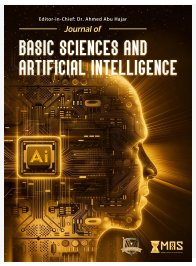
Journal of Basic Sciences and Artificial Intelligence (JBSAI)

DOI: 10.65098/jbsai.01.2026.46.52



RESEARCH ARTICLE

A Proposed Method for Predicting Civil Aircraft Trajectories Using the ADS-B System Based on Deep Tabular Learning

Noor Sharabati*

Department of Communications Engineering, Faculty of Electrical and Electronic Engineering, University of Aleppo, Aleppo 12212, Syria
Corresponding Author Email: noorsharabati1996@gmail.com

ARTICLE DETAILS

Article History:

Received: 27 May 2026
Accepted: 08 Sep 2026
Available online: 10 Sep 2026

Online Article Code

ABSTRACT

The continuous advancement of Air Traffic Control has resulted in the increasing importance of efficient and reliable airspace management for the safety and efficiency of modern air transportation. The rapid growth of global air traffic has created a great demand for intelligent systems to accurately monitor and control aircraft operations. Recent advances in artificial intelligence have opened up the possibility of data-driven approaches that can significantly improve the performance of air navigation and trajectory prediction systems. In this study, two deep tabular learning models, TabNet and TabNSA, were employed for aircraft trajectory prediction with Automatic Dependent Surveillance-Broadcast (ADS-B) data. The experimental results showed that the proposed models performed better than a number of popular deep learning approaches such as CNN-LSTM and CNN-BiLSTM. This performance gain was particularly clear in altitude prediction, which remains one of the most difficult tasks in aircraft trajectory forecasting. In particular, TabNSA outperformed TabNet with around 7% lower Root Mean Square Error (RMSE) and CNN-BiLSTM with 39% lower RMSE. TabNSA further improved the prediction accuracy in terms of both RMSE and Mean Absolute Error by around 50% over TabNet for latitude and longitude prediction. Additionally, qualitative comparisons of 2D and 3D flight trajectories showed that the predicted trajectories by TabNSA were highly similar to the actual flight trajectories, consistently better than the benchmark models and also exhibiting a strong correlation with the TabNet predictions.

KEYWORDS

Air Traffic Control; Automatic Dependent Surveillance-Broadcast; Aircraft Trajectory Prediction; Deep Tabular Learning; TabNet; TabNSA; CNN-LSTM; CNN-BiLSTM

1. INTRODUCTION

The Automatic Dependent Surveillance-Broadcast (ADS-B) system is a four-dimensional aircraft trajectory surveillance technology where a wireless communication link is used to periodically broadcast an aircraft position, identity, velocity, and other operational information at a transmission rate of approximately 6.2 messages per second. For aircraft and ground stations in the area of the aircraft, these broadcasts provide continuous monitoring of the aircraft's status. ADS-B technology was designed to enhance the precision of aircraft position reporting via Global Navigation Satellite Systems and is anticipated to supplant Secondary Surveillance Radar in future air traffic surveillance systems. However, ADS-B security is still vulnerable due to the absence of some security mechanisms such as authentication and encryption (Kim et al., 2016).

This study investigates the effectiveness of deep tabular learning models for aircraft trajectory prediction using real-world ADS-B data, in view of the rapid development of intelligent aviation technologies. A thorough comparative evaluation with standardized prediction accuracy metrics is used to assess the performance of the model. These findings underscore the need for advanced tabular learning architectures to improve the reliability and robustness of trajectory prediction systems in complex aviation environments (Olive et al., 2025).

2. RESEARCH PROBLEM AND OBJECTIVES

Conventional deep learning models often exhibit limited capability in handling heterogeneous tabular data, restricting their ability to capture complex feature interactions and consequently reducing both prediction accuracy and model interpretability. To address these limitations, this study aims to develop an efficient aircraft trajectory prediction framework based on ADS-B data using deep tabular learning models.

Specifically, the research focuses on implementing and comparing the TabNet and TabNSA architectures to improve trajectory prediction accuracy while minimizing prediction errors, as measured by the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) metrics. Furthermore, the proposed models are evaluated against conventional deep learning approaches, including CNN-LSTM and CNN-BiLSTM. The ultimate objective is to achieve an optimal balance between predictive accuracy and model interpretability by leveraging attention mechanisms to effectively capture complex feature relationships within aviation data.

3. DEEP TABULAR LEARNING

Although deep learning models have achieved remarkable success in processing unstructured data such as images, time-series data, and natural language, where clear spatial or temporal relationships exist, their performance remains limited when applied to tabular data. Unlike unstructured data, tabular datasets are typically heterogeneous, comprising both numerical and categorical features, and lack regular structural relationships among columns. Consequently, conventional deep learning models often introduce a large number of parameters while treating all features with equal importance, leading to redundant representations, suboptimal feature utilization, and limited model interpretability.

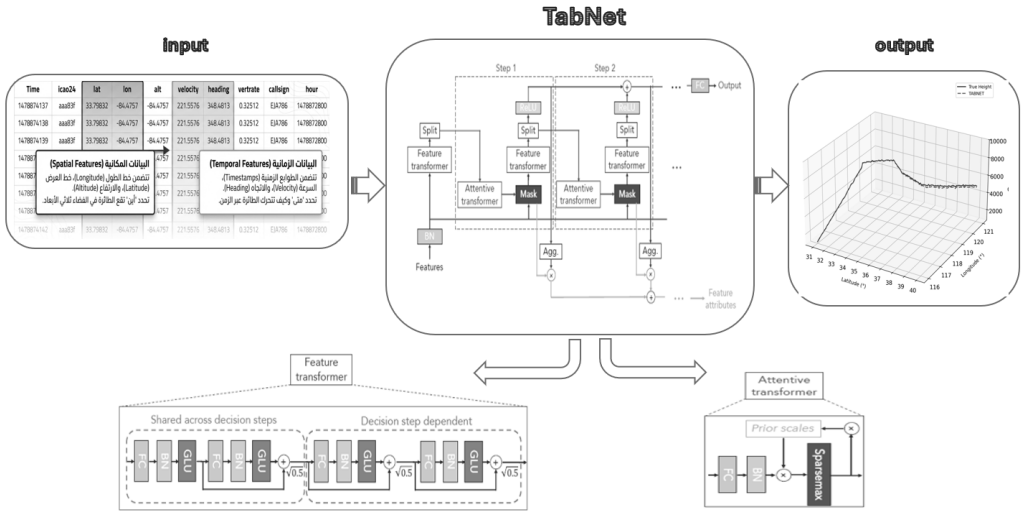


Figure 1 Overall Architecture of the Proposed Aircraft Trajectory Prediction Framework Based on the TabNet Algorithm

These limitations have motivated the development of specialized deep tabular learning models, as tabular data constitute the predominant data format in many real-world artificial intelligence applications, including finance, healthcare, and aviation. Compared with conventional deep learning architectures, deep tabular learning models are specifically designed to capture heterogeneous and long-range interactions among diverse feature types. They emulate the effectiveness of decision tree-based methods through attention-guided feature selection and sparse feature interactions, enabling automatic feature engineering and facilitating the integration of multimodal data. In contrast, conventional deep learning primarily relies on stacked neural layers to learn hierarchical feature representations. Furthermore, deep tabular learning supports self-supervised learning by predicting masked or hidden features, thereby improving predictive performance when labeled data are limited. These characteristics provide enhanced interpretability, greater computational efficiency through attention mechanisms that reduce computational complexity, and increased robustness to missing data and class imbalance (Kim et al., 2016; Olive et al., 2025).

4. RESEARCH METHODOLOGY

The present study proposes an aircraft trajectory prediction system for civil aviation in airport environments based on ADS-B data as a main input source. The proposed framework uses two state-of-the-art deep tabular learning models, TabNet and TabNSA, for the prediction of aircraft trajectories. The performance of the models is then evaluated and compared to previously published benchmark models. Figure 1 shows the overall architecture of the proposed TabNet-based trajectory prediction framework. Figure 2 shows the overall architecture of the proposed TabNSA-based framework.

5. TABNET ALGORITHM

The TabNet algorithm is based on an Encoder-Decoder architecture, where the Encoder is the main component responsible for supervised learning. It is based on the (N)-step sequential attention mechanism, where multiple decision steps are used to select features instance-wise. The model chooses the most informative features at each decision step and applies a non-linear transformation to generate the final prediction. This architecture mimics the decision-making process of Gradient Boosted Decision Trees, while preserving the benefits of gradient-based optimization. Therefore, TabNet outperforms conventional deep learning algorithms in terms of prediction performance as well as interpretability (Arik & Pfister, 2021).

Initially, the model receives tabular input data, where categorical features are transformed into learnable embedding representations, while numerical features are used directly without modification. The input is represented by the raw feature matrix $f \epsilon R^{B \times D}$ where (B) denotes the batch size and (D) represents the number of input features. The input features are subsequently passed through a Batch Normalization (BN) layer to normalize their distributions during training. Therefore, no additional feature normalization or preprocessing is required prior to this stage.

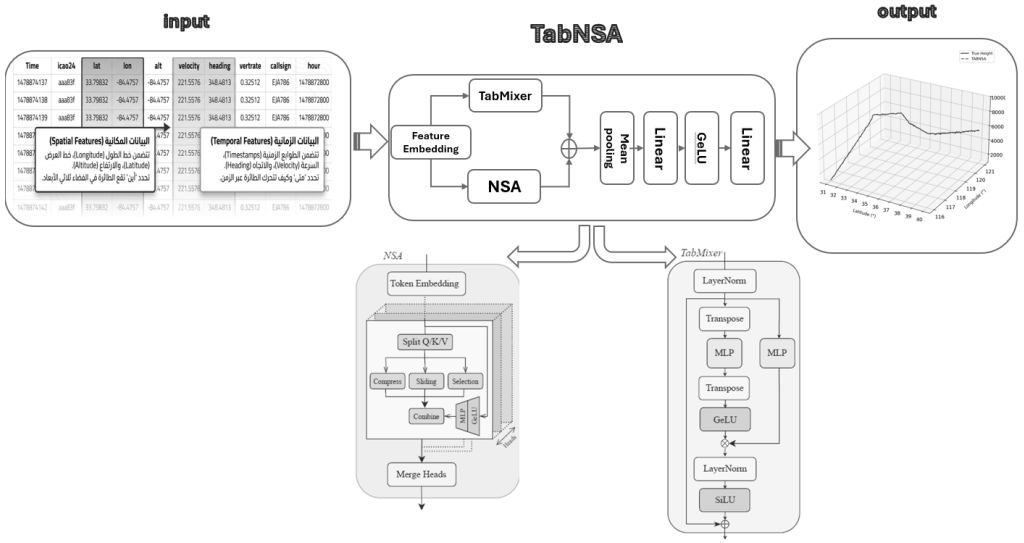


Figure 2 Overall Architecture of the Proposed Aircraft Trajectory Prediction Framework Based on the TabNSA Algorithm

6. SEQUENTIAL DECISION STEPS

In the second stage, the normalized input data are processed through a sequence of (N) consecutive decision blocks, where each decision step (i) dynamically selects a subset of informative features. This stage consists of the following components.

6.1 Attentive Transformer

The Attentive Transformer is responsible for feature selection by identifying the most relevant features for the current decision step based on the processed output from the previous decision step, $a[i-1]$together with the prior scale $P[i-1]$. The prior scale is updated at each decision step to prevent excessive reuse of the same features throughout the sequential decision process. It is computed as follows Arik & Pfister, (2021):

$$P[i] = \prod_{j=1}^{i} (\gamma - M[j])$$

Subsequently, a learnable feature mask, $M[i] \in R^{B\times D}$is generated using the Sparsemax activation function to enforce sparse feature selection while satisfying the following constraint $\sum_{j=1}^{D} M[i]_{b,j} = 1$ Here, $h_i$denotes a learnable transformation composed of a BN layer followed by a Fully Connected (FC) layer. The feature mask is computed as:

$$M[i] = sparsemax(P[i-1] \cdot h_i(a[i-1]))$$

6.2 Feature Masking

The generated feature mask is applied to the input features through element-wise multiplication, yielding the selected feature representation, $M[j].f$, thereby ensuring that the model focuses exclusively on the most informative features for each individual sample while suppressing less relevant information (Arik & Pfister, 2021).

6.3 Feature Transformer

The selected features are passed through a sequence of layers consisting of an FC layer, BN, and a Gated Linear Unit (GLU). The Feature Transformer contains both shared layers, which learn general patterns across all decision steps, and decision-specific layers, which capture the distinctive characteristics of each decision stage. The GLU layers act as gating mechanisms that regulate the flow of information, thereby enhancing the model's ability to learn complex nonlinear relationships while maintaining training stability. The output of the Feature Transformer is expressed as:

$$[d^{(i)}, a^{(i)}] = f_i(M_f^{(i)})$$

6.4 Split Block

The output of the Feature Transformer is divided into two components:

- $d^{(i)} \in R^{B\times N_d}$, which represents the decision contribution of the current decision block and contributes directly to the final prediction.
- $a^{(i)} \in R^{B\times N_a}$, which represents the feedback information passed to the Attentive Transformer in the subsequent decision step to guide the selection of relevant features (Arik & Pfister, 2021).

7. AGGREGATION AND FINAL OUTPUT

After all decision steps have been processed, the outputs of each decision block are aggregated to form the final decision representation, d_{out}. Before aggregation, the ReLU activation function is applied to the output of each decision block, as expressed by:

$$d_{out} = \sum_{i=1}^{N_{steps}} ReLU\left(d^{(i)}\right)$$

Finally, the Final Prediction stage applies a linear transformation to the aggregated representation to produce the model output. This transformation generates class scores for classification tasks or continuous numerical values for regression tasks (Arik & Pfister, 2021).

8. TABNSA ALGORITHM

The TabNSA algorithm is a hybrid framework that integrates the NSA mechanism with the TabMixer architecture. It employs a hierarchical sparse attention mechanism combined with parallel Multi-Layer Perceptron (MLP) networks to effectively handle high-dimensional heterogeneous tabular data while maintaining excellent scalability. This hybrid architecture dynamically selects the most informative feature groups for each sample and models complex nonlinear interactions by treating each individual feature as a token (Eslamian & Cheng, 2026).

8.1 Input Representation

The input to the model is a multidimensional tensor:

$$X \in R^{B\times N}$$

where Bdenotes the batch size and Nrepresents the number of input features. Each numerical feature is treated as an individual token.

8.2 Feature Expansion and Projection

During the Expansion stage, each numerical feature is projected into a higher-dimensional embedding space of dimension Dusing a shared linear embedding layer:

$$X \leftarrow Linear(1 \rightarrow D)$$

The embedded features are then projected into Query (Q), Key (K), and Value (V) representations through learnable projection matrices:

$$q = W_q X, k = W_k X, v = W_v X$$

where W_q, W_k, and $W_v$are learnable projection matrices.

Given the sequential input consisting of query vectors q_t, key vectors k_t, and value vectors v_t, the attention output is computed as:

$$o_t = Attn(q_t, k_{1:t}, v_{1:t}) = \sum_{i=1}^{t} \alpha_{t,i} v_i$$

8.3 Parallel Processing Blocks

The feature embeddings are processed through two parallel branches to capture both global contextual information and fine-grained feature interactions.

8.3.1 NSA Branch

Instead of employing the standard attention mechanism, whose computational complexity increases rapidly with the number of features, TabNSA replaces it with the NSA module. NSA reduces computational complexity by reconstructing the key-value pairs into compact representations:

$$\tilde{k}_t = f_K(q_t, k_{1:t}, v_{1:t}), \tilde{v}_t = f_V(q_t, k_{1:t}, v_{1:t})$$

Table 1 Comparison of the Architectural Parameters of the TabNet and TabNSA Models

Parameter	TabNet	TabNSA
Input Dimension	8	8
Output Dimension	3	3
Decision Dimension (n_d)	32	64
Attention Dimension (n_a)	32	64
Number of Decision Steps (n_{steps})	3	5
Relaxation Parameter (γ)	1.2	1.0
Epsilon (ϵ)	1×10^{-10}	1×10^{-10}
Number of Shared Layers (n_{shared})	3	2
Number of Independent Layers ($n_{independent}$)	3	2
Number of Chunks (n_{chunks})	64	64
Chunk Size ($chunk\_size$)	0	0
Ghost BN	True	True
BN Momentum	0.02	0.02
Track Running Statistics	False	False
Number of Attention Heads (n_{heads})	—	4
Attention Embedding Dimension ($attention\_dim$)	—	64

Token Compression: To eliminate redundancy among highly correlated features, consecutive feature blocks are compressed into block-level representations using a learnable **MLP** (ϕ)together with positional encoding within each block. The compressed key representation is defined as:

$$\tilde{k}_t^{cmp} = \left\{ \phi(k_{id+1:id+L}) \mid 0 \le i \le \lfloor \frac{t-L}{d} \rfloor \right\}$$

where Ldenotes the block length, $d$is the sliding stride between adjacent blocks, and $\phi$is a learnable MLP incorporating positional encoding (Eslamian & Cheng, 2026).

Token Selection: The compressed feature blocks are ranked according to their importance scores derived from the attention between the query vectors and the compressed keys. Only the top-n highest-ranked blocks are retained, enabling the model to ignore noisy or irrelevant feature dimensions. The importance scores are computed as:

$$p_t^{cmp} = Softmax(q_t^T \tilde{k}_t^{cmp})$$

Sliding Window: To capture local dependencies among logically related feature groups, a fixed-size sliding window of length $w$is maintained:

$$\tilde{k}_t^{win} = k_{t-w:t}, \tilde{v}_t^{win} = v_{t-w:t}$$

Gated Fusion: Finally, the compressed, selected, and sliding-window branches are combined using learnable gate scores to produce an attention-aware representation:

$$Y_{attn} = \sum_{c \in \{cmp, slc, win\}} g_t^c \cdot Attn(q_t, \tilde{K}_t^c, \tilde{V}_t^c)$$

where:

$$g_t^c \in [0, 1]$$

denotes the learnable gate score corresponding to branch c.

8.3.2 TabMixer Branch

In parallel with the NSA branch, the feature embeddings are processed by the TabMixer module, which employs parallel MLP branches with independent parameters to capture complex nonlinear feature interactions. The interaction is formulated as:

$$Z_{mixer} = SiLU\left[GeLU(MLP_1(X^T))^T \odot MLP_2(X)\right] + X$$

where:

- MLP_1performs **channel-wise mixing**.
- MLP_2performs **token-wise mixing**.

The combination of the GeLU and SiLU activation functions enables the model to simultaneously capture complex global dependencies across both features and samples (Eslamian & Cheng, 2026).

8.4 Feature Fusion and Aggregation

Following the parallel processing stage, the outputs of the NSA and TabMixer branches are fused to balance global contextual information with local feature interactions:

$$X_{fused} = Y_{attn} + Z_{mixer}$$

To obtain a single representation for each input sample, global mean pooling is applied:

$$X_{final} = MeanPooling(X_{fused})$$

Finally, the aggregated feature vector is passed through a two-layer MLP to

Table 2 Comparison of the Training Parameters of TabNet and TabNSA Models

Parameter	TabNet	TabNSA
Optimizer	Adam	Adam
Learning Rate	0.007	0.003
Learning Rate Scheduler	StepLR	StepLR
Scheduler Step Size	1	2
Scheduler Gamma	0.7	0.85
Loss Function	MSELoss	MSELoss
Sparsity Coefficient	1×10^{-3}	5×10^{-4}
Number of Epochs	8	15
Batch Size	6144	4096
Device	CPU	CPU / GPU
Data Loading Strategy	Chunked (MariaDB)	Chunked (MariaDB)
Evaluation Batch Size	6144	4096
Shuffle	True	True
Drop Last Batch	True	True

Table 3 Comparison of the Investigated Algorithms in Terms of RMSE and MAE

Metric	Variable (Unit)	TabNSA	TABnet	CNN-BILSTM	CNN-LSTM
RMSE	ALT (m)	47.70	51.20	78.06	310.13
	Lat (°)	0.013	0.027	0.043	0.189
	Lon (°)	0.056	0.090	0.053	0.098
MAE	ALT (m)	29.10	32.80	60.20	219.16
	Lat (°)	0.019	0.028	0.033	0.159
	Lon (°)	0.008	0.014	0.043	0.080

generate the final prediction:

$$Output = MLP(X_{final})$$

where the final layer employs the GeLU activation function for classification tasks (Eslamian & Cheng, 2026).

9. DATA SPLITTING AND CONSISTENT EXPERIMENTAL SETTINGS

To ensure a fair and reliable evaluation, the processed ADS-B dataset was divided into 80% training, 10% validation, and 10% independent test sets at the trajectory or flight-segment level to prevent temporal data leakage. The normalization parameters were calculated from the training set only and then applied to the validation and test sets. The CNN-LSTM and CNN-BiLSTM baselines were evaluated using the same dataset, data partitions, input features, preprocessing procedure, random seed, batch size, number of training epochs, and computational environment as the proposed TabNSA model. The validation set was used for model selection and hyperparameter tuning, while the test set remained unseen until final evaluation. This unified protocol ensures that the observed performance differences primarily reflect the learning architectures rather than differences in data or experimental conditions.

10. PRACTICAL COMPARISONS AND RESULTS

In this study, the proposed system was implemented using the two algorithms TabNSA and TabNet. Tables 1 and 2 present the training and testing parameters, as well as the specific architectures of both algorithms. The proposed methods were compared with several algorithms introduced in previous studies (Ding et al., 2022; Zhou et al., 2022), namely CNN-BiLSTM and CNN-LSTM, respectively.

To evaluate and compare the performance of these algorithms, two evaluation metrics were adopted: the RMSE and the MAE.

The RMSE represents the square root of the average squared differences between the model predictions and the actual target values, while the MAE represents the average absolute differences between the predicted results and the ground-truth values. They are defined by the following equations:

$$RMSE = \left[\frac{1}{N}\sum_{j=1}^{N}(Y_j - X_j)^2\right]^{\frac{1}{2}} \quad MAE = \frac{1}{N}\sum_{j=1}^{N}|Y_j - X_j|$$

where:

- N represents the number of samples.
- Y_j represents the predicted aircraft trajectory.
- X_j represents the actual aircraft trajectory.

A lower value of these two metrics indicates that the predicted trajectory is closer to the actual trajectory, which reflects higher predictive accuracy and better model performance.

Table 3 presents a comparison of the performance of the proposed TabNSA and TabNet algorithms with previous approaches, including BiLSTM and CNN-LSTM.

Figure 3 presents a two-dimensional comparison between the ground-truth aircraft trajectory and the trajectories predicted by the studied algorithms. This visualization provides a qualitative assessment of each model's prediction accuracy by illustrating the degree of overlap between the predicted flight paths and the actual trajectory in the horizontal plane.

To further illustrate the prediction performance, the three-dimensional flight trajectories generated by the investigated algorithms are plotted and compared with the corresponding ground-truth trajectory. The 3D visualization provides a more comprehensive representation of the aircraft's spatial movement by simultaneously considering latitude, longitude, and altitude, thereby facilitating a more intuitive evaluation of each model's ability to capture the aircraft's dynamic behavior. The resulting comparison is presented in Figure 4.

To analyze the distribution of feature importance across the decision steps, the interpretation masks of both TabNet and TabNSA are visualized, as shown in Figure 5. The interpretation mask of TabNet exhibits a sparse feature selection pattern, indicating that the model focuses on only a limited subset of features at each decision step. This sparse attention mechanism enhances the interpretability of the model by clearly identifying the most influential features; however, it may reduce the model's capacity to capture complex nonlinear interactions among multiple features.

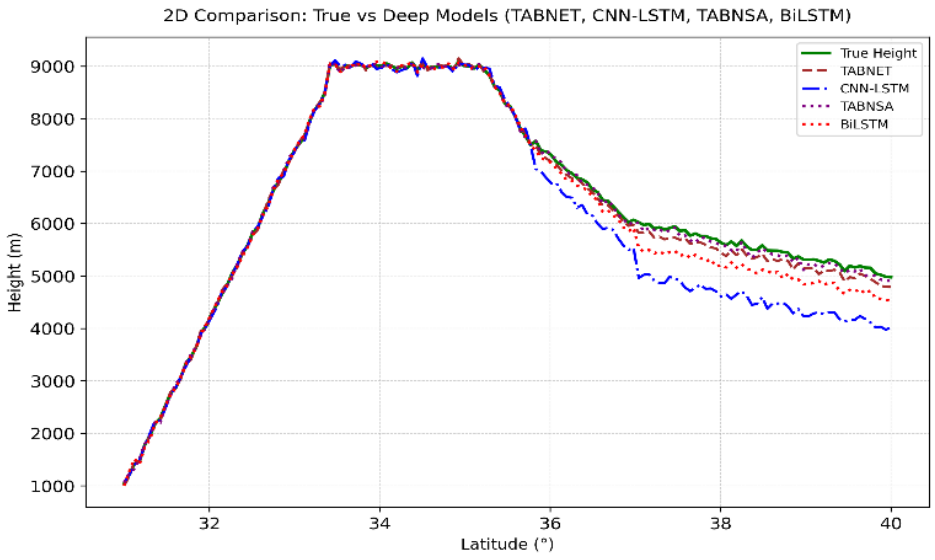
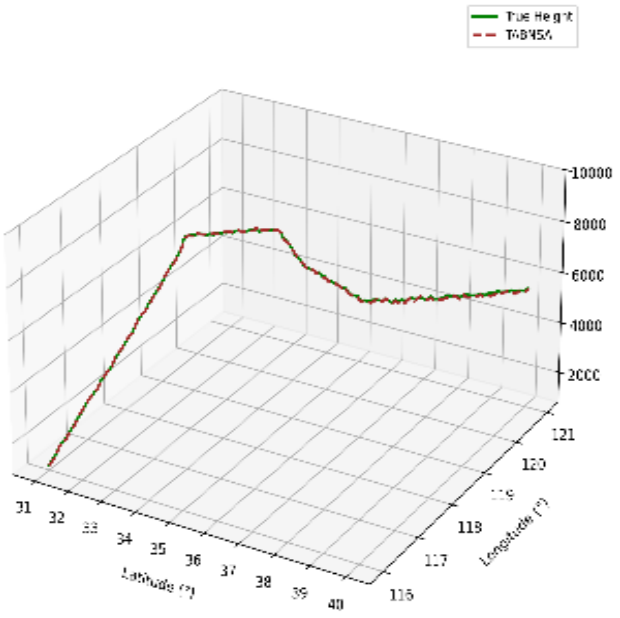
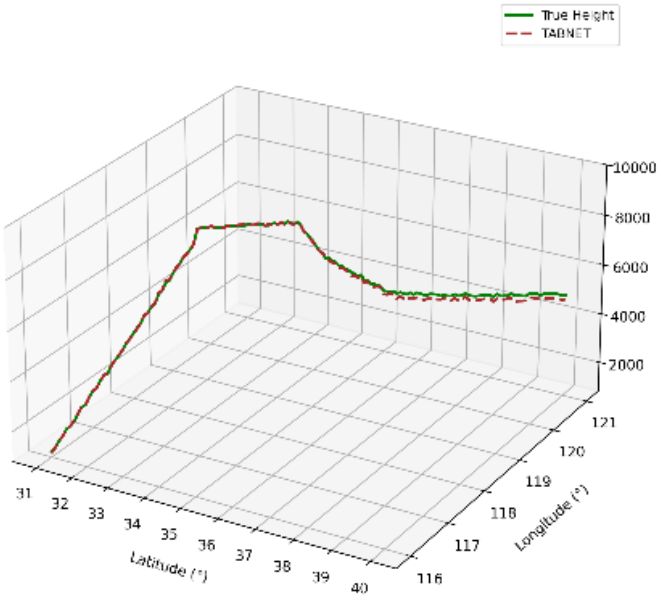


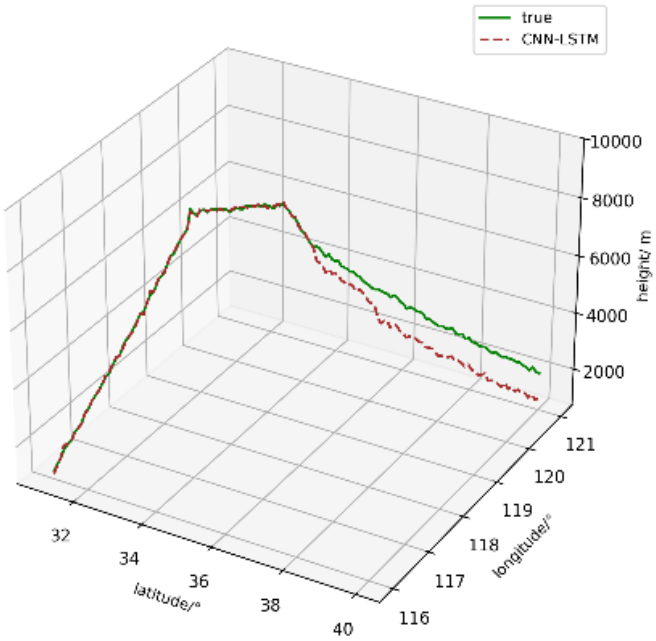
Figure 3 Two-Dimensional Comparison between the Ground-Truth Aircraft Trajectory and the Trajectories Predicted by the Studied Algorithms



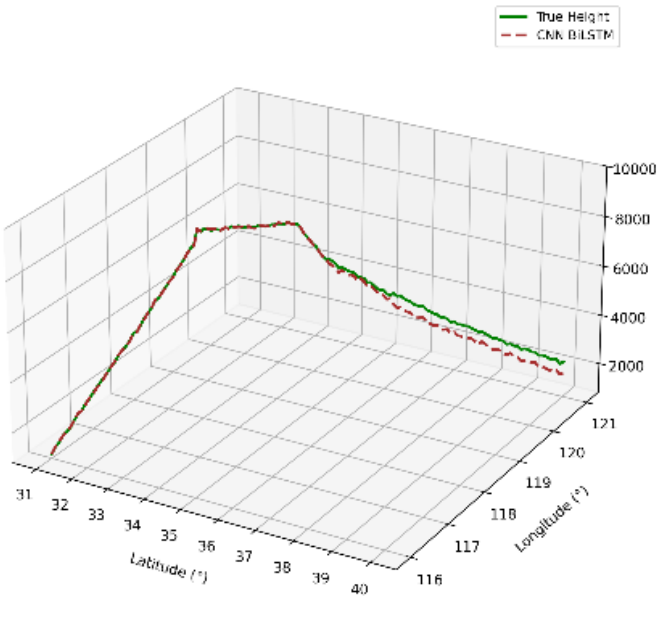
3D Flight Path: True vs TabNSA Predicted (Adjusted Ending)



3D Flight Path: True vs TabNet Predicted (Adjusted Ending)



3D Flight Path: True vs CNN-LSTM Predicted (Adjusted Ending)



3D Flight Path: True vs CNN-BiLSTM Predicted (Adjusted Ending)

Figure 4 Three-Dimensional Comparison between the Ground-Truth Trajectory and the Predicted Trajectories of the Studied Algorithms

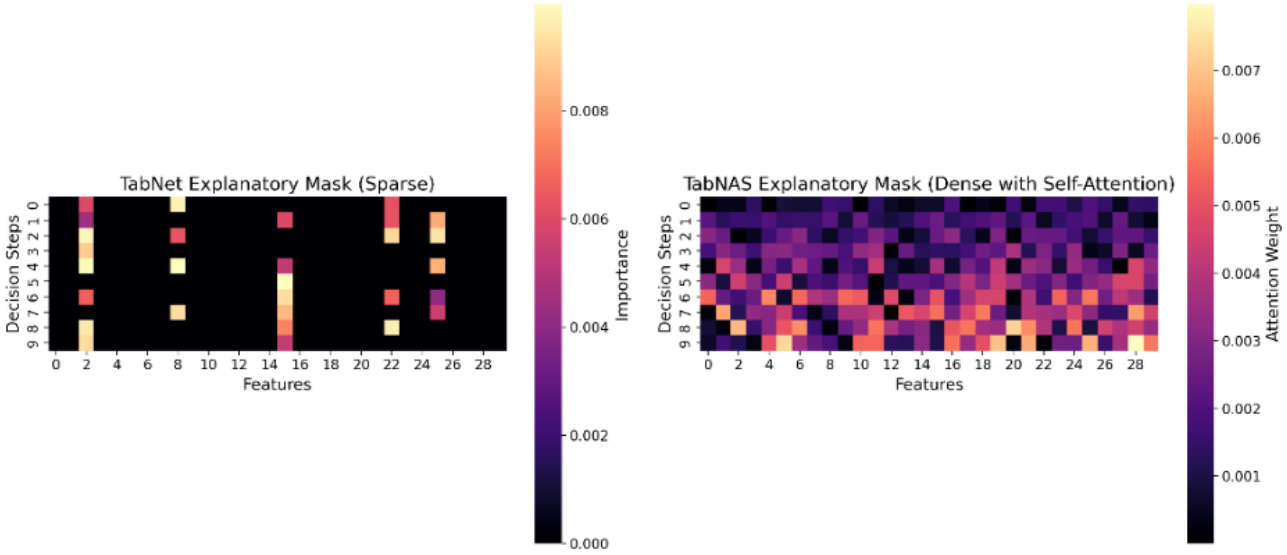


Figure 5 Interpretation Masks of the TabNet and TabNSA Models

In contrast, the interpretation mask of TabNSA demonstrates a denser attention distribution, where most features contribute with varying importance weights due to the integration of the NSA mechanism and self-attention operations. This richer attention distribution enables TabNSA to model more complex feature dependencies and better exploit the intrinsic structure of the data.

Consequently, while TabNet offers greater interpretability through explicit sparse feature selection, TabNSA provides a more expressive feature representation, allowing it to model richer feature interactions and achieve superior predictive performance on aircraft trajectory prediction tasks.

11. RESULTS

Based on Table 3, the proposed TabNSA model achieved the lowest RMSE and MAE values across all prediction variables compared with TabNet, CNN-BiLSTM, and CNN-LSTM. For altitude prediction, TabNSA obtained an RMSE of 47.7 m, compared with 51.2 m for TabNet, 78.06 m for CNN-BiLSTM, and 310.13 m for CNN-LSTM. This corresponds to an improvement of approximately 7% over TabNet and 39% over CNN-BiLSTM.

For altitude prediction, the MAE achieved by TabNSA was 29.1 m, compared with 32.8 m for TabNet, 60.2 m for CNN-BiLSTM, and 219.16 m for CNN-LSTM. Consequently, TabNSA reduced the prediction error by approximately 11% relative to TabNet and 52% relative to CNN-BiLSTM.

For the latitude and longitude coordinates, TabNSA consistently outperformed TabNet, reducing both RMSE and MAE by approximately 50%, thereby providing higher accuracy in predicting the aircraft's horizontal trajectory.

As illustrated in Figure 5, TabNSA employs a dense attention distribution in which most input features contribute to the prediction with varying importance weights. This enables the model to capture richer and more complex feature interactions. In contrast, TabNet relies on a sparse interpretation mask, selecting only a limited subset of features at each decision step. While this sparse mechanism improves model interpretability, it may limit its predictive capability by overlooking complex feature dependencies.

From Figures 3 and 4, it can be observed that the trajectories predicted by TabNSA closely match the ground-truth trajectories across all spatial coordinates. In comparison, the other investigated models exhibit larger deviations, particularly during rapid altitude changes and complex flight maneuvers.

Although TabNet provides greater interpretability through its sparse feature selection mechanism, TabNSA achieves a better balance between interpretability and predictive performance. Its enhanced representation learning capability and superior prediction accuracy make it a more suitable choice for aircraft trajectory prediction applications that require high predictive precision while still providing meaningful insights into feature importance.

AI USE DISCLOSURE

During the preparation of this manuscript, the authors used ChatGPT (OpenAI) to assist with the English translation and language refinement. The generated text was thoroughly reviewed, edited, and verified by the authors. The authors assume full responsibility for the final content of the manuscript.

REFERENCES

Arik, S. Ö., & Pfister, T. (2021). TabNet: Attentive interpretable tabular learning. P*roceedings of the AAAI Conference on Artificial Intelligence*, *35*(8), 6679–6687. https://doi.org/10.1609/aaai.v35i8.16826

Ding, W., Huang, J., Shang, G., Wang, X., Li, B., Li, Y., & Liu, H. (2022). Short-term trajectory prediction based on hyperparametric optimisation and a dual attention mechanism. *Aerospace*, *9*(8), 464. https://doi.org/10.3390/aerospace9080464

Eslamian, A., & Cheng, Q. (2026). TabNSA: Native sparse attention for efficient tabular data learning. *Neurocomputing*, *675*, 132928. https://doi.org/10.1016/j.neucom.2026.132928

Kim, B., Kim, J., Chae, H., Yoon, D., & Choi, J. W. (2016). Deep neural network-based automatic modulation classification technique. In 2*016 International Conference on Information and Communication Technology Convergence (ICTC)* (pp. 579–582). IEEE. https://doi.org/10.1109/ICTC.2016.7763537

Olive, X., Krummen, J., Figuet, B., & Alligier, R. (2025). Filtering techniques for ADS-B trajectory preprocessing. *Journal of Open Aviation Science*, *2*(2). https://doi.org/10.59490/joas.2024.7882

Zhou, J., Zhang, H., Lyu, W., Wan, J., Zhang, J., & Song, W. (2022). Hybrid 4-dimensional trajectory prediction model, based on the reconstruction of prediction time span for aircraft en route. *Sustainability*, *14*(7), 3862. https://doi.org/10.3390/su14073862