Human Trajectory Prediction on UAV Images: A Comparative Study

Rafael D. M. da Hora [1], Daniel R. Santos [1], Maurício C. M. de Paulo [2], Felipe Ferrari [2], Raul Q. Feitosa [3], Paulo F. F. Rosa [1]

[1] Dept. of Defense Eng., Military Institute of Engeneering, Rio de Janeiro, Brazil - (hora.rafael, daniel.rodrigues, rpaulo)@ime.eb.br
[2] Dept. of Cartographic Eng., Military Institute of Engeneering, Rio de Janeiro, Brazil - (mauricio.paulo, ferrari)@ime.eb.br
[3] Dept. Electrical Engineering, Pontifical Catholic University, Rio de Janeiro, Brazil – raul@ele.puc-rio.br

Keywords: Human trajectory prediction, UAV, object detection, deep learning models.

Abstract

Video human trajectory prediction is a fundamental research task for many civil and defense applications. Human trajectory prediction in videos, especially in the context of unmanned aerial vehicles (UAVs) platforms, presents unique challenges due to the temporal dynamics and motion patterns inherent in non-stationary video data. As frames in a video streaming are highly correlated, trajectory detection in UAV images is affected by particular factors such as oblique camera views and the platform motion. This study aims to evaluate the performance of deep learning model's predictions in the context of UAVs videos by comparing three distinct categories: classical machine learning, established deep learning architectures, and computationally efficient models based on Multi-layer Perceptrons (MLPs). We propose an analysis based on only bounding box center coordinates instead of image scenes. The results show that a simple linear architecture provided the best performance, highlighting the importance of these mechanisms in predicting human motion from trajectory data alone.

1. Introduction

Real-time trajectory prediction is a crucial task with broad interest across various fields, ranging from analyzing interpersonal behavior (Helbing and Molnár, 1995) to enhancing safety in autonomous systems (Huang et al., 2022). Real-time trajectory prediction involves simultaneously localizing the bounding boxes of objects and tracking their movement within an image.

A central challenge in real-time applications is tracking loss which could be caused by partial occlusion and motion blur. For instance, the tracking could be lost when a target is hidden for a long-term. Thanks to trajectory prediction approaches, such problems are mitigated by estimating the tracking of objects occluded, which enables reliable re-identification. However, as it is much more expensive to collect labels due to the vast volume of frames required to capture natural motion and ensure identity consistency over long sequences.

Compared with classical approaches, deep learning models can improve the performance of prediction methods, and bring additional challenges into focus. They can be prone to overfitting which could be caused by problems with simpler underlying dynamics, such as human motion. Although previous studies in human tracking have established that human trajectories can often be modeled as continuous and smooth lines (Wang et al., 2008, Park and Shi, 2015, Helbing and Molnár, 1995), they can be subject to sudden and unpredictable changes in direction due to interactions with the surrounding environment. For example, for event monitoring, surveillance, and pursuing individuals it cannot be consistently predictable. The solution demands a high number of parameters to be chosen, significant computational resources, and enormous datasets for pre-training. Thus, more attention must be given for to the development of light-weight and efficient prediction models to real-time tracking in video sequences.

In this paper, we performed a comparative benchmark of diverse deep learning paradigms to identify the most robust lightweight model for human trajectory prediction using only coordinate data on resource-constrained platform. Suppose that a human presence in disaster scenarios has been identified, and its 3D coordinate is demanded for rapid emergency task. Is it still possible to boost the performance for a deep learning human trajectory prediction method? Our solution is to introduces a new benchmark including images acquired from drones in motion and discusses key challenges faced in people tracking and future perspectives in this field.

The paper is organized as follows. In Section 2 a related work on human tracking prediction is provided. Section 3 reports the evaluated deep learning models for the before mentioned task. The experimental analysis is summarized in Section 4, while the conclusions are reported in Section 5.

2. Related Works

In this section, deep learning approaches used for the human tracking prediction related to our work will be introduced.

Trajectory prediction of objects in real-time have received much attention since the introduction of the deep learning. Previous studies in human tracking established that human trajectories can often be modeled as continuous and smooth lines (Wang et al., 2008, Park and Shi, 2015, Helbing and Molnár, 1995). Within this context, various works have been focused on improving prediction accuracy by incorporating rich contextual information. Most deep learning models used for human tracking tasks, analyze not only the path but also the agent's interaction with the environment through direct image analysis or complex map representations (Alahi et al., 2016, Coscia et al., 2016, Gupta et al., 2018, Hassan et al., 2021, Fadillah et al., 2025). Although these methods can achieve high accuracy, they suffer with the high computational cost, rendering them less practical for resource-constrained unmanned airborne vehicles (UAVs) platforms.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLIX-B2-2026-755-2026

An alternative approach is to simplify the problem by focusing exclusively on the time-series data from the target's bounding box center coordinates. It significantly reduces the computational requirement, making trajectory prediction a more feasible task for embedded systems. The proposed approach reduce a person's trajectory to a time series problem, which still retains complex and non-linear dynamics. Such characterization requires the use of time series prediction models. Thus, machine learning models such as XGBoost (Chen and Guestrin, 2016) and well-established deep learning architectures such as RNN-LSTM (Hochreiter and Schmidhuber, 1997) and Transformers are frequently used as benchmarks. Furthermore, a recent trend in the literature has explored models with lower complexity and computational cost, such as the MLP-based TiDE(Das et al., 2024), TSMixer(Chen et al., 2023), DLinear(Zeng et al., 2022), and N-HiTS(Challu et al., 2022). These simpler models have been benchmarked against more robust architectures and have often demonstrated surprisingly competitive or even superior performance.

This suggests that, for specific applications, the increase in complexity offered by larger models may not always translate into a proportional gain in terms of accuracy, which reinforce the value of exploring more efficient solutions.

3. PROPOSED METHOD

In this section, we will introduce an overview of our proposed structure Figure 1. The goal of this paper is explores a lightweight approach by conducting a comparative analysis of different prediction models that rely only on object bounding box center coordinates and identify an optimal balance between predictive accuracy and computational efficiency for this context.

Initially, a custom dataset was collected by recording human motion in open environments, ensuring natural and unbiased movements. For each video, the YOLOv12-m, combined with a Kalman Filter-based tracker, was employed to detect and maintain individual identities for coordinate extraction, extracting the center coordinates of their bounding boxes to represent trajectory data. This dataset was subsequently used to train and evaluate all models.

The evaluation of all the state-of-the-art models was conducted using the Darts library (Herzen et al., 2022) following a standardized experimental protocol. The prediction task was framed within a sliding-window paradigm, in which each model was trained to map an input window—comprising a sequence of historical observations—onto a corresponding prediction horizon. The investigated models are presented as follows.

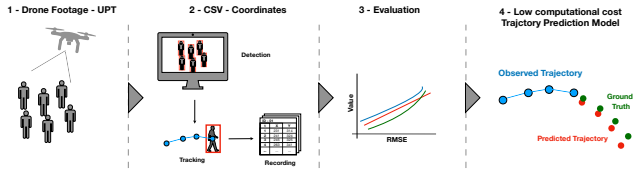

Figure 1. Graphic abstract: 1 - UPT dataset creation; 2 - Trajectory CSV Creation; 3 - Model Evaluation; 4 - Model Election

- **TIDE (Time-series Dense Encoder):** The TIDE model (Das et al., 2024) employs an encoder-decoder architecture based on Densely-Connected Neural Networks (MLPs) to process past observations and generate future predictions. Its selection is justified as it represents a recent and efficient deep learning approach that challenges the necessity of more complex architectures, such as Transformers, making it a strong candidate for tasks requiring low computational overhead.

- **TSMixer (Time-Series Mixer):** TSMixer (Chen et al., 2023) is a deep learning model based exclusively on MLPs, featuring alternating layers that mix information across the time-axis (time-mixing) and the feature-axis (feature-mixing). Its inclusion in this analysis allows for evaluating the effectiveness of an architecture that explicitly separates the processing of temporal dependencies from cross-variable correlations, offering a structurally distinct paradigm compared to other MLP-based models.

- **DLinear:** This model (Zeng et al., 2022) is based on decomposing the input time series into trend and seasonality components, applying a simple linear layer to each one, and including their outputs. DLinear is included to test the hypothesis that a purely linear model, when applied to well-defined components of the series, can be surprisingly effective. It serves as a benchmark for the minimum complexity required among the evaluated deep learning approaches.

- **N-HiTS (Neural Hierarchical Interpolation for Time Series):** N-HiTS (Challu et al., 2022) utilizes a hierarchical architecture of MLP blocks to analyze the time series at multiple resolution scales, combining the resulting information via interpolation. It was selected to investigate whether a multi-resolution analysis can better capture trajectory dynamics at different frequencies (e.g., short-term movements versus long-term directional changes), potentially offering an advantage over models that process the series at a single scale.

- **Recurrent Neural Network (LSTM-Based)**: The Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) cells (Salinas et al., 2019, Hochreiter and Schmidhuber, 1997) is a canonical architecture for sequential data modeling. It process the trajectory step-by-step while maintaining an internal memory to capture temporal dependencies. The LSTM is included as a fundamental deep learning baseline for time series, representing the standard recurrent approach against which the performance of newer MLP-based and attention-based models is measured.

- **TimesFM (Foundation Model)**: This architecture is a time-series transformer developed by Google Research as a foundation model for temporal forecasting. TimesFM (Das et al., 2023) utilizes a decoder-only Transformer design, pre-trained on a massive corpus comprising over 100 billion real-world time points. Unlike traditional models trained exclusively on task-specific data, this foundation model leverages universal patterns learned across diverse domains to provide robust zero-shot forecasting capabilities. Its application in this study allows for an evaluation of how a large-scale pre-trained model adapts to the specific spatial-temporal dynamics of pedestrian trajectories.

- **XGBoost (XGB) (Chen and Guestrin, 2016):** is a powerful gradient boosting algorithm that approaches prediction by transforming the time series into a tabular regression task using a sliding window and a multi-target regressor for the

10-point horizon, using past lagged values as input features. It was chosen to represent classical, non-neural-network-based machine learning methods. Its presence allows for a direct comparison between deep learning architectures and a robust, gradient boosting decision tree.

- **NAIVE (Naive Forecast):** This baseline model operates by fitting a line between the first and last points of the input series and extrapolating this line into the future. The Naive model is essential for the comparative analysis as it establishes a lower-bound performance threshold; any other model is only considered useful if it significantly outperforms this trivial forecast.

4. Experiments

The experimental framework was designed to evaluate and compare the performance of the selected models on the task of human trajectory prediction from UAV video streamings. First we created a novel dataset that contains human trajectories: time series of the positions of each individual in video streamings recorded by UAVs. The positions were obtained using Yolo v12 (Tian et al., 2025) and their bounding box image coordinates center trajectories were saved in csv files. Then, we trained the models to predict, at any given time, 10 future positions using the 10 previous positions as inputs.

The general guideline of the framework is composed of a novel dataset where people are detected by Yolov12 (Tian et al., 2025) .

4.1 Dataset Description

The experiments conducted in this paper required a dataset featuring specific features not present in state-of-the-art benchmarks for human trajectory, such as PIE (Rasouli et al., 2019) and ETH (Sun et al., 2021). These benchmarks often lack the complexities introduced by a non-stationary camera and oblique viewing angles, which are inherent to UAV-based data video streaming capture. To address this limitation, we introduced our own dataset called ”the UAV Person Tracking (UPT)” (Figure 2).

The UPT dataset consists of over an hour of video streaming, captured at 24 frames per second (FPS), depicting pedestrians in diverse, real-world scenarios. A key challenge is the variability in video resolution across different recordings. The process of trajectory extraction was automated using the YOLOv12-m object detection model. To ensure temporal consistency and maintain unique identities across sequences, the extraction process leveraged the native tracking capabilities provided by the YOLO framework, which incorporates a Kalman Filter-based tracker (Tian et al., 2025). For each identified pedestrian in every frame, the (x, y) coordinates of the bounding box center were extracted. This information was compiled into CSV files, where each file corresponds to a single pedestrian’s trajectory. To facilitate data normalization and account for varying resolutions, the frame’s height and width were recorded alongside each coordinate pair. Note that the dataset contain different cameras, so the image dimensions were registered to make normalization possible.

Since the original trajectories were recorded at 24 FPS, a frequency where displacement is often smaller than centroid noise; downsampling to 6, 3, and 2 FPS increases the temporal stride to better capture motion trends, the dataset was downsampled

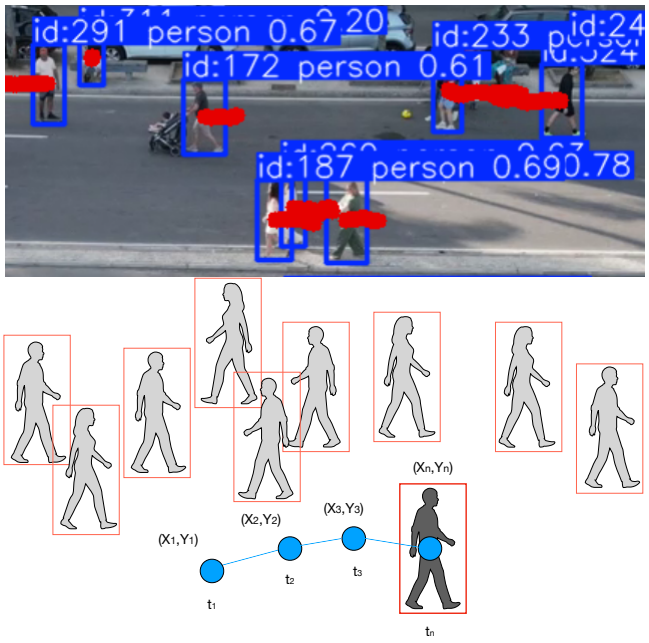

Figure 2. The UPT: more than an hour of movies from people walking on dynamic environment.

into three subsets. These subsets correspond to sampling rates of 6, 3, and 2 FPS. It allows the analysis of model performance across different temporal scales and apparent motion speeds.

4.2 Experimental Protocol

To ensure a robust model comparison, we developed an experimental protocol which consists of: a dataset normalization, a model setting, a training, and a performance analyses.

4.2.1 Dataset Normalization The initial dataset was a series of csv files where each file represents the list of image coordinates (x, y) from the bounding box center detected by YOLOv12-m detection. We applied a linear min-max scale obtaining mean 0 and range [-1,1] in both coordinates.

$$scale_X, scale_Y = \left[\frac{image_{high}}{2}, \frac{image_{width}}{2}\right] \quad (1)$$

For each pair (x,y) on the csv list, the the x_{scaled} and y_{scaled} values were computed as follows:

$$x_{scaled}, y_{scaled} = \left[\frac{x_{value}}{scale_X} - 1, \frac{y_{value}}{scale_Y} - 1\right] \quad (2)$$

4.2.2 Model Setting The model followed (Herzen et al., 2022) implementation and the hyperparameter set up to reach the best values for each proposed model.

- **TIDE:** For this model the encoder and decoder were composed by a single layer (encoder layers = 1, decoder layers = 1), with the decoder providing an output dimension of 16 (decoder_output_dim = 16). The hidden layers of the network have a size of 128 neurons (hidden_size = 128), defining the model’s representation capacity. For temporal processing, windows of 10 units are used for both past (observed) and future (to be predicted) data (temporal_width_past = 10, temporal_width_future = 10), and

the temporal decoder’s hidden layer has 32 neurons (temporal_decoder_hidden = 32). Layer normalization is not employed (use_layer_norm = False), while a dropout rate of 0.1 is applied for regularization (dropout = 0.1). Finally, the model incorporates static covariances (use_static_covariates = True) to enrich the prediction context.

- **TSMixer:** For this model, was used hidden_size=64, ff_size = 64, num_blocks=2 and activation=’Sigmoid to achieve the better results.
- **DLinear:** For this experiments, the best results were achieved from the default configuration using a kernel_size: int = 25 and static covariances.
- **N-HiTS:** This model uses MLP layers to model the time series jointly with ReLU activation functions, to predict a few points that are then interpolated to generate the full prediction for the input time series. The best results achieved for this model was obtained using 2 blocks of MLP layers.
- **LSTM:** For this RNN model, was used 50 hidden dimension and 10 layers (hidden_dim=50, n_rnn_layers=10). Also a sigmoid activation function was applied.
- **TimesFM:** For the transformer based model was applied the setup, as follows: 2 encoder/decoder layers with ReLU activation. This lightweight configuration was chosen to maintain computational parity with MLP-based models, adhering to the resource constraints of UAV platforms. During training task, the model processes trajectories in batches of 32.
- **XGB:** A simple XGB with no past or future covariances for this model.
- **NAIVE:** This model fits a line between the first and last point of the training series, and extends it in the future. For a training series of length T, can be written the following equation:

$$\hat{y}_{T+h} = y_T + h\left(\frac{y_T - y_1}{T-1}\right) \tag{3}$$

4.2.3 Training For the trajectory prediction stage, a sequence-to-sequence architecture was implemented. The model was configured to use an input sequence of 10 points from the past trajectory to predict an output sequence of 10 future points. This 10-point input strikes a balance between providing sufficient historical context to infer movement dynamics—such as velocity and direction—and maintaining manageable computational complexity. Although the original video was captured at 24 FPS, utilizing consecutive frames at this frequency introduces significant data redundancy. To effectively capture meaningful human motion, it is necessary to observe a broader temporal window by downsampling the data, thereby highlighting distinct movement patterns. Consequently, we evaluated three downsampled temporal resolutions—6, 3, and 2 FPS—to analyze different prediction horizons and assess model behavior during various speed transitions.

4.2.4 Metrics To evaluate and compare the before mentioned methods, the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) were applied to the prediction data, comparing them with the real sequence points. For each sample from the independent, chronological test set (the final 20% of the data), each model predicted 10 future points, and the RMSE and MAE were calculated for every complete 10-point predicted sequence. Although these metrics are usually assumed for accuracy evaluation, it is crucial to verify the prediction horizon, that is, how far the furthest prediction points are from the real points. To analyze this aspect, a mean error, in pixels, was calculated for each of the 10 predicted points individually within the sequence.

4.3 Results

We compared the selected video human trajectory prediction algorithms in different temporal resolutions.

4.3.1 6 frames per seconds These results evolve the paths with highest proximity between points in our analysis and consequently, a shortest horizon of prediction.

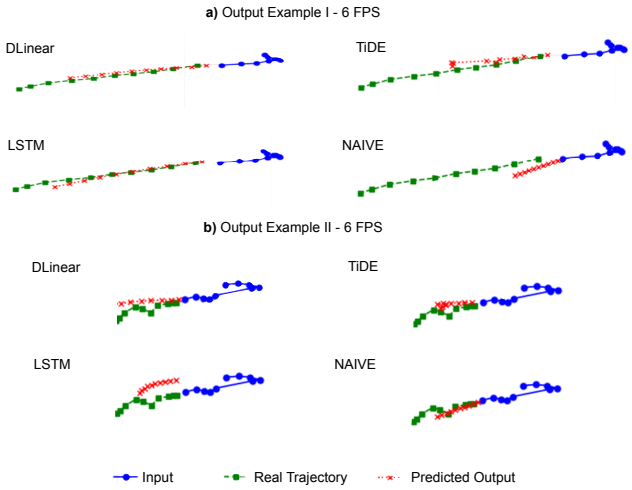

Figure 3. Output example on 6 FPS: a) High speed trajectory b)Low Speed trajectory

Fig. 8a presents the obtained results for the four best tested models at 6 FPS and Fig. 3 shows two examples for each one of the four best performed models. The example is composed by the observed trajectory (input) according 6 FPS, the ground truth (real trajectory) and predicted trajectory (output). Fig. 3 - a) represents a trajectory where a person was moving with higher speed than 3 - b).

4.3.2 3 frames per seconds The Fig. 8 - b) present the results for the four best performances.

Fig. 4 presents two examples for each one of the four best tested models. It includes both the predicted and the real target trajectory and analyzed an input.

4.3.3 2 frames per seconds Fig. 8 - c) presents the results for the four best performances. Fig. 6 - c) shows the results for all models considering 2 FPS.

Fig. 5 shows two examples for each one of the 4 best performance models. It includes both predicted and real target trajectory and reference input.

The overall results from RMSE and MAE for evaluated models at each considered frame rate are graphically presented at Fig. 6 and 7.

4.3.4 Computational Complexity Comparation To further assess the suitability of the models for real-time deployment on resource-constrained UAV platforms, we conducted a comparative analysis of their computational complexity. Table 1 summarizes the trainable parameters, estimated FLOPs, and average inference latency for each architecture.

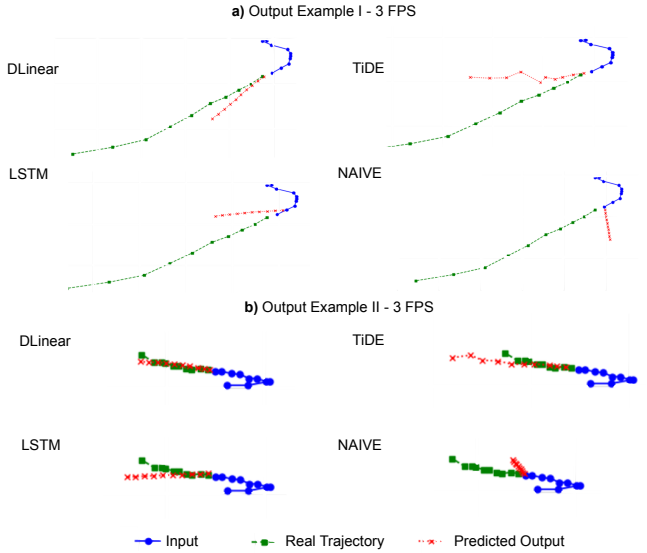

Figure 4. Output example on 3 FPS: a) High speed trajectory b)Low Speed trajectory

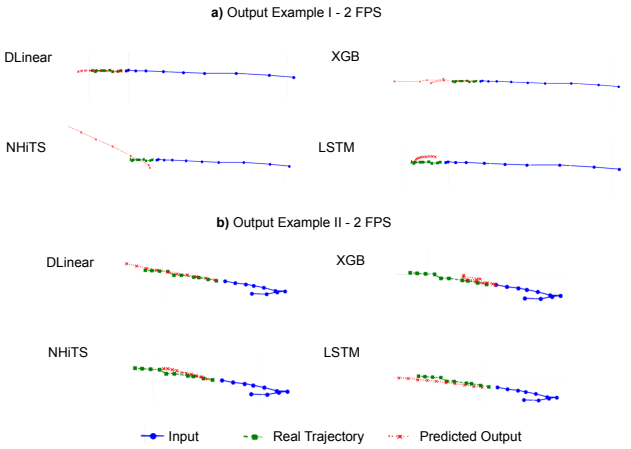

Figure 5. Output example on 2 FPS: a) High speed trajectory b)Low Speed trajectory

4.4 Discussion

The metrics presented show error values represented in pixels with a resolution of 640x480. All models achieved comparable RMSE and MAE values, indicating similar predictive accuracy; however, DLinear distinguished itself by attaining the lowest error with notably lower computational complexity

DLinear achieved the best results on all experiments, establishing it as a strong all-around performer. Analysis of the results indicates that the model performance varies considerably with the prediction horizon, as no single model for all scenarios. For short-term prediction horizons, the performance of the models was comparable as shown in Fig. 3. However, an increase in error was observed across all approaches, as expected. Thus, can be noted that the DLinear model showed highlighted stability, maintaining a more contained error growth and a more robust performance compared with other models across the entire prediction horizon.

As expected, the errors found for the first frames were the lowest in all models. The point-by-point error analysis on previously tables reinforces the best performance of the DLinear model. It

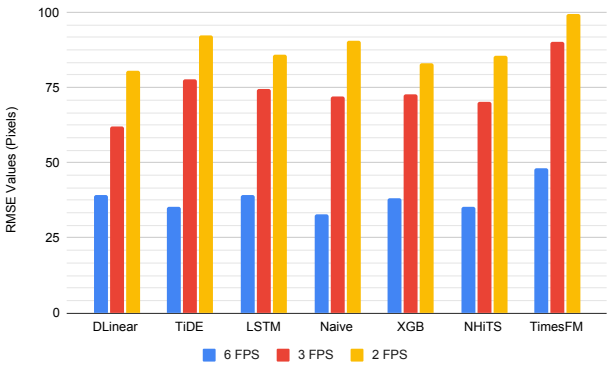

Figure 6. Overall RMSE values in Pixels at each frame rate

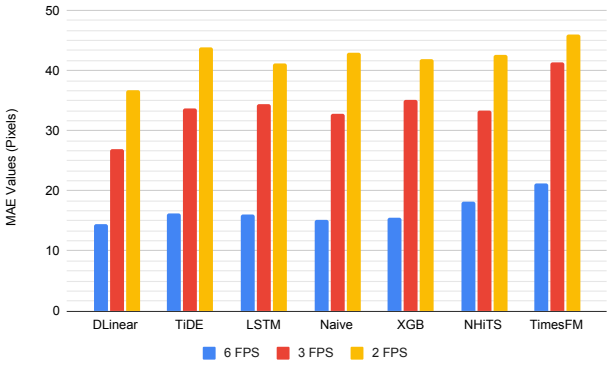

Figure 7. Overall MAE values in Pixels at each frame rate

not only starts with the lowest error at ”Point 1”, but it also maintains its advantage across the entire 10-point horizon. Its error growth is consistent and more controlled compared to its competitors as we can see on Fig. 3, Fig. 4 and Fig. 5. Specifically, DLinear operates with significantly fewer FLOPs (Floating Point Operations) compared to the Transformer or LSTM, making it the most viable candidate for real-time UAV deployment.

The trajectories shown in Fig. 3 - **b**), Fig. 4 - **b**), and Fig. 5 - **b)** are characterized non-smooth movements, with inflection points and non-smooth segments. Conversely, Fig. 5 - **a)** presents a more linear path with velocity changes. Since the models operate without image context, their ability to coherently predict the trajectory’s direction in line with the ground truth becomes a critical evaluation criterion. Thus, DLinear outperformed attention-based models by effectively isolating movement trends, whereas the Transformer was more susceptible to noise in detection centroids.

The nature of human trajectories helps explain such results. While paths are generally smooth—a characteristic well-suited to linear models—they can contain sudden changes or inflection points. Theoretically, a model with memory, as LSTM, should be better equipped to handle such non-linear, abrupt changes. However, the empirical results suggest that this theoretical advantage did not translate into superior overall performance in this context. This could be explained due to unpredictable changes are not frequent enough in the dataset to favor the complexity of the LSTM, or the 10-point loop closing window is not sufficient for it to learn effectively these complex dynamics effects.

Despite YOLOv12 low computational cost human detection, partial occlusions can cause significant instability in the resulting

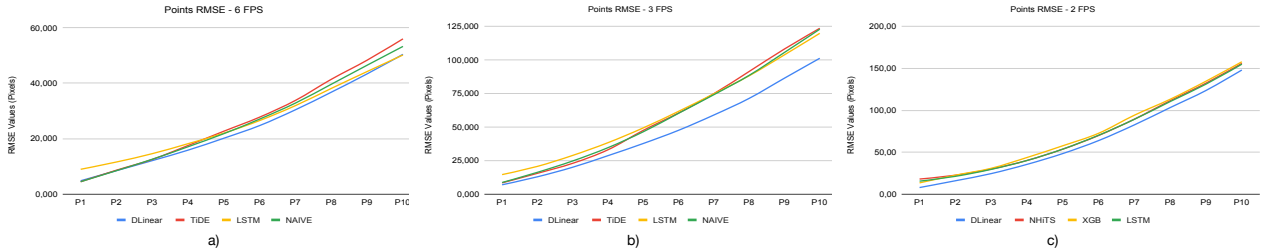

Figure 8. RMSE per point on: a) 6 FPS; b) 3 FPS; c) 2 FPS.

Table 1. Computational Complexity and Inference Latency Comparison

Model	Parameters	FLOPs	Inference (ms)
DLinear	840	740	92.26
TiDE	80.43 K	59.1 K	89.62
TSMixer	28.04 K	168.7 K	93.88
TimesFM	231.29 M	813.72 M	23.02†
NHiTS	1.67 M	1123.2 K	104.49
LSTM	195.42 K	3820 K	136.96†
XGBoost	213.8 K*	12 K*	91.14†

*Nodes/Comparisons for XGBoost. †Estimated values based on linear regression of experimental latency.

bounding boxes. This affects their centroids and corrupts the trajectory data, potentially leading to an incorrect representation of the problem.

5. Conclusion

In this work, we evaluated computationally efficient models for predicting human trajectories, using only coordinate data from our dataset UPT, which consists of UAV video streamings of people moving in real-world scenarios.

The DLinear model achieved the best overall performance on our dataset, suggesting that trend decomposition is more resilient than attention mechanisms for trajectory forecasting when training on resource-constrained, noisy UAV datasets. However, two primary challenges were identified: prediction accuracy significantly decreases as the predict horizon increases, and more complex trajectories with curves and inflections are harder to predict using the DLinear model.

We also noted that an additional limitation lies in the object detection stage. The Yolov12 network demonstrated instability in bounding boxes during partial occlusions. This noise in the input data affect prediction accuracy. Therefore, future work should consider using a more occlusion-resilient object detector to improve the overall system quality.

References

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S., 2016. Social LSTM: Human trajectory prediction in crowded spaces. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Las Vegas, NV, USA, 961–971.

Challu, C., Olivares, K. G., Oreshkin, B. N., Garza, F., Mergenthaler-Canseco, M., Dubrawski, A., 2022. N-HiTS: Neural hierarchical interpolation for time series forecasting.

Chen, S.-A., Li, C.-L., Yoder, N., Arik, S. O., Pfister, T., 2023. TSMixer: An all-MLP architecture for time series forecasting.

Chen, T., Guestrin, C., 2016. XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

Coscia, P., Castaldo, F., Palmieri, F. A., Ballan, L., Alahi, A., Savarese, S., 2016. Point-based path prediction from polar histograms. *2016 19th International Conference on Information Fusion (FUSION)*, 1961–1967.

Das, A., Kong, W., Leach, A., Mathur, S., Sen, R., Yu, R., 2024. Long-term forecasting with TiDE: Time-series dense encoder.

Das, A., Kong, W., Leach, A., Sen, R., Zhou, Y., 2023. A Decoder-Only Foundation Model for Time-Series Forecasting. *arXiv preprint arXiv:2310.10688*. https://arxiv.org/abs/2310.10688.

Fadillah, A., Lee, C.-L., Wang, Z.-X., Lai, K.-T., 2025. GoalNet: Goal Areas Oriented Pedestrian Trajectory Prediction. *IEEE Access*, 13, 132537–132546. http://dx.doi.org/10.1109/ACCESS.2025.3588812.

Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A., 2018. Social gan: Socially acceptable trajectories with generative adversarial networks.

Hassan, M. A., Khan, M. U. G., Iqbal, R., Riaz, R., Bashir, A.-A.-W., Tariq, U., 2021. Predicting humans future motion trajectories in video streams using generative adversarial network. *Multimedia Tools and Applications*, 80(25), 31599–31613.

Helbing, D., Molnár, P., 1995. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51, 4282–4286. https://link.aps.org/doi/10.1103/PhysRevE.51.4282.

Helbing, D., Molnár, P., 1995. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51(5), 4282–4286.

Herzen, J., LÃ¤ssig, F., Piazzetta, S. G., Neuer, T., Tafti, L., Raille, G., Pottelbergh, T. V., Pasieka, M., Skrodzki, A., Huguenin, N., Dumonal, M., KoÅ›cisz, J., Bader, D., Gusset, F., Benheddi, M., Williamson, C., Kosinski, M., Petrik, M., Grosch, G., 2022. Darts: User-Friendly Modern Machine Learning for Time Series. *Journal of Machine Learning Research*, 23(124), 1-6. http://jmlr.org/papers/v23/21-1177.html.

Hochreiter, S., Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Computation*, 9, 1735-1780.

Huang, Y., Du, J., Yang, Z., Zhou, Z., Zhang, L., Chen, H., 2022. A Survey on Trajectory-Prediction Methods for Autonomous Driving. *IEEE Transactions on Intelligent Vehicles*, 7(3), 652–674.

Park, H. S., Shi, J., 2015. Social saliency prediction. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4777–4785.

Rasouli, A., Kotseruba, I., Kunic, T., Tsotsos, J. K., 2019. Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. *International Conference on Computer Vision (ICCV)*.

Salinas, D., Flunkert, V., Gasthaus, J., 2019. DeepAR: Probabilistic forecasting with autoregressive recurrent networks.

Sun, M., Baldini, F., Trautman, P., Murphey, T., 2021. Move Beyond Trajectories: Distribution Space Coupling for Crowd Navigation. *Proceedings of Robotics: Science and Systems*, Virtual.

Tian, Y., Ye, Q., Doermann, D., 2025. YOLOv12: Attention-Centric Real-Time Object Detectors. *arXiv preprint arXiv:2502.12524*.

Wang, J. M., Fleet, D. J., Hertzmann, A., 2008. Gaussian Process Dynamical Models for Human Motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2), 283-298.

Zeng, A., Chen, M., Zhang, L., Xu, Q., 2022. Are transformers effective for time series forecasting?