International Journal of Robotics and Control Systems
Vol. 5, No. 1, 2025, pp. 625-639
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

Neural Network Architectures for UAV Path Planning: A Comparative Study with A* Algorithm as Benchmark

Gregorius Airlangga [a,1,*], Julius Bata [a,2], Oskar Ika Adi Nugroho [b,3], Lai Ferry Sugianto [c,4]

[a] Department of Information Systems, Universitas Katolik Indonesia Atma Jaya, Jakarta 12930, Indonesia
[b] Department of Electrical Engineering, National Chung Cheng University, Chiayi 621301, Taiwan
[c] Department of Business Administration, Fujen Catholic University, Taipei 24205, Taiwan
[1] gregorius.airlangga@atmajaya.ac.id; [2] julius.bata@atmajaya.ac.id; [3] oskar@alum.ccu.edu.tw; [4] 158325@mail.fju.edu.tw
* Corresponding Author

ARTICLE INFO

Article history
Received November 20, 2024
Revised January 22, 2025
Accepted February 02, 2025

Keywords
UAV Path Planning;
Neural Networks;
A* Algorithm;
Autonomous Navigation;
Comparative Analysis

ABSTRACT

Autonomous path planning for Unmanned Aerial Vehicles (UAVs) plays a critical role in applications ranging from disaster response to urban logistics. Traditional algorithms, such as A*, are widely recognized for their reliability in generating collision-free and efficient trajectories but often struggle with scalability in complex and dynamic environments. This study evaluates the performance of several neural network architectures, including MLP-LSTM, CNN-GRU, CNN-LSTM, CNN BILSTM, and others, as potential alternatives to classical methods. A dataset of trajectories generated by the A* algorithm was used to train and benchmark the models, enabling direct performance comparison across key metrics such as path length, smoothness, clearance, collisions, and waypoint density. The results demonstrate that the MLP-LSTM model outperforms other neural architectures, producing paths that closely resemble A* trajectories with high smoothness and waypoint granularity. While some models, such as CNN-GRU and CNN-BILSTM, show promise in generating feasible paths, their performance is inconsistent across different UAV scenarios. Models like Residual CNN and Hybrid CNN-MHA failed to generate meaningful trajectories, highlighting the critical importance of architectural choices. This study underscores the potential of neural network models for UAV path planning.

1. Introduction

Path planning plays a pivotal role in robotics, autonomous systems, and unmanned aerial vehicles (UAVs), enabling these systems to navigate complex environments efficiently and safely [1]-[3]. The objective is to determine an optimal route from a start to a destination while considering constraints such as obstacles, terrain features, and energy consumption [4]-[6]. Among the classical methods, the A* algorithm stands out due to its computational efficiency and heuristic-driven approach [7]-[9]. Despite its widespread adoption, A* and other traditional algorithms often struggle with dynamic environments and complex, high-dimensional constraints, necessitating innovative approaches to path planning [5], [10], [11]. Recent advances in machine learning, particularly neural networks, have revolutionized optimization and decision-making tasks across various domains [12]-[14]. Neural networks offer the ability to learn from data and generalize across different scenarios, providing a promising alternative to heuristic-based methods [15]-[17]. By leveraging supervised and reinforcement learning techniques, neural networks have shown potential to overcome the limitations of classical methods [18]-[20]. However, a comprehensive understanding of their applicability and performance in path planning tasks remains underexplored, especially when benchmarked against datasets generated by robust algorithms like A* [11], [21], [22]. Previous studies have predominantly focused on enhancing A* with heuristic modifications or hybridizing it with metaheuristic techniques such as genetic algorithms or particle swarm optimization [23]-[25]. While these methods improve performance under specific conditions, their scalability and adaptability to diverse scenarios remain limited [26]-[28]. Concurrently, research on neural networks for path planning has often been restricted to specific architectures or problem settings, leaving a gap in the comparative evaluation of various neural network models in terms of accuracy, computational efficiency, and robustness [29]-[31].

Addressing this gap, the present study aims to evaluate the performance of multiple neural network architectures for path planning tasks using datasets generated by the A* algorithm. By systematically analyzing the capabilities of different architectures, including feedforward neural networks, convolutional neural networks, and recurrent neural networks, this research seeks to identify the optimal model for specific path planning challenges. The dataset used in this study retains the inherent complexity of A*-based solutions, providing a robust baseline for evaluating neural network performance. The key contributions of this research are threefold. First, it introduces a novel experimental framework for benchmarking neural networks on datasets generated by A*, ensuring consistency and comparability of results. Second, it provides an in-depth analysis of model performance across critical metrics such as path smoothness, computational cost, and adaptability to unseen scenarios. Third, this study contributes to the growing body of knowledge on integrating machine learning and classical optimization techniques for autonomous navigation. The remainder of this article is structured as follows. Section 2 reviews the related work on path planning and neural network-based optimization. Section 3 details the methodology, including dataset generation, neural network architecture, and evaluation metrics. Section 4 presents and discusses the experimental results, highlighting the strengths and limitations of each model. Finally, Section 5 concludes the study, offering insights and directions for future research.

2. Related Work

Path planning has been a central topic in robotics, autonomous systems, and transportation, with methods ranging from classical deterministic approaches to modern machine learning-based solutions. This section reviews the key advancements in the field, focusing on traditional path planning techniques, neural network applications in optimization, and the intersection of these methodologies.

2.1. Traditional Path Planning Methods

Classical path planning algorithms, such as Dijkstra's algorithm, A*, and Rapidly exploring Random Trees (RRT), have long been the backbone of navigation systems [32]-[34]. Among these, the A* algorithm has gained significant attention due to its balance of computational efficiency and optimality when combined with an admissible heuristic [35]-[37]. A* operates by iteratively exploring nodes with the lowest cumulative cost, guided by a heuristic function that estimates the cost to the goal [38]-[40]. Despite its effectiveness, A* faces challenges in real-time applications, particularly in dynamic or large-scale environments, due to its high computational overhead and reliance on heuristic tuning [26], [41], [42].

Variations of A*, such as D* and Anytime A*, have attempted to address these limitations [43]-[45]. D* optimizes the algorithm for dynamic environments by allowing re-evaluation of paths when the environment changes [46]-[48]. Anytime A*, on the other hand, offers a trade-off between computational time and path optimality by generating suboptimal paths early and refining them over time [49]. However, these adaptations often involve significant algorithmic complexity and may not generalize well across diverse scenarios [50]. Metaheuristic approaches, including genetic algorithms (GA), particle swarm optimization (PSO), and ant colony optimization (ACO), have been explored to overcome the deterministic limitations of classical algorithms [51]-[54]. These methods rely on stochastic processes to explore the solution space, offering better scalability and adaptability in high-dimensional and dynamic environments [55]. However, their reliance on extensive parameter tuning and convergence time constraints poses practical challenges [56].

2.2. Neural Networks in Path Planning

The advent of deep learning has opened new avenues for path planning by enabling data-driven approaches that can learn complex patterns and generalize across varying scenarios [20], [57], [58]. Neural networks (NNs), particularly deep neural networks (DNNs), have been utilized to model and solve optimization problems in navigation tasks [59]-[61]. These models leverage large datasets to capture intricate relationships between input features and output paths, providing an alternative to heuristic and rule-based methods [62]. Feedforward neural networks (FNNs) have been employed in path planning tasks, typically for learning static navigation maps or predicting costs associated with specific routes [63]. While FNNs are computationally efficient, their inability to model spatial dependencies limits their performance in complex environments [64].

Convolutional neural networks (CNNs), originally developed for image processing, have demonstrated significant promise in path planning, particularly for grid-based and map-based representations [65]. CNNs excel in capturing spatial hierarchies and local patterns, making them suitable for tasks such as obstacle detection, cost map generation, and route optimization. Studies integrating CNNs with reinforcement learning have shown enhanced performance in dynamic environments, enabling real-time adaptability [66]. Recurrent neural networks (RNNs) and their variants, such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), have been explored for sequential decision-making tasks in navigation [67]. These models are particularly effective in scenarios requiring memory of past states, such as multi-step planning in partially observable environments [68]. However, their training complexity and susceptibility to vanishing gradient issues remain barriers to widespread adoption [69]. Hybrid models combining neural networks with classical algorithms or metaheuristics have also emerged as a promising research direction [70], [71]. For instance, CNNs and RNNs have been used to learn heuristic functions for A*, enhancing its efficiency and robustness [24]. Similarly, neural networks have been integrated with PSO and GA to optimize parameter selection and accelerate convergence [72].

2.3. Comparative Studies and Performance Evaluation

While individual applications of neural networks in path planning have been extensively explored, comparative studies evaluating the relative performance of different neural architectures are limited. Existing research often focuses on specific architecture or scenarios, providing fragmented insights into the broader applicability of neural networks for path planning tasks. For instance, studies evaluating CNNs have highlighted their superiority in tasks requiring spatial pattern recognition but noted their limitations in sequential decision-making [73]-[75]. Conversely, RNNs have shown strong performance in temporal planning tasks but struggle with spatial complexity [76]. This disparity underscores the need for systematic benchmarking to identify the strengths and weaknesses of each model type. Furthermore, the use of datasets generated by classical algorithms, such as A*, for training and evaluating neural networks remains underexplored [22]. Such datasets provide a structured and consistent baseline, enabling fair and reproducible comparisons across models. Research leveraging these datasets has demonstrated the potential for neural networks to approximate or even surpass classical methods under certain conditions, but these findings are often limited to specific problem settings.

2.4. Integration of Neural Networks and Classical Methods

The integration of neural networks with classical path planning algorithms represents a growing trend aimed at combining the strengths of both approaches [19]. Neural networks can serve as heuristic predictors, cost estimators, or path optimizers, augmenting the capabilities of classical algorithms [24]. For example, deep learning models have been employed to predict heuristic values for A*, reducing the algorithm's computational overhead in large-scale environments [77]. Similarly, hybrid

Vol. 5, No. 1, 2025, pp. 625-639

frameworks combining RNNs with RRT have shown improved efficiency in dynamic and partially observable settings [78]. Despite these advancements, challenges remain in achieving seamless integration. Issues such as model interpretability, scalability to high-dimensional spaces, and the robustness of neural networks in adversarial environments require further investigation. Moreover, the lack of standardized benchmarks and evaluation metrics hinders the ability to compare and generalize findings across studies.

2.5. Research Gap

While significant progress has been made in both classical and neural network-based path planning methods, critical gaps persist in the comparative evaluation of neural architectures, the integration of machine learning with classical techniques, and the scalability of these methods to diverse scenarios. This study addresses these gaps by systematically benchmarking multiple neural network architectures using datasets generated by the A* algorithm. By providing a comprehensive analysis of model performance across key metrics, this research aims to advance the understanding and application of neural networks in path planning.

3. Methodology

This section outlines the methodology employed in this research. As presented in the Fig. 1, this phase includes the generation of a dataset for training and testing, the design of multiple neural network architectures tailored for path planning, and the evaluation metrics used to comprehensively assess the performance of each model.

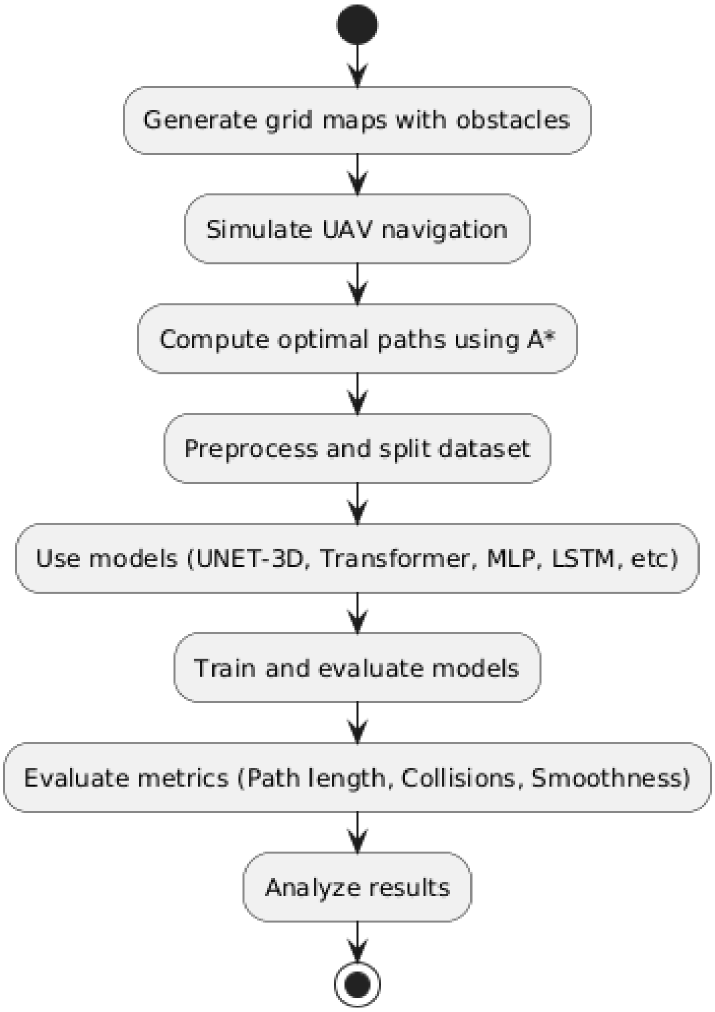

Fig. 1. Research Methodology

3.1. Dataset Generation

The dataset for this study was generated using synthetic grid-based maps, where each map represents a bounded two-dimensional space divided into discrete cells. Each cell could either be navigable or represent an obstacle. These maps simulate real-world scenarios of UAV navigation under varying levels of complexity, enabling the evaluation of path-planning models in diverse environments. The grid sizes varied, including 10×10, 50×50, and 100×100, to provide different spatial resolutions for the path-planning task. Each grid cell is associated with a state: free cells (value 0), obstacle cells (1), the start node (2), and the goal node (3). Obstacle densities of 10%, 30%, and 50% were introduced to create scenarios of varying navigational difficulty. To generate optimal paths, the A* algorithm was employed. This algorithm calculates paths by minimizing the cost function $f(n)$, defined as: $f(n) = g(n) + h(n)$, where $g(n)$ is the actual cost is from the start node to node n, and $h(n)$ is the heuristic cost estimate from n to the goal node. For this study, the heuristic function was selected as the Manhattan distance $h(n) = |x_{\text{goal}} - x_n| + |y_{\text{goal}} - y_n|$ where $(x_{\text{goal}}, y_{\text{goal}})$ and (x_n, y_n) denote the coordinates of the goal and the current node, respectively. The A* algorithm outputs an optimal path as a sequence of waypoints from the start to the goal. Each waypoint corresponds to a cell in the grid. The generated dataset contains input features such as grid encoding, obstacle locations, and heuristic values, while the outputs represent the optimal path generated by the algorithm. Each grid was flattened into a one-dimensional array to be used as input to the neural networks. Feature normalization was applied to scale the data between $[0,1]$: $x_i' = \frac{x_i - \min(x)}{\max(x) - \min(x)}$. This preprocessing step ensures that features are uniformly represented, facilitating faster convergence during model training. The dataset was split into training (80%), validation (10%), and test (10%) subsets. Stratified sampling was applied to ensure that obstacle densities and grid sizes were uniformly distributed across the splits, preventing bias.

3.2. Neural Network Architectures

To evaluate the suitability of neural networks for path planning, multiple architectures were implemented. These models were carefully selected to address various challenges of the path-planning task, such as spatial feature extraction, sequential dependencies, and global relationships.

3.2.1. UNET-3D

The UNET-3D architecture is designed for volumetric data and grid-based representations. It employs an encoder-decoder structure with symmetric skip connections. The encoder extracts hierarchical spatial features by progressively down-sampling the input grid, while the decoder reconstructs the output path by up-sampling the encoded features. The convolution operation in UNET-3D is defined as $O(i,j,k) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{p=0}^{P-1} K(m,n,p) \cdot I(i+m, j+n, k+p) + b$, where K is the kernel, I is the input feature map, and b is the bias term. The encoder and decoder are connected via skip connections, which preserve spatial details by transferring feature maps directly from the encoder to the decoder.

3.2.2. Transformer Model

Transformer architecture leverages a self-attention mechanism to capture global relationships in the input data. The self-attention mechanism is mathematically expressed as $\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V$ Here, Q, K, and V represent the query, key, and value matrices, respectively, and d_k is the dimensionality of the key vector. This architecture is particularly effective in handling long-range dependencies, making it suitable for path-planning tasks that require a global understanding of the grid.

3.2.3. MLP and LSTM

The Multi-Layer Perceptron (MLP) was implemented as a baseline model. It consists of multiple fully connected layers, with ReLU activation functions. The forward propagation in an MLP is defined as: $z^l = W^l a^{l-1} + b^l, \quad a^l = \sigma(z^l)$ where z^l is the pre-activation output at layer l, W^l and b^l are the weights and biases, and σ is the activation function. The LSTM model captures temporal dependencies in the sequence of waypoints. It uses gating mechanisms, defined as (1)-(5).

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{1}$$

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right) \tag{2}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \tag{4}$$

$$\tanh(W_c[h_{t-1}, x_t] + b_c) \tag{5}$$

where the hidden state h_t is given by $h_t = o_t \odot \tanh(c_t)$.

3.3. Evaluation Metrics

The performance of each model was evaluated using several metrics. Path length (L) was computed as $L = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$ .Collisions were counted as the number of instances where the path intersected with obstacles. Clearance was measured as the average and minimum distance from the path to the nearest obstacle $C_{\min} = \min_{o \in O} \sqrt{(x - x_o)^2 + (y - y_o)^2}$. Next, smoothness quantified the angular deviation between consecutive path segments: $S = \sum_{i=2}^{n} \arccos\left(\frac{v_{i-1} \cdot v_i}{|v_{i-1}||v_i|}\right)$. Travel time was directly proportional to path length and inversely proportional to UAV speed. Finally, the waypoint count measured the number of discrete waypoints in the planned path, reflecting the resolution of the solution.

4. Results and Discussion

This section presents the results of evaluating multiple neural network architectures on the multi-UAV path planning task as presented in Fig. 1 and Table 1, Table 2, Table 3, Table 4, Table 5, Table 6, Table 7, Table 8, Table 9, Table 10. The models analyzed include UNET 3D, Transformer Model, MLP-LSTM, CNN-GRU, CNN-LSTM, Residual CNN, MLP Model, Simple CNN, CNN-BiLSTM, and Hybrid CNN-MHA. The performance of each model was assessed based on key metrics, including path length, collisions, average clearance, minimum clearance, smoothness, travel time, and waypoint count. Across all UAVs, the model produced a path length of zero and recorded no collisions or travel time, suggesting that it failed to construct actionable trajectories. Despite these shortcomings, UNET-3D demonstrated moderate performance in maintaining average clearance, with UAV 2 achieving the highest value of 1.100, followed by UAV 3 at 0.934. This indicates that the model prioritized obstacle avoidance but lacked the structural capability to generate meaningful paths. The waypoint counts for all UAVs remained fixed at 10, which suggests that the model defaulted to a baseline output, further evidencing its inability to adaptively generate paths based on spatial configurations.

The Transformer model exhibited similar limitations to UNET-3D, as it also generated zero path lengths, collisions, and travel times for all UAVs. However, the model showed slightly improved average clearance values, particularly for UAV 2, which achieved the highest value of 1.143. This performance suggests that the Transformer model, leveraging its global attention mechanisms, was more effective in identifying safe regions within the grid. Nevertheless, the low clearance values for UAV 1 (0.442) and UAV 3 (0.652) reflect inconsistencies in its ability to uniformly handle obstacle-laden environments. The model's reliance on attention mechanisms, while beneficial for global obstacle recognition, appears insufficient for integrating local spatial features to construct usable paths. The MLP-LSTM model demonstrated a significant improvement in generating feasible paths compared to both UNET-3D and Transformer models. For UAV 1, the model produced a path length of 6.783, with no collisions and an average clearance of 2.034. The smoothness score of 1.394 for

UAV 1 was notably higher than the other models, indicating that the MLP-LSTM could generate relatively smooth paths. However, for UAV 2 and UAV 3, the model recorded one collision each, with minimum clearance values dropping to -0.022 and -0.123, respectively. These negative values indicate intersections with obstacles, suggesting that the model's obstacle avoidance capabilities were inconsistent across UAVs. While the MLP-LSTM excelled in smoothness and clearance metrics for UAV 1, the performance decline for UAV 2 and UAV 3 underscores the challenges of adapting to varying spatial configurations.

Table 1. UNET-3D Model

UAV	Path Length	Collisions	Avg Clearance	Min Clearance	Smoothness	Travel Time	Waypoint Count
UAV 1	0.0	0.0	0.634	0.634	0.0	0.0	10.0
UAV 2	0.0	0.0	1.1	1.1	0.0	0.0	10.0
UAV 3	0.0	0.0	0.934	0.934	0.0	0.0	10.0

Table 2. Transformer Model

UAV	Path Length	Collisions	Avg Clearance	Min Clearance	Smoothness	Travel Time	Waypoint Count
UAV 1	0.0	0.0	0.442	0.442	0.0	0.0	10.0
UAV 2	0.0	0.0	1.143	1.143	0.0	0.0	10.0
UAV 3	0.0	0.0	0.652	0.652	0.0	0.0	10.0

Table 3. MLP-LSTM Model

UAV	Path Length	Collisions	Avg Clearance	Min Clearance	Smoothness	Travel Time	Waypoint Count
UAV 1	6.783	0.0	2.034	0.045	1.394	6.783	50.0
UAV 2	6.729	1.0	2.075	-0.022	1.189	6.729	50.0
UAV 3	6.644	1.0	2.015	-0.123	1.165	6.644	50.0

Table 4. CNN-GRU Model

UAV	Path Length	Collisions	Avg Clearance	Min Clearance	Smoothness	Travel Time	Waypoint Count
UAV 1	6.314	2.0	0.575	-0.549	0.194	6.314	10.0
UAV 2	6.689	1.0	0.645	-0.118	0.334	6.689	10.0
UAV 3	6.324	0.0	1.025	0.556	0.703	6.324	10.0

Table 5. CNN-LSTM Model

UAV	Path Length	Collisions	Avg Clearance	Min Clearance	Smoothness	Travel Time	Waypoint Count
UAV 1	6.354	2.0	0.55	-0.553	0.693	6.354	10.0
UAV 2	6.583	1.0	0.615	-0.135	0.61	6.583	10.0
UAV 3	6.509	0.0	1.033	0.6	0.963	6.509	10.0

Table 6. Residual-CNN Model

UAV	Path Length	Collisions	Avg Clearance	Min Clearance	Smoothness	Travel Time	Waypoint Count
UAV 1	0.0	0.0	0.675	0.675	0.0	0.0	10.0
UAV 2	0.0	0.0	1.032	1.032	0.0	0.0	10.0
UAV 3	0.0	0.0	0.565	0.565	0.0	0.0	10.0

The CNN-GRU model also demonstrated the ability to generate feasible paths but showed limitations in avoiding collisions. For UAV 1, the model produced a path length of 6.314 with two collisions, an average clearance of 0.575, and a smoothness score of 0.194. UAV 2 showed slightly better performance, with a path length of 6.689, one collision, and an average clearance of 0.645.

UAV 3 achieved the best clearance among the three UAVs, with an average value of 1.025 and no collisions. However, the relatively low smoothness scores across all UAVs indicate that the CNN-GRU struggled to balance path feasibility and trajectory consistency. The integration of convolutional layers for spatial feature extraction and GRU layers for sequential modeling proved moderately effective but insufficient for ensuring high-quality paths. The CNN-LSTM model exhibited a similar trend to CNN-GRU, with feasible paths generated for all UAVs but with persistent collision rates. For UAV 1, the path length was 6.354 with two collisions and an average clearance of 0.550. UAV 2 and UAV 3 showed slight improvements in obstacle avoidance, with average clearance values of 0.615 and 1.033, respectively. The smoothness scores for CNN-LSTM were higher than those of CNN-GRU, particularly for UAV 3, which achieved a smoothness score of 0.963. This suggests that the LSTM layers contributed positively to the sequential modeling of trajectories. However, the persistent collision rates and the model's struggle to consistently maintain high clearance values highlight its limitations in effectively integrating spatial and temporal dependencies. The Residual CNN model exhibited similar deficiencies to the UNET-3D and Transformer models, producing zero path lengths, collisions, smoothness, and travel times for all UAVs. The average clearance values were moderate, with UAV 2 achieving the highest value of 1.032, followed by UAV 1 at 0.675. These results suggest that the residual connections in the model preserved spatial features but failed to translate these features into actionable trajectories. The zero values for key metrics such as path length and travel time indicate that the model was unable to generate paths, rendering it ineffective for the path-planning task. Path planning results using neural network + A* shown in Fig. 2.

Table 7. MLP Model

UAV	Path Length	Collisions	Avg Clearance	Min Clearance	Smoothness	Travel Time	Waypoint Count
UAV 1	0.0	0.0	0.739	0.739	0.0	0.0	10.0
UAV 2	0.0	0.0	0.862	0.862	0.0	0.0	10.0
UAV 3	0.0	0.0	0.614	0.614	0.0	0.0	10.0

Table 8. Simple CNN Model

UAV	Path Length	Collisions	Avg Clearance	Min Clearance	Smoothness	Travel Time	Waypoint Count
UAV 1	0.0	0.0	0.816	0.816	0.0	0.0	10.0
UAV 2	0.0	0.0	1.038	1.038	0.0	0.0	10.0
UAV 3	0.0	0.0	0.632	0.632	0.0	0.0	10.0

Table 9. CNN-BiLSTM Model

UAV	Path Length	Collisions	Avg Clearance	Min Clearance	Smoothness	Travel Time	Waypoint Count
UAV 1	7.132	3.0	0.626	-0.685	0.407	7.132	10.0
UAV 2	6.594	1.0	0.633	-0.124	0.581	6.594	10.0
UAV 3	6.038	0.0	1.104	0.684	1.271	6.038	10.0

Table 10. Hybrid CNN-MSA Model

UAV	Path Length	Collisions	Avg Clearance	Min Clearance	Smoothness	Travel Time	Waypoint Count
UAV 1	0.0	0.0	0.498	0.498	0.0	0.0	10.0
UAV 2	0.0	0.0	1.01	1.01	0.0	0.0	10.0
UAV 3	0.0	0.0	0.696	0.696	0.0	0.0	10.0

The MLP Model, like the Residual CNN, failed to generate feasible paths, producing zero path lengths, collisions, smoothness, and travel times for all UAVs. The average clearance values were slightly higher than those of the Residual CNN, with UAV 2 achieving a value of 0.862. However, the model's inability to generate paths highlights the limitations of fully connected layers in capturing the complex spatial dependencies required for path planning. The Simple CNN model performed similarly to the MLP and Residual CNN models, producing zero path lengths and travel times for all UAVs. The average clearance values were moderate, with UAV 2 achieving the highest value of 1.038. While the convolutional layers allowed the model to identify obstacle-free areas, the lack of sequential modeling capabilities prevented it from generating feasible paths. The CNN-BiLSTM model demonstrated substantial improvements in path feasibility compared to other models. For UAV 1, the model produced a path length of 7.132 with three collisions and an average clearance of 0.626. UAV 2 and UAV 3 showed better obstacle avoidance, with average clearance values of 0.633 and 1.104, respectively. The smoothness scores for UAV 2 and UAV 3, at 0.581 and 1.271, respectively, were among the highest recorded, indicating that the BiLSTM layers effectively modeled the temporal dependencies in the trajectories. Despite these advancements, the collision rate for UAV 1 highlights the challenges in integrating spatial and sequential dependencies consistently across all UAVs.

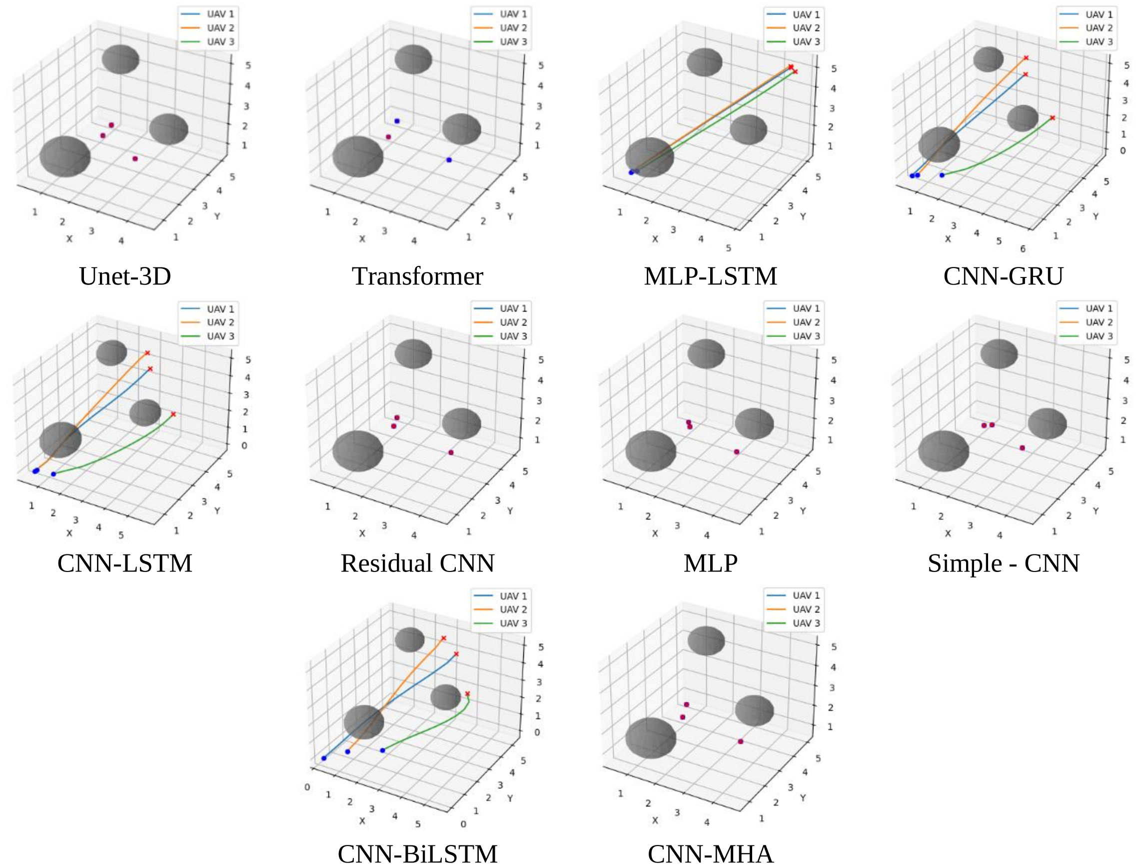

Fig. 2. Path planning results using neural network + A*

The Hybrid CNN-MHA model exhibited performance trends like the UNET-3D and Transformer models, producing zero path lengths, collisions, and travel times for all UAVs. The average clearance values were moderate, with UAV 2 achieving 1.010, but the model's inability to generate paths underscores the limitations of combining convolutional layers with attention mechanisms in this context. Overall, the results reveal that while some models, such as MLP-LSTM and CNN-BiLSTM, could generate feasible paths with relatively high smoothness and clearance, others, like UNET-3D, Transformer, and Hybrid CNN-MHA, struggled to construct usable trajectories. The findings emphasize the need for hybrid architectures that balance spatial feature extraction, temporal modeling, and global attention mechanisms to address the challenges of multi-UAV path planning effectively.

5. Conclusion

This study evaluated the performance of various neural network architectures for multi-UAV path planning, including UNET-3D, Transformer, MLP-LSTM, CNN-GRU, CNN-LSTM, Residual

CNN, MLP, Simple CNN, CNN-BiLSTM, and Hybrid CNN-MHA. The results revealed significant differences in the capabilities of these models, with certain architectures excelling in specific metrics while others struggled to generate feasible paths. The MLP-LSTM and CNN-BiLSTM models demonstrated the most promise, producing feasible paths with reasonable smoothness and clearance values. However, both models exhibited challenges in maintaining consistent obstacle avoidance, as evidenced by occasional collisions. Their ability to handle spatial and sequential dependencies made them more effective than simpler models such as MLP and Simple CNN, which failed to generate any meaningful paths. Despite their improved performance, these models still require enhancements to achieve robust path-planning capabilities across diverse scenarios.

On the other hand, models like UNET-3D, Transformer, and Hybrid CNN-MHA were able to maintain moderate clearance values, reflecting an emphasis on obstacle avoidance. However, these models consistently failed to generate traversable paths, resulting in zero path lengths and travel times across all UAVs. Their reliance on either global attention mechanisms or convolutional operations, without adequate integration of sequential dependencies, appears to limit their effectiveness in path generation. The CNN-GRU and CNN-LSTM models performed moderately, producing feasible paths but struggling with smoothness and collision avoidance. The Residual CNN model preserved spatial information effectively through skip connections but failed to utilize this information to construct actionable trajectories.

Overall, the findings highlight the need for hybrid architectures that combine the strengths of convolutional, recurrent, and attention-based mechanisms. Future research should focus on developing models that integrate spatial feature extraction, temporal dependency modeling, and global attention to address the limitations observed in this study. Additionally, improving model generalization to handle diverse grid sizes, obstacle densities, and complex navigation scenarios is essential for advancing UAV path-planning capabilities. This study underscores the potential of neural networks in solving path-planning problems while identifying critical areas for improvement. The insights gained from this research provide a foundation for the development of more robust and efficient path-planning solutions for multi-UAV systems.

Acknowledgment: The authors would like to express their deepest gratitude to Atma Jaya Catholic University of Indonesia for their invaluable support throughout the course of this research. This support includes administrative assistance, access to essential research facilities, and technical guidance, all of which have greatly contributed to the successful completion of this study.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] R. Chai, Y. Guo, Z. Zuo, K. Chen, H. Shin, and A. Tsourdos, “Cooperative motion planning and control for aerial-ground autonomous systems: Methods and applications,” *Progress in Aerospace Sciences*, vol. 146, p. 101005, 2024, https://doi.org/10.1016/j.paerosci.2024.101005.

[2] M. T. R. Khan, M. Muhammad Saad, Y. Ru, J. Seo, and D. Kim, “Aspects of unmanned aerial vehicles path planning: Overview and applications,” *International Journal of Communication Systems*, vol. 34, no. 10, p. e4827, 2021, https://doi.org/10.1002/dac.4827.

[3] C. Gómez Arnaldo, M. Zamarreño Suárez, F. Pérez Moreno, and R. Delgado-Aguilera Jurado, “Path Planning for Unmanned Aerial Vehicles in Complex Environments,” *Drones*, vol. 8, no. 7, p. 288, 2024, https://doi.org/10.3390/drones8070288.

[4] M. Chodnicki, B. Siemiatkowska, W. Stecz, and S. St\kepień, “Energy efficient UAV flight control method in an environment with obstacles and gusts of wind,” *Energies*, vol. 15, no. 10, p. 3730, 2022, https://doi.org/10.3390/en15103730.

[5] D. Debnath, F. Vanegas, J. Sandino, A. F. Hawary, and F. Gonzalez, “A Review of UAV Path-Planning Algorithms and Obstacle Avoidance Methods for Remote Sensing Applications,” *Remote Sensing*, vol. 16, no. 21, p. 4019, 2024, https://doi.org/10.3390/rs16214019.

[6] L. Zhang, A. Celik, S. Dang and B. Shihada, "Energy-Efficient Trajectory Optimization for UAV-Assisted IoT Networks," *IEEE Transactions on Mobile Computing*, vol. 21, no. 12, pp. 4323-4337, 2022, https://doi.org/10.1109/TMC.2021.3075083.

[7] Y. Chu *et al*., “Evolution of Unmanned Surface Vehicle Path Planning: A Comprehensive Review of Basic, Responsive, and Advanced Strategic Pathfinders,” *Drones*, vol. 8, no. 10, p. 540, 2024, https://doi.org/10.3390/drones8100540.

[8] J. S. S. Pavithra, A. Harini Karthika, J. Fusic S and P. Krishnapriya, "Contrastive Analysis of Path Planning Algorithms for Mobile Robots-A case study," *2023 International Conference on Energy, Materials and Communication Engineering (ICEMCE)*, pp. 1-6, 2023, https://doi.org/10.1109/ICEMCE57940.2023.10434168.

[9] Z. Hu *et al*., “Optimal Path Planning With Minimum Inspection Teams and Balanced Working Hours For Power Line Inspection,” *Frontiers in Physics*, vol. 10, p. 955499, 2022, https://doi.org/10.3389/fphy.2022.955499.

[10] M. Jones, S. Djahel, and K. Welsh, “Path-planning for unmanned aerial vehicles with environment complexity considerations: A survey,” *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–39, 2023, https://doi.org/10.1145/3570723.

[11] J. Luo, Y. Tian, and Z. Wang, “Research on Unmanned Aerial Vehicle Path Planning,” *Drones*, vol. 8, no. 2, p. 51, 2024, https://doi.org/10.3390/drones8020051.

[12] H. Taherdoost, “Deep Learning and Neural Networks: Decision-Making Implications,” *Symmetry*, vol. 15, no. 9, p. 1723, 2023, https://doi.org/10.3390/sym15091723.

[13] A. Goel, A. K. Goel, and A. Kumar, “The role of artificial neural network and machine learning in utilizing spatial information,” *Spatial Information Research*, vol. 31, no. 3, pp. 275–285, 2023, https://doi.org/10.1007/s41324-022-00494-x.

[14] S. Mishra, “A reinforcement learning approach for training complex decision making models,” *International Journal of Science and Research*, vol. 11, no. 5, pp. 2130-2137, 2022, https://www.doi.org/10.21275/SR22051120903.

[15] A. A. Juan *et al*., “A review of the role of heuristics in stochastic optimisation: From metaheuristics to learnheuristics,” *Annals of Operations Research*, vol. 320, no. 2, pp. 831–861, 2023, https://doi.org/10.1007/s10479-021-04142-9.

[16] R. Tyasnurita, E. Özcan and R. John, "Learning heuristic selection using a Time Delay Neural Network for Open Vehicle Routing," *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1474-1481, 2017, https://doi.org/10.1109/CEC.2017.7969477.

[17] C.-A. Cheng, A. Kolobov, and A. Swaminathan, “Heuristic-guided reinforcement learning,” *Machine Learning*, 2021, https://doi.org/10.48550/arXiv.2106.02757.

[18] Y. Zhang, W. Zhao, J. Wang, and Y. Yuan, “Recent progress, challenges and future prospects of applied deep reinforcement learning: A practical perspective in path planning,” *Neurocomputing*, vol. 608, p. 128423, 2024, https://doi.org/10.1016/j.neucom.2024.128423.

[19] L. Yang *et al*., “Path Planning Technique for Mobile Robots: A Review,” *Machines*, vol. 11, no. 10, p. 980, 2023, https://doi.org/10.3390/machines11100980.

[20] X. Xiao, B. Liu, G. Warnell, and P. Stone, “Motion planning and control for mobile robot navigation using machine learning: a survey,” *Autonomous Robots*, vol. 46, no. 5, pp. 569–597, 2022, https://doi.org/10.1007/s10514-022-10039-8.

[21] C. A. Moore, “Uncertainty-aware path planning on aerial imagery and unknown environments,” *Mississippi State University*, 2024, https://scholarsjunction.msstate.edu/td/6145/.

[22] Z. Zhao *et al*., "A Survey of Optimization-Based Task and Motion Planning: From Classical to Learning Approaches," *IEEE/ASME Transactions on Mechatronics*, 2024, https://doi.org/10.1109/TMECH.2024.3452509.

[23] A. G. Gad, "Particle swarm optimization algorithm and its applications: a systematic review," *Archives of Computational Methods in Engineering*, vol. 29, no. 5, pp. 2531–2561, 2022, https://doi.org/10.1007/s11831-021-09694-4.

[24] M. Abd Elaziz *et al*., "Advanced metaheuristic optimization techniques in applications of deep neural networks: a review," *Neural Computing and Applications*, vol. 33, pp. 14079-14099, 2021, https://doi.org/10.1007/s00521-021-05960-5.

[25] A. Akter *et al*., "A review on microgrid optimization with meta-heuristic techniques: Scopes, trends and recommendation," *Energy Strategy Reviews*, vol. 51, p. 101298, 2024, https://doi.org/10.1016/j.esr.2024.101298.

[26] L. E. Lwakatare, A. Raj, I. Crnkovic, J. Bosch, and H. H. Olsson, "Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions," *Information and Software Technology*, vol. 127, p. 106368, 2020, https://doi.org/10.1016/j.infsof.2020.106368.

[27] M. H. Nasir, J. Arshad, M. M. Khan, M. Fatima, K. Salah, and R. Jayaraman, "Scalable blockchains—A systematic review," *Future Generation Computer Systems*, vol. 126, pp. 136–162, 2022, https://doi.org/10.1016/j.future.2021.07.035.

[28] M. S. Aslanpour, S. S. Gill, and A. N. Toosi, "Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research," *Internet of Things*, vol. 12, p. 100273, 2020, https://doi.org/10.1016/j.iot.2020.100273.

[29] C. S. Tan, R. Mohd-Mokhtar and M. R. Arshad, "A Comprehensive Review of Coverage Path Planning in Robotics Using Classical and Heuristic Algorithms," *IEEE Access*, vol. 9, pp. 119310-119342, 2021, https://doi.org/10.1109/ACCESS.2021.3108177.

[30] J. R. Sanchez-Ibanez, C. J. Perez-del-Pulgar, and A. Garc\'\ia-Cerezo, "Path planning for autonomous mobile robots: A review," *Sensors*, vol. 21, no. 23, p. 7898, 2021, https://doi.org/10.3390/s21237898.

[31] C. Zhou, B. Huang, and P. Fränti, "A review of motion planning algorithms for intelligent robots," *Journal of Intelligent Manufacturing*, vol. 33, no. 2, pp. 387-424, 2022, https://doi.org/10.1007/s10845-021-01867-z.

[32] N. AbuJabal, T. Rabie, M. Baziyad, I. Kamel, and K. Almazrouei, "Path Planning Techniques for Real-Time Multi-Robot Systems: A Systematic Review," *Electronics*, vol. 13, no. 12, p. 2239, 2024, https://doi.org/10.3390/electronics13122239.

[33] N. Abujabal, R. Fareh, S. Sinan, M. Baziyad, and M. Bettayeb, "A comprehensive review of the latest path planning developments for multi-robot formation systems," *Robotica*, vol. 41, no. 7, pp. 2079–2104, 2023, https://doi.org/10.1017/S0263574723000322.

[34] C. Wong, "Adaptive task planning and motion planning for robots in dynamic environments," *STAX*, 2020, https://stax.strath.ac.uk/concern/theses/b8515n50j.

[35] W. Wang, B. Jia, S. P. Simonovic, S. Wu, Z. Fan, and L. Ren, "Comparison of representative heuristic algorithms for multi-objective reservoir optimal operation," *Water Resources Management*, vol. 35, pp. 2741–2762, 2021, https://doi.org/10.1007/s11269-021-02864-w.

[36] F. Geißer, P. Haslum, S. Thiébaux, and F. Trevizan, "Admissible heuristics for multi-objective planning," *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 32, pp. 100–109, 2022, https://doi.org/10.1609/icaps.v32i1.19790.

[37] A. S. Alsagri and A. A. Alrobaian, "Optimization of combined heat and power systems by meta-heuristic algorithms: An overview," *Energies*, vol. 15, no. 16, p. 5977, 2022, https://doi.org/10.3390/en15165977.

[38] C. Ntakolia and D. K. Iakovidis, "A swarm intelligence graph-based pathfinding algorithm (SIGPA) for multi-objective route planning," *Computers & Operations Research*, vol. 133, p. 105358, 2021, https://doi.org/10.1016/j.cor.2021.105358.

[39] C. G. Correa, M. K. Ho, F. Callaway, N. D. Daw, and T. L. Griffiths, "Humans decompose tasks by trading off utility and computational cost," *PLoS Computational Biology*, vol. 19, no. 6, p. e1011087, 2023, https://doi.org/10.1371/journal.pcbi.1011087.

[40] Y. Zhuang *et al*., "Toolchain*: Efficient action space navigation in large language models with a* search," *Computation and Language*, 2023, https://doi.org/10.48550/arXiv.2310.13227.

[41] R. Yu *et al.*, "Ransomware detection using dynamic behavioral profiling: A novel approach for real-time threat mitigation," *TechRxiv*, 2024, https://doi.org/10.36227/techrxiv.173047864.44215173/v1.

[42] S. MahmoudZadeh, A. Yazdani, Y. Kalantari, B. Ciftler, F. Aidarus, and M. O. Al Kadri, "Holistic Review of UAV-Centric Situational Awareness: Applications, Limitations, and Algorithmic Challenges," *Robotics*, vol. 13, no. 8, p. 117, 2024, https://doi.org/10.3390/robotics13080117.

[43] H. Lin, S. Wan, W. Gan, J. Chen and H. -C. Chao, "Metaverse in Education: Vision, Opportunities, and Challenges," *2022 IEEE International Conference on Big Data (Big Data)*, pp. 2857-2866, 2022, https://doi.org/10.1109/BigData55660.2022.10021004.

[44] D. Foead, A. Ghifari, M. B. Kusuma, N. Hanafiah, and E. Gunawan, "A systematic literature review of A* pathfinding," *Procedia Computer Science*, vol. 179, pp. 507–514, 2021, https://doi.org/10.1016/j.procs.2021.01.034.

[45] S. F. Ahmed *et al*., "Deep learning modelling techniques: current progress, applications, advantages, and challenges," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 13521–13617, 2023, https://doi.org/10.1007/s10462-023-10466-8.

[46] H. Peng, C. Mei, S. Zhang, Z. Luo, Q. Zhang, and Z. Wu, "Multi-strategy dynamic multi-objective evolutionary algorithm with hybrid environmental change responses," *Swarm and Evolutionary Computation*, vol. 82, p. 101356, 2023, https://doi.org/10.1016/j.swevo.2023.101356.

[47] Y. Wang, Y. Ma, Q. Li, and Y. Zhao, "A dynamic multi-objective optimization evolutionary algorithm based on classification of environmental change intensity and collaborative prediction strategy," *The Journal of Supercomputing*, vol. 81, no. 1, pp. 1–52, 2025, https://doi.org/10.1007/s11227-024-06480-4.

[48] H. Sun, C. Wang, X. Li, and Z. Hu, "A decision variable classification strategy based on the degree of environmental change for dynamic multiobjective optimization," *European Journal of Operational Research*, vol. 313, no. 1, pp. 296–311, 2024, https://doi.org/10.1016/j.ejor.2023.08.023.

[49] B. Shen, M. A. Cheema, D. D. Harabor, and P. J. Stuckey, "Fast optimal and bounded suboptimal euclidean pathfinding," *Artificial Intelligence*, vol. 302, p. 103624, 2022, https://doi.org/10.1016/j.artint.2021.103624.

[50] C. Wang, M. Sui, D. Sun, Z. Zhang, and Y. Zhou, "Theoretical analysis of meta reinforcement learning: Generalization bounds and convergence guarantees," *Proceedings of the International Conference on Modeling, Natural Language Processing and Machine Learning*, pp. 153–159, 2024, https://doi.org/10.1145/3677779.3677804.

[51] P. K. Mandal, "A review of classical methods and Nature-Inspired Algorithms (NIAs) for optimization problems," *Results in Control and Optimization*, vol. 13, p. 100315, 2023, https://doi.org/10.1016/j.rico.2023.100315.

[52] R. Bhavya and L. Elango, "Ant-inspired metaheuristic algorithms for combinatorial optimization problems in water resources management," *Water*, vol. 15, no. 9, p. 1712, 2023, https://doi.org/10.3390/w15091712.

[53] A. M. Nassef, M. A. Abdelkareem, H. M. Maghrabie, and A. Baroutaji, "Review of metaheuristic optimization algorithms for power systems problems," *Sustainability*, vol. 15, no. 12, p. 9434, 2023, https://doi.org/10.3390/su15129434.

[54] V. Tomar, M. Bansal, and P. Singh, "Metaheuristic Algorithms for Optimization: A Brief Review," *Engineering Proceedings*, vol. 59, no. 1, p. 238, 2024, https://doi.org/10.3390/engproc2023059238.

[55] H. Long, G. Li, F. Zhou, and T. Chen, "Cooperative dynamic motion planning for dual manipulator arms based on RRT* Smart-AD Algorithm," *Sensors*, vol. 23, no. 18, p. 7759, 2023, https://doi.org/10.3390/s23187759.

[56] X.-S. Yang, "Nature-inspired optimization algorithms: Challenges and open problems," *Journal of Computational Science*, vol. 46, p. 101104, 2020, https://doi.org/10.1016/j.jocs.2020.101104.

[57] M. Popović, J. Ott, J. Rückin, and M. J. Kochenderfer, "Learning-based methods for adaptive informative path planning," *Robotics and Autonomous Systems*, vol. 179, p. 104727, 2024, https://doi.org/10.1016/j.robot.2024.104727.

[58] H. Kurunathan, H. Huang, K. Li, W. Ni and E. Hossain, "Machine Learning-Aided Operations and Communications of Unmanned Aerial Vehicles: A Contemporary Survey," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 1, pp. 496-533, 2024, https://doi.org/10.1109/COMST.2023.3312221.

[59] J. Song, D. Rondao, and N. Aouf, "Deep learning-based spacecraft relative navigation methods: A survey," *Acta Astronautica*, vol. 191, pp. 22–40, 2022, https://doi.org/10.1016/j.actaastro.2021.10.025.

[60] M. Capra, B. Bussolino, A. Marchisio, G. Masera, M. Martina and M. Shafique, "Hardware and Software Optimizations for Accelerating Deep Neural Networks: Survey of Current Trends, Challenges, and the Road Ahead," *IEEE Access*, vol. 8, pp. 225134-225180, 2020, https://doi.org/10.1109/ACCESS.2020.3039858.

[61] G. de Alteriis *et al.*, "The use of artificial intelligence approaches for performance improvement of low-cost integrated navigation systems," *Sensors*, vol. 23, no. 13, p. 6127, 2023, https://doi.org/10.3390/s23136127.

[62] H. Poddar, "Unravelling the secrets of artificial neural networks and perceptrons," *Deep Learning in Engineering, Energy and Finance*, vol. 1, p. 25, 2024, https://doi.org/10.1201/9781003564874-2.

[63] Z. Zeng, Q. Cheng, and Y. Si, "Logical Rule-Based Knowledge Graph Reasoning: A Comprehensive Survey," *Mathematics*, vol. 11, no. 21, p. 4486, 2023, https://doi.org/10.3390/math11214486.

[64] H. Wu, R. Lei, Y. Peng, and L. Gao, "AAGNet: A graph neural network towards multi-task machining feature recognition," *Robotics and Computer-Integrated Manufacturing*, vol. 86, p. 102661, 2024, https://doi.org/10.1016/j.rcim.2023.102661.

[65] M. Benzaoui, A. Benbekhma, and H. E. Taibi, "Indoor Obstacle Avoidance System Design and Evaluation Using Deep Learning and SLAM-Based Approaches," *Science and Information Conference*, pp. 262–276, 2024, https://doi.org/10.1007/978-3-031-62281-6_19.

[66] S. R. Budur, "Artificial neural network driven fault tolerance for redundant wheeled mecanum robots: Ensuring navigation despite wheel (s) failure," *Electronic Theses and Dissertations for Graduate School*, 2024, https://etda.libraries.psu.edu/catalog/27818skb6406.

[67] F. Leon and M. Gavrilescu, "A review of tracking and trajectory prediction methods for autonomous driving," *Mathematics*, vol. 9, no. 6, p. 660, 2021, https://doi.org/10.3390/math9060660.

[68] I. D. Mienye, T. G. Swart, and G. Obaido, "Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications," *Information*, vol. 15, no. 9, p. 517, 2024, https://doi.org/10.3390/info15090517.

[69] S. Xie, Z. Zhang, H. Yu, and X. Luo, "Recurrent prediction model for partially observable MDPs," *Information Sciences*, vol. 620, pp. 125–141, 2023, https://doi.org/10.1016/j.ins.2022.11.065.

[70] H. Ruan *et al.*, "GSNN: A Neuromorphic Computing Model for the Flexible Path Planning in Various Constraint Environments," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 1081-1099, 2025, https://doi.org/10.1109/TASE.2024.3359641.

[71] K. Veeranjaneyulu, M. Lakshmi, and S. Janakiraman, "Swarm Intelligent Metaheuristic Optimization Algorithms-Based Artificial Neural Network Models for Breast Cancer Diagnosis: Emerging Trends, Challenges and Future Research Directions," *Archives of Computational Methods in Engineering*, vol. 32, pp. 381–398, 2024, https://doi.org/10.1007/s11831-024-10142-2.

[72] M. G. M. Abdolrasol *et al.*, "Artificial neural networks based optimization techniques: A review," *Electronics*, vol. 10, no. 21, p. 2689, 2021, https://doi.org/10.3390/electronics10212689.

[73] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on Convolutional Neural Networks (CNN) in vegetation remote sensing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 173, pp. 24–49, 2021, https://doi.org/10.1016/j.isprsjprs.2020.12.010.

[74] K. Alibabaei *et al*., "A review of the challenges of using deep learning algorithms to support decision-making in agricultural activities," *Remote Sensing*, vol. 14, no. 3, p. 638, 2022, https://doi.org/10.3390/rs14030638.

[75] R. G. Goriparthi, "Deep Learning Architectures for Real-Time Image Recognition: Innovations and Applications," *Revista de Inteligencia Artificial en Medicina*, vol. 15, no. 1, pp. 880–907, 2024, https://redcrevistas.com/index.php/Revista/article/view/219.

[76] Y. He, P. Huang, W. Hong, Q. Luo, L. Li, and K.-L. Tsui, "In-Depth Insights into the Application of Recurrent Neural Networks (RNNs) in Traffic Prediction: A Comprehensive Review," *Algorithms*, vol. 17, no. 9, p. 398, 2024, https://doi.org/10.3390/a17090398.

[77] M. Wang, W. Fu, X. He, S. Hao, and X. Wu, "A survey on large-scale machine learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 6, pp. 2574–2594, 2020, https://doi.ieeecomputersociety.org/10.1109/TKDE.2020.3015777.

[78] F. Noroozi, M. Daneshmand, and P. Fiorini, "Conventional, Heuristic and Learning-Based Robot Motion Planning: Reviewing Frameworks of Current Practical Significance," *Machines*, vol. 11, no. 7, p. 722, 2023, https://doi.org/10.3390/machines11070722.