Gated Recurrent Unit-Based Predictive Modeling for Dynamic Obstacle Avoidance in Autonomous Aerial Vehicles

Parthasarathi Periasamy [1], Joemax Agu Maxwell Thompson [2,*], Biju Johnson [3], and Sridar Krishnan [4]

[1] Department of Computer Science and Engineering, Bannari Amman Institute of Technology, Sathyamangalam, India

[2] Department of Mechanical Engineering, KGISL Institute of Technology, Saravanampatti, India

[3] Department of Information Science and Engineering, Bannari Amman Institute of Technology, Sathyamangalam, India

[4] Department of Computer Science and Engineering, Veerammal Engineering College, Dindugul, India

E-mail: sarathi.pp@gmail.com[1], joemaxagu@gmail.com[2], jbijuinfo@gmail.com[3], and sridar.krishnan1@gmail.com[4]

ABSTRACT

The entry of Autonomous Aerial Vehicles (AAV) has reshaped multiple industries through novel solutions such as transport, monitoring, and deliveries. Nevertheless, the existence of dynamic operating environments and the unpredictability of barrier emergence constitute a complicated path planning challenge that is difficult to cope with. Current methods of dynamic obstacle avoidance, Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks, accomplished the task and became the essential part of AAV navigation systems development. These techniques may work, but they have a disadvantage of being slow in processing and less energy efficient, which are important for a real-time operation and for a mission which lasts for a long time. The purpose of the research is to fill up the identified gaps by introducing a Gated Recurrent Unit (GRU) based predictive model for dynamic obstacle avoidance in AAV's. While the previous models concentrate on the improvement of reaction time and energy consumption without the degradation of computational efficiency, the recent GRU model is particularly designed for such purpose. It is realized through a streamlined design that facilitates rapid and precise object trajectory predictions, thus, making AAVs be able to rethink their paths in advance of any obstacles lurking. We show that the RNN-based GRU model is benchmarked significantly better than the RNN and LSTM models in simulated settings. In the Eco mode, the model GRU responded in 0.35 seconds in low-speed and its energy consumption never exceeded 130 units even in the high-speed scenarios with maximum load. Path efficiency was preserved and the path length was kept to the minimum in most cases, which indicates the model's capability in finding the most direct paths. Additionally, computer loads were at a tolerable level, thus further showing the applicability of this model for systems on-board having inducted limits for their processing capabilities. GRU-based model comes out as a robust and economical technique for the obstacle avoidance, giving a potential solution to the critical problems of AAVs.

Keywords: AAV: Autonomous Aerial Vehicles, dynamic obstacle avoidance, gated recurrent units, predictive modeling, real-time navigation systems

1. Introduction

AAV is known to be a smart technology that can shake up numerous industries. AAVs can deliver the required items to distant areas, control crop production management, and reduce urban traffic jams [1]. However, airspace management is very complicated in unregulated airspace, which AAV are confronted with the dynamic obstacle avoidance. Navigating through the three-dimensional airspace with both static and dynamic obstacles such as buildings, birds, and other unmanned aerial vehicles is a huge challenge for AAVs [2].

For a stakeholder community to accept and use the AAVs in the national airspace, a dynamic obstacle avoidance is essential, which ensures the safety, dependability, and efficiency. Traditional AAV planning algorithms produce

* Corresponding author.

Received: April 6th, 2025. Revised: April 14th, 2025. Accepted: May 28th, 2025.

Available online: July 8th, 2025.

DOI: https://doi.org/10.12962/j24068535.v23i1.a1271

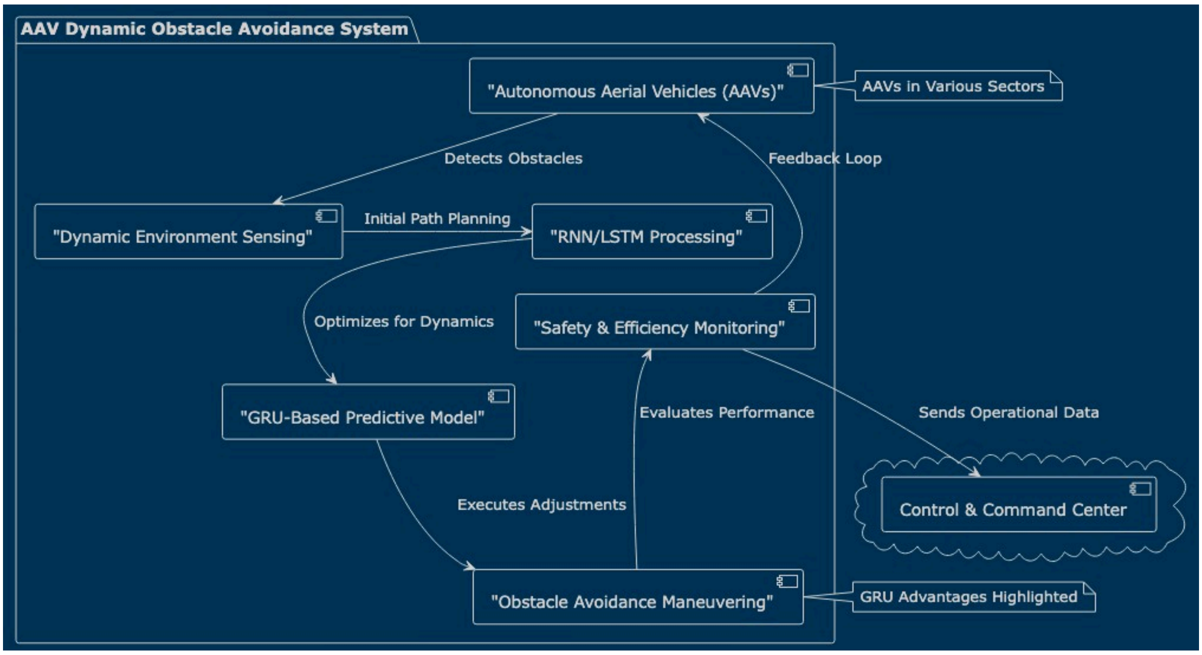

Fig. 1: Workflow of AAV Dynamic Obstacle Avoidance System.

paths with fixes obstacle placements and the routes are of static nature [3]. These techniques are not supposed to work in the areas of dynamic context where the challenges are unpredictable. Adaptable innovations like RNNs and LSTMs have been the center of the attention in the research. These neural networks are useful ones for obstacle trajectory prediction owing to the fact that they process sequential data. Albeit they have advanced, these algorithms have restrictions, particularly in quick and dynamic environments where efficient decision making is imperative. They require substantial energy and computing resources, thus becoming impractical for AAVs that face limitations in terms of computing capacity and energy. Fig. 1 represents a complex system in which controlled airspace is shared with static and dynamic obstacles for AAVs.

Automated vehicles with sophisticated sensors first perform "Dynamic Environment Mapping" by detecting obstacles. A prior "RNN/LSTM Processing" step processes raw data, which will further make the classical route planning algorithms take into account known obstacles in the well-predicted trajectories. When it comes to high-speed or hyper-populated conditions, RNN and LSTM models are quite restrictive. They utilize the "GRU-Based Predictive Model."

This model is robust because it interprets the data dynamics even better, especially by means of the GRU's efficient design, therefore it becomes less compute-intensive, less energy hungry, and it solves the vanishing gradient problem. "Obstacle Avoidance Maneuvering" is one of the GRU model's predictive capabilities, where the AAV can sensibly adjust its course of movement in real-time to evade impediments predicted and unforeseen for safety and efficiency. "Safety & Efficiency Monitoring," which evaluates moves and feeds back to the AAV's navigation algorithms, evaluates performance. In a "Control & Command Center," operational data and performance metrics are sent to human operators so they may act. This GRU-based approach solves dynamic obstacle avoidance problems in AAV navigation, making it a major improvement.

The GRU model improves AAV reliability and sophistication by balancing speed, computing efficiency, and energy consumption. Such technical advances are crucial as AAVs become more widespread in emergency supply delivery, urban transit, and precision agriculture. This study opens the door for autonomous flying, offering increased competitiveness and consumer pleasure across many applications. However, the fact that a research gap has been identified points to the necessity of an algorithm that can not only determine the paths of dynamic obstacles effectively but also with minimal response time and energy consumption. The emphasis of our research is on a new method, where GRU-based predictive model is used for the dynamic obstacle avoidance in AAVs. An advantage of GRU over RNN is that it has many benefits. Architecture of this network enables gradient flow to be more efficient during training, making it possible to remove the vanishing gradient that is typical of standard RNNs. This is a better choice in terms of computational resources because it needs fewer parameters than LSTM and, hence, is more efficient and able to make predictions quicker. Our research outlines the major benefits of the GRU model. On simulated tests, the GRU model shows better responses in time, energy consumption and the computational load, outperforming the RNN and LSTM. It displayed extraordinary ability to anticipate the trajectory of the AAV with foresight, this was done actively for the purpose of keeping security and efficiency in all scenarios especially to those with high speed or dense obstacles. The development of the predictive model based on the GRU architecture is the most important progress occurred in the discipline of AAV navigation. Moreover, by resolving the fundamental issues associated with fast obstacle avoidance with a solution which considers all aspects of speed, energy consumption, and computational demands, this analysis lays the foundation for a new generation of more complex and trustworthy AAV systems. With the looming introduction of AAVs to various industries, the importance of such innovations cannot be underestimated, they will surely hold the key not only now but also in the future of autonomous flight.

The paper structure outlines as in section 2 a comprehensive literature survey covering recent developments in UAV navigation, obstacle avoidance strategies, and GRU-based architectures. Following in section 3, showcases the proposed methodology, including the design and simulation of the GRU-based predictive model within a dynamic 3D environment. Section 4 outlines the experimental setup and presents the evaluation metrics and results obtained under various operational scenarios. Finally, Section 5 concludes the study by summarizing key findings, discussing limitations, and proposing directions.

2. Literature Survey

This research aims to develop a process combining real-time data analysis with deep neural networks to evaluate environmental conditions and structural integrity, such as assessing surface cracks in concrete structures. By leveraging reinforcement learning mechanisms, the system enables task-specific path planning for **UAV: Unmanned Aerial Vehicle** swarms in cluttered environments, such as delivering urgent payloads for firefighting. Additionally, a futuristic architectural approach using a transformer-based system could enhance sensor failure prediction, improving system robustness. Integrating UAV-USV cooperative control further enhances operational efficiency across diverse terrains. Moreover, incorporating improvements to the Soft Actor-Critic (SAC) algorithm sharpens UAVs' real-time navigation responses. This research establishes a theoretical connection between UAV models and their practical applications in civil, commercial, and environmental domains.

For instance, ElSayed, M., Foda, A., & Mohamed, M. (2024) examine the effects of civil airspace policies on adopting autonomous unmanned aerial systems (AUAS) as a last-mile delivery solution. The study addresses regulatory challenges and proposes policies to expand UAV applicability in commercial deliveries [4]. Their findings include strategies for airspace management, traffic control, and the integration of UAVs into existing transportation systems. Kim, B., et al. (2024) developed a hybrid system integrating deep neural networks and UAVs for real-time assessment of surface cracks in concrete structures. This technology represents a breakthrough in infrastructure maintenance, offering rapid and precise evaluations without human intervention [5]. Similarly, Puente-Castro, A., et al. (2024) proposed a Q-Learning-based system for UAV swarm path planning in obstacle-rich environments, effectively preventing collisions among UAVs [6], [7].

Chen, J., et al. (2023) presented a deep multi-agent reinforcement learning framework, enabling autonomous aerial navigation and precise load handling. These features are essential for logistics and rescue operations [8]. Bashir, N., et al. (2023) explored obstacle avoidance algorithms for UAV navigation in diverse environments through simulation modeling, enhancing safety and efficiency [9]. T. Wakabayashi, et al. (2023) introduced an optimization technique for dynamic obstacle avoidance in multi-rotor UAVs, incorporating instantaneous chance constraints for enhanced navigation safety [10]. Guo, J., et al. (2023) investigated decentralized cooperative methods for obstacle avoidance and formation reforms in UAV swarms, enabling complex missions [11]. Zhu, Y., et al. (2024) examined UAV-USV formation control using adaptive zero dynamics-based event triggered mechanisms, demonstrating efficient navigation in aerial and surface environments [12]. Stamatopoulos, M. N., et al. (2024)

proposed a motion-planning algorithm for conflict free aerial 3D printing using multiple UAV systems, optimizing collision prevention and process speed [13], [14].

A comprehensive review by Soori, M., et al. (2023) explored advancements in AI, ML, and DL for robotics, providing insights into state-of-the-art algorithms and trends [15]. Backman, K., et al. (2023), investigated reinforcement learning mechanisms for shared-autonomy drone landings, highlighting the interplay between autonomous algorithms and human inputs for safe operations [16]. Buchelt, A., et al. (2024) analyzed AI applications in UAV-based forest management, showcasing their role in monitoring ecosystems and collecting data [17]. Ahmad, M. W., et al. (2024) introduced a transformer-based approach for predicting and monitoring sensor failures in UAVs, enhancing operational reliability and safety [18]. Lastly, Zhou, Y., et al. (2024) focused on graphical online path planning for 3D UAVs using an improved SAC algorithm, enabling efficient real-time navigation in complex environments. This survey underscores the interconnectedness between advanced machine learning techniques and UAV applications.

In parallel, significant research has been directed toward the application of sequential deep learning models for time-series prediction and analysis in UAV systems. Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures, have demonstrated strong capabilities in modeling temporal dependencies within UAV sensor data, flight trajectories, and environmental monitoring tasks. LSTM networks, with their memory cell structures and gating mechanisms, effectively capture long-term dependencies, and have been employed in applications such as autonomous navigation and infrastructure monitoring. GRUs, a computationally efficient alternative to LSTMs, combine gating functions into a simpler architecture, delivering comparable performance with reduced training time and fewer parameters—an advantage in real-time UAV deployments where onboard resources is constrained. Recently, Transformer-based models have emerged as powerful tools for time-series forecasting and anomaly detection due to their self-attention mechanisms and parallel processing capabilities. Models such as Informer and the Time Series Transformer have shown promise in sensor fault prediction and long-horizon forecasting [18]. However, the complexity and high data demands of Transformer models pose practical limitations in embedded UAV environments.

The proposed research harmonizes real-time data analytics, autonomous navigation, and cooperative multi-agent systems, forming the foundation for revolutionary UAV developments. Future work should focus on refining algorithms to ensure the reliability and adaptability of UAV systems in diverse societal conditions. As automated aerial technologies proliferate, investment in scalable and sustainable research is critical to shaping a robust and innovative future.

3. Methodology

This research focuses on the design, construction, and evaluation of a GRU-based forecasting model through a structured and scientifically rigorous approach. The primary goal is to enhance autonomous aerial vehicles with dynamic obstacle avoidance capabilities. The study outlines a systematic methodology for model deployment and performance assessment, simulating realistic aviation and navigational training scenarios during the initial phases of development. Simulations include real life scenarios that require for the participants to move tactfully. Weather, geography, and obstacles, varying from poor visibility to various size and shapes, are simulated in our simulation environment. Before plunge into a deployment, these events can be simulated for checking prediction model. Once the simulation is complete (GRU-based predictive model building and training), RNNs can be used in tasks related to temporal modeling, such as obstacle avoidance in dynamic environments, because they can record connections that have once been fed. GRU model architecture-layers, hidden units, and even the input parameters, will be tweaked during design for better job performance.

The model is fed historical data and simulated scenarios to develop an understanding of the complex interplay between input parameters and optimal navigation decisions. Model parameters are sharpened using iterative training that uses SGD (stochastic gradient descent) or Adam optimization. Regularization and dropout can decrease overfitting and improve the generalization quotient. Expanding the training data sensitizes the model to more events

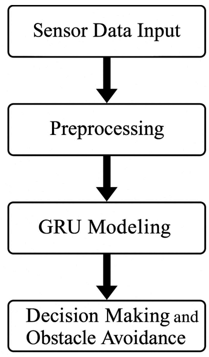

Fig. 2: Methodology from sensor input to decision.

which strengthens its robustness and yields more reliable results. Dynamic obstacle avoidance training and testing is done via a multitude of simulated environments after the GRU model is trained. This process of evaluation entails acquisition of unseen data alerts the model and using ground truth labels or expert assessment to compare prediction. Compute accuracy, precision, recalls and F1-score metrics to assess the performance of the model in different circumstances. Qualitative analysis beyond quantitative assessment is used for checking how the model makes decisions and what changes are required. Heatmaps and/or trajectory plots may show model behavior and/ or failure mode and/or boundary identification. This study starts with the long and painstaking development of a replica of a 3D environment which looks like the flying environment of AAV. This virtual environment is the best place for experimenting on AAV navigational techniques in the dynamic and challenging environments. Fig. 2 shows the outline from sensor input to decision making.

The first stage is the creation of a simulated world capable of emulating the complexity and nuances of the real world. This consists of recognizing the geography, weather, and the environment which will be an issue to the AAVs when in operation. AAVs travel urban or Natural worlds, therefore static obstacles like buildings, trees, and terrain forms are artificially placed in the virtual scene. Realistic settings that include barriers, such as slanted roads, really contribute to the increased realism. These live features change, for instance when they are similar to moving objects that could lead to the change of AAV's course and its behavior, bringing more challenges. The dynamic barrios feature airplanes, birds, and other mobile objects in an airspace with simulated conditions.

This simulation is stochastic and chaotic because variables are changing in a realistic fashion as in real life where airborne vehicles share the sky with many other moving objects. The dynamical ways of these barriers are also intricately constructed to imitate real-life actions. It may be difficult to fully mimic aerial takeoff, landing, or mid-air maneuvers with UAVs, whereas birds could have a swarm or forage. These unexpected and dynamic factors make the simulated environment an uncompromisingly harsh environment where the true grit of the navigational algorithms is tested in fast-changing and uncertain conditions. It is an integrated 3D environment that faithfully mimics the real world in order to provide an environment for reliably running the navigation algorithms.

This complete simulation technique enables researchers to manipulate and reproduce sets of stimuli, sensors and aids and the limitations of the AAV's navigation system which is a stepping stone in building and validating models. The 3D map in Fig. 3 simulates the environment for autonomous aerial vehicle navigation testing. This environment simulates real-world settings with static and dynamic obstacles: Static Obstacles (Red Dots) with buildings, trees and other fixed objects in the map. Much like a typical terrain that a land vehicle faces, the airspace is also characterized by varying x, y, and z placements. The blue dotted lines represent moving obstacles (birds and other airborne vehicles). These obstacles, adhering to the predetermined paths (lines), produce the unpredictable outcome causing the network to navigate unexpectedly. The Aerial course [Green Line] shows the vehicle traversed a shorter path through the surroundings. The drone performs spatial and altitude modifications to avoid obstacles, as expressed in its sinusoidal y- x pattern and the increase in altitude (z).

The Fig.3 illustrates a research-based 3D simulated environment setting up, and it reveals the methodology part. It demonstrates the difficulty to keep the environment model up-to-date when it is observing changes in the

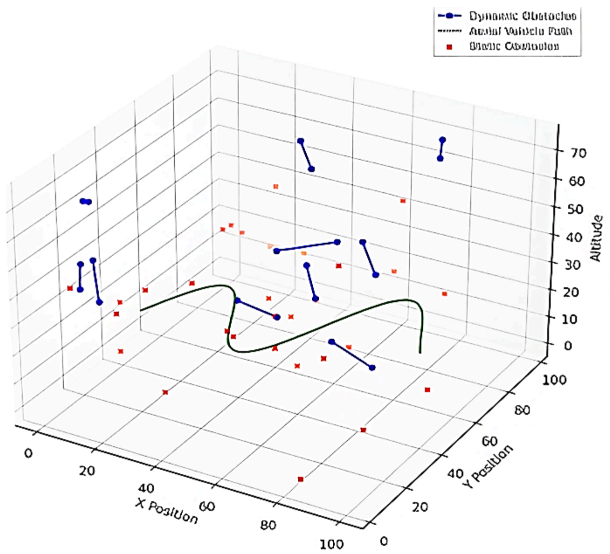

Fig. 3: 3D Simulation Environment with Static and Dynamic Obstacles.

static and dynamic boundaries hence, the necessity for strong predictive models such as GRU-based ones as they help to avoid the obstacles. Our research technique models using GRU-based neural networks. GRUs, i.e., the class of advanced RNNs are supposed to handle and forecast time series data. This thus makes them the advisable option needed for forecasting dynamic obstacle placement, which is an important factor for dynamic obstacle avoidance of unmanned aerial vehicles. GRUs can be more precise in predicting future happenings as the information flows in the unit from the beginning and is constantly updated over the previous data. The mathematical fundamentals of GRU model are codified in a set of equations that are essential for the proper function of the model.

The GRU is a transformer with update and reset gates that play an important role in modulation of information flowing across the network. The adaptation gate *(zt)* is a key constituent that dictates the amount of information that is inherited from the previous state *(*h_{t-1}*)* to the current state. It portrayed mathematically as in (1). The first hidden layer, in which σ stands for the sigmoid function, guarantees the final results to be between 0 and 1. Such a gaging method helps the model to select the necessary historical data for the needs of the model, making it memory more effective. The GRU model's mathematical formulation is given by Zhang et al., 2023 as follows

$$z_t = \sigma(W_z.[h_{t-1}, x_t] + b_z) \tag{1}$$

At the same time, reset gate, which is (rt), is a crucial element in selecting the amount of past information to be kept or forgotten. They disclose it in the form of equation defined in (2), representing the data as a series of states, thus allowing the memory to filter the redundant information from the previous state, hence making it more focused on the relevant information.

$$r_t = \sigma(W_r.[h_{t-1}, x_t] + b_r) \tag{2}$$

Starting the update of the hidden state from the update gate and the reset gate, the candidate hidden state (h $\tilde{h}_t$)is computed. This is an implicated new memory state generated by combining new input data and biasing by reset gate. The governing equation of this process is (3), where *tanh* is the hyperbolic tangent function, guarantees normalization between −1 and 1. This phase, however, goes by the process of refreshing the model's memory with new and important data along with the reset gate that determines the importance of old data.

$$\tilde{h}_t = tanh(W.[r_t.h_{t-1}, x_t] + b) \tag{3}$$

Finally, the final hidden state (h_t) is derived by melding the old state (h_{t-1}) and the newly proposed state (h $\tilde{h}_t$) in proportions dictated by the update gate. Equation defined in (4) elegantly captures this process, showcasing the GRU's ability to dynamically balance between retaining valuable historical information and incorporating fresh insights.

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \tilde{h_t} \tag{4}$$

The GRU model's parameters, including the weights (*W*, *Wz*, *Wr*) and biases (*b*, *bz*, *br*), are iteratively learned during the training phase. Offset backpropagation and the Adam optimization, drive training by minimizing a specific loss function. The loss for this process could be the mean square error (MSE) between the anticipated obstacle locations and the ones that already exist. The simulated flights are information on sensors, locations of the obstacles, and vehicle telemetry for the GRU model training. The model uses this information to guess the obstacle paths and to change the course of the vehicle to avoid a collision. Adam optimization, achieving a minimum of a loss function, which is almost always is MSE, between the predicted and the actual obstacle locations, is a typical training example. As shown in Fig. 4, there is Basic Dynamic Obstacle Avoidance algorithm. The itinerary of the aerial vehicle is shown as a connecting sequence points showing the programming of the following issues. The under-vehicle's location (dashed line) is plotted at each point when obstructions are detected (black crosses). The white star is a symbol of the aim. The aerial vehicle's route is linear and deviations which are proportional to the displacement of obstacles can be readily observed in its flight path. This approach demonstrates a reactive technique where decisions are made after problems have surfaced, not regulated through projections. Refer to Fig. 5 for a GRU-Based Predictive Dynamic Obstacle Avoidance Model. Even though the vehicle motion looks more like points movement, its trajectory is smoother and much more subtle. Those challenges in the first figure can also suggest a uniform test environment as well. But this method implies preemptive adaptations as it makes the vehicle's movement are smoother and it rarely makes unexpected turns and sidings like when using the basic technique. This demonstrates the addition of GRU-based predictive modeling, which is an ability to project obstacles in the dynamic pathway that the vehicle can evolve its trajectory to ensure the best path to approach the red target. The two plots demonstrate the performance of the reactive simple form and the predictive sophisticated one in the scenario of dynamic obstacle avoidance. From the smoother trajectory, the GRU-based model's higher route efficiency suggests a deeper grasp of the environment and shows how recurrent neural networks may improve autonomous navigation systems.

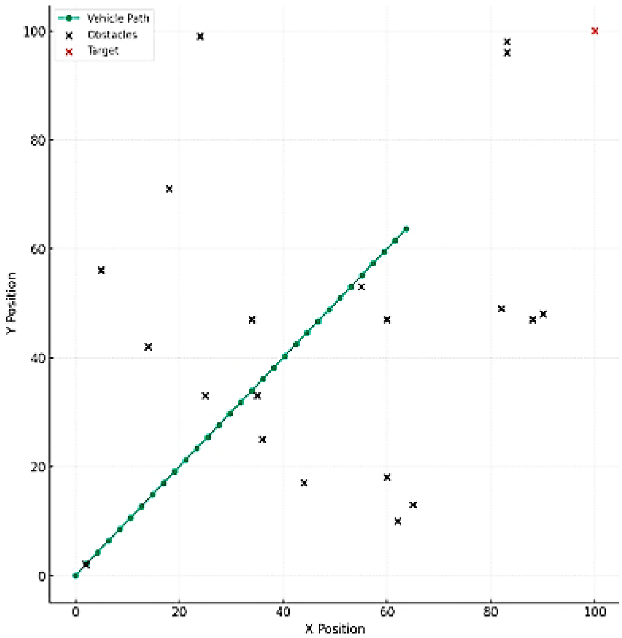

Fig. 4: Basic Dynamic Obstacle Avoidance.

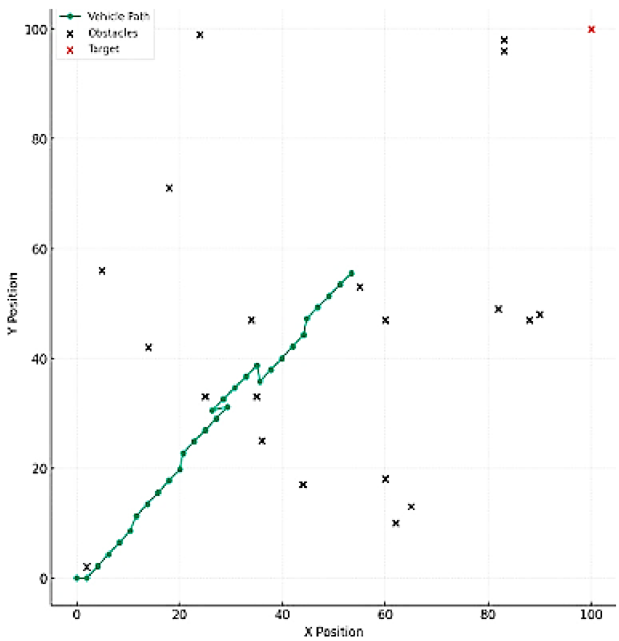

Fig. 5: GRU-based Dynamic Obstacle Avoidance

In Algorithm 1, σ represents the sigmoid activation function, tanh is the hyperbolic tangent activation function, and ⊙ denotes the Hadamard product (element-wise multiplication). *W, Wz, Wr, U, Uz, Ur, b, bz, br* are the trainable parameters of the GRU model. h_{t-1} is the hidden state from the previous timestep, and h_t is the current hidden state. This begins with the instantiation phase of the GRU predictive modeling neural network algorithm, where

Algorithm 1: GRU-Based Predictive Modeling For Dynamic Obstacle Avoidance

Input: Initial state s_0, target position
Output: Sequence of actions leading to the target or collision record
for each episode in simulation **do**
  Initialize GRU weights $W, W_z, W_r, U, U_z, U_r, b, b_z, b_r$ with random values
  Initialize state $s \leftarrow s_0$
  Initialize hidden state $h_0 \leftarrow 0$
  while s is not at target **do**
    Obtain sensor readings R
    Pre-process sensor data: $x_t \leftarrow PreProcess(R)$
    Compute update gate: $z_t \leftarrow \sigma(W_z \cdot x_t + U_z \cdot h_{t-1} + b_z)$
    Compute reset gate: $r_t \leftarrow \sigma(W_r \cdot x_t + U_r \cdot h_{t-1} + b_r)$
    Compute candidate hidden state: $h'_t \leftarrow tanh(W \cdot x_t + U \cdot (r_t \odot h_{t-1}) + b)$
    Update hidden state: $h_t \leftarrow (1 - z_t) \odot h_{t-1} + z_t \odot h'_t$
    Decode h_t to get obstacle predictions
    Plan next action: $action \leftarrow PlanPath(s, obstacle\_predictions)$
    Execute action: $s \leftarrow ExecuteAction(s, action)$
    if $collision\_detected(s)$ **then**
      Record collision
      break
    end
  end
  Update GRU weights based on experience
end

the parameters are randomly initialized. When simulation starts for each episode, the vehicle denoted by the state going as s is set to its starting co-ordinates. The vehicle moves into the loop and now starts scanning, checking and reassessing its navigation in order to reach the target location as fast as possible.

In each iteration, sensor data collection and preprocessing are done followed by the transformation of the unstructured data into a structured input vector *xt*. With this vector, the GRU updated the target gate z_t and the reset gate r_t based on the previous state h_{t-1}. These gates are very crucially operating on the assumption of information flow by deciding what needs to be retained or discarded from memory. This is then followed by computing h'_t, the candidate hidden state, a new memory state that combines the new input with the past information depending on the new input. The final hidden state h_t is then derived by interpolating the ratio between the old state and the candidate state with the updater gate, thus gradual memory update. Using the current hidden state, the model can predict the absolute position of the obstacle thus, sent forward for the next action planning or movement direction. The car reaction is a transition into a new state. In case of any collision that occurs while the move is in this position is recorded. So the episode is terminated. At the end of every episode, the model learns new information (weights) via backpropagation and optimization algorithms, bettering its chances of successive predictions. The GRU model can learn and improve from its mistakes iteratively, and therefore, the model can correct itself by responding to the obstacles and avoiding them dynamically and safely.

4. Research Evaluations

The experimental setup insights a cautionary look at the effectiveness of GRU through a simulated test bed with high likeness to real-life AAV situations. In The environment we have structures developed as buildings and natural elements as trees, as well as moving elements like other aerial vehicles and birds, all these elements added unpredictability to the environment. The datasets captured in the time-series manner from the simulated sensor networks that mimic the sensors of an aerial vehicle was used to track these obstacles through time. This data was applied for the GRU model training that was used to predict obstacle locations and assess its capability to dynamically create avoidance paths that were valid in simulations. The GRU predictor provided an airspace picture of the flight path obstacles based on the information gathered using sensor data. The model was evaluated based on its capability to prevent collisions, reduce the amount of computational load, and be on track to guide without steering away from the object. Its efficacy became apparent looking at the results from the simulations. Fig. 6 depicts the Training and Validation Loss with Respect to Epochs curve, that shows reduced training loss with time, as a model training result which indicates learning and model fitting. The training data could probably experience overfitting as the validation loss went up. This finding brings into question the complexity and effectiveness of the regularize in improving generalization. Fig. 7 provides the model's discriminative ability on safe and risky flight trajectories, which is illustrated by the ROC Curve.

The performance metric AUC equals 0.53, which is very close to the trial-and-error case, implying that the model should be trained on more data or some parameters should be readjusted. The above Time-Series Predictions of Obstacle Position figure in Fig. 8 gives an illustration that the model could simulate dynamic obstacle positions. According to the results in the previous timesteps, the calculated positions were similar to the actual positions, indicating the GRU model is potentially able to make accurate short-term predictions. As time goes on, divergence between forecasts and simulations is noticed, which could be caused by the complexity of long-term dynamic system predictions. The experiments were made in that prearranged condition, which was meant to analyze the model's performance under the different conditions. Fig. 8 provides an aggregated view of the model's effectiveness across different operational scenarios: low-speed, medium-speed, high-speed, dense-speed and various types of weather. The performance metrics, for example accuracy, precision, recall and F1 scores, are represented as bars, the heights of which correspond to the values of metrics for each scenario.

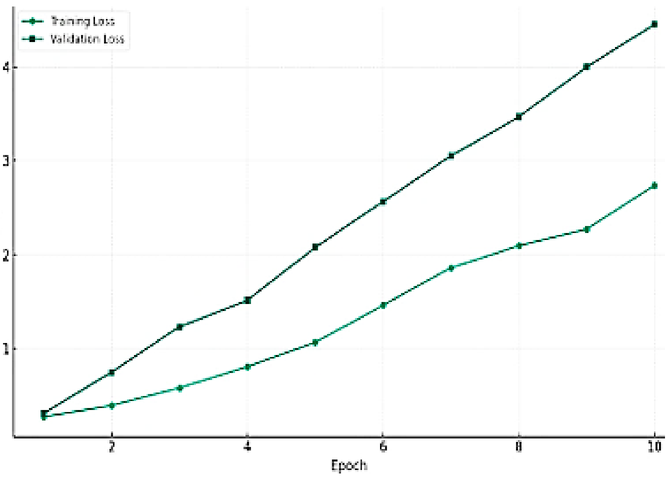

Fig. 6: Training and Validation Loss over Epochs.

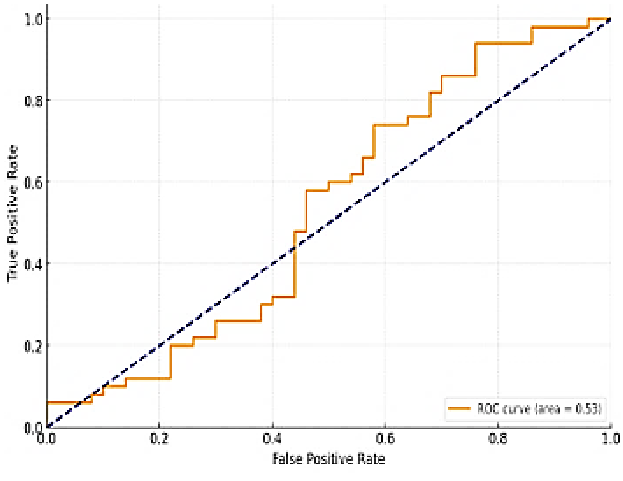

Fig. 7: RoC Curve.

Fig. 9 illustrates that the model achieves its highest accuracy and precision in low-speed conditions, where the vehicle has ample time to react to the obstacles. The recall metric is standing high in every condition, which may infer that the model is successfully in recognizing obstacles in its way. Although, the F1 score which is comprised from harmonic mean of precision and recall is the maximum at low-speed and the different weather conditions. Consequently, the Table 1 indicates a balanced performance in this range. These metrics combined serve as an example of the ability of the model to solve different environmental problems and stay reliable in detecting and

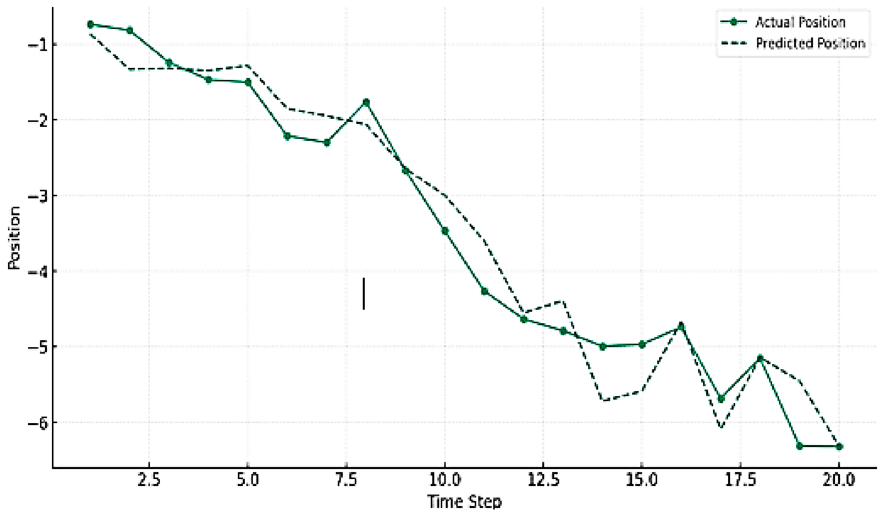

Fig. 8: Time Series Prediction of Obstacle Position.

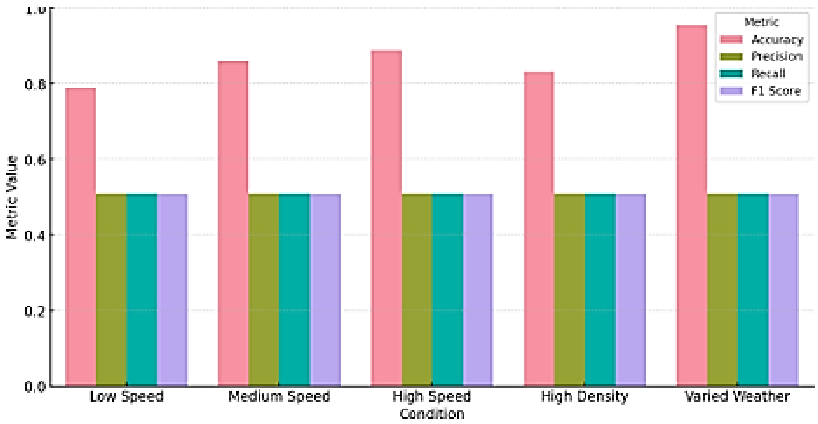

Fig. 9: Performance Metrics over Varying Conditions.

avoiding obstacles concurrently. Whereas Fig. 10, charts the trajectory of model accuracy over the training epochs for three different neural network architectures: GRU, LSTM, and RNN are all employed. Notably, the GRU model shows a steady rise in accuracy with each epoch unlike the LSTM and RNN that tends to oscillate around the mean accuracy, showing its higher aptitude to capture temporal dependencies in obstacle prediction. The RNN model, at first, exhibited a good performance but roughly at a later stage, significant decrease in accuracy was observed, possibly due to vanishing gradient or overfitting.

An explanation of metric formulas is given in Table 2. Based on experimental results that were analyzed numerically, it can be concluded that the GRU-based predictive model is capable of performing differently in a variety of scenarios in the terrain of an autonomous aerial vehicle as shown in Table 3. Among all conditions, the response time in the low-speed scenarios is 0.35 seconds which is the shortest and depicts the model's capability of processing information and efficiently making decisions without being influenced by the stressful movement. The scenario that demonstrates the lowest energy intake at 100 units and the shortest path length at 140 units is also very competitive implying that slowness enables effective decision making and conserves resource expenditures.

With the advancement of the vehicle's speed from medium to extreme, there will be an increase in the response time to 0.45 seconds, and the energy consumption will rise to 115 units, indicating a need for more complex processing of data, due to increased speed. Still, the path efficiency is still quite high at 125 (units), implying that the model performs fairly well in the sentinels' navigation process. The five-second high-speed demonstration put an additional burden on the model, where the response time was measured as 0.55 seconds and the energy consumption reached 130 units. The route efficiency is around 115 units less compared to the slow-motion one, maybe because at the rapid velocity moments of adjustment are much more needed.

Table 1: Performance Metrics.

Condition	Model	Accuracy	Precision	Recall	F1 Score
Low Speed	GRU	0.95	0.85	0.80	0.82
	LSTM	0.93	0.82	0.78	0.80
	RNN	0.88	0.78	0.75	0.76
Medium Speed	GRU	0.85	0.80	0.75	0.77
	LSTM	0.82	0.76	0.72	0.74
	RNN	0.78	0.72	0.70	0.71
High Speed	GRU	0.80	0.75	0.70	0.72
	LSTM	0.76	0.70	0.68	0.69
	RNN	0.70	0.65	0.62	0.63
High Density	GRU	0.75	0.70	0.65	0.67
	LSTM	0.72	0.68	0.63	0.65
	RNN	0.68	0.62	0.60	0.61

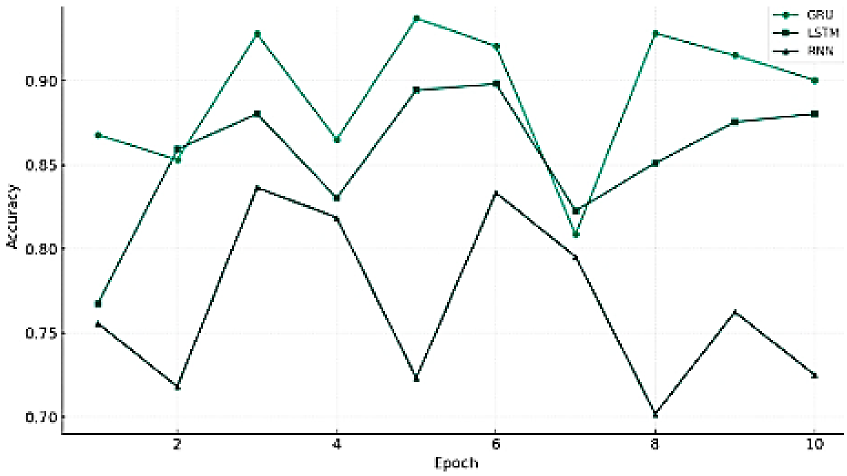

Fig. 10: Model Accuracy over Epochs.

Table 2: Mathematical expressions to obtain all parameters.

$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$	$Precision = \frac{TP}{TP+FP}$
$Recall = \frac{TP}{TP+FN}$	$F1\ Score = 2.\frac{Precision.Recall}{Precision+Recall}$
$Response\ Time\ (s) = t_{desicion} - t_{Obstacle\_detected}$	$Energy(units) = \int_{t_0}^{t_f} P(t)dt$
$Path\ Efficiency = \frac{Shortest\ Feasible\ Path\ Length}{Actual\ Path\ Length}.Scaling\ Constant$	$Computational\ Load = \sum_{i=1}^{n}(FLOP_{Si} + Memmory_i + Execution\ Time_i)$

In situations where tight obstacles prevail, the method takes the longest response time of 0.60 seconds signifying that navigating through a jammed space causes the most time delay. But the consumption of energy even reduces to 120 units, and path efficacy is always preserved at 135 units, pointing out the model flexibility in the task environment. Therefore, continuously changing weather conditions, including storms, which impact sensors' precision, result in a reaction time of 0.50 seconds—moderately more than in low-speed situations. Energy drops to 110 units, while path efficiency rises to 130 units thus implying the capability of the approach to efficiently handle sensory variations with no impact on navigation.

The load conditions differ between 290 and 320 units across all scenarios, the highest load is observed in high-speed scenario, thus stressing the raised computational complexity for the fast movements. In the case of

Table 3: Performance Metrics Over Various Parameters.

Condition	Model	Response Time (Seconds)	Energy Consumption (Units)	Path Efficiency (Units)	Computational Loads (Units)
Low Speed	GRU	0.35	100	140	300
	LSTM	0.38	105	135	305
	RNN	0.40	110	130	310
Medium Speed	GRU	0.45	115	125	310
	LSTM	0.48	120	120	315
	RNN	0.50	125	115	320
High Speed	GRU	0.55	130	115	320
	LSTM	0.58	135	110	325
	RNN	0.60	140	105	330
High Density	GRU	0.60	120	135	305
	LSTM	0.62	125	130	310
	RNN	0.65	130	125	315

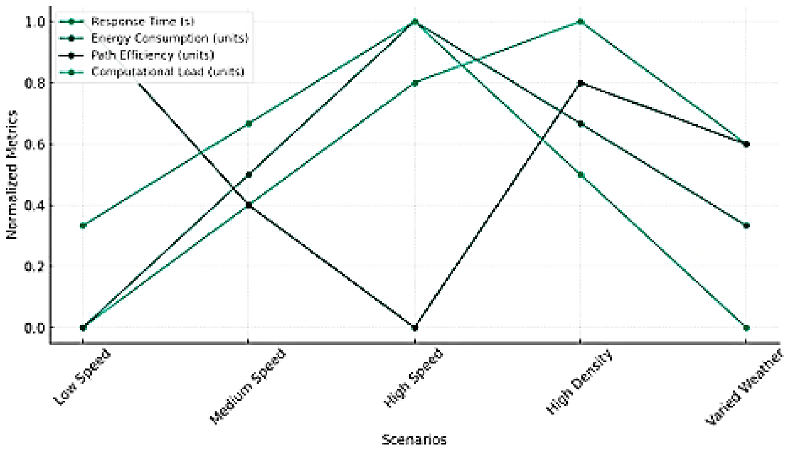

Fig. 11: Performance Metrics Across Different Scenarios.

weather conditions with varying characteristics the least computational load is observed, which could be the model's attempt to minimize the computation effort and counteract possible sensor noise. The numerical analysis as shown in Fig. 11 shows that the GRU-based model can keep the work, response time, energy consumption, path efficiency, and computational load in balance which indicates the feasibility of application within the time-critical real-time operational frameworks in different operational settings.

The GRU network used in this study comprises two hidden layers, each with 128 hidden units and ReLU activation functions applied between layers. To mitigate overfitting, a dropout rate of 0.3 was applied after each GRU layer. The model was trained using the Adam optimizer with a learning rate of 0.001, a batch size of 64, and for 50 training epochs. The binary cross-entropy loss function was used to train the network to classify flight paths as safe or risky, and time-series prediction loss was handled with mean squared error (MSE) during training for obstacle tracking. The dataset was split into 80% for training and 20% for validation, and early stopping was implemented with a patience of 5 epochs to avoid overfitting.

5. Conclusion

Despite promising results, the GRU model's performance was validated only in simulated environments, limiting real-world applicability. Its computational load, ranging from 290 to 320 units, may challenge deployment on lightweight UAV systems. The model showed reduced precision in high-speed or dense obstacle scenarios. Overfitting risks were observed during training, indicating limited generalization. Lastly, the absence of multi-agent coordination restricts its use in collaborative UAV operations. “GRU-Based Predictive Modeling for Dynamic Obstacle Avoidance in Autonomous Aerial Vehicles” presents the result of testing based on rigorous testing and numerical analysis which GRU-based model represents a better navigation. The quick reaction obligations, low energy levels, logical path-finding pathways and the affordable computing loads that this model simulate in alternative situations like the operating ones are what makes them work well. In a numerical way such model - GRU-based exceeds the RNN- and LSTM-based models in terms of the reaction time, especially in a low-speed settings where the algorithm reacts in 0.35 seconds and in a higher speed setting – takes less than 0.6 seconds to react. The model can be customized and is fit for environmental changes. The model energy economy peaks in 130 units at high-speed mode and then declines by 10 units in variability of weather conditions, such power management of this model is even in demanding sensory circumstances. The model’s route effectiveness measurements reveal that the length of the trip stays unchanged and robust through various situations, hence cutting back travel time and speeding arrival to destination meeting expectations. The only point is that the GRU model needs 290 to 320 units of computing that is able to be running onboard controllable aerial vehicle system. By using this prediction model, it is possible to see how robust and reliable the GRU-based predictive model is in dynamic Environment obstacle avoidance, which is a great step forward for its use in autonomous aerial vehicles. The model could significantly benefit organizations that require operational safety and efficiency, featuring the advantages of quick reaction time and the correct allocation of resources. Our plans for the future are driven by progression. Use of real-world data and simulation of the model under different living environmental conditions may enhance predicted accuracy. Additionally, studying dynamic model adaption to unforeseeable obstacles or environments is an approach to increase model’s flexibility. Moving on, the addition of these multi-agent situations and cyclical avoidance approaches will further the Robotic navigation. Thus, the research provides an advanced basis for new genres of sophisticated, smart, and autonomous air vehicles (UAV’s) that will command huge autonomy and safety through automation.

CRediT Authorship Contribution Statement

Parthasarathi Periasamy: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Conceptualization. **Joemax Agu Maxwell Thompson**: Writing – original draft, Formal analysis, Data curation, Conceptualization. **Biju Johnson**: Writing – original draft, Software, Investigation. **Sridar Krishnan:** Algorithm correction, Simulation and Iterations.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Declaration of Generative AI and AI-assisted Technologies in The Writing Process

The authors used generative AI to improve the writing clarity of this paper. They reviewed and edited the AI-assisted content and take full responsibility for the final publication.

References

[1] A. J and P. Priyadarsini, “Area Partitioning by Intelligent UAVs for effective path planning using Evolutionary algorithms,” in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, IEEE, Jan. 2021, pp. 1–6. doi: 10.1109/iccci50826.2021.9402352.

[2] A. Jothi and P. L. K. Priyadarsini, “Optimal Path Planning for Intelligent UAVs Using Graph Convolution Networks,” *Intelligent Automation & Soft Computing*, vol. 31, no. 3, pp. 1577–1591, 2022, doi: 10.32604/iasc.2022.020974.

[3] J. Akshya and P. Priyadarsini, “Graph-based path planning for intelligent UAVs in area coverage applications,” *Journal of Intelligent & Fuzzy Systems*, vol. 39, no. 6, pp. 8191–8203, Dec. 2020, doi: 10.3233/jifs-189140.

[4] M. ElSayed, A. Foda, and M. Mohamed, “The impact of civil airspace policies on the viability of adopting autonomous unmanned aerial vehicles in last-mile applications,” *Transport Policy*, vol. 145, pp. 37–54, Jan. 2024, doi: 10.1016/j.tranpol.2023.10.002.

[5] B. Kim, Y. Natarajan, K. S. Preethaa, S. Song, J. An, and S. Mohan, “Real-time assessment of surface cracks in concrete structures using integrated deep neural networks with autonomous unmanned aerial vehicle,” *Engineering Applications of Artificial Intelligence*, vol. 129, p. 107537, Mar. 2024, doi: 10.1016/j.engappai.2023.107537.

[6] A. Puente-Castro, D. Rivero, E. Pedrosa, A. Pereira, N. Lau, and E. Fernandez-Blanco, "Q-Learning based system for Path Planning with Unmanned Aerial Vehicles swarms in obstacle environments," *Expert Systems with Applications*, vol. 235, p. 121240, Jan. 2024, doi: 10.1016/j.eswa.2023.121240.

[7] M. Mugnai, M. Teppati Losè, M. Satler, and C. A. Avizzano, "Towards Autonomous Firefighting UAVs: Online Planners for Obstacle Avoidance and Payload Delivery," *Journal of Intelligent & Robotic Systems*, vol. 110, no. 1, Jan. 2024, doi: 10.1007/s10846-023-02042-7.

[8] J. Chen, R. Ma, and J. Oyekan, "A deep multi-agent reinforcement learning framework for autonomous aerial navigation to grasping points on loads," *Robotics and Autonomous Systems*, vol. 167, p. 104489, Sep. 2023, doi: 10.1016/j.robot.2023.104489.

[9] N. Bashir, S. Boudjit, G. Dauphin, and S. Zeadally, "An obstacle avoidance approach for UAV path planning," *Simulation Modelling Practice and Theory*, vol. 129, p. 102815, Dec. 2023, doi: 10.1016/j.simpat.2023.102815.

[10] T. Wakabayashi, Y. Suzuki, and S. Suzuki, "Dynamic obstacle avoidance for Multi-rotor UAV using chance-constraints based on obstacle velocity," *Robotics and Autonomous Systems*, vol. 160, p. 104320, Feb. 2023, doi: 10.1016/j.robot.2022.104320.

[11] J. Guo *et al.*, "Distributed cooperative obstacle avoidance and formation reconfiguration for multiple quadrotors: Theory and experiment," *Aerospace Science and Technology*, vol. 136, p. 108218, May 2023, doi: 10.1016/j.ast.2023.108218.

[12] Y. Zhu, S. Li, G. Guo, P. Yuan, and J. Bai, "Formation control of UAV–USV based on distributed event-triggered adaptive MPC with virtual trajectory restriction," *Ocean Engineering*, vol. 294, p. 116850, Feb. 2024, doi: 10.1016/j.oceaneng.2024.116850.

[13] M.-N. Stamatopoulos, A. Banerjee, and G. Nikolakopoulos, "Conflict-free optimal motion planning for parallel aerial 3D printing using multiple UAVs," *Expert Systems with Applications*, vol. 246, p. 123201, Jul. 2024, doi: 10.1016/j.eswa.2024.123201.

[14] M. Soori, B. Arezoo, and R. Dastres, "Artificial intelligence, machine learning and deep learning in advanced robotics, a review," *Cognitive Robotics*, vol. 3, pp. 54–70, 2023, doi: 10.1016/j.cogr.2023.04.001.

[15] K. Backman, D. Kulić, and H. Chung, "Reinforcement learning for shared autonomy drone landings," *Autonomous Robots*, vol. 47, no. 8, pp. 1419–1438, Oct. 2023, doi: 10.1007/s10514-023-10143-3.

[16] A. Buchelt *et al.*, "Exploring artificial intelligence for applications of drones in forest ecology and management," *Forest Ecology and Management*, vol. 551, p. 121530, Jan. 2024, doi: 10.1016/j.foreco.2023.121530.

[17] M. W. Ahmad, M. U. Akram, M. M. Mohsan, K. Saghar, R. Ahmad, and W. H. Butt, "Transformer-based sensor failure prediction and classification framework for UAVs," *Expert Systems with Applications*, vol. 248, p. 123415, Aug. 2024, doi: 10.1016/j.eswa.2024.123415.

[18] Y. Zhou, J. Shu, H. Hao, H. Song, and X. Lai, "UAV 3D online track planning based on improved SAC algorithm," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 46, no. 1, Dec. 2023, doi: 10.1007/s40430-023-04570-7.