An Attention-Based Temporal Graph Neural Network for Enhanced Multi-Unmanned Aerial Vehicle Obstacle Prediction

Samah Alzanin
Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Kharj, Saudi Arabia
s.alzanin@psau.edu.sa (corresponding author)

Received: 18 November 2025 | Revised: 22 December 2025 and 10 January 2026 | Accepted: 13 January 2026

 DOI: https://doi.org/10.48084/etasr.16356

ABSTRACT

In recent years, Unmanned Aerial Vehicles (UAVs) have shown promise for autonomous sensing. UAVs are employed for many applications that comprise mapping, surveillance, tracking, and searching operations. Discovering an effective path between available resources and a target goal is a vital concern and has been the focus of recent research. Various path-planning models are employed to find an effective path for a UAV to navigate from a resource to a goal with obstacle avoidance. Artificial Intelligence (AI) is a growing technology finding applications in multiple industries. The incorporation of AI into UAVs is inducing rapid growth in this area by improving efficacy and flight safety. Machine Learning (ML) models enable UAVs to make real-time decisions in complex environments, achieving optimal solutions under hardware constraints. Various analyses of UAVs have utilized numerous ML approaches to improve the performance of UAVs. This article introduces a novel framework titled Attention-based Temporal Graph Neural Network for Multi-Unmanned Aerial Vehicle Obstacle Prediction (ATGN-MUAVOP). The aim is to accurately classify potential obstacles to improve the efficiency of autonomous UAV navigation. Initially, z-score normalization is used to standardize the input features. For effective obstacle prediction, a Graph Convolutional Network (GCN) method is employed. Furthermore, a Temporal Convolutional Network with an Attention Mechanism (TCN-AM) is applied for classification. The performance validation of the ATGN-MUAVOP methodology illustrates a superior accuracy of 98.28% over existing methods on the UAV Autonomous Navigation dataset.

Keywords-obstacle prediction; Temporal Convolutional Network (TCN); Attention Mechanism (AM); Unmanned Aerial Vehicle (UAV); Deep Learning (DL)

I. INTRODUCTION

An Unmanned Aerial Vehicle (UAV) flies either autonomously by executing its program or via radio remote control. UAVs offer several advantages, including efficient utilization, lower cost, and high maneuverability. They are widely applied in civilian, military, and commercial domains. In the civilian sector, UAVs can perform high-risk and high-altitude tasks, namely pesticide spraying, antenna inspection, and express transportation [1]. In the military domain, UAVs may be utilized to perform complex tasks involving decoy fire, intelligence gathering, reconnaissance, and target positioning. In the commercial domain, UAVs can perform comprehensive aerial photography from multiple angles to capture optimal imagery.

With the rapid development of UAV technologies, their safety issues have become more critical, and UAV safety supervision methods have evolved. UAV trajectory prediction is a core technique for UAV safety supervision and represents an essential step toward more precise and automatic UAV method. UAV trajectory prediction refers to the procedure of approximating the upcoming UAV trajectory using the UAV's local data. Trajectory prediction provides precise navigation information to enhance UAV control and path planning [2].

Recently, Machine Learning (ML) models, a subfield of Deep Learning (DL), have demonstrated strong performance in predictive tasks and are widely utilized in UAV-related applications. Long Short-Term Memory (LSTM) networks and Recurrent Neural Networks (RNNs) are particularly effective at capturing temporal dependencies in sequential data. These models are well suited for forecasting time-sequential information, such as UAV trajectories, where patterns unfold over time. DL-based trajectory prediction has become increasingly important, not only for UAVs in the airspace but also for ground vehicles and maritime vessels. The trajectory prediction of vehicles benefits from the structured nature of road traffic systems, which creates strong correlations with neighboring trajectories [3]. Therefore, vehicle trajectories are comparatively easier to predict. Multiple authors have employed the social Generative Adversarial Network (GAN) for forecasting automobile trajectories.

Advanced Neural Network (NN) architectures [4], including Graph Neural Networks (GNNs), RNNs, LSTM, Transformers, and multi-drone frameworks, can be combined with multimodal sensor data, such LiDAR, vision, and communication signals, to develop a comprehensive understanding of operational conditions. These approaches enable the prediction of highly complex and often multimodal trajectories, allowing UAV swarms to make collaborative, proactive, and intelligent decisions for path planning, task coordination, and collision avoidance.

Several studies have developed advanced NN frameworks for trajectory prediction. Authors in [5] presented the Heterogeneous Driving Graph Transformer (HDGT), a backbone modeling the driving scene as a heterogeneous graph with various kinds of edges and nodes. Multiple node types are linked according to their semantic relations. To encode spatial relations, the coordinates of each edge and its in-nodes are represented in a local node-centric coordinate system. For aggregation within the GNN, a hierarchical transformer framework is employed to handle the heterogeneous nature of the inputs. Authors in [6] suggested the Graph-driven Interaction-aware Multimodal Trajectories Prediction (GIMTP) framework, intended for probabilistically forecasting future vehicle trajectories by capturing interactions effectively. To holistically capture both temporal and spatial dependencies embedded in the dynamic adjacency matrix, this method integrates the Diffusion Graph Convolutional Network (DGCN), producing a graph embedding that encodes both past and future states. Authors in [7] explored multi-agent trajectory prediction by introducing a graph-driven social feature extractor and a patching-driven temporal feature extractor, allowing efficient feature extraction and cross-situation generalization. They further proposed an approach based on explicit modality modulation to integrate social and temporal characteristics, creating an effective single-phase inference pipeline.

Authors in [8] presented a map-free trajectory prediction technique termed Trajectories Relative dual-frame Prediction (TR-Pred). In this approach, LSTM embeds the trajectories of each agent, and GNNs extract latent traffic information, such as drivable areas, traffic control states, and lane lines. The relative frame uses a temporal transformer to capture local relative movement between agents, followed by GNN-based extraction of communication data for each target agent. Authors in [9] developed a driving behavior intention detection mechanism and a trajectory prediction mechanism using DL. This system employs an encoder-decoder structure, where the encoder encodes past states of nearby agents as a context vector, and the decoder, together with a Mixture Density Network (MDN), integrates the context vector with classified driving actions to forecast the probability distribution of future trajectories. In [10], a Multi-Granularity Situations Understanding (MGSU) framework was presented to study scene layout at various granularities. This framework consists of three mechanisms: (1) an inverse reinforcement learning mechanism generates optimal paths through grid-driven policy sampling and outputs multiple candidate scene paths; (2) a coarse-tuned fusion mechanism utilizes cross-attention to fuse the monitored trajectories with semantic scene data; (3) a fine-tuned fusion mechanism combines the monitored trajectories with the scene paths to generate multiple future trajectories. Authors in [11] presented the Variational Non-Autoregressive Graph Transformer (VNAGT), which integrates a conditional variational autoencoder with a non-autoregressive method, enabling simultaneous prediction of multiple trajectories with reduced latency. A unified graph attention-driven mechanism captures temporal and social interactions, suitable for heterogeneous and homogeneous multi-agent scenarios, facilitating seamless integration of available class data.

Although these studies demonstrate effective performance, high computational cost and limited real-time scalability remain significant challenges, as many models rely on heterogeneous graph constructions, heavy map dependence, or multi-stage pipelines. Spatial–temporal modeling, interaction learning, and temporal dependencies are often inadequately balanced, and limitations exist in handling uncertainty and adapting to dense environments. A clear research gap exists in developing a unified, lightweight graph–temporal framework with effective attention, capable of modeling interactions and temporal dependencies without relying on external maps or multi-stage inference.

To address this gap, this paper proposes the Attention-based Temporal Graph Neural Network for Multi-Unmanned Aerial Vehicle Obstacle Prediction (ATGN-MUAVOP), designed to classify potential obstacles and enhance the efficiency of autonomous UAV navigation. The key contributions of this work are:

- Z-score normalization is applied to standardize input features.
- A Graph Convolutional Network (GCN) effectively captures obstacle structures and their interconnections to support accurate obstacle prediction.
- A Temporal Convolutional Network with an Attention Mechanism (TCN-AM) is employed to analyze sequential UAV data. The Attention Mechanism (AM) allows the model to focus on significant temporal features, improving classification accuracy.

II. MATERIALS AND METHODS

This article presents a systematic approach for precise obstacle classification that integrates data preprocessing, obstacle prediction, and classification. Figure 1 illustrates the workflow of the ATGN-MUAVOP technique.

A. Z-Score Normalization

In the initial step, z-score normalization is employed to preprocess the input data, ensuring uniform data distribution [12]. This standardizes features to zero mean and unit variance, preventing scale dominance among variables. It also provides stable and faster model convergence. Compared to min–max or max normalization, z-score normalization is more robust to varying data distributions and better preserves relative feature variations, which are crucial for learning.

The features are standardized by subtracting mean values and dividing by the standard deviation:

$$X^* = \frac{X - \overline{X}}{\sigma_X} \quad (1)$$

This process transforms the attributes to have a mean of 0 and a standard deviation of 1.

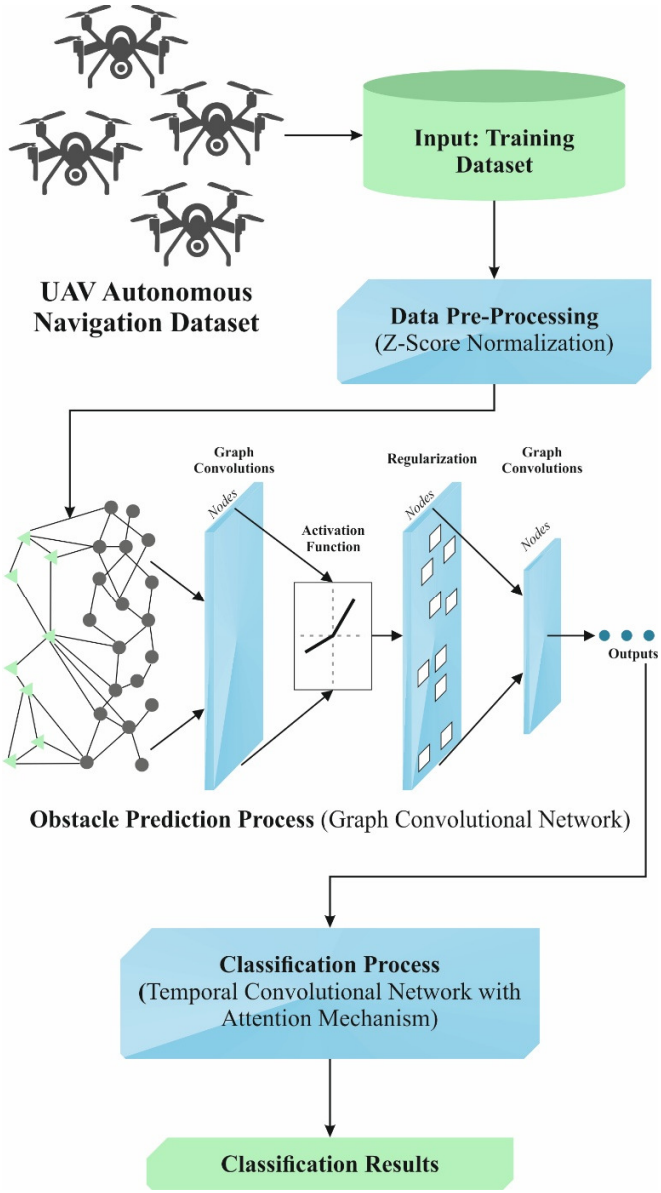

Fig. 1. Workflow of the ATGN-MUAVOP technique.

Once training is over, while the NN learns the mapping from standardized input to output attributes, the original mapping is recovered by de-normalization:

$$X = X^* \odot \sigma_X + \overline{X} \quad (2)$$

where $\odot$ signifies element-wise multiplication. Z-score standardization is widely utilized and shares similarities with other normalization approaches. It scales NN attributes to a comparable level, assuring uniform contribution from every feature to the learning process. Thus, it more precisely represents latent relations in the data and improves learning performance. In practical NN applications, mapping relations are usually determined based on normalized input and output attributes.

The spatiotemporal coordinates of the flow $\{x, y, t\}$ act as input, whereas the corresponding flow quantities $\{u, v, p\}$ are the output. The prediction error is measured as:

$$R_{L_2} = \frac{\|\hat{U} - U\|}{\|U\|} \quad (3)$$

Here, $\|U\|$ signifies the L_2 norm of the training data, and $\|\hat{U} - U\|$ represents the L_2 norm of the deviation between predicted and true data. A smaller R_{L_2} value implies more accurate predictions. The fitting of training data using normalized input-output attributes highlights the significance of normalization model when training an NN.

B. Proposed Graph Convolutional Network Architecture

In the prediction step, a GCN is leveraged to enable secure and effective drone navigation [13]. GCNs build upon Convolutional Neural Networks (CNNs), which capture the spatial correlations of Euclidean data. While CNNs perform well on structured data, they struggle with non-Euclidean structures, such as graphs, where data are represented as nodes and edges defining communication network topologies. GCNs address this limitation by directly managing the spatial correlations of graph-structured data, working on the edges and nodes in the graph. They achieve feature aggregation by performing graph convolution operations among the central node and its adjacent nodes. The operation can be expressed as:

$$H^{(l+1)} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \quad (4)$$

where $H^{(l)}$ signifies the node features in Hidden Layer (HL) l, and $H^{(l+1)}$ denotes the node features in HL $l+1$. The matrix $\hat{A} = \widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}$ is the symmetric normalized adjacency matrix, derived from $\tilde{A}$, which incorporates self-loops for every node. The adjacency matrix with self-loops is computed as $\tilde{A} = A + I_N$, where I_N indicates an N-dimensional identity matrix representing the self-loop of each node. $W^{(l)}$ denotes the weight matrix in HL l, and $\widetilde{D}$ is the degree matrix of $\tilde{A}$. The GCN updates the feature representation of each central node by computing a weighted average of its neighbors' features across multiple layers. After l layers of graph convolution, the features of a central node aggregate not only its own information but also the features from its first to l-th order neighbors. This stacking of graph convolution layers captures the global network topology and improves the feature representation of every node.

C. Obstacle Classification Model

In the final step, the TCN-AM is implemented for a precise classification process [14]. TCN is a convolution-based technique intended for sequence modeling, capable of acquiring both short- and long-range dependencies in time-series data. Unlike RNNs like LSTM and Gated Recurrent Unit (GRU), TCNs leverage dilated convolutions that enable a huge receptive area without increasing the number of parameters. In this work, the TCN comprises four temporal convolutional

layers with kernel size 3, dilation rates {1, 2, 4, 8}, 128 hidden channels, ReLU activation, and a dropout of 0.2. A single-head AM is applied to the final TCN output.

1) Dilated Convolutions

Let the input sequence be $X = [x_1, x_2, \dots, x_T]$, where T denotes the sequence length. The output of a 1D convolution with filter size k, dilation rate d, and weight vector $W = [w_1, w_2, \dots, w_k]$ at time step t is given by:

$$y_t = \sum_{i=1}^{k} w_i \cdot x_{t-d\cdot(i-1)} \quad (5)$$

The dilation rate d regulates the spacing among the components in the input sequence considered by the convolution, permitting the receptive area to increase exponentially as the number of layers increases. This allows the TCN to acquire long-term dependencies without increasing network depth.

2) Residual Connections

To mitigate the vanishing gradient problem and enhance information flow in deep TCNs, residual connections are leveraged. The residual connection at layer l is expressed as:

$$y_t^{(l)} = f\left(X_t^{(l)}\right) + X_t^{(l-1)} \quad (6)$$

where $X_t^{(l-1)}$ signifies the input to the previous layer, and $f\left(X_t^{(l)}\right)$ denotes the output of the dilated convolution at layer l. Residual connections facilitate adequate data flow and allow the network to capture both short- and long-term dependencies.

3) Attention Mechanism

At the same time, the AM concentrates on the most informative time steps to improve prediction accuracy. Applied to the output of the final TCN layer, it refines temporal features. Let h_t denote the TCN output at each time step. The AM calculates a context vector c that is a weighted sum of hidden states:

$$e_t = v^T \tanh(W_h h_t + b_h) \quad (7)$$

$$\alpha_t = \frac{\exp(e_t)}{\sum_{i=1}^{T} \exp(e_i)} \quad (8)$$

$$c = \sum_{t=1}^{T} \alpha_t \, h_t \quad (9)$$

where e_t denotes the unnormalised attention score; W_h and b_h refer to learnable parameters, and α_t indicates the normalised attention weight for time step t. The context vector c acquires the most significant information from the overall sequence and is employed as input to the final classification layers.

III. RESULTS AND DISCUSSION

In this section, the performance of the ATGN-MUAVOP approach is evaluated using the UAV Autonomous Navigation dataset [15]. The method runs on Python 3.6.5 with an Intel i5-8600k CPU, 4 GB GPU, 16 GB RAM, 250 GB SSD, and 1 TB HDD, using a learning rate of 0.01, ReLU, 50 epochs, 0.5 dropout, and a batch size of 5. The dataset contains a total of 5,000 cases divided into two classes for obstacle detection. The "Has_Obstacle" class comprises 4,764 cases, signifying scenarios where an obstacle is present, whereas the "Obstacle_No" class consists of 236 cases, representing instances where no obstacle is detected. This distribution shows a class imbalance, with the majority of examples belonging to the obstacle-present category. The data are suitable for training and testing ML models for autonomous navigation, obstacle detection, or robot safety systems. Figure 2 illustrates the class distribution.

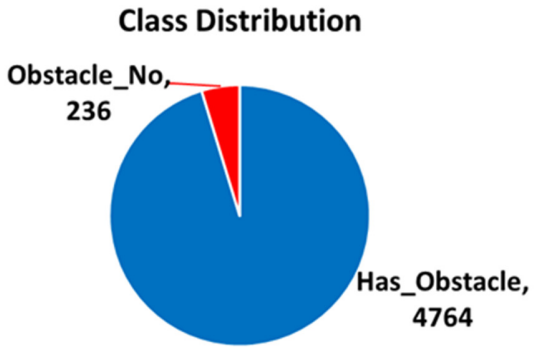

Fig. 2. Class distribution of the UAV Autonomous Navigation dataset.

Table I presents the prediction results of ATGN-MUAVOP compared with existing techniques using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) metrics. Among all approaches, ATGN-MUAVOP achieves the lowest values: MSE = 0.0341, RMSE = 0.0341, and MAE = 0.0341. Meanwhile, the existing models, including NRAT, Attention-LSTM, FlightBERT++, Inverted Transformer, PSTT, CNN-LSTM-SE, and CNN-LSTM, yield higher error values.

TABLE I. PREDICTION PERFORMANCE OF ATGN-MUAVOP AND EXISTING METHODS

Method	MSE	RMSE	MAE
NRAT	0.2533	0.5033	0.0263
Attention-LSTM	0.2582	0.5081	0.0240
FlightBERT++	0.2463	0.4963	0.0465
Inverted Transformer	0.1750	0.4183	0.0482
PSTT	0.2546	0.5046	0.0295
CNN-LSTM-SE	0.2341	0.4838	0.0167
CNN-LSTM	0.1920	0.4382	0.0426
ATGN-MUAVOP (proposed)	0.0341	0.0341	0.0341

Figure 3 compares the battery levels under two states: obstacle absent (0), and obstacle present (1). Both distributions have similar median battery levels, but the spread and range vary slightly. The obstacle-absent state exhibits a broader interquartile range, signifying higher variability in battery levels, whereas the obstacle-present state has a narrower distribution, implying more constant battery readings. Overall, battery levels remain relatively stable through both states.

Figure 4 presents the correlation matrix of the ATGN-MUAVOP model, demonstrating effective classification and accurate obstacle detection.

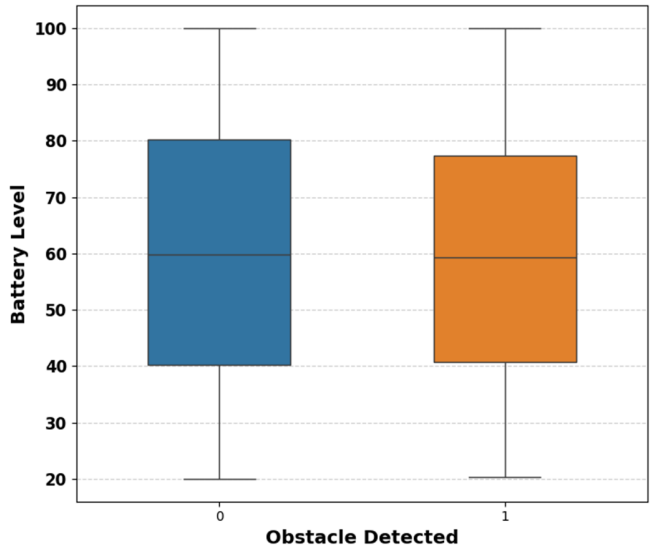

Fig. 3. Battery levels when obstacles are absent (0) and present (1).

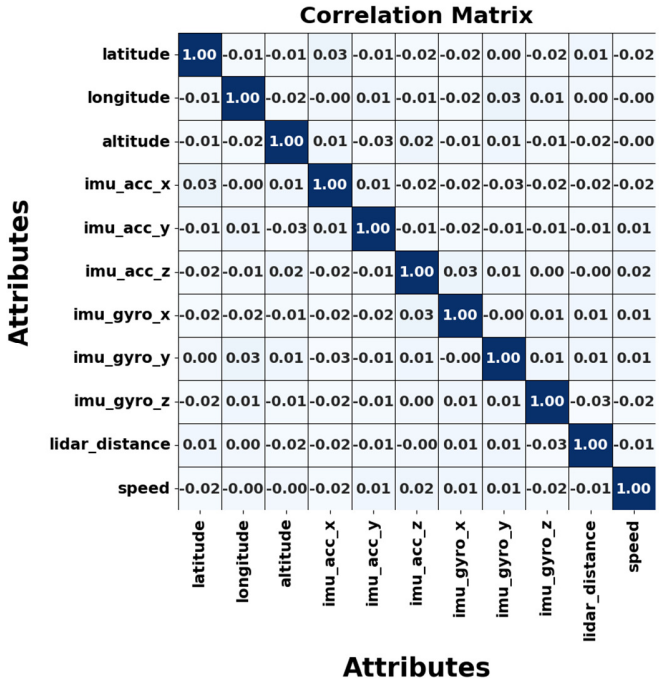

Fig. 4. Correlation matrix of the ATGN-MUAVOP model.

Table II presents the classifier performance of the ATGN-MUAVOP method under a 70% training and 30% testing split using multiple evaluation metrics. During training, the model achieves strong performance, with average values of accuracy ($accur_y$) = 98.06%, precision ($preci_n$) = 96.50%, recall ($recal_l$) = 98.06%, $F1_{score}$ = 97.26%, and area under the curve (AUC_{score}) = 98.06%. During testing, the model reaches average values of accuracy $accur_y$ = 98.28%, $preci_n$ = 95.64%, $recal_l$ = 98.28%, $F1_{score}$ = 96.92%, and AUC_{score} = 98.28%.

TABLE II. CLASSIFIER PERFORMANCE OF ATGN-MUAVOP UNDER 70%:30%

Class label	$Accur_y$	$Preci_n$	$Recal_l$	$F1_{score}$	AUC_{score}
Training phase (70%)					
Has_Obstacle	99.64	99.82	99.64	99.73	98.06
Obstacle_No	96.47	93.18	96.47	94.80	98.06
Average	98.06	96.50	98.06	97.26	98.06
Testing phase (30%)					
Has_Obstacle	99.58	99.86	99.58	99.72	98.28
Obstacle_No	96.97	91.43	96.97	94.12	98.28
Average	98.28	95.64	98.28	96.92	98.28

Table III presents a comparison analysis of ATGN-MUAVOP with existing models [16]. The results show that ATGN-MUAVOP achieves the highest $accur_y$ (98.28%), outperforming HOG+Cascading (82.50%), CNN (AlexNet) (72.00%), YoloV9 (93.30%), TOOCM (96.70%), Faster R-CNN (91.00%), CenterNet (90.50%), and MMA (88.90%). Moreover, based on $F1_{score}$, ATGN-MUAVOP also achieves the highest value (96.92%). These results demonstrate that the proposed method efficiently captures multi-UAV spatiotemporal interactions and intrinsic obstacle dynamics, providing robust and consistent predictions across diverse and cluttered environments where existing methods struggle. The compared models were evaluated under identical experimental conditions.

TABLE III. COMPARATIVE ANALYSIS OF ATGN-MUAVOP WITH EXISTING MODELS

Approach	$Accur_y$	$Preci_n$	$Recal_l$	$F1_{score}$
HOG+Cascading [16]	82.50	83.86	91.93	89.34
CNN (AlexNet) [16]	72.00	87.58	83.41	84.77
YoloV9 [16]	93.30	92.41	85.80	86.00
TOOCM [16]	96.70	94.27	96.28	95.20
Faster R-CNN [16]	91.00	88.50	86.90	87.70
CenterNet [16]	90.50	88.00	86.00	87.00
MMA [16]	88.90	86.50	84.70	85.60
ATGN-MUAVOP (proposed)	98.28	95.64	98.28	96.92

IV. CONCLUSION

This study introduced the Attention-based Temporal Graph Neural Network for Multi-Unmanned Aerial Vehicle Obstacle Prediction (ATGN-MUAVOP) model to accurately classify potential obstacles and enhance the efficacy of autonomous Unmanned Aerial Vehicle (UAV) navigation. The model integrates z-score normalization, Graph Convolutional Network (GCN)–based obstacle prediction, and Temporal Convolutional Network with Attention Mechanism (TCN-AM)–based classification. Performance validation demonstrates that ATGN-MUAVOP achieves a superior accuracy of 98.28% compared with existing methods on the UAV Autonomous Navigation dataset.

The model's limitations include reliance on controlled datasets, which may reduce generalization in highly dynamic and previously unseen real-world scenarios. Performance may

also degrade in environments with high agent density due to increased interaction complexity. Additionally, the model is sensitive to noisy or incomplete sensory data. Future work should focus on improving robustness under sparse observations and extending scalability for large-scale, real-time deployments.

FUNDING AND DATA AVAILABILITY STATEMENT

The authors extend their appreciation to Prince Sattam bin Abdulaziz University for funding this research work through project number PSAU/2025/01/34645. The data supporting the findings of this study are openly available in the Kaggle repository at https://www.kaggle.com/datasets/ziya07/uav-autonomous-navigation-dataset.

REFERENCES

[1] S. A. H. Mohsan, M. A. Khan, F. Noor, I. Ullah, and M. H. Alsharif, "Towards the Unmanned Aerial Vehicles (UAVs): A Comprehensive Review," *Drones*, vol. 6, no. 6, June 2022, Art. no. 147, https://doi.org/10.3390/drones6060147.

[2] A. Ahmed and R. N. Farhan, "Autofocus Vision System Enhancement for UAVs via Autoencoder Generative Algorithm," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 18867–18872, Dec. 2024, https://doi.org/10.48084/etasr.8519.

[3] P. Shukla, S. Shukla, and A. Kumar Singh, "Trajectory-Prediction Techniques for Unmanned Aerial Vehicles (UAVs): A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 3, pp. 1867–1910, June 2025, https://doi.org/10.1109/COMST.2024.3471671.

[4] S. Rezwan and W. Choi, "Artificial Intelligence Approaches for UAV Navigation: Recent Advances and Future Challenges," *IEEE Access*, vol. 10, pp. 26320–26339, 2022, https://doi.org/10.1109/ACCESS.2022.3157626.

[5] X. Jia, P. Wu, L. Chen, Y. Liu, H. Li, and J. Yan, "HDGT: Heterogeneous Driving Graph Transformer for Multi-Agent Trajectory Prediction via Scene Encoding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 11, pp. 13860–13875, Nov. 2023, https://doi.org/10.1109/TPAMI.2023.3298301.

[6] K. Wu, Y. Zhou, H. Shi, X. Li, and B. Ran, "Graph-Based Interaction-Aware Multimodal 2D Vehicle Trajectory Prediction Using Diffusion Graph Convolutional Networks," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 2, pp. 3630–3643, Feb. 2024, https://doi.org/10.1109/TIV.2023.3341071.

[7] L. Huajian, D. Wei, F. Kunpeng, W. Chao, and G. Yongzhuo, "PMM-Net: Single-Stage Multi-Agent Trajectory Prediction With Patching-Based Embedding and Explicit Modal Modulation," *IEEE Robotics and Automation Letters*, vol. 10, no. 7, pp. 6664–6671, July 2025, https://doi.org/10.1109/LRA.2025.3568623.

[8] Z. Wang, J. Zhang, J. Chen, and H. Zhang, "Spatio-Temporal Context Graph Transformer Design for Map-Free Multi-Agent Trajectory Prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 1369–1381, Jan. 2024, https://doi.org/10.1109/TIV.2023.3329885.

[9] Z. Gao, M. Bao, F. Gao, and M. Tang, "Probabilistic multi-modal expected trajectory prediction based on LSTM for autonomous driving," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 238, no. 9, pp. 2817–2828, Aug. 2024, https://doi.org/10.1177/09544070231167906.

[10] B. Yang, J. Yang, R. Ni, C. Yang, and X. Liu, "Multi-granularity scenarios understanding network for trajectory prediction," *Complex & Intelligent Systems*, vol. 9, no. 1, pp. 851–864, Feb. 2023, https://doi.org/10.1007/s40747-022-00834-2.

[11] X. Chen, H. Zhang, Y. Hu, J. Liang, and H. Wang, "VNAGT: Variational Non-Autoregressive Graph Transformer Network for Multi-Agent Trajectory Prediction," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 10, pp. 12540–12552, Oct. 2023, https://doi.org/10.1109/TVT.2023.3273230.

[12] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Burlington, MA, USA: Morgan Kaufmann Publishers, 2012.

[13] T. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[14] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal Convolutional Networks: A Unified Approach to Action Segmentation," in *14th European Conference on Computer Vision*, Amsterdam, Netherlands, 2016, pp. 47–54, https://doi.org/10.1007/978-3-319-49409-8_7.

[15] "UAV Autonomous Navigation Dataset." Kaggle. [Online]. Available: https://www.kaggle.com/datasets/ziya07/uav-autonomous-navigation-dataset.

[16] L. Wang *et al.*, "Image-based obstacle detection methods for the safe navigation of industrial unmanned aerial vehicles," *Scientific Reports*, vol. 15, no. 1, Oct. 2025, Art. no. 36063, https://doi.org/10.1038/s41598-025-19904-9.