An Intelligent Multi-Drone Navigation-Based Trajectory Prediction and Classification Framework Using a Hybrid Recurrent Neural Network Model

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: 13 November 2025 | Revised: 29 November 2025 and 11 December 2025 | Accepted: 13 December 2025

 DOI: https://doi.org/10.48084/etasr.16258

ABSTRACT

Unmanned Aerial Vehicles (UAVs), commonly referred to as drones, have drawn attention from various fields. Navigation is a core module of the drone guidance system, along with control and navigation methods, which are vital subsystems of the drones. Numerous drone navigation technologies have been proposed, comprising GPS, reference frames, and models. Additionally, computer vision models have been effectively verified and improved in real-world applications. One of the most noticeable advances is the application of computer vision in drones, which provides the technology for visual navigation and obstacle avoidance, enabling unmanned control. Recent developments and the advances of Deep Learning (DL)-based solutions have also made the navigation of autonomous vehicles more feasible. This study presents an Intelligent Multi-Drone Navigation-Based Trajectory Prediction and Classification System (IMDN-TPCS) model, in which data pre-processing is performed using min-max normalization. This is followed by the employment of a Bidirectional Gated Recurrent Unit with an attention mechanism (BiGRU-Attn) method for an accurate trajectory prediction. Finally, the Deep Belief Network (DBN) method is utilized for classification. The comparative study of the IMDN-TPCS method demonstrated a superior accuracy of 98.80% compared to other models on the UAV Autonomous Navigation dataset.

Keywords-drones; trajectory prediction; deep belief network; bidirectional gated recurrent unit; attention mechanism

I. INTRODUCTION

A UAV is an aircraft without a human pilot onboard, controlled autonomously or using a remote operator employing pre-programmed instructions [1]. The growth of UAVs has benefited from advances in computer science, sensor technology, electronics, and aviation technology. UAVs have gradually become ubiquitous in civilian applications due to technical improvements and cost reduction [2]. The former play an important role in cargo transportation, rescue operations, agriculture, disaster monitoring, aerial photography, scientific research, and film production [3]. UAV trajectory forecasting supervises the aircraft movement parameters, allowing observation and control of relevant parameters during operation. It is a technology that improves UAV safety management and provides more precise navigation information for UAV control [4]. The conventional trajectory forecast approach is mainly based on the Bird's Eye View (BEV) method, which offers different observational viewpoints within a unified feature space [5]. This approach unifies BEV features with intra-group interaction and sparse feature extraction. Conventional methods struggle in multi-drone remote sensing due to spatial heterogeneity and sparse targets, challenging trajectory forecasting. DL techniques, particularly Generative Adversarial Networks (GAN), Convolutional Neural Networks (CNN), and recurrent models, are reliable methods for forecasting trajectories.

Authors in [6] presented an LSN-GTDA model, containing a pair of symmetrical U-Net networks. The structure symmetrically allocates trajectories, forecasting mechanisms, and semantic segmentation. Authors in [7] introduced a coarse-to-fine feature fusion network, C2F-Net, using coarse-tuned data communication for guiding the fusion of fine-tuned features. This method involves a selective coarse-to-fine feature collaboration mechanism. Authors in [8] proposed an innovative structure by employing ML on feature extraction from aerial imaging. The developed information, integrated with GPS coordinates attained in signal accessibility, is utilized for training a linear ML (Linear Regression) to approximate longitude and latitude in the event of GPS signal damage. Authors in [9] presented a region-driven relation learning model, which social communication through region-wise

dynamics of joint condition, the variations in crowd density. Authors in [10] employed data of only the present condition and designed direction for predicting the upcoming trajectory of many vehicles at intersections.

In [11], an enhanced LSTM trajectories forecast network, CNN-Attention-LSTM, which depends on a convolution fusion framework as well as attention enhancement, is presented. Initially, the native You Only Look Once (YOLO) version 5 is enhanced to improve its recognition capability. Authors in [12] enhanced UAV navigation utilizing YOLOv8-StrongSort for obstacle detection, Dynamic Data Memory (DDM) for Deep Q-Networks (DQN) learning, and an Artificial Potential Field (APF)-based reward for improved obstacle avoidance. Authors in [13] examined how ML and Reinforcement Learning (RL) improve UAV navigation and obstacle avoidance. Authors in [14] utilized KiteRunner, which integrates UAV orthophoto-based global planning, Diffusion Model (DM)-based local path generation, and Contrastive Language-Image Pretraining (CLIP) with Generative Pretrained Transformer (GPT) for natural language understanding.

Authors in [15] presented a model utilizing a dynamic Spatio-Temporal Graph Convolutional Network (ST-GCN) with a weighted adjacency matrix and Temporal Convolutional Network (TCN) to capture spatial and temporal dependencies. Authors in [16] developed a model by integrating Faster R-CNN, YOLO models for detailed feature extraction, and Fine-Tuned Transformer (FT-Transformer) for classification. They also used Multi-Channel Attention YOLOv7 (MCA-YOLOv7), Depth-View Detection Transformer (DV-DETR), and Hierarchical DL for Intelligent Surveillance in UAV Imagery (HDL-ISTUAVI) for comparison. Although the existing studies are efficient, they perform poorly when integrated with multi-drone spatial-temporal dependencies and real-time decision-making under dynamic conditions, highlighting a research gap in accurately predicting trajectories while maintaining robustness in intrinsic environments.

The present study introduces an IMDN-TPCS for efficient trajectory prediction and classification using state-of-the-art DL techniques. The major contributions of this study are:

- Multi-drone trajectory data are scaled by the min-max normalization, improving the input quality and mitigating computational complexity to improve the accuracy and efficiency of trajectory prediction.
- The BiGRU-Attn method is employed for accurately predicting multi-drone trajectories, thus capturing intrinsic spatiotemporal dependencies. It also utilizes bidirectional data flow and Attention Mechanisms (AMs) to focus on temporal features.
- The DBN is implemented for classifying predicted multi-drone trajectories, enabling the effective data of movement patterns. It captures hierarchical features from trajectory data for enhanced decision-making and improves the reliability and accuracy of multi-drone behavior analysis.

The novelty of the study lies in the integrated framework combining BiGRU-Attn for precise trajectory prediction and DBN for classification. This approach is specifically tailored for intelligent multi-drone navigation, improving both prediction accuracy and classification performance in intrinsic aerial environments.

II. PROPOSED MODEL

In this study, the IMDN-TPCS model follows a systematic process to achieve accurate trajectory prediction and classification. The model comprises min-max normalization, BiGRU-Attn-based trajectory prediction, and DBN-based classification. Figure 1 illustrates the IMDN-TPCS workflow.

A. Data Pre-Processing

The min-max normalization is employed as the data pre-processing model intended to scale and standardize the variables or features in the data [17]. The objective of min-max normalization is to place every variable on a uniform scale, assisting in similarities and improving the execution of ML models. This technique is chosen for its efficiency in scaling all features to a uniform range, preserving relative relationships and enhancing model convergence, unlike standardization or z-score methods, which may distort data distribution in bounded trajectory data. The values of the numerical variables are transformed into a pre-determined range, typically between zero and one. To minimize the impact of outliers, min–max normalization helps create comparable data, ensuring that diverse variables are on a similar scale. A single variable is standardized using:

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

where X is the original value of the variable, $X_{normalized}$ is the normalized value of X, and X_{max} and X_{min} denote the maximum and minimum values of variable X, respectively.

After normalization, the transformed values lie between 0 and 1; here, 0 represents the minimum value of the variable, and 1 represents its maximum value. The relative locations of every data point in the primary range were represented using values ranging from 0 to 1. After standardization, each value is transformed to fit within a similar range, between 0 and 1. Min-max normalization is preferred for standardization as it maintains the original distribution shape and maps features to a common scale [0, 1], which is specifically beneficial for distance-based models such as KNN. It helps in maintaining attributes with a larger numeric range from dominating distance evaluations, thus improving the classifier outcome.

B. Trajectory Prediction Algorithm

The prediction task is based on time-series analysis, which requires advanced DL approaches that are capable of modeling sequential dependency [18]. This helps capture both past and future dependencies in sequential data, while the AM highlights critical temporal features, giving higher prediction accuracy compared to standard Recurrent Neural Networks (RNN) or unidirectional Gated Recurrent Unit (GRU) models. RNNs and their advanced versions, such as BiGRU-Attn and Bidirectional-RNN (Bi-RNN), are particularly effective for such tasks. These models improve the capability of the model by processing temporal data and managing dynamic patterns effectively.

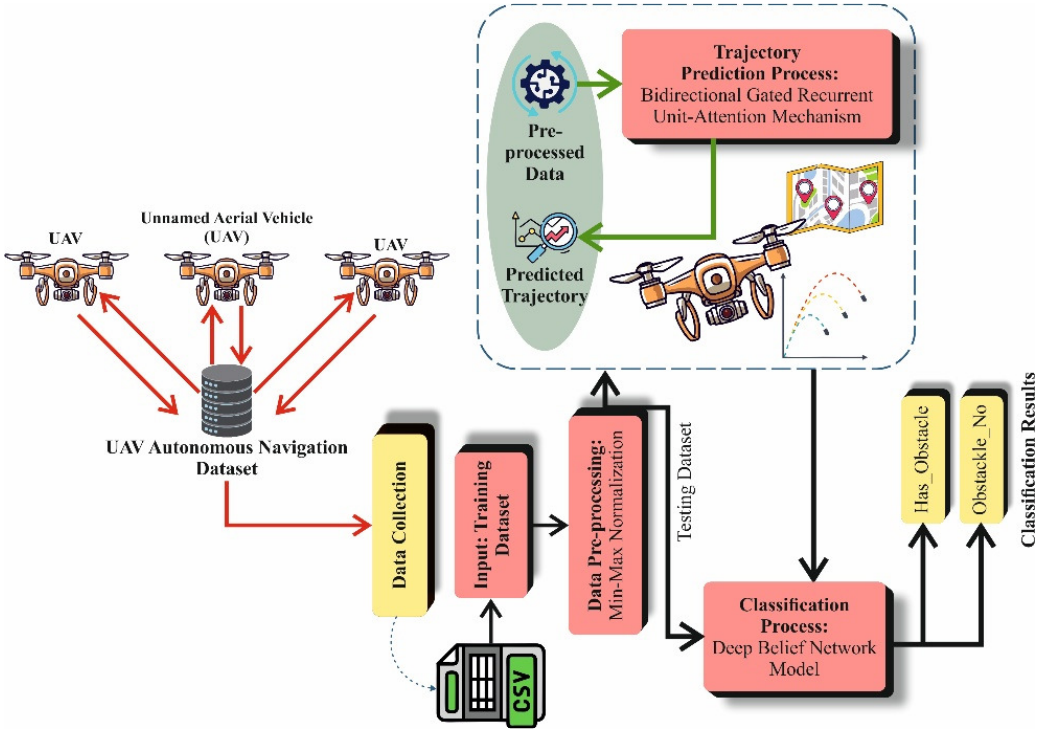


Fig. 1. Architecture of the IMDN-TPCS model.

Bidirectional RNNs extend conventional RNNs by processing a sequence of input in two directions: forward and backward. Unlike traditional RNNs that rely on past states for predictions, Bi-RNNs leverage both past and upcoming information to acquire intricate patterns in time-series data. This dual-direction method is specifically useful in a prediction task. The Bi-RNN framework is defined as:

$$\tilde{h}_t = f\left(W_{xh}^{+} x_t + W_{hh}^{+} * \tilde{h}_{t-1} + b_h^{+}\right) \quad (2)$$

$$\tilde{h} = f\left(W_{xh}^{+} * x_t + W_{hh}^{-} * \tilde{h}_{t+1} + b_h^{-}\right) \quad (3)$$

$$y_t = \left(W_{hy}^{+} * \tilde{h}_t + W_{hy}^{-} * \tilde{h}_t + b_y\right) \quad (4)$$

where f denotes the non-linear activation function such as $tanh$, $\tilde{h}_t$ and $\tilde{h}_t$ are Hidden Layers (HL) of forward and backward sequence, correspondingly, b_h^{+}, b_h^{-}, b_y is the biased vector, and W_{xh}^{+}, W_{hh}^{+}, W_{xh}^{+}, W_{hh}^{-}, W_{hy}^{-}, and W_{hy}^{-} denote weighted matrices.

The GRU is a version of RNNs, often used for addressing the challenges of vanishing gradients and imitations of conventional RNNs. It combines dual gate reset and update, which control the data flow, enabling the model to concentrate on appropriate time-steps while effectively processing longer sequences.

$$r_t = \sigma(W_{xr} x_t + W_{hr} h_{t-1} + b_r) \quad (5)$$

$$\widetilde{h_t} = tanh(W_{xh} x_t + W_{hh}(r_t \odot h_{t-1}) + b_h) \quad (6)$$

$$u_t = \sigma(W_{xu} x_t + W_{hu} h_{t-1} + b_u) \quad (7)$$

$$h_t = (1 - u_t) \odot h_{t-1} + u_t \odot \widetilde{h_t}) \quad (8)$$

where $\widetilde{h_t}$ denotes candidate HL; u_t and r_t are the update and reset gates, respectively; b_r, b_h, and b_u are biased vectors; h_{t-1} and h_{th} refers to HL at preceding and current time-steps, respectively; $\tanh(\dot{s})$ represents the hyperbolic tangent function; W_{xr}, W_{hr}, W_{xh}, W_{hh}, W_{xu}, and W_{hu} are weighted matrices; $\odot$ represents the Hadamard product; and $\sigma(s)$ denotes the activation function of the sigmoid.

To further improve GRU performance, AMs are incorporated. AM dynamically assigns weight to the order of input, allowing the GRU to concentrate on the most appropriate time-steps for prediction. The AM is defined as:

$$e_t = score(h_t, s) = h_t^T W_a s \quad (9)$$

$$\alpha_t = \frac{\exp(e_t)}{\sum_{k=1}^{T} exp(e_k)} \quad (10)$$

$$c = \sum_{t=1}^{T} \alpha_t h_t \quad (11)$$

where W_a is the attention weighted matrix; e_t denotes the alignment score among HL h_t; s and c are the context vector, denting the weighted sum of HL; and α_t denotes the attention weight for every HL.

C. Classification via DBN

After trajectory predictions, the DBN is applied for classification to proficiently categorize trajectory behaviors and navigation patterns [19]. This is chosen for its ability in learning hierarchical feature representations from intrinsic trajectory data, enabling more accurate classification of navigation patterns compared to conventional classifiers like SVM or shallow neural networks. DBN is a generative probabilistic method composed of many stochastic latent

variables. The former is developed by stacking restricted Boltzmann machines, where every RBM learns the hierarchical model of the input data in an unsupervised manner. By greedy layer-wise pretraining followed by supervised fine-tuning, DBN progressively removes features from unprocessed data. This hierarchical framework allows DBN to address issues like non-linearity, higher dimension, and inadequate labeled data in various situations. To integrate generative learning with distinctive fine-tuning, DBN is employed in various domains such as image classification. The major advantage of DBNs lies in their ability to model probability distributions and in eliminating deeper feature models without necessitating large-scale labeled data. In contrast with shallow or conventional ML approaches, DBNs establish strong generality, robustness to noise, and improved performance.

Every RBM comprises a Visible Layer (VL) and an HL, where the hidden activation from a single RBM acts as the visible input. The output layer is connected to the final HL to assist supervised tasks like a classifier. This hierarchical model enables DBN to acquire both local models and high-level dependencies in data. The DBN depends on RBM, which defines the relations between hidden and visible variables. Equation (12) presents the function of RBM for each joint configuration of HL and VL.

$$E(v,h) = -v^T W h - b^T v - c^T h \quad (12)$$

where h represents a hidden vector, v denotes the visible vector, c and b are biased vectors for HL and VL, and W represents a weighted matrix connecting HL and VL, respectively.

$$P(v,h) = \frac{1}{Z} exp\big(-E(v,h)\big) \quad (13)$$

$$Z = \sum_{v,h} \exp\big(-E(v,h)\big) \quad (14)$$

where Z represents the partition function.

$$P(v) = \frac{1}{Z} \sum_h \exp\big(-E(v,h)\big) \quad (15)$$

This expression measures the probability of an identified input v in RBM to add every possible hidden configuration. The conditional probability of HL specified in the visible vector is defined by:

$$P\big(h_j = 1 \big| v\big) = \sigma\big(\sum_i v_i W_{ij} + c_j\big) \quad (16)$$

where $\sigma(\cdot)$ denotes the logistic sigmoid function. Similarly, the reconstruction of VL from HL is:

$$P\big(v_j = 1 \big| h\big) = \sigma\big(\sum_j h_j W_{ij} + b_j\big) \quad (17)$$

Finally, to stack many RBMs, a DBN creates the hidden activation of a single layer that acts as observable subsequent inputs.

$$P(v, h^1, h^2, \ldots, h^L) = P(v|h^1) \prod_{l=1}^{L-1} P\,(h^l|h^{l+1}) P(h^L) \quad (18)$$

where $h^1, h^2, \ldots, h^L$ denotes the HL of DBN with maximum layers. Table I presents the key hyperparameters of the DBN model.

TABLE I. KEY HYPERPARAMETERS OF THE DBN MODEL

Hyperparameter	Value
RBM layers	3
Hidden unit layer	512, 256, 128
Learning rate	0.01
Batch size	64
Epoch per layer	50
Activation function	Sigmoid

III. EXPERIMENTAL RESULTS

The experimental evaluation of the IMDN-TPCS technique uses the UAV Autonomous Navigation dataset [20]. This dataset is utilized to study obstacle detection for DL-driven autonomous flight. The dataset combines multi-sensor information, involving LiDAR, GPS, UAV telemetry, and IMU, to provide obstacle avoidance and real-time path planning. It contains 5,000 instances belonging to two classes: Has_Obstacle and Obstacle_No. The complete details of this dataset are presented in Table II.

TABLE II. DATASET SUMMARY

Classes	Instances
Has_Obstacle	4764
Obstacle_No	236
Total	5000

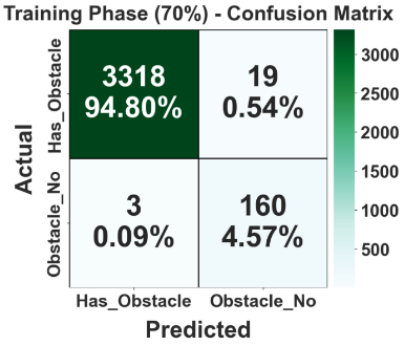
Table III compares the prediction outcome of the IMDN-TPCS method with existing techniques using MSE, RMSE, and MAE values. The IMDN-TPCS method obtained an MSE of 0.021, whereas the AttConv LSTM, GRU, CNN+LSTM-Attention, LSTM-Attention, GRU-Attention, RNN, and Bi-LSTM methods obtained an MSE of 0.080, 0.094, 0.052, 0.052, 0.053,1.235, and 3.098, respectively. Similarly, the IMDN-TPCS method obtained minimum RMSE and MAE values of 0.145 and 0.963, respectively, compared to AttConv LSTM, GRU, CNN+LSTM-Attention, LSTM-Attention, GRU-Attention, RNN, and Bi-LSTM models, which had maximum RMSE and MAE, as presented in Table III.

Figure 2 shows the confusion matrices generated by the IMDN-TPCS method using 70% data in the Training Phase (TRPH) and 30% data in the Testing Phase (TSPH). The results indicate that the IMDN-TPCS method effectively classifies and detects both classes.

TABLE III. PREDICTION PERFORMANCE OF THE IMDN-TPCS MODEL COMPARED WITH EXISTING TECHNIQUES

Model	MSE	RMSE	MAE
AttConv LSTM	0.080	0.282	17.530
GRU Model	0.094	0.307	10.199
CNN+LSTM-Attention	0.052	0.227	21.000
LSTM-Attention	0.052	0.229	12.167
GRU-Attention	0.053	0.230	30.650
RNN Algorithm	1.235	1.111	49.540
Bi-LSTM	3.098	1.760	52.110
IMDN-TPCS	0.021	0.145	0.963

(a)



(b)

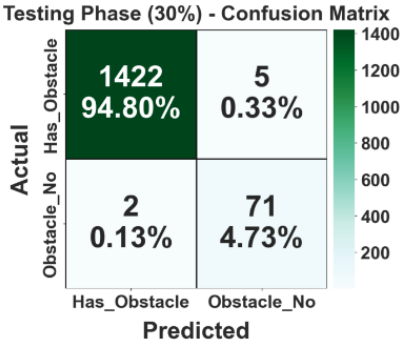


Fig. 2. Confusion matrices for: (a) TRPH (70%), and (b) TSPH (30%).

Table IV illustrates the classifier performance of the IMDN-TPCS method on a 70:30 split of TRPH/TSPH. On 70% of TRPH, the IMDN-TPCS method achieved average accuracy, precision, recall, F1-score, and AUC values of 98.80%, 94.65%, 98.80%, 96.62%, and 98.80%, respectively. Similarly, on 30% of TSPH, the IMDN-TPCS model achieved average accuracy, precision, recall, F1-score, and AUC values of 98.45%, 96.64%, 98.45%, 97.53%, and 98.45%, respectively.

Table V displays the comparative analysis of the IMDN-TPCS technique with the existing model using several evaluation metrics [20, 22, 23] on the UAV Autonomous Navigation dataset. The result indicates that the IMDN-TPCS method has achieved the highest performance, with accuracy, precision, recall, and F1-score values of 98.80%, 94.65%, 98.80%, and 96.62%, respectively. Meanwhile, the existing models, such as SVM, Decision Tree (DT), RF-OOB, PointNet++, OOB-PointNet, YOLOv7, and YOLOv8, demonstrated the lowest performance, making the IMDN-TPCS the most efficient and useful method for the classification and multi-drone trajectory prediction.

TABLE IV. CLASSIFIER OUTCOME OF IMDN-TPCS MODEL UNDER 70:30 OF TRPH/TSPH

Class labels	Accuracy	Precision	Recall	F1-score	AUC
TRPH (70%)					
Has_Obstacle	99.43	99.91	99.43	99.67	98.80
Obstacle_No	98.16	89.39	98.16	93.57	98.80
Average	98.80	94.65	98.80	96.62	98.80
TSPH (30%)					
Has_Obstacle	99.65	99.86	99.65	99.75	98.45
Obstacle_No	97.26	93.42	97.26	95.30	98.45
Average	98.45	96.64	98.45	97.53	98.45

TABLE V. COMPARATIVE PERFORMANCE OF IMDN-TPCS MODEL WITH EXISTING METHODS ON NAVIGATION DATASET

Models	Accuracy	Precision	Recall	F1-score
SVM Classifier [20]	98.60	86.56	90.74	92.05
DT [20]	96.50	93.82	95.04	95.86
RF-OOB Model [20]	98.20	85.29	86.74	94.11
PointNet++ [20]	91.00	90.61	94.73	87.58
OOB-PointNet [20]	94.40	86.28	93.58	86.20
YOLOv7 [21]	75.89	91.77	89.58	89.44
YOLOv8 [21]	89.12	91.03	96.24	94.69
IMDN-TPCS	98.80	94.65	98.80	96.62

Table VI displays the comparison evaluation of the IMDN-TPCS approach with existing models on the UAV Multi-Modal Target Tracking dataset [21, 16]. The compared Faster R-CNN, Fine-tuned YOLOv8m, MCA-YOLOv7, DV-DETR, and HDL-ISTUAVI demonstrated superior UAV-based object detection performance, while the IMDN-TPCS model achieved accuracy, precision, recall, and F1-score values of 97.86%, 97.90%, 97.90%, and 97.86%, respectively.

TABLE VI. COMPARATIVE PERFORMANCE OF IMDN-TPCS MODEL WITH EXISTING METHODS ON THE TARGET TRACKING DATASET

Methods	Accuracy	Precision	Recall	F1-score
Faster R-CNN [16]	89.52	96.55	89	90.05
Fine-tuned YOLOv8m [16]	90.22	92.78	89.53	90.13
MCA-YOLOv7 [16]	88.26	89.22	85.58	95.33
DV-DETR [16]	88.11	97.45	86.9	96.49
HDL-ISTUAVI [16]	85.89	97.11	96.63	91.74
IMDN-TPCS	97.86	97.9	97.9	97.86

TABLE VII. COMPARISON OF COMPUTATIONAL EFFICIENCY, GPU USAGE, AND INFERENCE TIME

Models	FLOPs (M)	GPU (M)	Inference time (ms)
Bilstm	6.97	4343	31.13
Transformer Encoder	16.73	2677	32.35
EGCN	428.97	3220	18.88
ADDM-STFTP	106.82	4528	45.32
IMDN-TPCS	2.09	745	10.99

Table VII presents the computational efficiency of the IMDN-TPCS model [24]. BiLSTM and Transformer Encoder have moderate FLOPs and inference times of approximately 31-32 ms. EGCN is computationally heavy but faster at 18.88 ms, while ADDM-STFTP is slower at 45.32 ms. The IMDN-TPCS model demonstrated the lowest FLOPs and GPU usage, and achieved the fastest inference time of 10.99 ms, making it suitable for real-time applications.

IV. CONCLUSION

This study presents an Intelligent Multi-Drone Navigation–Based Trajectory Prediction and Classification System (IMDN-TPCS). The model includes min-max normalization, Bidirectional Gated Recurrent Unit with an attention mechanism (BiGRU-Attn)-based trajectory prediction, and Deep Belief Network (DBN)-based classification. The

comparison study of the IMDN-TPCS method demonstrated a superior performance with an accuracy of 98.80% compared to other models on the Unmanned Aerial Vehicles (UAV) Autonomous Navigation dataset. However, the study has some limitations, including the limited dataset diversity, model generalizability, and scalability when applied to larger drone swarms. Furthermore, the interpretability and robustness of the model in highly dynamic or cluttered environments remain limited. Additionally, the method is limited to a controlled testing environment and is not deployable in a real-world application. Future work should concentrate on testing the framework on real-world flight data, integrating Reinforcement Learning (RL) for adaptive decision-making, and applying Transfer Learning (TL) to improve performance across diverse operational scenarios. Future studies should also concentrate on broadening testing scenarios and integrating more contextual factors for more robust trajectory prediction and classification.

DATA AVAILABILITY STATEMENT

The dataset used in this study is publicly available at https://www.kaggle.com/datasets/ziya07/uav-autonomous-navigation-dataset. and https://www.kaggle.com/datasets/ziya07/uav-multi-modal-target-tracking-dataset.

FUNDING

The authors extend their sincere appreciation to Prince Sattam bin Abdulaziz University for funding this research work through the project No. PSAU/2025/01/34645.

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. A. Laghari, A. K. Jumani, R. A. Laghari, and H. Nawaz, "Unmanned Aerial Vehicles: a Review," *Cognitive Robotics*, vol. 3, pp. 8–22, 2023, https://doi.org/10.1016/j.cogr.2022.12.004.

[3] S. A. H. Mohsan, N. Q. H. Othman, Y. Li, M. H. Alsharif, and M. A. Khan, "Unmanned Aerial Vehicles (UAVs): Practical Aspects, Applications, Open Challenges, Security Issues, and Future Trends," *Intelligent Service Robotics*, vol. 16, no. 1, pp. 109–137, Mar. 2023, https://doi.org/10.1007/s11370-022-00452-4.

[4] R. A. Firmansyah, S. Muharom, I. Masfufiah, A. H. Kisyarangga, and D. F. M. A. F. Rosyad, "Drone Localization using Global Navigation Satellite System and Separated Feature Visual Odometry Data Fusion," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 19466–19471, Feb. 2025, https://doi.org/10.48084/etasr.9130.

[5] M. A. Alohali *et al.*, "Modelling of a Multi-drone Framework for Trajectory Prediction and Deep Reinforcement Learning-based Obstacle Classification in Dynamic Environments," *Engineering Applications of Artificial Intelligence*, vol. 163, 2025 (online), Art. no. 112924, https://doi.org/10.1016/j.engappai.2025.112924.

[6] L. Mei *et al.*, "LSN-GTDA: Learning Symmetrical Network via Global Thermal Diffusion Analysis for Pedestrian Trajectory Prediction in Unmanned Aerial Vehicle Scenarios," *Remote Sensing*, vol. 17, no. 1, Jan. 2025, Art. no. 154, https://doi.org/10.3390/rs17010154.

[7] M. Chen, Z. Wang, Z. Wang, L. Zhao, P. Cheng, and H. Wang, "C2F-Net: Coarse-to-Fine Multidrone Collaborative Perception Network for Object Trajectory Prediction," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 18, pp. 6314–6328, 2025, https://doi.org/10.1109/JSTARS.2025.3541249.

[8] C. Petridis, A. Shrivastava, M. Vacic, and Z. Obradovic, "PixelPath: Predicting UAV Trajectories in GPS-Restricted Environments Using Image Feature Extraction and Machine Learning," in *Artificial Intelligence Applications and Innovations*, vol. 757, I. Maglogiannis, L. Iliadis, A. Andreou, and A. Papaleonidas, Eds. Cham, Switzerland: Springer Nature Switzerland, 2026, pp. 270–284, https://doi.org/10.1007/978-3-031-96231-8_20.

[9] C. Zhou, G. AlRegib, A. Parchami, and K. Singh, "TrajPRed: Trajectory Prediction With Region-Based Relation Learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 8, pp. 9787–9796, Aug. 2024, https://doi.org/10.1109/TITS.2024.3381843.

[10] D. Zhu, Q. Khan, and D. Cremers, "Multi-Vehicle Trajectory Prediction and Control at Intersections Using State and Intention Information," *Neurocomputing*, vol. 574, p. 127220, Mar. 2024, https://doi.org/10.1016/j.neucom.2023.127220.

[11] Z. Dang, B. Sun, C. Li, S. Yuan, X. Huang, and Z. Zuo, "CA-LSTM: An Improved LSTM Trajectory Prediction Method Based on Infrared UAV Target Detection," *Electronics*, vol. 12, no. 19, Sept. 2023, Art. no. 4081, https://doi.org/10.3390/electronics12194081.

[12] J. Liu, W. Luo, G. Zhang, and R. Li, "Unmanned Aerial Vehicle Path Planning in Complex Dynamic Environments Based on Deep Reinforcement Learning," *Machines*, vol. 13, no. 2, Feb. 2025, Art. no. 162, https://doi.org/10.3390/machines13020162.

[13] W. Skarka and R. Ashfaq, "Hybrid Machine Learning and Reinforcement Learning Framework for Adaptive UAV Obstacle Avoidance," *Aerospace*, vol. 11, no. 11, Oct. 2024, Art. no. 870, https://doi.org/10.3390/aerospace11110870.

[14] S. Huang *et al.*, "KiteRunner: Language-Driven Cooperative Local-Global Navigation Policy with UAV Mapping in Outdoor Environments," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hangzhou, China, Oct. 2025, pp. 18512–18519, https://doi.org/10.1109/IROS60139.2025.11247564.

[15] H. Sadid and C. Antoniou, "Dynamic Spatio-temporal Graph Neural Network for Surrounding-aware Trajectory Prediction of Autonomous Vehicles," *IEEE Transactions on Intelligent Vehicles*, pp. 1–14, 2024, https://doi.org/10.1109/TIV.2024.3406507.

[16] D. Borovyk, O. Barmak, P. Radiuk, and I. Krak, "Hierarchical Deep Learning Model for Identifying Similar Targets in UAV Imagery," *Drones*, vol. 9, no. 11, Oct. 2025, Art. no. 743, https://doi.org/10.3390/drones9110743.

[17] J. Han, M. Kamber, and J. Pei, "Data Preprocessing," in *Data Mining*, Amsterdam, Netherlands: Elsevier, 2012, pp. 83–124.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Representations by Back-propagating Errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, https://doi.org/10.1038/323533a0.

[19] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, July 2006, https://doi.org/10.1162/neco.2006.18.7.1527.

[20] Ziya, "UAV Autonomous Navigation Dataset." Kaggle, Mar. 2025, [Online]. Available: https://www.kaggle.com/datasets/ziya07/uav-autonomous-navigation-dataset.

[21] Ziya, "UAV Multi-Modal Target Tracking Dataset." June 2025, [Online]. Available: https://www.kaggle.com/datasets/ziya07/uav-multi-modal-target-tracking-dataset.

[22] Y. Xu, D. Wu, M. Zhou, and J. Yang, "Deep Learning-Based Point Cloud Classification of Obstacles for Intelligent Vehicles," *World Electric Vehicle Journal*, vol. 16, no. 2, Feb. 2025, Art. no. 80, https://doi.org/10.3390/wevj16020080.

[23] A. Padia *et al.*, "Object Detection and Classification Framework for Analysis of Video Data Acquired from Indian Roads," *Sensors*, vol. 24, no. 19, Sept. 2024, Art. no. 6319, https://doi.org/10.3390/s24196319.

[24] Y. Luo, A. Sun, and J. Hong, "Autonomous Driving Decision-Making Method Based on Spatial-Temporal Fusion Trajectory Prediction," *Applied Sciences*, vol. 14, no. 24, Dec. 2024, Art. no. 11913, https://doi.org/10.3390/app142411913.