Received 20 November 2023, accepted 13 December 2023, date of publication 18 December 2023, date of current version 22 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3343800

RESEARCH ARTICLE

Long-Term Trajectory Prediction Model Based on Transformer

QIANG TONG[1,2], JINQING HU[1,2], YULI CHEN[1,2,3], DONGDONG GUO[1,2], AND XIULEI LIU[1,2]

[1]Laboratory of Data Science and Information Studies, Beijing Information Science and Technology University, Beijing 100101, China
[2]Beijing Advanced Innovation Center for Materials Genome Engineering, Beijing Information Science and Technology University, Beijing 100101, China
[3]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Qiang Tong (tongq85@bistu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFB2600600, in part by the Research and Development Program of Beijing Municipal Education Commission under Grant KM202111232003, and in part by the Key Research and Cultivation Project in the Promotion of University Classification Development under Grant 2121YJPY225.

ABSTRACT Recurrent neural network models have problems such as memory loss and gradient disappearance when dealing with long time series data. This paper proposes a long-term trajectory prediction model based on Transformer to process long-term sequence information. Firstly, the position encoding is used to preserve the relative positional relationship between trajectory points. Secondly, the multi-head attention mechanism is used to fully learn the feature information between different trajectories, and the trajectory data can be encoded at one time. Finally, the encoder and decoder mechanism is used to predict future trajectory data. Compared with the long-term trajectory prediction benchmark method TrajAirNet, the average displacement error, absolute displacement error of the proposed model on the long-term trajectory dataset are reduced by about 8.2% and 51.4%, respectively. The experimental results show that the proposed model has higher accuracy and robustness on long-term trajectory prediction dataset.

INDEX TERMS Trajectory prediction, aircraft trajectory, transformer, ADS-B.

I. INTRODUCTION

The conflict between air traffic volume and airspace resources is becoming increasingly intense. Traditional air traffic control only focuses on the control of the safe distance between each aircraft, can not meet the needs of subsequent development. Relying solely on the flight plan in air traffic control, without future trajectory awareness, can easily lead to flight conflicts. In situations with high air traffic volume, failure to adjust manually in a timely manner can lead to congestion in the airspace. This not only increases fuel consumption for aircraft but also adds to the workload of air traffic control personnel. Therefore, the air traffic control system is in urgent need of transformation. To alleviate the problem of limited airspace resources, enhance the efficiency of flight control coordination, and improve flight safety, the International Civil Aviation Organization (ICAO) has introduced the concept of Trajectory Based Operation (TBO) and recognized it as the fundamental technology for the future generation of air traffic control systems [1]. The smooth operation of TBO is closely related to the four-dimensional trajectory prediction of aircraft. Four-dimensional trajectory prediction refers to the evaluation and estimation of the changing longitude, latitude, altitude, and other positional information of an aircraft over time. It aims to reduce potential conflicts during flights and accurately understand the overall operational status of the entire airspace, thereby reducing the workload of air traffic control personnel. It also supports automated flight control and management, providing technical support for air traffic safety and traffic flow management.

Trajectory prediction can be divided into short-term trajectory prediction and long-term trajectory prediction based on the predicted time scale [2]. Long-term trajectory prediction, in contrast to short-term prediction, provides flight path estimates over a more extensive time frame, enabling its application in air traffic flow prediction, flight planning optimization, and fuel consumption optimization [3].

In the case of time series data, there is a specific temporal order and correlation between the current moment and past moments. Therefore, when using neural networks to process time series data, it is necessary to consider the patterns of certain random variables that change over time. Recurrent Neural Network (RNN) is a widely used model for processing

The associate editor coordinating the review of this manuscript and approving it for publication was Juan A. Lara.

time series data [4]. RNN has the ability to retain information in its memory, but it fails to effectively preserve long-range dependencies in time series data due to issues such as vanishing or exploding gradients.

In order to alleviate the gradient problem caused by Recurrent neural networks, researchers have proposed methods to control information accumulation, such as the widely used models: Long Short-Term Memory (LSTM) [5] and Gate Recurrent Unit (GRU) [6].

A. MOTIVATION

Models based on recurrent neural networks are trained with sequential input and can only compute in a left-to-right or right-to-left order. These leads to two problems. First, the current result relies on the computation of the previous moment, which means that the training data needs to be inputted sequentially. This affects the efficiency of model computation and increases training time. Information loss occurs during computation. Although LSTM, GRU, and similar models with gated mechanisms alleviate the gradient issues caused by long-term dependencies, they cannot eliminate them completely.

To address the aforementioned problems and leverage the benefits of Transformers [7] in handling long time series data, this paper proposes a long-term trajectory prediction model named Trajectory Embedding Transformer (TET). The model first utilizes Positional Encoding (PE) to capture the relative positional relationships among trajectory points. Then, by using the attention mechanism, the embedding operation between the trajectory points is completed in a single step. At the same time, the use of multi-head attention allows for the learning of diverse features between trajectory points across various representation spaces. Overall, the model predicts the future trajectory based on the encoder-decoder structure.

In summary, our contributions are as follows:

1) To our knowledge, the proposed TET model is the first to introduce transformer into long-term trajectory prediction. By combining multi-head attention mechanism with residual networks, it effectively captures both long-term and short-term features of trajectory data, mitigating the limitations of RNN-based models in feature extraction during training and inference.

2) Two datasets, Traj-60 and Traj-120, were constructed and experimentally validated the advantage of our proposed model in long-track prediction. On the long-term trajectory data set, the average and absolute displacement errors of the model are reduced by 8.2% and 51.4%, respectively.

3) We conducted comparative experiments between the proposed model and the baseline model in terms of accuracy, efficiency, and practicality. The experimental results validated the advantages of our model in these aspects.

The rest of the paper is organized as follows. In section II, we review related works in recurrent neural network models for trajectory prediction and Transformer. Section III presents our proposed methodology, starting with the problem formulation and followed by the introduction of our TET model architecture. Section IV presents the experimental setup, including the dataset, implementation details, evaluation metrics, and the results and analysis. Section V offers conclusions, the limitation of our model and suggestions for future research.

II. RELATED WORK

A. RECURRENT NEURAL NETWORK MODELS FOR TRAJECTORY PREDICTION

The increasing prominence of artificial intelligence has led to the widespread adoption of LSTM and GRU as prominent techniques for trajectory prediction. Shi et al. [5] implemented an LSTM-based trajectory prediction model, incorporating a sliding window technique to maintain trajectory continuity and enhance prediction accuracy. Zhang et al. [8] presented a novel LSTM network optimized with the Ant Lion Optimizer (ALO) algorithm for trajectory prediction. By controlling the initial weight values of the LSTM network using the ALO algorithm, the convergence speed was improved. Han et al. [9] introduced a 4D trajectory prediction model based on deep learning techniques. In their study, LSTM was used to model the aircraft's motion state. Zhang et al. [10] proposed a hybrid model that combined Deep Neural Networks (DNN) and LSTM. The DNN was used to refine the trajectory predicted by LSTM, thereby improving trajectory prediction accuracy. Furthermore, Zhao et al. [11] proposed a Deep Long Short-Term Memory (D-LSTM) neural network for aircraft trajectory prediction, which improved the accuracy of predictions in complex flight environments. Pang et al. [12] presented a novel network architecture that embedded convolutional layers within the repeating modules of LSTM. This allowed for extracting useful features from the Weather Cube and addressed the issue of predicting aircraft trajectories related to pre-takeoff convective weather. Shi et al. [13] proposed a combined model for short-term trajectory prediction. They used two methods to predict altitude and obtained the final altitude value through weighted averaging, which improved the accuracy of trajectory prediction. Shi et al. [14] introduced an online-updating short-term trajectory prediction model. First, the LSTM model was trained using historical trajectory data, and the model parameters were saved. The model was then further trained and updated using real-time trajectory data, improving trajectory prediction accuracy. Liu et al. [15] proposed a deep generative model that includes an LSTM encoder network and a mixture density LSTM decoder network, which improved trajectory prediction accuracy. Zhang et al. [16] presented an aircraft trajectory prediction model based on GRU, which enhanced the accuracy of aircraft flight trajectory prediction.

Although LSTM and GRU alleviate the vanishing or exploding gradient issues caused by long-term dependencies and are effective for short-term trajectory prediction, they do not completely solve the gradient problem and cannot be directly applied to long-term trajectory prediction.

B. TRANSFORMER

The groundbreaking Transformer model, introduced by Google in 2017 for machine translation tasks [7], diverges from traditional models by eliminating the use of recurrent neural networks (RNNs) and instead solely relying on the attention mechanism for sequence-to-sequence generation [17].

The application of the attention mechanism originally emerged within the field of computer vision, inspired by the observation of animals focusing their attention on key information when perceiving external objects [18]. The attention mechanism mimics the way biological systems observe information by selectively focusing and directing attention towards the relevant aspects of the observed stimuli. This allows for the quick identification of desired information from a complex set of inputs.

Many researchers have applied it to trajectory prediction. Zhang et al. [19] proposed an attention-based Convolutional LSTM (AttConvLSTM) network that converts the trajectory prediction problem into a classification problem by segmenting the reachable area. It calculates the arrival probabilities for each spatial location within the reachable area of the target aircraft. Jia et al. [20] introduced a trajectory prediction model based on the attention mechanism and LSTM. LSTM is used to extract temporal features from trajectories and improve accuracy, while the attention mechanism is employed to capture important factors that influence the variations at the current point for trajectory prediction. Nathan et al. [21] developed a hybrid model based on the attention mechanism, demonstrating that it can significantly improve the performance of convolutional layers and enhance the dimensional capacity of the learning model.

In Transformer, only use attention mechanical can not capture the sequential characteristic of series. The sinusoidal version of position embedding is used to solve this problem [7]. By summing the position encodings with the input embeddings and feeding the result the model, it can give the model a sense of the order in which the token is currently processed [22].

The Transformer model, at its core, employs an encoder-decoder structure with the attention mechanism module as its primary component. Its computational approach aligns with parallel processing, making it well-suited for modern GPU frameworks. In contrast to RNN-based models that rely on sequential training and inference [23], the Transformer model offers shorter training and inference times. Consequently, the Transformer model not only addresses the issue of long-term dependencies but also showcases higher computational efficiency when compared to RNN-based models. These attributes establish the Transformer model as a compelling choice in diverse research studies [24], [25].

III. METHODOLOGY

A. PROBLEM FORMULATION

Assuming $\mathbf{X}$ represents all the trajectories, including the trajectories of N aircraft, it can be denoted as $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$. When the time step is $T = 1, 2, \ldots, t_{obs}$, the input trajectory of aircraft i is represented as $\mathbf{X}_i = (x_i^t, y_i^t, z_i^t)$, where t_{obs} denotes the length of the observation sequence and x_i^t represents the longitude of aircraft i at time t, y_i^t represents the latitude of aircraft i at time t, and z_i^t represents the altitude of aircraft i at time t. When the time step is $T = t_{obs+1}, t_{obs+2}, \ldots, t_{pred}$, the real trajectory of aircraft i is denoted as $\mathbf{Y}_i = (x_i^t, y_i^t, z_i^t)$, and the predicted trajectory is represented as $\hat{\mathbf{Y}}_i = (\hat{x}_i^t, \hat{y}_i^t, \hat{z}_i^t)$. $\hat{x}_i^t$ denotes the predicted longitude of aircraft i at time t, $\hat{y}_i^t$ represents the predicted latitude of aircraft i at time t, and $\hat{z}_i^t$ represents the predicted altitude of aircraft i at time t.

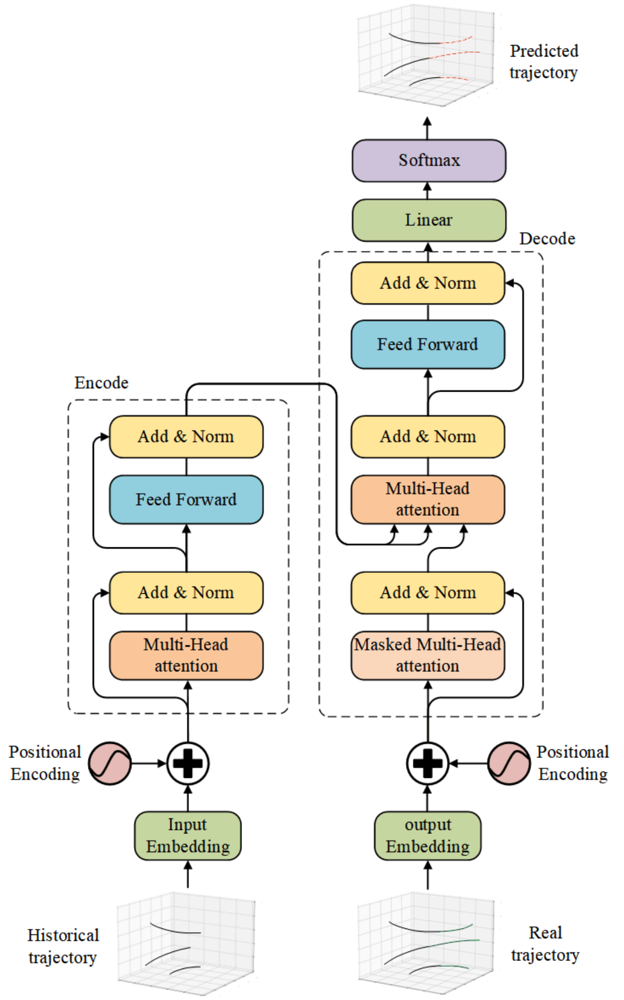


FIGURE 1. Model framework of TET. After Embedding and Positional Encoding, the input trajectory data is sent to the encoder in the left dotted box to get the representation of the trajectory, and then to the decoder on the right to make the trajectory prediction.

B. TET MODEL ARCHITECTURE

The architecture of TET, as shown in Fig. 1, primarily consists of position encoding, encoder modules, and decoder modules. The historical trajectory is encoded with positional information, incorporating the relative positions between trajectory points. This encoded information is then fed into the encoder, where feature learning takes place, resulting in intermediate encoding information. The intermediate encoding information is subsequently passed to the decoder for decoding. The decoder combines the intermediate encoding information with the position-encoded real trajectory information and decodes it. Finally, the decoded information undergoes linear vector projection and softmax probability transformation operations to generate the predicted trajectory.

1) POSITIONAL ENCODING

Due to the varying number of trajectory points in each trajectory, before inputting the aircraft data $\mathbf{X}_i = (x_i^t, y_i^t, z_i^t), t = 1, 2, \ldots, t_{obs}$ into the network, an embedding operation is performed to embed the trajectory data. This involves mapping the trajectory data to a high-dimensional vector space $\mathbf{ed}_i^t$ using an embedding layer.

$$\mathbf{ed}_i^t = \zeta(x_i^t, y_i^t, z_i^t; \mathbf{W_{ed}}) \tag{1}$$

where $\zeta(\cdot)$ represents the linear embedding function, and $\mathbf{W}_{ed}$ denotes the weight parameters of $\zeta(\cdot)$ during the embedding process.

RNNs possess the capability to capture the temporal order of input data, as the sequence of input data represents its positional information. Nevertheless, the utilization of attention mechanisms instead of RNNs may result in a potential drawback of losing temporal information across trajectory points. Consequently, the model becomes unable to comprehend the relative positional information among trajectory points is compromised. Therefore, it is necessary to incorporate position encoding in the trajectory embeddings, forming a new representation $\mathbf{en}_i^t$ that is input to the encoding layer of the model. This enables the model to learn the relative positional temporal information. The formula for position encoding is shown in Equation (2), (3) and (4).

$$\mathbf{pe}(t, 2c) = \sin(\frac{t}{10000^{\frac{2c}{d}}}) \tag{2}$$

$$\mathbf{pe}(t, 2c+1) = \cos(\frac{t}{10000^{\frac{2c}{d}}}) \tag{3}$$

$$\mathbf{en}^t = \mathbf{ed}^t + \mathbf{pe}^t \tag{4}$$

where d represents the dimensionality of the vector $\mathbf{pe}^t$ after position encoding, which is equal to the dimensionality of the embedded vector $\mathbf{ed}_i^t$. $2c$ and $2c+1$ correspond to the even and odd columns of the dimension, respectively.

2) ENCODER

The model consists of multiple encoders that share the same structure. Each encoder consists of multiple components, including a multi-head self-attention layer, a normalization layer, residual connections (Add & Norm), and a feed-forward fully connected layer. The role of the encoder layer is to map all input sequences into a vector space that contains learned information about the entire sequence. The structure of the encoder can be represented as follows:

$$\mathbf{e}_l = EncoderLayer(\mathbf{e}_{l-1}), l \in [1, n] \tag{5}$$

where $EncoderLayer(\cdot)$ represents an encoder layer, n denotes the total number of encoder layers, and $\mathbf{e}_l$ represents the output of the encoder layer.

The model captures both long-term and short-term dependencies in trajectory data through the attention mechanism. Since attention weights are not shared, the multi-head attention mechanism can effectively learn different features from the training data. It computes the attention weights for the input and generates an output vector that contains encoded information on how each trajectory point should attend to other points in the sequence.

Another important component in the encoder is the Feed Forward Layer. After performing attention calculations, a fully connected feed-forward network is applied to the vector $\mathbf{e}_m$ at each position. The vector undergoes two linear transformations, followed by activation functions, and is then added to the output of the attention mechanism. Finally, the result is passed through a normalization function.

$$\begin{aligned} FFN(\mathbf{e}_m) = Norm(\mathbf{e}_m + Dropout(relu(\mathbf{e}_m\mathbf{W}_1 \\ + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2)) \end{aligned} \tag{6}$$

where $Norm(\cdot)$ represents the normalization function, $Dropout(\cdot)$ is the dropout function, $relu(\cdot)$ is the activation function, and $\mathbf{W}_1$ and $\mathbf{W}_2$ are weight vectors, while $\mathbf{b}_1$ and $\mathbf{b}_2$ are position offset vectors.

In the attention mechanism module, the output of each time step has already integrated the information of all time steps, so in the following fully connected feed-forward network, each time step is only a further integration of its own characteristics, independent of other time steps.

The normalization function serves the purpose of constraining the feature values within a reasonable range. This is necessary because as the path length of the model's input and output increases, the calculations involved may result in excessively large or small values, which can hinder the convergence rate of the model.

3) DECODER

The decoder, similar to the encoder, accepts the real trajectory as input, following position encoding. RNNs are commonly used in traditional Seq2Seq models because the input processing of the model is sequential. This sequential processing requires the computation at the current time step to rely on the result from the previous time step. During the training process, the model is not allowed to perceive the future data at time step t. The data at time step $t+1$ can only be observed after the computation at time step t has been completed. However, the RNN is replaced by an attention mechanism in the encoder module, which poses a problem. During training, the whole real trajectory truth is exposed to the encoder. To solve this problem, masked multi-head attention is employed in the decoder. The decoder is trained by utilizing a fixed length of trajectory data as computation units and is solely influenced by historical trajectory data, disregarding any future trajectory data. Therefore, in order to prevent any potential interference with the training outcomes, the subsequent real data is masked using a triangular matrix.

In the multi-head attention module of the encoder, the query vector $\mathbf{Q}$ comes from the previous sub-layer, while the keys $\mathbf{K}$ and values $\mathbf{V}$ come from the output of the encoder. It establishes the relationship between the encoder and the decoder, and uses the predicted trajectory information from the decoder as $\mathbf{Q}$ to extract relevant information and integrate it into the current track features extracted by the encoder, to make predictions.

4) LOSS FUNCTION

The TET model utilizes the L2 loss method as its loss function to evaluate the magnitude of the error between the real trajectory $\mathbf{Y}_i$ and the predicted trajectory $\hat{\mathbf{Y}}_i$. The relevant formula is as follows:

$$Loss_{L2} = \frac{1}{-}n \sum_{i=1}^{n} (\mathbf{Y}_i - \hat{\mathbf{Y}}_i)^2 \tag{7}$$

IV. EXPERIMENTS

A. DATASET

The experimental dataset of this study is from the Automatic Dependent Surveillance-Broadcast (ADS-B) system. Common sources of ADS-B data include the OpenSky Network [26], ADSB Exchange [27], and ADSBHub [28]. The OpenSky Network constructs a large-scale trajectory database using real-time data uploaded by volunteers, aviation enthusiasts, and academic organizations. It provides researchers with access to the data to enhance airspace safety, reliability, and efficiency. ADSB Exchange is the world's largest unfiltered public source website for flight data, containing not only civilian aircraft data but also data from military and certain private aircraft. ADSBHub offers real-time ADS-B data sharing and exchange services, but data access is subject to certain restrictions, granting access to trajectory data for those who contribute data. Considering the convenience of data acquisition, The experimental data of this study is from the OpenSky Network.

The ADS-B data presents problems such as missing and duplicate trajectory points. The data is represented in timestamps, making it difficult to visually interpret specific time information, which hampers subsequent processing. Moreover, the trajectories have varying lengths, with some containing only a few points, rendering them unsuitable for subsequent experiments. Furthermore, since the dimensions of longitude, latitude and altitude are different, the numerical differences are large, which may cause the model to be unable to correctly measure the importance of features in the sample, affecting the accuracy of model training.

To address the aforementioned issues, we preprocess the trajectory data, including eliminating duplicates and completing missing data. The fixed time interval between adjacent trajectory points in each trajectory is 10 seconds.

Two different datasets, Traj-60 and Traj-120, were constructed in this study. The Traj-60 dataset consists of trajectories with no fewer than 60 points each, while the Traj-120 dataset consists of trajectories with no fewer than 120 points each. The Traj-60 dataset contains 2000 trajectories, and the Traj-120 dataset contains 4000 trajectories. In the experiments, the dataset was partitioned into training, validation, and testing sets following a 7:2:1 ratio. The training set was utilized for optimizing the trajectory prediction model's hyperparameters, while the validation played a crucial role in monitoring the convergence of the model during the training process. Subsequently, the testing set was employed to evaluate the accuracy of the trajectory predictions generated by the trained model.

B. IMPLEMENTATION AND EVALUATION METRICS

This experiment employed batch training, with each batch containing 64 data samples. Each data sample consists of multiple aircraft trajectories. During training, the input observation length for each trajectory was set to 10, and the prediction length was set to 60. The model was trained for 1500 iterations, with a model dimension of 512 and 8 attention mechanism heads. The Adam optimization function was used. The training and testing of this experiment were conducted on the Ubuntu 18.04 operating system. The GPU used was the NVIDIA GeForce RTX 3090, and the CPU was the Intel(R) Xeon(R) Platinum 8358P CPU @ 2.60GHz. The experiment utilized Python (version 3.8) and the PyTorch deep learning framework (version 1.11.0), with CUDA (version 11.6).

The evaluation metrics used to assess the accuracy of the model's predictions, as described in reference [29], include the following:

1) Average Displacement Error (ADE): It measures the average Euclidean distance between each predicted trajectory coordinate point and the corresponding ground truth trajectory coordinate point. The calculation formula is:

$$ADE = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{t_f} \sum_{t=1}^{t_f} |\mathbf{p}_{i_p}^{t} - \mathbf{p}_{i_{GT}}^{t}| \tag{8}$$

2) Final Displacement Error (FDE): It represents the Euclidean distance between the last predicted trajectory coordinate point and the last ground truth trajectory coordinate point. The calculation formula is:

$$FDE = \frac{1}{n} \sum_{i=1}^{n} |\mathbf{p}_{i_p}^{t_f} - \mathbf{p}_{i_{GT}}^{t_f}| \tag{9}$$

3) Maximum Displacement Error (MDE): It measures the maximum Euclidean distance difference between all predicted trajectory coordinate points and the corresponding ground truth trajectory coordinate points. The calculation formula is:

$$MDE = \frac{1}{n} \sum_{i=1}^{n} \max_{t=1,2,\ldots,t_f} |\mathbf{p}_{i_p}^{t} - \mathbf{p}_{i_{GT}}^{t}| \tag{10}$$

In equations (8)-(10), t_f represents the prediction time length, and $p_{i_p}^{t_f}$ and $p_{i_{GT}}^{t_f}$ represent the predicted and ground truth positions, respectively, at the final time step t_f for the i-th trajectory.

C. NUMBER OF LAYERS IN ENCODER AND DECODER

Both the encoder and decoder are important components of the TET model and hold equal importance in the model. Therefore, the encoder and decoder were configured with an equal number of layers. To improve the model's performance on the dataset, this study tested the number of layers N_x for both the encoder and decoder while keeping other conditions consistent. The results of the test are presented in Table. 1. and Fig. 2.

TABLE 1. Prediction errors of different N_x on Traj-60 dataset.

N_x	ADE	FDE	MDE
2	0.021959	0.019042	0.046227
4	0.021638	0.018963	0.045963
6	0.020170	0.018570	0.043912
8	0.022368	0.020937	0.048923
10	0.022570	0.021674	0.049046

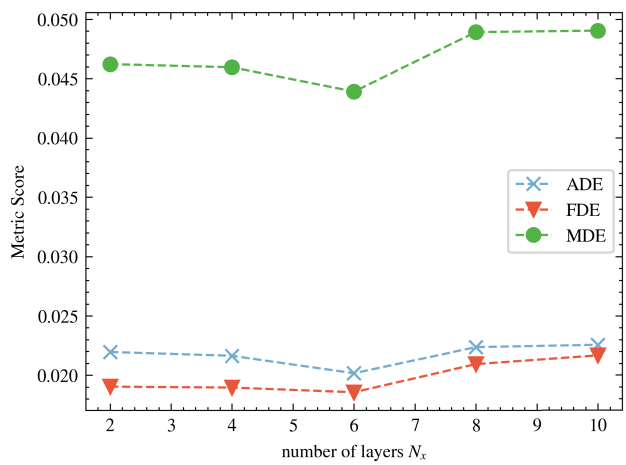


FIGURE 2. Prediction errors of different N_x on Traj-60.

From the results in Fig. 2, it can be observed that initially, as the number of layers N_x increases, the ADE, FDE, and MDE metrics decrease, indicating that increasing the number of layers in the encoder and decoder can lead to better model fitting. When N_x is equal to 6, the ADE, FDE, and MDE values are optimal. If N_x continues to increase, the computational complexity increases, and the performance metrics show an upward trend, indicating larger errors between the predicted and ground truth trajectory points. Therefore, the TET model's N_x is set to 6.

D. RESULTS AND ANALYSIS

To assess the efficacy of the TET model on long-term trajectories and compare it with other existing aircraft

TABLE 2. Prediction errors of different models on Traj-120 dataset.

Models	ADE	FDE	MDE
BP	0.543972	0.728848	0.728850
LSTM	0.020885	0.038698	0.039957
GRU	0.032141	0.061959	0.062385
TrajAirNet	0.010171	0.017699	0.019122
TET	0.009340	0.008602	0.021769

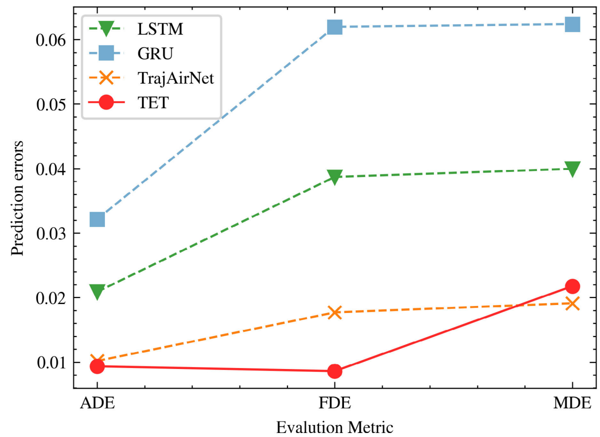


FIGURE 3. Prediction errors of different models.

TABLE 3. Time comparison of different models on Traj-120 dataset.

Models	Training time/h	Inference time/ms
BP	0.91	0.24
LSTM	1.39	0.69
GRU	1.18	0.14
TrajAirNet	23.61	2.41
TET	54.18	2.82

prediction methods, the prediction errors were calculated for all trajectories in the test set. The BP network utilized the method described in [30] and consisted of a single hidden layer. The LSTM network employed the method described in reference [9] with a single layer of 64 neurons in the hidden layer and a learning rate of 0.001. The GRU network followed the method described in [16], with the other parameters kept consistent with LSTM. TrajAirNet [31] served as the benchmark method for long-term trajectory prediction. It combined temporal convolutional networks, graph attention networks, and conditional variational autoencoders. Weather factors were not included in this experiment, and the learning rate was set to 0.001. The Traj-120 dataset was used in this experiment. The prediction errors obtained by training and testing the aforementioned models on the Traj-120 dataset are shown in Table. 2. and Fig. 3.

From the results in the table, it can be observed that compared to the LSTM and GRU models, the TET model achieves higher prediction accuracy. This is because the TET series of neural network models employ attention

TABLE 4. Comparison of models under different prediction lengths.

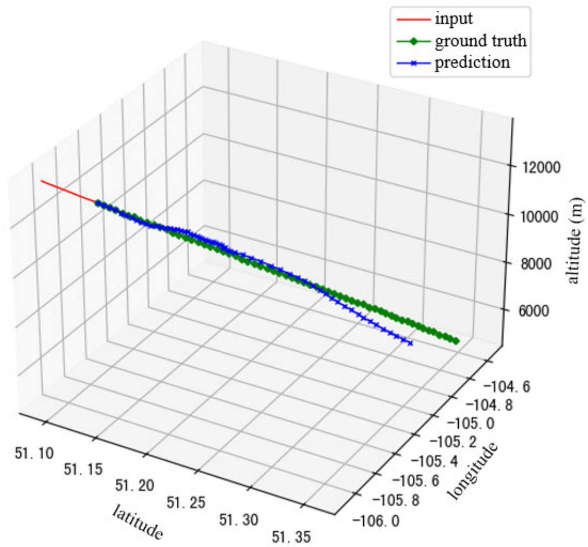
Models	Metric	70	80	90	100	110	120
TrajAirNet	ADE	0.015942	0.025451	0.033419	0.022591	0.025451	0.043752
	FDE	0.021831	0.054216	0.062722	0.047959	0.055326	0.095943
	MDE	0.027405	0.055326	0.066017	0.048839	0.054216	0.097671
TET	ADE	0.011207	0.012995	0.014838	0.016503	0.018073	0.020413
	FDE	0.011516	0.015694	0.023404	0.025936	0.032049	0.040755
	MDE	0.025910	0.030348	0.034474	0.038443	0.042552	0.048212

mechanisms instead of RNN types, allowing for holistic learning of position-encoded trajectory information without the issue of losing historical information. This results in a stronger ability to learn features from long time series data. Compared to the benchmark model TrajAirNet for long-term trajectory prediction, the TET model performs slightly worse in the MDE metric, indicating some individual trajectory prediction points with larger errors. However, it reduces the ADE and FDE metrics by 8.2% and 51.4%, respectively. This demonstrates that the proposed TET model better fits the overall distribution of trajectory data and exhibits robustness in long-term trajectory prediction.

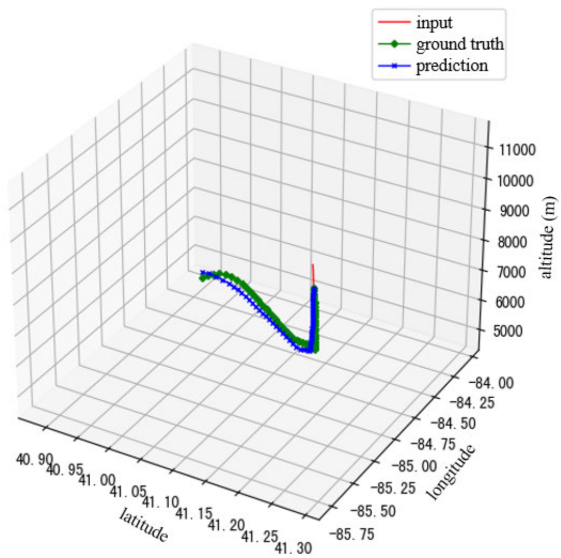
Examples of trajectory predictions by TET are shown in Fig. 4. The total number of input trajectory points is 10, and the total number of output trajectory points is 60. Fig. 4(a) and 4(b) depict the comparison between the predicted and ground truth results for two randomly selected data trajectories from the Traj-120 test dataset. In the early stages, Fig. 4(a) exhibit good prediction performance, but deviations can be observed in the later predictions. Fig. 4(b) captures the overall trend of the trajectory, but there are still some discrepancies in the later predictions. Overall, the proposed model can capture the data distribution for simple trajectory variations. However, as the time steps increase, the deviation between the predicted and ground truth trajectories gradually increases, indicating that there is still room for improvement in long-term trajectory prediction for the model.

To test the practicality of the TET model, this study conducted experiments to compare the time consumption of different trajectory prediction models on the Traj-120 dataset. The observation length was set to 10, and the prediction length was set to 60. The training time refers to the duration of model iteration training, while the prediction time refers to the average time taken to compute predictions for a set of test data after loading the model. The time comparison results for different trajectory prediction models are presented in Table. 3.

From the results in the table, it can be observed that the BP model requires the least training and prediction times compared to the other models. This is because the BP model has fewer parameters and a simpler network structure. Following the BP model, the LSTM and GRU models have similar training and prediction times. However, the GRU model demonstrates slightly shorter times compared



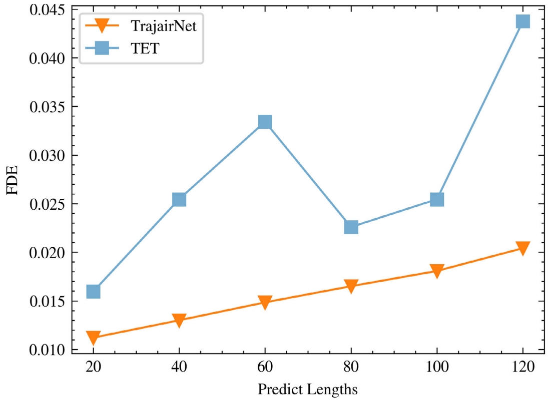
(a) Sample Trajectory from Traj-120 Dataset, ID: 124.



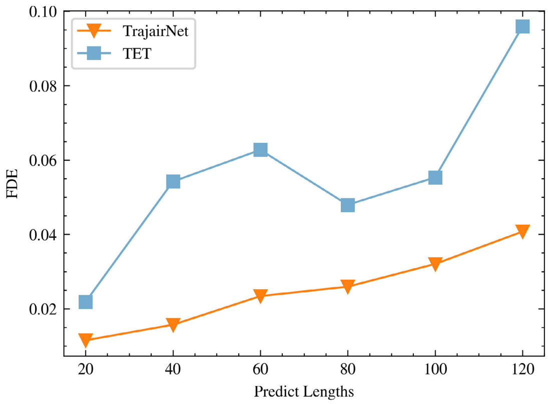
(b) Sample Trajectory from Traj-60 Dataset, ID: 218.

FIGURE 4. Example graph predicted by the TET model(The red line, green line and blue line indicate the observed trajectory points, the real trajectory points and the predicted trajectory points respectively).

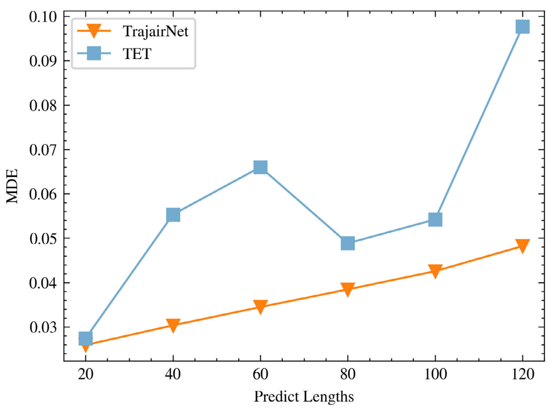
to LSTM due to its simpler network structure. In comparison to TrajAirNet, the training time for TET is longer. This is due to the larger number of computational parameters in



(a) ADE



(b) FDE



(c) MDE

FIGURE 5. Prediction errors of the model under different prediction lengths.

TET and the longer path length from input to output for the same set of data. The prediction times for both models are similar and fall within a reasonable range. Overall, the TET model ensures efficient prediction, meeting the requirements of practical engineering applications.

Furthermore, this study conducted experiments to compare the model accuracy of TrajAirNet and TET at different prediction lengths. Different prediction lengths were set during model training, while other conditions remained consistent. The observation length was fixed at 10, and the minimum prediction length was set to 60, while the maximum prediction length was set to 120. The prediction length was increased by 10 units each time, and the performance metrics of the models were evaluated on the test set. The experimental results are presented in Table. 4, and a visualization of the experimental results is shown in Fig. 5.

From the experimental results, it can be observed that the TET model achieves higher prediction accuracy compared to the LSTM and GRU models. The TET series of neural network models employ attention mechanisms instead of RNN types. This allows for holistic learning of position-encoded trajectory information without the problem of losing historical information. This results in a stronger ability to learn features from long time series data. Compared to the benchmark model TrajAirNet for long-term trajectory prediction, the TET model performs slightly worse in the MDE metric, indicating that there are some individual trajectory prediction points with larger errors. However, it reduces the ADE and FDE metrics by 8.2% and 51.4%, respectively. This demonstrates that the proposed TET model better fits the overall distribution of trajectory data and exhibits robustness in long-term trajectory prediction.

V. CONCLUSION

Aiming at the problems of memory loss and gradient vanishing that occur in RNN-based models when dealing with long time series data. This paper propose a Transformer-based long-term trajectory prediction model named TET. Through experimental comparisons and performance evaluations, we validate the accuracy, effectiveness, and practical applicability of TET. Compared to the benchmark method TrajAirNet, the TET model demonstrates higher accuracy and robustness for longer prediction lengths. However, due to the large number of model parameters, TET has a drawback on training time. Therefore, future work will focus on optimizing the trajectory prediction model to reduce training time and improve training efficiency.

REFERENCES

[1] K. Leiden, A. Fernandes, and S. Atkins, "Managing aircraft by trajectory: Literature review and lessons learned," in *Proc. Integr. Commun., Navigat., Surveill. Conf. (ICNS)*, Apr. 2018, pp. 3E1-1–3E1-16.

[2] X. Guan, X. Zhang, D. Han, Y. Zhu, J. Lv, and J. Su, "A strategic flight conflict avoidance approach based on a memetic algorithm," *Chin. J. Aeronaut.*, vol. 27, no. 1, pp. 93–101, Feb. 2014.

[3] R. A. Zachariah, S. Sharma, and V. Kumar, "Systematic review of passenger demand forecasting in aviation industry," *Multimedia Tools Appl.*, vol. 82, no. 30, pp. 46483–46519, Dec. 2023.

[4] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Phys. D, Nonlinear Phenomena*, vol. 404, Mar. 2020, Art. no. 132306.

[5] Z. Shi, M. Xu, Q. Pan, B. Yan, and H. Zhang, "LSTM-based flight trajectory prediction," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.

[6] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks," in *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2017, pp. 1597–1600.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6000–6010.

[8] Z. Zhang, R. Yang, and Y. Fang, "LSTM network based on on antlion optimization and its application in flight trajectory prediction," in *Proc. 2nd IEEE Adv. Inf. Manag.,Communicates,Electron. Autom. Control Conf. (IMCEC)*, May 2018, pp. 1658–1662.
[9] P. Han, J. Yue, C. Fang, Q. Shi, and J. Yang, "Short-term 4D trajectory prediction based on LSTM neural network," in *Proc. 2nd Target Recognit. Artif. Intell. Summit Forum*, Jan. 2020, pp. 146–153.
[10] X. Zhang and S. Mahadevan, "Bayesian neural networks for flight trajectory prediction and safety assessment," *Decis. Support Syst.*, vol. 131, Apr. 2020, Art. no. 113246.
[11] Z. Zhao, W. Zeng, Z. Quan, M. Chen, and Z. Yang, "Aircraft trajectory prediction using deep long short-term memory networks," in *Proc. CICTP*, Jul. 2019, pp. 124–135.
[12] Y. Pang, N. Xu, and Y. Liu, "Aircraft trajectory prediction using LSTM neural network with embedded convolutional layer," in *Proc. Annu. Conf. PHM Soc.*, vol. 11, 2019, pp. 1–8.
[13] Q. Shi, J. Yue, P. Han, and W. Wang, "Short-term flight trajectory prediction based on LSTM-ARIMA model," *J. Signal Process.*, vol. 35, no. 12, pp. 2000–2009, 2019.
[14] Q. Shi, W. Wang, and P. Han, "Short-term 4D trajectory prediction algorithm based on online-updating LSTM network. J," *Signal Process*, vol. 37, pp. 66–74, Jan. 2021.
[15] Y. Liu and M. Hansen, "Predicting aircraft trajectories: A deep generative convolutional recurrent neural networks approach," 2018, *arXiv:1812.11670*.
[16] Z. Hongpeng, H. Changqiang, X. Yongbo, and T. Shangqin, "Real-time prediction of air combat flight trajectory using gru," *Syst. Eng. Electron.*, vol. 42, no. 11, pp. 2546–2552, 2020.
[17] A. Postnikov, A. Gamayunov, and G. Ferrer, "Transformer based trajectory prediction," 2021, *arXiv:2112.04350*.
[18] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, Sep. 2021.
[19] A. Zhang, B. Zhang, W. Bi, and Z. Mao, "Attention based trajectory prediction method under the air combat environment," *Int. J. Speech Technol.*, vol. 52, no. 15, pp. 17341–17355, Dec. 2022.
[20] P. Jia, H. Chen, L. Zhang, and D. Han, "Attention-LSTM based prediction model for aircraft 4-D trajectory," *Sci. Rep.*, vol. 12, no. 1, Sep. 2022, Art. no. 15533.
[21] N. Schimpf, E. J. Knoblock, Z. Wang, R. D. Apaza, and H. Li, "Flight trajectory prediction based on Hybrid- recurrent networks," in *Proc. IEEE Cognit. Commun. Aerosp. Appl. Workshop (CCAAW)*, Jun. 2021, pp. 1–6.
[22] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. Int. Conf. Mach. Learn.,Int. Conf. Mach. Learn.*, May 2017, pp. 1243–1252.
[23] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 3104–3112.
[24] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," 2021, *arXiv:2106.13008*.
[25] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, Sep. 2022, pp. 11106–11115.
[26] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic, and M. Wilhelm, "Bringing up OpenSky: A large-scale ADS-B sensor network for research," in *Proc. IPSN 13th Int. Symp. Inf. Process. Sensor Netw.*, Apr. 2014, pp. 83–94.
[27] R. Salcido, A. Kendall, and Y. Zhao, "Analysis of automatic dependent surveillance-broadcast data," in *Proc. AAAI Fall Symp. Ser.*, 2017, pp. 225–230.
[28] K. Patroumpas, N. Pelekis, and Y. Theodoridis, "On-the-fly mobility event detection over aircraft trajectories," in *Proc. 26th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2018, pp. 259–268.
[29] Y. Chen, Q. Tong, T. Chen, S. Hou, and X. LIU, "Short-term trajectory prediction model of aircraft based on attention mechanism and generative adversarial network," *J. Comput. Appl.*, vol. 42, no. 10, p. 3292, 2022.
[30] K. Qian, Y. Zhou, L. Yang, R.-P. Xie, and X.-D. He, "Aircraft target track prediction model based on bp neural network," *Command Inf. Syst. Technol.*, vol. 8, no. 3, pp. 54–58, 2017.
[31] J. Patrikar, B. Moon, J. Oh, and S. Scherer, "Predicting like a pilot: Dataset and method to predict socially-aware aircraft trajectories in non-towered terminal airspace," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 2525–2531.

QIANG TONG received the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, in 2012. Since August 2018, he has been a Lecturer with the Computer School, Beijing Information Science and Technology University, China. His research interests include image recognition, computer vision, and machine learning.

JINQING HU received the B.S. degree from the Tianjin University of Science and Technology, Tianjin, China, in 2020. He is currently pursuing the master's degree with the School of Computer Science, Beijing Information Science and Technology University. His interests include machine learning and time series prediction.

YULI CHEN received the M.S. degree from Beijing Information Science and Technology University, Beijing, China, in 2022. He is currently pursuing the Ph.D. degree with the State Key Laboratory of networking and switching technology, Beijing University of Posts and Telecommunications. His research interests include time series prediction and natural language processing.

DONGDONG GUO received the Ph.D. degree from the School of Artificial Intelligence, Beijing Normal University, in 2022. Since September 2022, he has been a Lecturer with the Computer School, Beijing Information Science and Technology University, China. His research interests include natural language processing and computational linguistics.

XIULEI LIU received the Ph.D. degree in computer science from the Beijing University of Posts and Telecommunications, in March 2013. Since January 2022, he has been a Professor with the Computer School, Beijing Information Science and Technology University, China. He was a Visiting Ph.D. Student with CCSR, University of Surrey, from October 2008 to October 2010. His research interests include semantic sensor, semantic web, knowledge graph, and semantic information retrieval.