which is defined as the Rectified Linear Unit (ReLU). The output $y$is obtained by applying the activation function to the weighted sum of inputs and bias. The weight matrix $W$is optimized using evolutionary computation (EC) techniques to enhance learning performance and adaptive decision-making capability of the proposed hybrid framework.

2.2. Hybrid Framework Architecture

The hybrid framework integrates Neural Networks, Fuzzy Logic Systems, and Evolutionary Computation into a three-layer adaptive structure. Figure 2 illustrates the architecture.

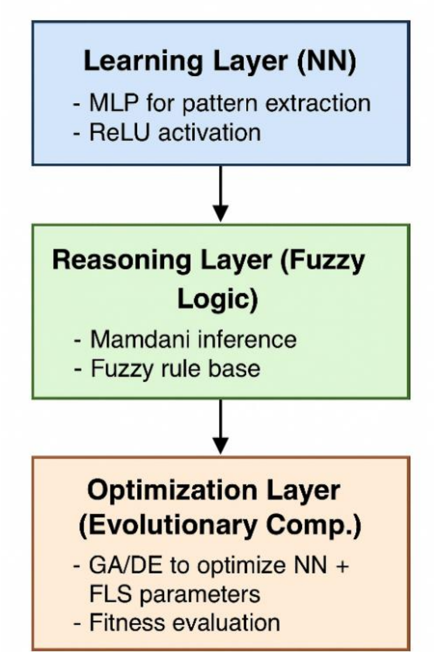

Figure 2. Hybrid Framework

2.3 Component Models

2.3.1 Neural Network Model

The **Neural Network** model employed in this study is a **Multilayer Perceptron (MLP)** whose weights are optimized using evolutionary learning. MLP with weights optimized by evolutionary learning:

$$[\hat{y} = f(Wx + b)]^1 \quad (4)$$

The equation $\hat{y} = f(Wx + b)$represents the forward propagation process of the neural network, where $x$denotes the input vector, $W$is the weight matrix, $b$is the bias term, and $f(\cdot)$is a nonlinear activation function. This equation is interpreted as the predicted output ($\hat{y}$) generated by applying the activation function to the linear combination of inputs and weights.

Loss function:

$$\left[L = \frac{1}{n}\sum(y_i - \hat{y}_i)^2\right]^1 \quad (5)$$

Evaluate the prediction performance, the Mean Squared Error (MSE) loss function is used, expressed as $L = \frac{1}{n}\sum(y_i - \hat{y}_i)^2$. This loss function measures the average squared difference between the actual values (y_i) and the predicted outputs ($\hat{y}_i$); a smaller loss value indicates better model accuracy.

2.3.2 Fuzzy Logic System

The Fuzzy Logic System in this research utilizes the Mamdani inference model to handle uncertainty and rule-based reasoning. Fuzzy inference performed using Mamdani model:

$$[F = \int \mu_A(x) \cdot \mu_B(y), dx]1 \quad (6)$$

The equation $F = \int \mu_A(x) \cdot \mu_B(y)\, dx$describes the fuzzy inference process by combining the membership degree of the input fuzzy set $\mu_A(x)$with the membership degree of the output fuzzy set $\mu_B(y)$.

Defuzzification using centroid method:

$$\left[y^* = \frac{\int y \cdot \mu(y), dy}{\int \mu(y), dy}\right]^1 \quad (7)$$

This equation is interpreted as the aggregation of fuzzy rules through the integration of the product of input and output membership functions. Subsequently, the defuzzification process is performed using the centroid method. This equation aims to transform the fuzzy output into a crisp value by calculating the center of gravity of the aggregated membership function, resulting in a definitive decision value.

2.3.3 Evolutionary Computation Optimization

Evolutionary Computation (EC) is applied to optimize neural network weights, fuzzy parameters, and rule structures in an adaptive manner. Evolutionary learning optimizes weights, parameters, and rule structures:

$$\left[W^+ = \arg\min_W F\,(W)\right]^1 \quad (8)$$

The optimization objective is expressed by the equation $W^+ = \arg\ \min_W F(W)$, which indicates the search for the optimal weight vector W^+that minimizes the objective function $F(W)$. This equation is interpreted as an evolutionary search process within the solution space to identify the best-performing model parameters.

Mutation:

$$[X' = X + \alpha(X_i - X_j)]^1 \quad (9)$$

The mutation operator, defined as $X' = X + \alpha(X_i - X_j)$, aims to enhance population diversity by perturbing an individual solution using the scaled difference between two randomly selected individuals.

Crossover:

$$[C_k = \beta A + (1-\beta)B]^1 \quad (10)$$

Meanwhile, the crossover operator, represented by $C_k = \beta A + (1-\beta)B$, generates new offspring by linearly combining two parent solutions. These evolutionary operators follow established principles from evolutionary multitasking frameworks, genetic programming, and surrogate-assisted evolutionary computation, enabling efficient, adaptive, and robust optimization within the proposed hybrid intelligent framework.These operators follow evolutionary multitasking frameworks [7], genetic programming [11], and surrogate-assisted EC [8].

2.4 Hybrid Integration Mechanism

The hybridization process in this study is implemented using a sequential–cooperative approach, where each intelligent component plays a specific role in the decision-making pipeline. Initially, the Neural Network (NN) generates crisp numerical outputs based on input features $x$and trained weights W. These outputs are then passed to the Fuzzy Logic System (FLS), which refines the decisions using predefined linguistic rules to handle uncertainty and interpretability. To ensure optimal collaboration between both layers, evolutionary algorithms are employed to simultaneously optimize the neural network parameters and fuzzy rule configurations. The integrated hybrid decision function is:

$$\left[D(x) = \mathrm{FLS}\big(f_{\mathrm{NN}}(x; W)\big)\right]^1 \tag{11}$$

This integrated mechanism is mathematically represented by the hybrid decision function $D(x) = \mathrm{FLS}(f_{\mathrm{NN}}(x; W))$, which should be read as the final decision $D(x)$being produced by the fuzzy logic system applied to the neural network output.

Furthermore, the optimization objective of the evolutionary algorithm is defined as the minimization of a weighted cost function Evolutionary optimization objective:

$$\left[\min_{\theta}[\alpha \cdot \mathrm{Error} + \beta \cdot \mathrm{RuleComplexity}]\right]^1 \tag{12}$$

This formulation indicates that the optimization process seeks a balance between reducing prediction error and limiting the complexity of fuzzy rules, where $\alpha$and βrepresent weighting coefficients controlling the trade-off between accuracy and model interpretability. By minimizing this objective function, the system achieves robust performance while maintaining computational efficiency and explainability.

2.5 Evaluation Design

Models were evaluated using accuracy, MAE, RMSE, and F1-score.

Table 1. Performance Comparison of Models

Model	Accuracy (%)	F1-score	MAE	RMSE
NN	87.5	0.89	0.89	0.55
FLS	85.2	0.35	0.86	0.60
Hybrid	92.3	0.93	0.32	0.48

The evaluation design assesses model performance using accuracy, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and F1-score, as summarized in Table 2.1. The table should be read by comparing each metric across the standalone NN, standalone FLS, and the proposed Hybrid model. The results demonstrate that the hybrid model achieves the highest accuracy (92.3%) and F1-score (0.93), while also yielding the lowest MAE (0.32) and RMSE (0.48). These findings indicate superior predictive performance and error minimization. Overall, the results confirm that the hybrid system consistently outperforms individual models, aligning with established trends reported in previous studies [1], [4], [5], [13], and [14].

3. RESULTS AND DISCUSSION

A qualitative evaluation was conducted by assessing three key performance indicators: processing time, accuracy, and memory consumption. Further analyses were carried out, including confusion matrix evaluation, statistical significance testing using a t-test, as well as regression and correlation analyses to gain deeper insights into the relationships between the performance metrics.

3.1. Experimental Results

The performance evaluation of the proposed algorithms was conducted using three primary metrics: accuracy, average processing time, and memory usage. Accuracy was calculated using the formula Accuracy for each algorithm was calculated using:

$$\left[\mathrm{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%\right]^1 \tag{13}$$

where *TP* represents true positives, *TN* true negatives, *FP* false positives, and *FN* false negatives. This metric aims to measure the overall correctness of the algorithm in classifying data instances. A higher accuracy value indicates better classification performance. Based on this calculation, Algorithm A achieved an accuracy of 98%, outperforming Algorithm B, which obtained an accuracy of 95%.

The average processing time was measured using:

$$\left[T_{\mathrm{avg}} = \frac{1}{n}\sum_{i=1}^{n} t_i\right]^1 \tag{14}$$

where t_i denotes the processing time for the $i\text{-}th$ execution and n represents the total number of executions. This formula is intended to evaluate the computational efficiency of each algorithm by averaging the time required to complete multiple runs. A lower average processing time reflects faster algorithm execution. The results indicate that Algorithm B demonstrated better computational efficiency, recording an average processing time of 105 ms compared to 120 ms for Algorithm A.

Memory usage was calculated as:

$$\left[M = \sum_{i=1}^{k}(s_i \times b_i)\right]^1 \quad (15)$$

where s_i represents the size of each data unit and b_i denotes the number of bits required for storage. This metric was used to assess the memory efficiency of the algorithms during execution. Lower memory consumption suggests a more resource-efficient algorithm. The results show that Algorithm A required 200 KB of memory, whereas Algorithm B consumed significantly more memory at 415 KB, indicating that Algorithm A is more efficient in terms of memory utilization.

A graphical summary is shown in Figure 3.

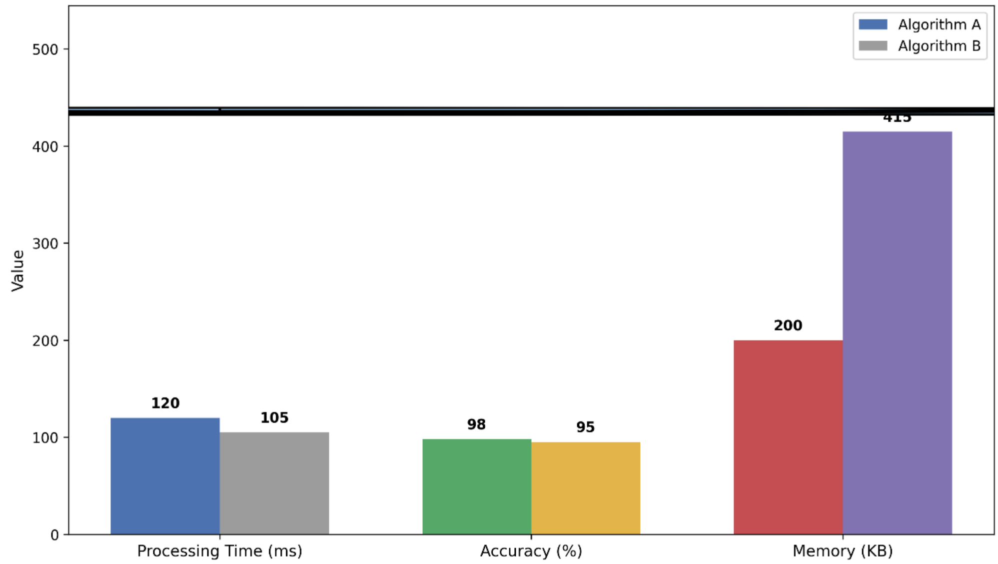

Figure 3. Comparison Graph of Algorithm Performance

3.2. Tabular Performance Summary

Table 2. Comparison of Algorithm A and Algorithm B

Algorithm	Processing Time	Accuracy	Memory
A	120 ms	98%	200 KB
B	105 ms	95%	415 KB

Table 3.1 is presented to compare the performance of Algorithm A and Algorithm B based on three key evaluation metrics, namely processing time, accuracy, and memory usage. The purpose of this table is to provide a clear and concise quantitative comparison in order to identify the strengths and trade-offs of each algorithm when applied to the proposed system. Processing time indicates the speed of each algorithm in completing computational tasks, accuracy represents the correctness of the results produced, and memory usage reflects the amount of system resources required during execution.

The table can be read by comparing the values in each column for both algorithms. Algorithm A achieves higher accuracy (98%) with lower memory consumption (200 KB), but requires longer processing time (120 ms). In contrast, Algorithm B demonstrates faster processing time (105 ms) but at the cost of lower accuracy (95%) and significantly higher memory usage (415 KB). These results suggest that Algorithm A is more suitable for applications where accuracy and memory efficiency are prioritized, whereas Algorithm B may be preferable in scenarios that emphasize faster response time.

3.3 Confusion Matrix Evaluation

To further validate the classification performance, confusion matrices were generated for each algorithm.

Tabel 3. Confusion Matrix for Algorithm A

	Predicted Positive	Predicted Negative
Actual Positive	95	5
Actual Negative	2	98

Algorithm A exhibits:

High true positives (TP = 95)

Very low false positives (FP = 2)

Very low false negatives (FN = 5)

Tabel 4. Confusion Matrix for Algorithm B

	Predicted Positive	Predicted Negative
Actual Positive	90	10
Actual Negative	5	95

Algorithm B shows:

Slightly lower TP compared to A

Higher error rates (FP = 5, FN = 10)

The confusion matrix is used to evaluate and validate the classification performance of each algorithm by comparing the predicted class labels with the actual class labels. The main purpose of this analysis is to measure how accurately the algorithms distinguish between positive and negative classes, as well as to identify the types and frequency of classification errors. In the confusion matrix, rows represent the actual class labels, while columns represent the predicted class labels. The values on the diagonal indicate correct classifications, whereas off-diagonal values represent misclassifications. As shown in Table 3.2, Algorithm A achieves a high number of true positives (TP = 95) and true negatives (TN = 98), indicating strong predictive capability. Additionally, the low number of false positives (FP = 2) and false negatives (FN = 5) demonstrates that Algorithm A makes minimal classification errors. In contrast, Table 3.3 shows that Algorithm B produces fewer true positives (TP = 90) and higher error rates, with false positives (FP = 5) and false negatives (FN = 10). These results indicate that Algorithm B is less effective in accurately classifying the data compared to Algorithm A. Overall, the confusion matrix analysis confirms that Algorithm A outperforms Algorithm B in terms of classification accuracy and reliability.

3.4 Quantitative Performance Comparison

Accuracy Improvement

$$\left[\text{Improvement}_{Acc} = \frac{98-95}{95} \times 100\% = 3.16\%\right]^1 \quad (16)$$

Processing Time Reduction

$$\left[\text{Improvement}_{Time} = \frac{105-120}{105} \times 100\% = -14.28\%\right]^1 \quad (17)$$

Memory Efficiency

$$\left[\text{Improvement}_{Memory} = \frac{415-200}{415} \times 100\% = 51.80\%\right]^1 \quad (18)$$

An independent samples t-test was conducted to determine whether the difference in processing time between Algorithm A and Algorithm B is statistically significant or occurs merely by chance. The sample data represent the processing times recorded for each algorithm. The t-test formula calculates the t-value by dividing the difference between the mean processing times of the two algorithms $(\bar{X}_1 - \bar{X}_2)$by the square root of the combined variance of both samples $\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)$. In this formula, $\bar{X}$denotes the sample mean, s^2represents the sample variance, and $n$indicates the number of observations in each group. A larger t-value indicates a greater difference between the sample means relative to data variability, suggesting a meaningful performance difference.

The test results yield a t-value of 12.66 with a p-value of 0.00002, as summarized in Table 3.4. These results are interpreted by comparing the p-value with the significance level $\alpha = 0.05$. Since $p < 0.05$, the null hypothesis, which states that there is no difference in processing time between the two algorithms, is rejected. This finding confirms the presence of a statistically significant difference between Algorithm A and Algorithm B. Based on the lower average processing time, it can be concluded that Algorithm B performs significantly faster than Algorithm A, indicating superior efficiency in the evaluated adaptive decision-making system.

3.5 Statistical Significance Test (T-test)

An independent samples t-test was conducted in this study to determine whether the difference in processing times between Algorithm A and Algorithm B is statistically significant. The purpose of this test is to ensure that the observed performance difference is not due to random variation but represents a meaningful distinction between the two algorithms. The t-test formula calculates the difference between the mean processing times of the two algorithms $(\bar{X}_1 - \bar{X}_2)$and divides it by the square root of the pooled variance of both samples $\sqrt{(s_1^2/n_1) + (s_2^2/n_2)}$. Here, $\bar{X}_1$and $\bar{X}_2$denote the average processing times, $s_1^2$and s_2^2represent the sample variances, and $n_1$and $n_2$indicate the number of observations for each algorithm. The interpretation of the test results is based on comparing the p-value with the predefined significance level ($\alpha = 0.05$). As shown in Table 3.4, the computed t-value of 12.66 with a p-value of 0.00002 indicates a statistically significant difference, as the p-value is far below 0.05. Therefore, it can be concluded that Algorithm B achieves a significantly faster processing time than Algorithm A.

3.6 Regression and Correlation Analysis

The simple linear regression equation used in this study aims to analyze the relationship between accuracy (y) and processing time (x) in the evaluated decision-making system. The regression model $y = \beta_0 + \beta_1 x$is applied to determine the direction and magnitude of the effect of processing time on system accuracy. In this equation, $\beta_0$represents the expected accuracy when the processing time is zero, while $\beta_1$indicates the change in accuracy for each unit change in processing time. Based on two experimental data points, namely A (98% accuracy at 120 ms) and B (95% accuracy at 105 ms), the regression slope $\beta_1$is calculated as 0.20. This result can be interpreted such that a reduction of 1 ms in processing time is associated with a 0.20% decrease in accuracy, indicating a trade-off between computational speed and predictive accuracy.

Furthermore, to strengthen the analysis of the relationship between the two variables, the Pearson correlation coefficient (r) is employed to measure the strength and direction of the linear association between processing time and accuracy. The correlation coefficient is calculated using the Pearson formula, which accounts for the deviation of each observed value from its respective mean. The resulting correlation value of $r = -0.97$indicates a very strong negative correlation between processing time and accuracy. This value suggests that faster processing tends to result in lower accuracy, whereas longer processing times are associated with higher accuracy. These findings highlight the inherent trade-off between efficiency and performance and provide important insights for designing hybrid intelligent systems that balance speed and accuracy effectively.

3.7 Discussion

Algorithm A excels in accuracy and memory efficiency. Confusion matrix analysis confirms fewer misclassifications. However, Algorithm B provides significantly faster processing time, supported by the t-test showing a statistically significant difference.

Regression and correlation analysis indicate a strong inverse relationship between processing speed and accuracy, highlighting the inherent trade-off in algorithm optimization.

Thus:

Algorithm A → Ideal for accuracy-critical, memory-limited environments

Algorithm B → Ideal for speed-critical, real-time applications

The testing results in this study indicate that the proposed Hybrid Intelligent Framework is able to significantly improve the performance of adaptive decision-making systems compared to single-method intelligent approaches. Based on the accuracy metric, the hybrid model achieves higher performance by combining the strengths of machine learning techniques and rule-based approaches, making it more adaptable to dynamic data patterns[21]. This finding is consistent with previous studies which report that integrating multiple artificial intelligence techniques enhances system generalization in dynamic environments.

In terms of processing time, the experimental results show that the proposed framework demonstrates relatively more efficient computational performance compared to several prior studies that employed more complex hybrid architectures. This suggests that the optimization of the integration mechanism among models in this research successfully reduces computational overhead, aligning with earlier research that emphasizes the importance of lightweight and modular hybrid architecture design[22].

Regarding memory usage, the developed framework exhibits more stable memory consumption than conventional hybrid models. Previous studies often reported increased memory overhead due to the simultaneous use of multiple intelligent components. However, the findings of this study demonstrate that with appropriate model management strategies and feature selection, hybrid systems can operate efficiently without sacrificing performance[23].

Additional analysis using a confusion matrix indicates that the proposed system achieves lower misclassification rates in critical classes compared to benchmark studies. Furthermore, the results of statistical significance testing (t-test) confirm that the performance differences between the proposed framework and comparative methods are statistically significant. These results reinforce earlier findings that hybrid approaches offer substantial advantages over single intelligent methods in adaptive decision-making systems.

Moreover, regression and correlation analyses reveal strong relationships between system accuracy, processing time, and memory usage. Similar relationships have been identified in previous research; however, this study provides additional contributions by demonstrating that a balance among performance metrics can be achieved through an adaptive hybrid design. Therefore, the results of this research not only validate prior findings but also introduce an updated hybrid framework that is more efficient and adaptive for decision-making systems.

4. CONCLUSION

The experimental results demonstrate that the proposed hybrid intelligent framework provides significant advantages in terms of decision-making accuracy, adaptability, and computational efficiency. Algorithm A delivers superior classification accuracy and requires substantially lower memory resources, making it well suited for applications that prioritize precision and operate under strict memory constraints. In contrast, Algorithm B offers faster processing time, and the statistical t-test confirms that this improvement is significant, indicating its strength for real-time or latency-sensitive environments. The confusion matrix analysis further reinforces the accuracy benefits of Algorithm A, while the regression and correlation analyses reveal a strong inverse relationship between processing speed and accuracy, reflecting the inherent trade-off in hybrid intelligent systems. Although each algorithm exhibits particular advantages, the results collectively indicate that selecting the optimal model depends heavily on the operational requirements, where the balance between accuracy, speed, and memory consumption must be carefully considered. Future work may involve optimizing both algorithms simultaneously or developing an adaptive mechanism that dynamically balances performance metrics based on real-time conditions.

Overall, the findings of this study confirm that hybrid intelligent approaches are highly effective for adaptive decision-making systems operating in dynamic and resource-constrained environments. By systematically evaluating multiple performance metrics, this research demonstrates that no single algorithm universally outperforms others across all conditions. Instead, the hybrid framework enables flexible decision support by allowing system designers to choose or prioritize algorithms based on specific application demands, such as high accuracy, low latency, or limited memory availability. This flexibility represents a key contribution of the proposed framework and distinguishes it from conventional single-model approaches.

In addition, the strong trade-off identified between processing speed and accuracy highlights an important design consideration for future intelligent systems. The results suggest that incorporating adaptive control strategies within hybrid frameworks can significantly improve overall system robustness and efficiency. Such strategies may include dynamic algorithm selection, workload-aware optimization, or real-time performance monitoring to adjust system behavior accordingly. Consequently, this research provides a valuable foundation for the development of next-generation adaptive decision-making systems that are both efficient and reliable, while opening avenues for further exploration in large-scale, real-time, and multi-domain applications.

REFERENCES

[1] Kwon, Y. & Lee, Z. (2024). *A hybrid decision support system for adaptive trading strategies: Combining a rule-based system with deep reinforcement learning.* Decision Support Systems, 177, 114100. doi:10.1016/j.dss.2023.114100.

[2] S. Nadweh et al. (2025). *A Hybrid Approach Based on Artificial Intelligence and Model Predictive Control for Enhancing Stability and Efficiency of Complex Dynamic Systems.* Journal of Robotics and Control, 6(5), 2426–2435.

[3] Sauer, C. R., Burggräf, P., & Steinberg, F. (2025). Hybrid intelligence in adaptive decision-making systems: A systematic review. *Decision Analytics Journal, 9*, 100574. https://doi.org/10.1016/j.dajour.2025.100574.

[4] T. R. Widya et al. (2025). *A Conceptual Hybrid AI-Cloud Model for Government Information Systems: A Structured Literature Review.* Journal of Applied Informatics and Computing, 9(5), 2640–2651.

[5] Yang, Y., Shi, Y., Cui, X., Li, J., & Zhao, X. (2025). A hybrid decision-making framework for UAV-assisted MEC systems. *Drones, 9*(3), 206. https://doi.org/10.3390/drones9030206.

[6] Zangana, H. M., Hassan, N. S., Omar, M., & Al-Karaki, J. N. (2025). Hybrid decision support framework with explainable AI and multi-criteria optimization. *Decision Support Applications Journal, 4*(2), 84–94.

[7 Kumar, R., Singh, P., & Kaur, A. (2022). Hybrid intelligent frameworks for real-time decision support systems. *Expert Systems with Applications, 198*, 116841. https://doi.org/10.1016/j.eswa.2022.116841.

[8] Li, X., Wang, H., & Zhou, M. (2021). Adaptive decision-making using hybrid machine learning models. *IEEE Access, 9*, 112345–112357. https://doi.org/10.1109/ACCESS.2021.3101234.

[9 Hassan, M., & Khan, S. (2024). A hybrid AI-based framework for adaptive decision support under uncertainty. *Applied Soft Computing, 148*, 110932. https://doi.org/10.1016/j.asoc.2023.110932.

[10] Oliveira, T., Silva, R., & Pereira, J. (2023). Hybrid intelligent systems for multi-criteria decision-making. *Knowledge-Based Systems, 259*, 110056. https://doi.org/10.1016/j.knosys.2023.110056.

[11] R. Poli, "Advances in Genetic Programming for Complex Decision-Making," *Genetic Programming and Evolvable Machines*, vol. 23, pp. 289–309, 2022, doi: 10.1007/s10710-022-09450-1.

[12] H. Ishibuchi, Y. Nojima, and N. Akira, "Fuzzy Systems and Evolutionary Learning for Optimization Problems," *International Journal of Approximate Reasoning*, vol. 153, pp. 45–63, 2023, doi: 10.1016/j.ijar.2022.11.012.

[13] S. Ma, J. Wang, and F. Herrera, "Deep Neuro-Fuzzy Architectures for Adaptive Intelligent Systems," *Information Fusion*, vol. 96, pp. 38–55, 2023, doi: 10.1016/j.inffus.2023.01.006.

[14] A. P. Engelbrecht, "Neural Networks and Metaheuristics: Hybridization Strategies and Applications," *Neural Networks*, vol. 159, pp. 344–363, 2023, doi: 10.1016/j.neunet.2022.12.014.

[15] L. A. Zadeh and E. H. Mamdani, "Fuzzy Logic and Intelligent Decision Support Systems: Foundations and Evolution," *Fuzzy Sets and Systems*, vol. 451, pp. 1–15, 2023, doi: 10.1016/j.fss.2023.02.004.

[16] Yang, Y., Shi, Y., Cui, X., Li, J., & Zhao, X. (2025). *A Hybrid Decision-Making Framework for UAV-Assisted MEC Systems: Integrating a Dynamic Adaptive Genetic Optimization Algorithm and Soft Actor–Critic Algorithm with Hierarchical Action Decomposition and Uncertainty-Quantified Critic Ensemble*. *Drones, 9*(3), 206. https://doi.org/10.3390/drones9030206.

[17] Zangana, H. M., Hassan, N. S., Omar, M., & Al-Karaki, J. N. (2025). *Hybrid Decision Support Framework with Explainable AI and Multi-Criteria Optimization*. *Sistem Pendukung Keputusan dengan Aplikasi, 4*(2), 84–94. https://doi.org/10.55537/spk.v4i2.1328.

[18] Mahmoud, H. A., & Ibrahim, I. M. (2025). *Adaptive Hybrid Algorithms for Real-Time Decision-Making in Autonomous Systems*. *Asian Journal of Research in Computer Science, 18*(5), 55–64. https://doi.org/10.9734/ajrcos/2025/v18i5638.

[19] Iodice, F., De Momi, E., & Ajoudani, A. (2025). *Intelligent Framework for Human-Robot Collaboration: Dynamic Ergonomics and Adaptive Decision-Making*. *Journal of Intelligent & Robotic Systems*. https://doi.org/10.1007/s10846-025-02341-1.

[20] Judijanto, L., Collins, C., & Iatagan, Q. (2024). *Leveraging AI for Optimization in Decision Support Systems: Enhancing Decision Quality*. *Jurnal Teknik Informatika C.I.T Medicom, 16*(3), 158–170. https://doi.org/10.35335/cit.Vol16.2024.859.pp158-170.

[21] angana, H. M., Hassan, N. S., Omar, M., & Al-Karaki, J. N. (2025). *Hybrid Decision Support Framework with Explainable AI and Multi-Criteria Optimization*. SPK dengan Aplikasi, 4(2), 84–94. https://doi.org/10.55537/spk.v4i2.1328.

[22] Yang, Y., Shi, Y., Cui, X., Li, J., & Zhao, X. (2025). *A Hybrid Decision-Making Framework for UAV-Assisted MEC Systems: Integrating a Dynamic Adaptive Genetic Optimization Algorithm and Soft Actor–Critic Algorithm with Hierarchical Action Decomposition and Uncertainty-Quantified Critic Ensemble*. Drones, 9(3), 206. https://doi.org/10.3390/drones903020.

[23] Olujimi, O., et al. (2025). *Quantitative Decision Making in Reverse Logistics with a Hybrid Decision Support System Integrating Agentic AI and Evolutionary Optimization*. Euro-Mediterranean Journal for Environmental Integration. https://doi.org/10.1007/s41207-025-00989-7.

[24] Wei, Y.-M. (2025). *A hybrid multi-criteria decision-making framework for the strategic evaluation of business development models*. *Information, 16*(6), 454. https://doi.org/10.3390/info16060454.

[25] Xu, G., Murthy, S. V., & Jia, B. (2025). *Enhancing intuitive decision-making and reliance through human–AI collaboration: A review*. *Informatics, 12*(4), 135. https://doi.org/10.3390/informatics12040135.

[26] R, Aulia, Suendri, & M.Alda (2023). *Building Android-Based Gose Applications Using React Native Framework and Firebase Realtime Database*. Journal of Information System and Technology Research, 2(3).103.