REAL-TIME RADAR TRACKING SYSTEM WITH DEEP LEARNING

DERİN ÖĞRENME İLE GERÇEK ZAMANLI RADAR TAKİP SİSTEMİ

MUHAMMED EMİR ÇAKICI

PROF. DR. SUAT ÖZDEMİR
Supervisor

Submitted to
Graduate School of Science and Engineering of Hacettepe University
as a Partial Fulfillment to the Requirements
for the Award of the Degree of Master of Science
in Computer Engineering

January 2023

ABSTRACT

REAL-TIME RADAR TRACKING SYSTEM WITH DEEP LEARNING

Muhammed Emir ÇAKICI
Master of Science, Computer Engineering
Supervisor: Prof. Dr. SUAT ÖZDEMİR
January 2023, 72 pages

Real-time data tracking plays an essential role in the flight test processes of an aircraft. The data flowing from the aircraft to the ground control center must be real-time and uninterrupted. However, sometimes ground control systems can cause disconnection with the aircraft, making it difficult to track them. This thesis firstly gives a brief survey of real-time aircraft tracking systems and then proposes a deep learning-based, real-time 3D prediction of the next location named DeepAT for uninterrupted real-time data tracking. Our DeepAT model uses an Encoder-Decoder GRU model to predict the next location of the aircraft. Thus, in case of any disconnection, the tracking of the aircraft can be sustainable. In the experiments, real flight test sensor data collected with the telemetry system are used. Experimental analyzes are performed for two structurally different aircraft, one of which is a highly maneuverable fixed-wing propeller aircraft and the other an Unmanned Air Vehicle (UAV). The efficiency and superiority of the proposed method is demonstrated by comparing it with state-of-the-art methods in terms of Mean Absolute Error (MAE) and Mean Squared Error (MSE) metrics. The results show that our proposed method outperforms the state-of-the-art and gives better prediction of the next location of aircraft.

Keywords: deep learning, GRU, Encoder-Decoder, aircraft, tracking system, UAV

ÖZET

DERİN ÖĞRENME İLE GERÇEK ZAMANLI RADAR TAKİP SİSTEMİ

Muhammed Emir ÇAKICI
Yüksek Lisans, Bilgisayar Mühendisliği
Danışman: Prof. Dr. SUAT ÖZDEMİR
Ocak 2023, 72 sayfa

Gerçek zamanlı veri takibi, hava araçlarının uçuş testi süreçlerinde önemli bir rol oynar. Uçaktan yer kontrol merkezine akan verinin gerçek zamanlı ve kesintisiz olması gerekmektedir. Ancak yer kontrol sistemlerinden ya da hava aracından kaynaklı kopukluklar yaşanabilir. Bu tez, öncelikle gerçek zamanlı hava aracı takip sistemleri hakkında yapılan çalışmalar ile ilgli kısa bir inceleme sunar ve ardından kesintisiz gerçek zamanlı veri takibi için, DeepAT adlı bir sonraki konumun derin öğrenmeye dayalı, gerçek zamanlı 3 Boyutlu tahminini yapan bir model önerir. Veri seti olarak telemetri sistemi ile toplanan gerçek hava aracı sensör verilerini kullandık. Biri yüksek manevra kabiliyetine sahip sabit kanatlı pervaneli uçak ve diğeri İnsansız Hava Aracı (İHA) olmak üzere yapısal olarak çok farklı iki uçak için ayrı testler yapıldı. Önerilen yöntem, zaman tabanlı tahmin alanında çok kullanılan yöntemlerle karşılaştırılmıştır. Bu karşılaştırmalarda değerlendirme metrikleri olarak Mean Absolute. Error (MAE) ve Mean Squared Error (MSE) kullanılmıştır. Sonuçlar, önerdiğimiz yöntemin son teknoloji yöntemlerden daha iyi performans ve daha iyi tahmin sonuçları verdiğini göstermektedir.

Keywords: derin öğrenme, GRU, Encoder-Decoder, hava aracı, takip sistemi, İHA

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Prof. Dr. Suat ÖZDEMİR, for his constant guidance and support throughout this journey. His invaluable insights, encouragement, and patience have been instrumental in the completion of this thesis. I would also like to thank my mentor Asst. Prof. Feyza YILDIRIM OKAY, who was in constant contact with me during my thesis process and helped me complete this journey with her vast knowledge. I am also grateful to Assoc. Prof. Adnan ÖZSOY, Assoc. Prof. Ali Seydi KEÇELİ, Assoc. Prof. Mehmet DEMİRCİ and Asst. Prof. Yılmaz AR for their valuable feedback and support. I am deeply grateful to my study participant Burakhan ÇORAK for his time and willingness to participate in this research.

I would also like to thank Turkish Aerospace for providing the resources and cooperation and my superiors Mahmut Kürşad ARPACIOĞLU and Onur YÖRÜK for the support needed to complete this research.

I would like to extend my heartfelt thanks to my family and friends for their unwavering love and support throughout this journey. Their encouragement and belief in me kept me motivated and on track, even during the toughest times. My parents Taner ÇAKICI and Sevilay ÇAKICI, have always been my role models and their constant love and support have been a source of strength throughout my life. I would also like to thank my sister Sümeyye Ceren ÇAKICI, for her understanding and support. They have been a sounding board for my ideas and have always been there to listen and offer valuable advice. I am forever grateful to each and every one of you.

Finally, I would like to express my deepest gratitude to my girlfriend, Süveydanas KOCACIK, for her unwavering love and support throughout this journey. Her encouragement, understanding, and patience have been invaluable in the completion of this thesis. She has been my rock, my confidant and my inspiration, and I couldn't have done this without her. I am forever grateful for her love, support and for being an integral part of my life.

CONTENTS

Page

TABLES

FIGURES

ABBREVIATIONS

3D	:	**3 D**imension
DeepAT	:	**Deep A**ircraft **T**racking
GRU	:	**G**ated **R**ecurrent **U**nit
UAV	:	**U**nmanned **A**ir **V**ehicle
MAE	:	**M**ean **A**bsolute **E**rror
MSE	:	**M**ean **S**quared **E**rror
RMSE	:	**R**oot **M**ean **S**quared **E**rror
MAPE	:	**M**ean **A**bsolute **S**quared **E**rror
İHA	:	**İ**nsansız **H**ava **A**racı
LSTM	:	**L**ong **S**hort **T**erm **M**emory
RNN	:	**R**ecurrent **N**eural **N**etwork
NLP	:	**N**atural **L**anguage **P**rocessing
ADS-B	:	**A**utomatic **D**ependent **S**urveillance–**B**roadcast
IQR	:	**I**nter**Q**uartile **R**ange
StEncDecLSTM	:	**St**acked **Enc**oder **Dec**ecoder **LSTM**
EncDecGRU	:	**Enc**oder **Dec**ecoder **GRU**
EncDecLSTM	:	**Enc**oder **Dec**ecoder **LSTM**

1. INTRODUCTION

Aircraft Tracking Systems are used in different areas in the aviation industry such as military, civil, or development phase of aircraft. It is important to be informed about the current position and the trajectory of an aircraft to predict and control the traffic in airports. This is why many types of Aircraft Tracking Systems are used in airports. As in civil use cases, it may be crucial to know the current position and the trajectory of an ally or enemy aircraft. Knowing the current position and trajectory gives superiority to the user. In aircraft development phases, knowing the place of an aircraft is mandatory. After the initial prototype is produced, the flight test phases begin immediately to determine the boundaries of the prototype using a variety of sensors deployed at various locations by Flight Test Instrumentation teams [1]. During the testing phase, telemetry devices are utilized to track the position of the aircraft and to receive data from it. Following the airplane in all conditions allows the user to do their duties more effectively and with fewer errors.

Real-time aircraft tracking is a long-studied topic that has not lost its appeal. It still needs to improve as the aircraft evolves. There are different types of use cases for real-time aircraft tracking methods. One of its most popular uses is as an air traffic management system in airport traffic control systems. To avoid any negative consequences, real-time tracking of aircraft is vital. Among traditional methods, real-time tracking of an aircraft is generally sufficient to determine where the aircraft is. However, ongoing research is being conducted to take aircraft tracking to a higher phase. The next phase includes not only tracking the aircraft but also predicting the next location of the aircraft. In the past, mathematical methods have been used to estimate the position of the aircraft [2, 3]. With the enhancements in machine learning [4] and deep learning [5] methods, however, mathematical methods also lost their validity and became a supportive position alongside machine learning and deep learning methods. Especially for real-time prediction problems, deep learning-based methods have gained too much attraction with their accurate results [6].

Flight test processes are another area where real-time tracking of the aircraft is mandatory.

In flight test processes, data is landed from the air to the ground in real-time by telemetry systems [7]. The ground-based telemetry antenna tracks the aircraft in the air to collect all data from the aircraft. Data is collected via a telemetry antenna and transmitted to a real-time telemetry visualization system for processing and visualization. In this process, it is critical to obtain all data from aircraft, so losing aircraft during flight tests can be a challenging issue for test phases. According to the size of the loss, either the flight test or the maneuver of the aircraft can be restarted, which costs organizations both time and money.

1.1. Scope of the Thesis

In this thesis, we aim to develop a novel deep learning-based system called DeepAT. This system predicts the next 3D position and altitude on highly maneuverable, fixed-wing propeller aircraft and UAVs and then decides the trajectory accordingly in real-time. Thus, possible data loss that may arise from problems such as the aircraft being out of sight of the antenna during the test is minimized and the aircraft becomes traceable throughout all flights. Data loss may occur when the aircraft moves faster than the antenna rotation angle or with the corruption of the data reflected from surrounding buildings when flying too close to the runway. In our previous work, we introduced DeepAT model [8] by focusing on its advantages of real-time tracking systems. In this thesis, the proposed model is extended by implementing two different aircraft data sets including a propeller fixed-winged aircraft and an Unmanned Air Vehicle (UAV). The success of the proposed model is demonstrated by comparing it with five different deep learning methods.

1.2. Contributions

Other contributions of the thesis can be listed as follows.

- To the best of our knowledge, this is the first real-time 3D next position prediction system of a highly maneuverable aircraft.

- We propose a new approach for aircraft tracking systems using the stacked Encoder-Decoder GRU.

1.3. Organization

The organization of the thesis is as follows:

- Chapter 1 presents the motivation, contributions, and scope of the thesis.
- Chapter 2 gives brief background information.
- Chapter 3 provides detailed information about aircraft tracking systems and their proposed methods.
- Chapter 4 proposes DeepAT method for aircraft tracking systems is explained.
- Chapter 5 performs experimental analysis and the results of the proposed DeepAT model are discussed.
- Chapter 6 concludes the thesis and presents possible future directions.

2. BACKGROUND INFORMATION

2.1. Long Short Term Memory (LSTM)

Recurrent Neural Network (RNN) is a neural network that has short-term memory, allowing it to process time-series problems better. Even though RNN had great success with the prediction of short-term time-series problems, it has a problem with long-term dependency problems with long series [9]. LSTM has emerged to solve the RNN's long-series dependency problem. LSTM is a type of RNN [10] that learns long-term dependencies with an innovative memory. Furthermore, LSTM is a powerful technique that specifically addresses the time-series estimation problem [11]. When compared to RNN, the main strength of LSTM is the gating mechanism. It also has memory blocks that have gates: the input gate to update the current memory with the new memory state, the forget gate to forget the previous memory state, and the output gate to handle the current memory state and output [12]. Cell state is the key to LSTM. In forget gate, LSTM determines what kind of information and how much of the information will be discarded. In order to consolidate the data, a forget gate combines h_{t-1} and X_t into a single vector, normalizing the values from 0 to 1, where 1 is total pass and 0 is total discard using the sigmoid activation function. After discarding with forget gate, the input gate gives new information to the cell state. In the input gate, tanh function represents the current information and sigmoid function is used to catch the necessary information and discard the useless information. The output of the sigmoid function multiplies with tanh function output and is added to the current cell state. Lastly, the output gate determines which information is going to be the output, which filters the data effectively and prevents pointless calculations. The structure of LSTM can be seen in Fig. 2.1. The structure of the LSTM model gates can be formulated as [13]:

$$i_t = \sigma_g(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{1}$$

$$f_t = \sigma_g(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{2}$$

$$o_t = \sigma_g(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{3}$$

$$\widetilde{c_t} = \tanh(W_{xc}x_t + W_{hc}h_{t-1}] + b_c) \tag{4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c_t} \tag{5}$$

$$h_t = o_t \odot \tanh(c_t) \tag{6}$$

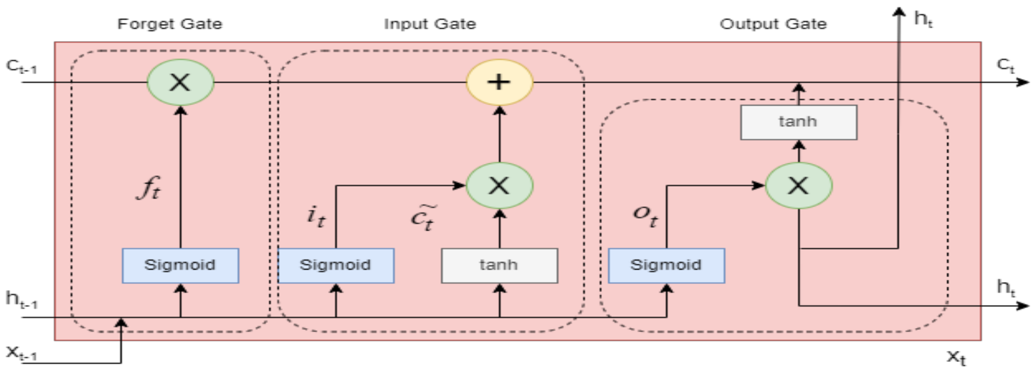

Figure 2.1 LSTM Structure

where i_t, f_t, and o_t are input, forget, and output gates, respectively. σ is the sigmoid function. W_i, W_f, W_o, W_c represent the weight matrices used in the gates and block input, and b_i, b_f, b_o, b_c represent the bias vectors. x_t is the current time. Also, c_t is the state of memory at time t, $\widetilde{c_t}$ is the candidate memory state at time t, h_{t-1} is the output of the previous memory in time $t-1$, and h_t is the output in time t.

2.2. Gated Recurrent Unit (GRU)

GRU is proposed by Chung et al. [14] to allow each recurring unit to adaptively capture the dependencies of different timespans. When compared to LSTM, GRU has a simpler structure and has a similar performance to LSTM. Hence, GRU can be thought of as a simpler version of LSTM [14]. Instead of forget gate, input gate, and output gate, in GRU networks, there is an update gate that allows the current hidden state to be updated with the new hidden state and a reset gate that allows deleting the previous hidden status information. Unlike LSTM, in GRU structure, the input and forget gates are combined in the update gate. The update gate

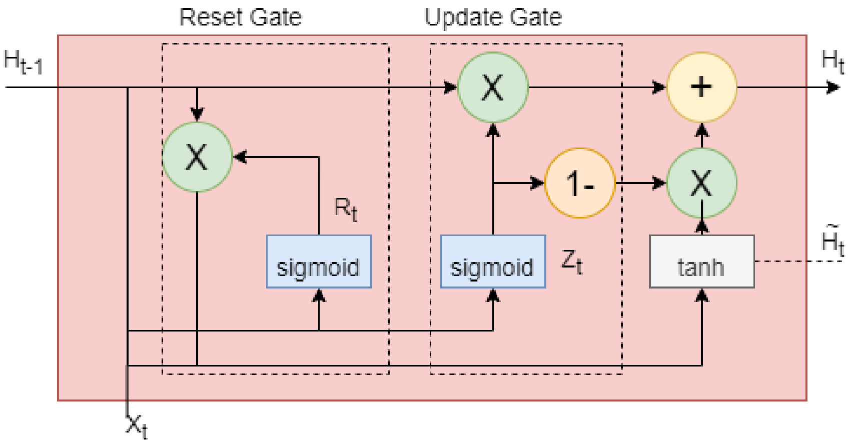

Figure 2.2 GRU Structure

can be described as the combination of the forget and input gates in LSTM. Like LSTM's forget and input gate, the main responsibility of the update gate is processing memory and information. More state information is fetched sooner when the value of the update gate is larger. The reset gate is used to regulate how much status information from the previous moment is discarded. The reset gate is discarded more often, the smaller its value. The advantage of GRU over LSTM is that GRU needs fewer parameters and fewer samples than LSTM. The structure of GRU can be seen in Fig.2.2 and the general structure of GRU models can be formulated as follows [13]:

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \tag{7}$$

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \tag{8}$$

$$\widetilde{h}_t = \tanh(W_{xh}x_t + W_{hh}(r_t \odot h_{t-1}) + b_h) \tag{9}$$

$$h_t = (z_t) \odot h_{t-1} + (1 - z_t) \odot \widetilde{h}_t \tag{10}$$

where z_t is the update gate and r_t is the reset gate which determines the amount of the historical information to return, respectively. $\widetilde{h}_t$ is the last information in the candidate hidden layer. h_{t-1} is the previous hidden state and h_t is the current hidden state. W_z, W_r, W_h represent weight matrices and b_u, b_r, b_h represent bias vectors. σ is the Sigmoid function.

2.3. Encoder-Decoder Neural Network

In sequence-to-sequence NLP applications like language translation, machine translation, and encoder-decoder models have shown cutting-edge outcomes [15]. Multivariate time-series data problems can also be seen as sequence-to-sequence learning problems. Hence encoder-decoder model can be used. Traditionally, RNN models are used to predict time series data. Simply by forcing the network to memorize the sequence, a well-tuned LSTM layer may cause the entire network to function correctly with the sequential knowledge in the data. Using encoder-decoder architecture with the time-series data is a wise choice due to its superior performance with sequential data. Encoder-Decoder model generally consists of three different parts [16]. The first part is the encoder part. The encoder part is an RNN that receives the input sequence data and summarizes the information gathered from the input sequence to internal state vectors. The second part is the feature vector. Feature vector in an intermediate state that is responsible for keeping the information of the input which is collected from the encoder. Moreover, it decides which information is useful and needs to stay and which information is useless and needs to be discarded. The last part is the decoder part. This part is an RNN whose initial states are initialized to the final states of the encoder RNN. With these initial states, the decoder part generates the output sequence. The general structure of Encoder-Decoder models can be seen in Fig. 2.3.

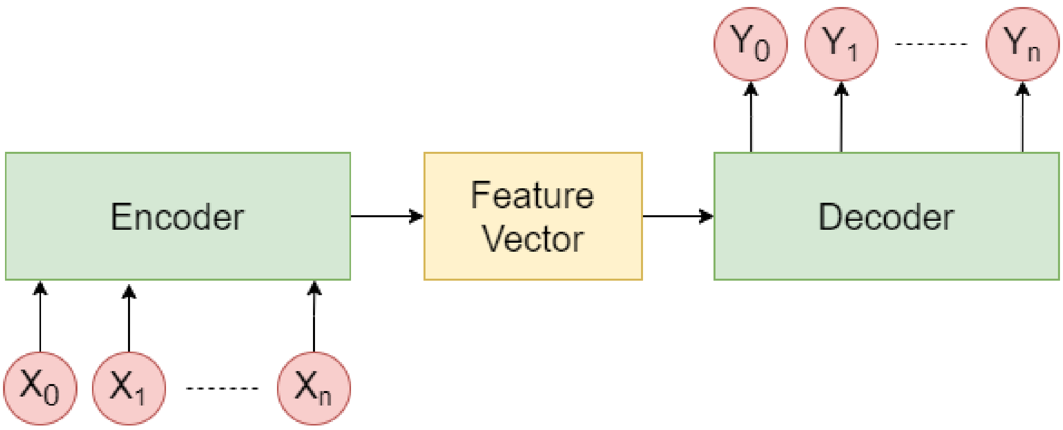

Figure 2.3 Encoder Decoder Neural Network

3. RELATED WORK

Real-time aircraft tracking systems become important with the new involvements in the aircraft industry. Especially, deep learning-based tracking systems have the capability of more accurate predictions to avoid data loss during the connection of air to the ground station. There are many studies in the literature focusing on aircraft tracking systems. These systems generally aim to provide traffic management using different approaches. In this section, we classify existing aircraft tracking systems into ML, DL, and mathematical models according to which approach is used. This section firstly classifies existing aircraft tracking systems into ML, DL and mathematical approaches summarized in Table 3.1 that focuses on the real-time deep learning-based models in aircraft as summarized and demonstrated in Table 3.2. The studies carried out were grouped under fixed-wing aircraft and UAV.

Different from current studies, the proposed DeepAT focuses on predicting two different aircraft that have different structural designs. These aircraft have different flexibility and different maneuvering capabilities. Apart from these studies, the UAV that DeepAT is trying to predict is a large aircraft with propellers, apart from the quadrotor architecture in the studies. At the same time, the other aircraft DeepAT are trying to predict is a fixed-wing propeller aircraft with high maneuverability and proven acrobatic capabilities.

3.1. Mathematical Approaches

Mathematical approaches have been used for a long time in trajectory prediction studies. However, with the development of machine learning and deep learning methods, mathematical models could not keep their old popularity and started to be used as an auxiliary factor in machine learning and deep learning methods. In most studies, the estimation is performed using Markov models. Although Markov models are generally good in short-term predictions, they perform poorly in long-term predictions due to computational weight. Some studies combine Markov models with machine learning methods. Also, some studies focus on using Bayesian Filters. Although the use of Kalman filters seems to be used alone in relatively old studies, it has been used in machine learning and deep learning models in recent studies. However, Kalman filters do not provide successful results when uncertainties are high. For this reason, it may not be used for prediction in turboprop and jet-engine aircraft. Baek et al. [3] introduce an Interaction of Multiple models to predict conflict between aircraft and try to create a no-conflict zone. In the experiments, ADS-B data is used. They split the flight with the first-order Markov chain and estimate the position with Interaction Multiple Model. They divide the flight into four phases according to their horizontal and vertical movements. Dividing phase similar to clustering methods. According to the results, the model gives good results under divided phases. Wang et al. [2] propose a Bayesian updating model to predict possible accidents in National Airspace System and improve safety. In the experiments, flight track data from Sherlock Data Warehouse is used. They use a bayesian entropy network to reduce uncertainty, and then use Bayesian updating for trajectory estimation. According to the results, kinematic models can be used to predict trajectory. Bayesian updating gives good results in landing phases. Ayhan et al. [17] employ two different models to improve efficiency at ATM. They proposed HMM for flight data and Viterbi model for weather data. In the experiments, raw trajectory database and weather database are used. They split the area into cubes and calculate each one in their cube considering the weather conditions. When comparing the predicted trajectories with ground truth values, prediction results are successful. Banerjee et al. [18] propose a Bayesian filtering algorithm for UAV traffic management and generation of possible trajectories for UAVs. Flight UAV data is used. They generated the trajectory with non-uniform rational B-splines and use Kalman filters for prediction. According to the results, with non-uniform rational B-splines (NURBS), the trajectory can be generated without in-flight parameters. However, sharp turns that UAVs needed could not be generated with this algorithm.

3.2. Machine Learning Based Approaches

As in other fields, Machine Learning (ML) offers promising solutions to aircraft tracking systems. In clustering-based examples, the aircraft is included in a group according to its flight time, sequence duration, flight phases, or landing and take-off positions, and it is expected to converge to the behavior of that group. In these examples, k-means and k-nn methods are used for clustering. The data generated after clustering were used for estimation using both DL-based and ML-based methods. Leege et al. [19] propose a generalized linear model to reduce runway traffic and avoid conflict in aircraft landing processes. In the experiments, commercial Aircraft and meteorological data are used. The meteorological data and aircraft data are combined before the training phase by adding the wind parameters. According to the results, each aircraft created its own line for descending. Also, the wind has a strong effect on aircraft while descending. Another study on commercial aircraft data is performed by Fernandes et al. [20]. The authors propose a Hybrid clustering Model and a two-phased Hidden Markov Model (HMM) to help Air Traffic Management (ATM) and reduce complicity on ATM. The authors, first, cluster the trajectories and then use HMM to predict the future trajectories. According to the results, HMM can give satisfying outcomes when combined with clustering methods. Baratt et al. [4] employ an unsupervised K-means clustering algorithm and Gaussian Mixture Model (GMM) to solve the air traffic problem on airways in the situations of takeoffs and landings. In the experiments, the only position from ADS-B receivers information of the commercial aircraft is used. The reason for choosing only position data is to ensure that the model to be created would be used in a way that would appeal to the majority of aircraft. They, first, cluster the takeoff and landing situations, and then they use GMM to predict the next location of the aircraft. According to the results, clustering is very effective when trying to predict a maneuver of the aircraft. However, it can be hard to predict the whole flight with only clustering and Gaussian methods. Julio et al. [21] offer a Support Vector Regression model to estimate the trajectory of bus travel with in-time information about traffic conditions. In the experiments, the historical data and telemetry data obtained from the bus every 30 seconds are used. They made grid cells of the road and predict travel in each grid on their own. According to results without altitude value, machine learning models can predict trajectories of vehicles pretty well but artificial neural networks used in this project show that in trajectory prediction situations, Artificial Intelligence solutions work better than Machine Learning ones. Wang et al. [22] present a multi-cell neural network with clustering using DBSCAN to predict the trajectory of terminal maneuvering areas. In the experiments, ADS-B data taken from the airport is used. They split the model into two phases. In the first phase, they cluster with DBSCAN. Then, in the second phase, they use Multi-Cell Neural Network with the aim of predicting the trajectories. According to the results, clustering is the best way to predict trajectory in machine learning applications. On the other hand, in real-time applications, prediction can be hard with clustering.

Approach	Reference	Model	Purpose	Dataset	Key Finding	Real-time
Mathematical Model	Baek et al. [3]	Interacting Multiple Model	To predict conflict between aircraft and create a no-conflict zone	ADS-B data	- The flight is divided into four phases according to their horizontal and vertical movements. - Dividing phases by movements similar to clustering. - The model gives good results under divided phases.	-
	Wang et al. [2]	Bayesian Updating	For safety and predict possible accidents in National Airspace System	Flight track data from Sherlock data warehouse	- Kinematic models can be used to predict trajectory. - Bayesian updating gives good results on landing phases.	-
	Ayhan et al. [17]	Hidden Markov Model for flight data Viterbi for weather data	Efficiency at Air Traffic Management	Raw trajectory database Weather database	- When comparing the predicted trajectories with ground truth values viterbi and gives good results.	-
	Banerjee et al. [18]	Bayesian filtering algorithm	UAV Traffic management and generation of possible trajectories for UAVs	In flight UAV data	- With non-uniform rational B-splines, trajectory can be generated without in-flight parameters. - With this algorithm, sharp turns cannot be generated.	-
Machine Learning	Leege et al. [19]	Generalized Linear Models	Reduce runway traffic and avoid conflict of aircraft landing processes	Commercial Aircraft Meteorological data	- Each aircraft created its own line for descending. - The wind has a strong effect on aircraft while descending.	No
	Fernandes et al. [20]	Hybrid clustering HMM two-phase algorithm	Air Traffic Management	Commercial Aircraft data	- Hidden Markov Models can give satisfying outcomes when combining with clustering.	No
	Baratt et al. [4]	Unsupervised K-means clustering Gaussian Mixture Model	Air Traffic Management in takeoff and landing situations	Only position data from ADS-B receivers of Commercial aircraft	- Clustering is very effective when trying to predict a maneuver of the aircraft but it can be hard to predict the whole flight with only clustering and Gaussian methods	No
	Julio et al. [21]	Support Vector Regression	Bus travel trajectory estimation using real-time information about traffic conditions	Historical data and Telemetry data obtained from the bus every 30 second	- Without altitude value, machine learning models can predict the trajectories of vehicles pretty well. - But ANN model used in this project also shows that in trajectory prediction situations, AI solutions work better than ML ones.	Yes
	Wang et al. [22]	Clustering with DBSCAN Multi cell NN	4D trajectory prediction on Terminal Maneuvering Area	ADS-B data taken from Airport	- Clustering is the best way to predict the trajectory in machine learning applications. - But in real-time prediction clustering can be hard	No

Table 3.1 Aircraft Tracking Systems with Traditional Approaches

3.3. Deep Learning Based Approaches

Deep Learning (DL) algorithms are innovative and powerful approaches for prediction problems. Recently, the use of DL algorithms for aircraft tracking systems has increased since the structure of these algorithms is suitable for solving real-time trajectory problems.

3.3.1. Fixed-wing Aircraft

Shafienya et al. [23] proposed a hybrid deep learning approach to predict the trajectory in Airports. In the experiments, historical ADS-B data is used. Accordingly, they first used a hybrid CNN-GRU model to extract features of flight trajectories and after that, a 3D-CNN is used to predict the spatial-temporal features. They used a sliding window to split the data into trajectories. They used MAE and RMSE to measure the performance. According to experimental results, the proposed methods give much better results with low maneuver capacity commercial aircraft. Wu et al. [24] proposed a Back Propagation Neural Network to help Airport Traffic Management in real-time. They used commercial aircraft ADS-B data in experiments. Since commercial aircraft have limited maneuverability and are heavy, predictions are performed on small data. They employ the backpropagation neural network after using the K-means clustering approach first. According to the results, backpropagation methods efficiently support real-time trajectory estimation. Han et al. [25] proposed a hybrid model with K-means clustering for clustering the trajectories and GRU model to predict the trajectories to solve the aircraft traffic problem. They used 12 days worth of ADS-B data. They combine historical and real-time datasets to obtain similarities. After getting clusters from K-Means, they put historical data into historical GRU model and real-time data into online-updating GRU model. This gives them the advantage of predicting in real-time. Commercial aircraft generally have four different types of maneuvers in a flight including Takeoff, Ascending, Descending, and Landing. Due to their heaviness, they have limited maneuvering capacity. With that kind of aircraft, clustering is generally easy and very efficient. Ma et al. [26] proposed a hybrid CNN-LSTM model for Air Traffic Management. They used historical ADS-B data. They used CNN to extract the spatial features and then they used LSTM to predict the trajectory. They selected six prior samples to predict the next latitude, longitude, and altitude values. After getting the experimental results, they compare their CNN-LSTM network with vanilla LSTM and BP networks. They used RMSE, MAE, and MAPE as evaluation metrics. According to the results, the authors show that CNN-LSTM network tends to learn better and give better results. Zhao et al. [27] proposed a Deep LSTM model to solve the Air Traffic Management problem. They used real flight data from ADS-B. They discovered that the limitations of the LSTM cause it to lose its learning capacity while predicting multidimensional values. Therefore, they presented a Deep LSTM model to overcome this problem. They used MAE, MRE, and MSE to evaluate the performance. The proposed D-LSTM model has slightly better performance than existing methods when compared to LSTM, Elman Neural Network, and BP Neural Network. Zhang et al. [28] proposed a Bayesian neural network-based trajectory prediction model to help ease the safety concerns in civil aviation. They trained two different deep-learning models to predict short-term and long-term trajectories of aircraft. They used

DNN to predict short-term and LSTM to predict long-term trajectories. After training these two models, they combine them to get better results. Even though LSTM gives good results in long-term prediction, DNN gives better results in short term, so they used DNN to correct the prediction of LSTM in long term. They used RMSE and MAE as evaluation metrics. Experimental results show that the combined method gives good results in full trajectory prediction. Pang et al. [29] proposed a trajectory prediction method with consideration of the weather impact on the trajectory of aircraft. They focused on the effects of environmental changes. They used weather air traffic and weather data from Sherlock Warehouse. They used RNN and FCNN as their prediction models and CNN for feature extraction. As a result, the results show that the weather affects the flight trajectories, and adding weather parameters while doing trajectory prediction reduces the error rate of prediction. Shi et al. [5] proposed an LSTM network-based trajectory prediction method with the aim to create a safer air traffic system and oversee the potential dangers in flight routes. They used ADS-B data collected from the aircraft. They created an LSTM network with sliding windows to predict trajectories. The results show that RNN networks are good at predicting trajectories in real-time. However, LSTM has better accuracy due to its complex and gated network structure with more parameters. LSTM also has higher accuracy compared to Hidden Markov Models and Weighted Markov Models. Adding sliding windows gives the model the ability to track every phase of the flight. Cheng et al. [30] proposed a machine learning-aided trajectory prediction method with the aim to prevent conflict in Air Traffic. They proposed an LSTM model to estimate the trajectories of aircraft. Then, they fed this prediction to their conflict detection algorithm to discover in case of any conflict. They used ADS-B data. They compare their LSTM-based conflict detection method with CNN-based and LS-based conflict detection methods. They used RMSE to evaluate the prediction performance. They compared these three algorithms in three different time spans. They conducted the test with short-term, mid-term, and long-term trajectory prediction. According to the experimental results, LSTM-based trajectory prediction gives better results in three different time spans. Hashemi et al. [31] compared six different conventional and state-of-the-art models for aircraft trajectory prediction. In their work, they compared Logistic Regression, Support Vector Regression, DNN, CNN, RNN, and LSTM. After comparing, they tested the security of the models. They used traffic flow management system (TFMS) data. The results show that even deep learning-based models give better results, these models have low resiliency against adversarial attacks. Zhang et al. [32] proposed an Attention-based LSTM model for trajectory prediction with aim of solving air traffic management problems. Their proposed system deals with the trajectory prediction problem in three different phases which are climbing, cruising, and approaching phases. They are focusing on short-term trajectory prediction where the estimation period is nearly 10 minutes. They used ADS-B data, weather data, and flight profiles as their dataset. They applied a soft attention mechanism to LSTM so they can get higher accuracy scores. They used MAE as their evaluation metric. They compared their methods with Kalman Filters, Hidden Markov Model, LSTM, and S-LSTM. The result shows that adding attention to LSTM gives better results in three different phases. Liu et al. [33] proposed an Encoder-Decoder LSTM network model for predicting 4D trajectories to solve Air Traffic problem. They used the flight tracks dataset gathered from FAA Traffic Flow Management System. In their model, they first used encoder LSTM to create a fixed-size feature vector from flight plans. Then, they used decoder LSTM to map the fixed-size feature vector to the target trajectory. Lastly, they used a convolutional layer in decoder network to add weather-related data to trajectory prediction. The results show that their encoder-decoder network is good at predicting full path trajectories. Han et al. [34] proposed a cyclic GRU network for trajectory prediction with the aim of easing Air Traffic Control. They used ADS-B data. They divide datasets into historical and real-time data. They had 12 days worth of data in total and they used 11 days of data as history and 1 day as real-time. They trained GRU model with historical data and tested it on real-time data. They used RMSE as an evaluation metric. They compared their model with ARIMA, Holt-Winters, GRU, and LSTM models. According to the results, RMSE values of their method are much lower than other ones. Wang et al. [22] proposed a Multi-cells Neural Network model with the aim of predicting trajectory in Terminal Maneuvering Areas. They used ADS-B data. They used DBSCAN and PCA to preprocess the data so they can use it in MCNN model. They cluster the data with DBSCAN and fed these clusters to MCNN model. They cluster data into five different clusters. The training phase worked on each different cluster. They fed the trajectory data in this phase of the model. While predicting trajectory, their main focus is to estimate the time of arrival of aircraft to airports. They used K-fold cross-validation method. They used Multiple Linear Regression (MLR) to compare their methods. They used MAE and MRSE as their evaluation metrics. The results show that their method has higher performance results than MLR but MAE and RMSE values differ in each different cluster. You et al. [35] proposed a sequence-to-sequence Encoder-Decoder GRU-RNN model to predict the trajectory of ships. They use AIS data as their dataset. They first preprocess raw data for trajectory segmentation. They used Encoder-Decoder model for prediction. Encoder part is a single-layer GRU model. After encoding, Decoder part which is also a single-layer GRU gets the data and creates output. They used RMSE as their evaluation metric. They used Adam optimizer for hyperparameter tuning. They compare their Seq2Seq model with LSTM and GRU models. The results show that Seq2Seq model has a significant impact on trajectory prediction.

3.3.2. UAV

Yang et. al [36] proposed a bidirectional GRU for real-time trajectory prediction of quad rotors. They used historical data on ten different types of quad rotors. They compared their bidirectional GRU model with GRU in terms of loss and evaluation time. They used MAE, RMSE, and MAPE as evaluation metrics. The results show that bidirectional GRU gives better performance than vanilla GRU model. Zhang et al. [37] proposed a GRU model with optimization of HOA to predict UAV trajectories for military usage. They used HOA to prevent GRU from falling into the local optimization problem. They compared their HOA-GRU model with RNN, LSTM, GRU, and HOA-LSTM. They used RMSE as an evaluation metric. Results show that HOA optimization reduces RMSE in LSTM and GRU. However, traditional GRU works better than LSTM. When comparing the prediction time, GRU outperforms HOA-GRU, which is expected due to the complexity of HOA optimization. Even though the HOA optimization is time-consuming, the HOA-GRU gives better results and the forecast time of the HOA-GRU is still below the maximum tolerable prediction time. Xie et al. [38] proposed a maneuver prediction model on UAVs based on deep learning models and supported this maneuver prediction with short-term trajectory prediction, which is based on LSTM. They created a library of maneuvering units and predicted the maneuvers in this library. They proposed an Adaptive boosting-Auto encoder-Deep Echo State Network to predict maneuver in long-term. They proposed an LSTM model with the support of Gaussian walking algorithm to predict trajectory in short-term. They combine these two models with a layered strategy to predict maneuvering with better accuracy. Firstly, they used trajectory features to classify maneuvers with the existing maneuvers inside the library. Then, they created two different layers one of which is long-term prediction and the other one is short-term prediction models. Later that, they kept track of historical movements and created characteristics to identify the maneuvers. They combine these characteristics with current trajectory prediction results to determine the maneuver in real-time. The results show that combining trajectory prediction with maneuver prediction gives high accuracy in maneuver prediction. Tang et al. [39] proposed a GRU-based trajectory prediction method for UAVs. In experiments, they used 800 historical data on quadrotor UAVs. They used MAE as their evaluation metric. They compared their GRU model with ARIMA, SVR, and LSTM. The results show that GRU and LSTM give better performance than ARIMA and SVR. However, due to LSTM's complex structure, the prediction time of LSTM is higher than GRU.

Reference	Model	Purpose	Dataset	Key Finding
Shafienya et. al. [23]	Hybrid CNN-GRU and 3D-CNN	Predict the aircraft trajectory in Airports	Historical ADS-B Data	- Extracting trajectories with CNN-GRU is a good option. - CNN gives good results in low maneuver capacity aircrafts
Wu et. al. [24]	Back Propagation Neural Network	Air Traffic Management	Commercial Aircraft ADS-B Data	- Small data is enough to predict low maneuver capacity aircrafts
Han et. al. [25]	K-Means Clustering and GRU	Air Traffic Management	Historical And Real Time ADS-B Data	- Clustering is easy and very efficient with commercial aircrafts
Ma et. al [26]	Hybrid CNN-LSTM	Air Traffic Management.	Historical ADS-B data.	- Extracting spatial features with CNN is giving good results.
Zhao et. al. [27]	Deep LSTM model	Air Traffic Management	Real flight data from ADS-B	- Limitation of the LSTM causes LSTM to drop the learning capacity while predict multidimensional values.
Zhang et. al. [28]	Bayesian NN	Ease the safety concerns on civil aviation.	Raw Flight Information Exchange Model Messages	- Using DNN to correctify LSTM in long term prediction helps LSTM to give better results.
Pang et. al. [29]	RNN and FCNN	Air Traffic Management	Air traffic and Weather data from Sherlock Warehouse	- Weather is affecting the flight trajectories. - Adding weather parameters while doing trajectory prediction reduces the error rate of prediction.
Shi et. al. [5]	LSTM	Create a safer air traffic system. Oversee the potential dangers in flight routes.	ADS-B data	-Adding sliding windows gives the model the ability to track every phase of the flight.
Cheng et. al. [30]	LSTM	Prevent conflict on Air Traffic.	ADS-B data	- Adding trajectory prediction in conflict prevention algorithm makes conflict prediction reliable in different timespans.
Zhang et. al. [32]	Attention-based LSTM	Air Traffic Management	ADS-B Data	- Splitting flight to phases helps predicting trajectories. - Adding attention to LSTM gives better results while predicting in real time
Liu et. al. [33]	Encoder-Decoder LSTM	Air Traffic Management	Flight Tracks dataset from FAA Traffic Flow Management System	- While predicting full path, encoder decoder model helps focusing.
Han et. al. [34]	Cyclic GRU	Easing the Air Traffic Control	ADS-B data	- Dividing dataset as historical and real-time data while training, fastens the prediction.
Wang et. al. [22]	Multi-cells Neural Network	Predicting trajectory in Terminal Maneuvering Areas	ADS-B data	- Dividing data to clusters and training model in each cluster differently gives variety when selecting best model.
You et. al. [35]	Encoder-Decoder GRU-RNN	Predict trajectory of ships	AIS data	- Encoder-Decoder model gives highly good results when compared to LSTM or GRU models.
Yang et. al [36]	Bidirectional GRU	Quadrotor Trajectory Prediction	10 different types of Quadrotor data	- GRU is good with high maneuverable aircrafts.
Zhang et. al. [37]	GRU model with optimization of HOA	Predict UAV trajectories for military usage	Historical UAV Data	- Using HOA to optimize GRU and LSTM prevent these models to fall local optimization problems. - Optimization reduces the error rate but with complexity causes the higher prediction times
Xie et. al. [38]	Adaptive boosting-Auto encoder-Deep Echo State Network for long term prediction LSTM model with Gaussian walking algorithm for short term predicton	Predict UAV Trajectory and maneuvers for military usage	Simulated Data from UCAV Kinematic model	-Creating maneuver dataset to predict from is good solution when predicting maneuvers. - Adding trajectory prediction in maneuver prediction lowers the error rate in maneuver prediction.
Tang et. al. [39]	GRU	UAV trajectory prediction	800 historical data of quadrotor UAVs.	- Due to LSTM's complex structure prediction time of LSTM is higher than GRU.

Table 3.2 Aircraft Tracking Systems with Deep Learning Approaches

4. PROPOSED METHOD

In this thesis, we propose a novel deep learning-based real-time Stacked Encoder-Decoder GRU model called DeepAT that will predict the next position of the aircraft in 3D (latitude, longitude, and altitude). The general structure of Stacked Encoder-Decoder GRU model is shown in Fig. 4.1. We aim to predict the latitude, longitude, and altitude values of an aircraft at time t with respect to the old values. The main purpose of selecting Encoder-Decoder GRU model is its superior performance in especially time-series problems. Our real-time problem can be called a sequence-to-sequence problem. GRU and LSTM models have high reliability in time-series prediction and combining RNN-based methods with Encoder-Decoder architecture strengthens the prediction of RNN-based methods.

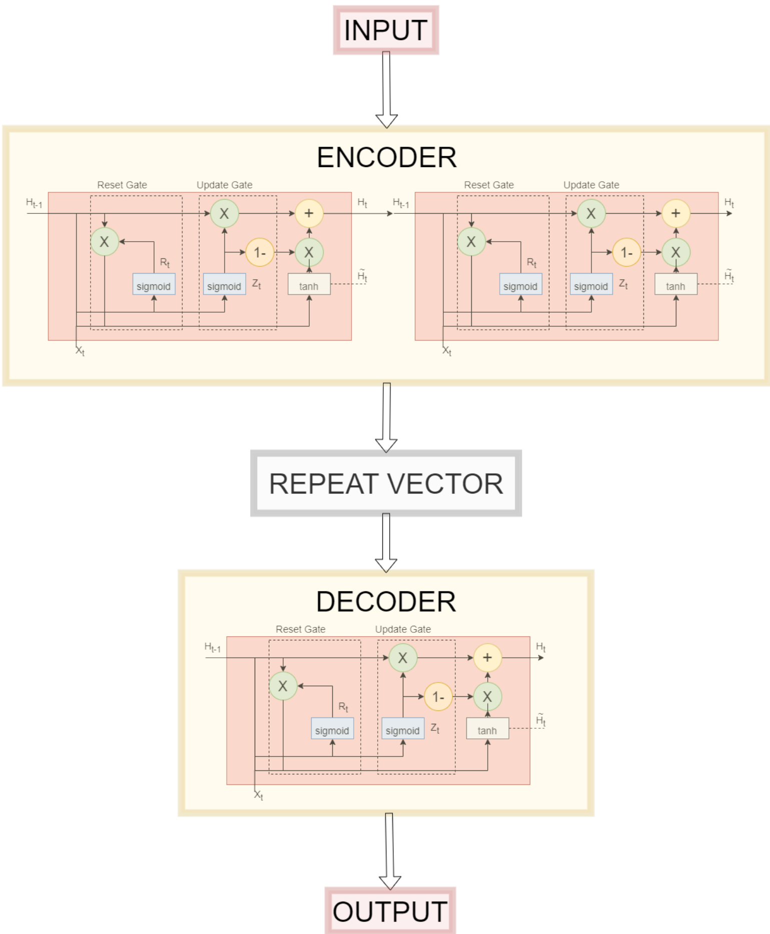

Figure 4.1 Stacked Encoder-Decoder GRU

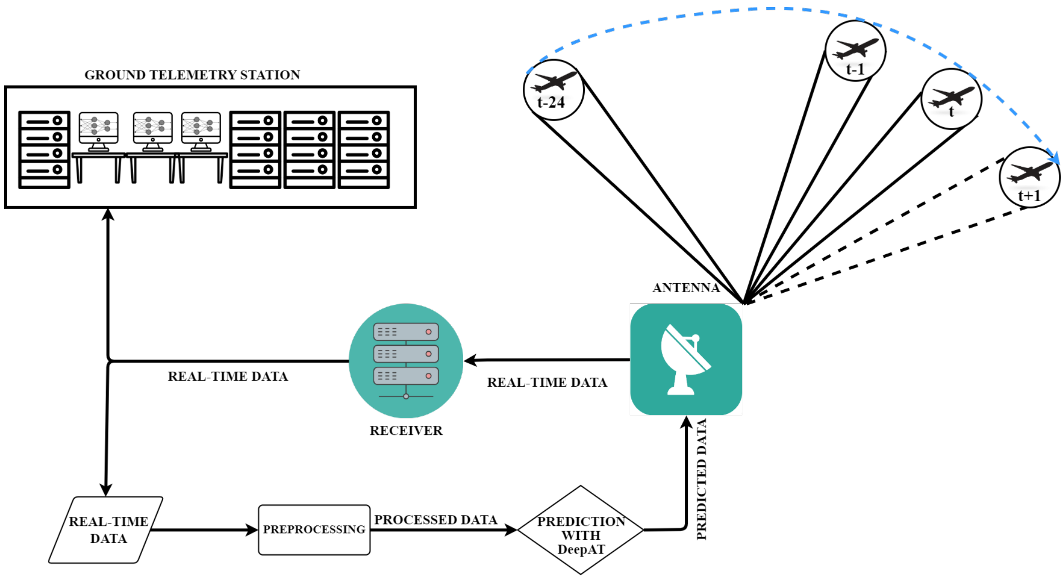

Figure 4.2 The System Model of Proposed DeepAT

The general flow of DeepAT is shown in Fig. 4.2. In DeepAT, our aim is to predict the next location of an aircraft in 3D and solve the data loss problem in flight tests. In the aviation industry, every aircraft needs to be tested before rolling out. Although different types of aircraft, such as fighter jets and UAVs, have different structural designs, testing phases are crucial for a safe and accurate aviation operation. This testing phase consists of many different areas. The final part of this testing phase is the flight test. Many sensors are placed inside the aircraft by flight test instrumentation engineers for flight tests. Real-time tracking of these sensors from the ground is crucial. To track these sensors and track the aircraft, a ground control center called a telemetry station is used. In the telemetry station, the aircraft is tracked by a radar antenna, and data is collected from this antenna. With this capability, any kind of error can be detected and a failure recording procedure can be started by an engineer in the control center with the real-time data flow from the aircraft to the ground. While in flight tests, any kind of loss in data flow can be big trouble and may affect the test phase in very different ways. The aircraft's speed, the antenna's or radar's angle of view, the rotational speed of the antenna or radar, the aircraft's proximity to the runway, and the device all have the potential to cause data loss. One scenario where data loss occurs is when an aircraft moves outside the radar's coverage area. Repetition of the maneuver or the flight test may be necessary, depending on the degree of the loss. To overcome this problem, we propose our DeepAT where the next location of aircraft is going to be predicted. For any kind of data loss problem, the coordinates and altitude values obtained from DeepAT will be used to find the aircraft with a radar antenna as soon as possible.

Parameter	**Values**
Sliding Window Size	24
Number of Epochs	20
Patience	4
Optimizer	Adam
Activation Function	ReLU
Number of GRU in Encoder	2
Number of GRU in Decoder	1
Number of Units in GRU	32

Table 4.1 Hyper Parameters

The hyper parameters used for DeepAT are given in Table 4.1. 20 epochs are determined for the training phase. The patience value was chosen as 4. The patience value is followed by the validation loss value. Since Adam optimizer has a faster calculation time than others and needs less parameters, Adam optimizer is selected. Also, ReLU is determined as the activation function since it reduces the probability of encountering gradient vanishing problems. We used 2 GRU in Encoder and 1 GRU in decoder, our repeat vector number was 1.

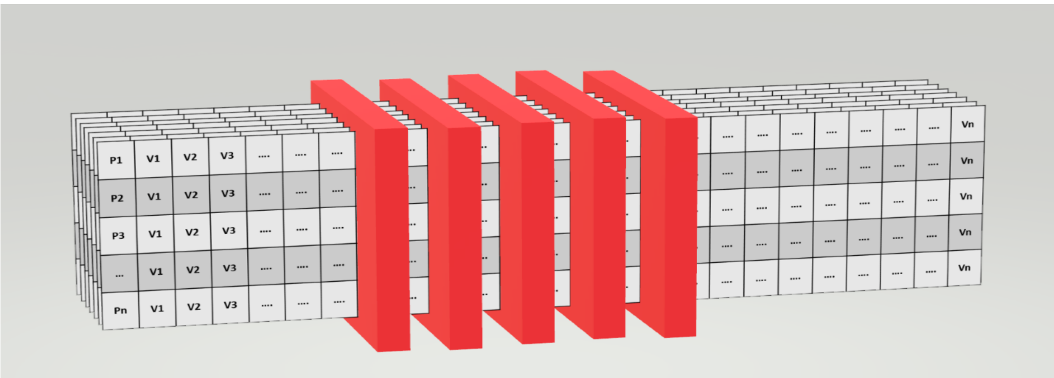

Figure 4.3 Sliding Window

We use a sliding window (Fig. 4.3) to predict the next location in real-time. We split windows to a specific size and slide them one by one to predict the next location. In our DeepAT model, this sliding window size is 24. We are proposing a Stacked Encoder-Decoder GRU model in our DeepAT model.

We compare our model with five different methods including LSTM, GRU, Encoder-Decoder LSTM, Encoder-Decoder GRU, and Stacked Encoder-Decoder LSTM models. The reason that we select GRU model and Encoder-Decoder architecture is the outstanding performance of RNN with Encoder-Decoder architecture in time-series data. As we said before, real-time time-series problems can be treated like sequence-to-sequence problems. The reason that we choose GRU over LSTM is that GRU has a simpler structure, fewer parameters, and the same performance as LSTM. In the short-term prediction of the next position, the complexity of LSTM causes the model to give higher error rates.

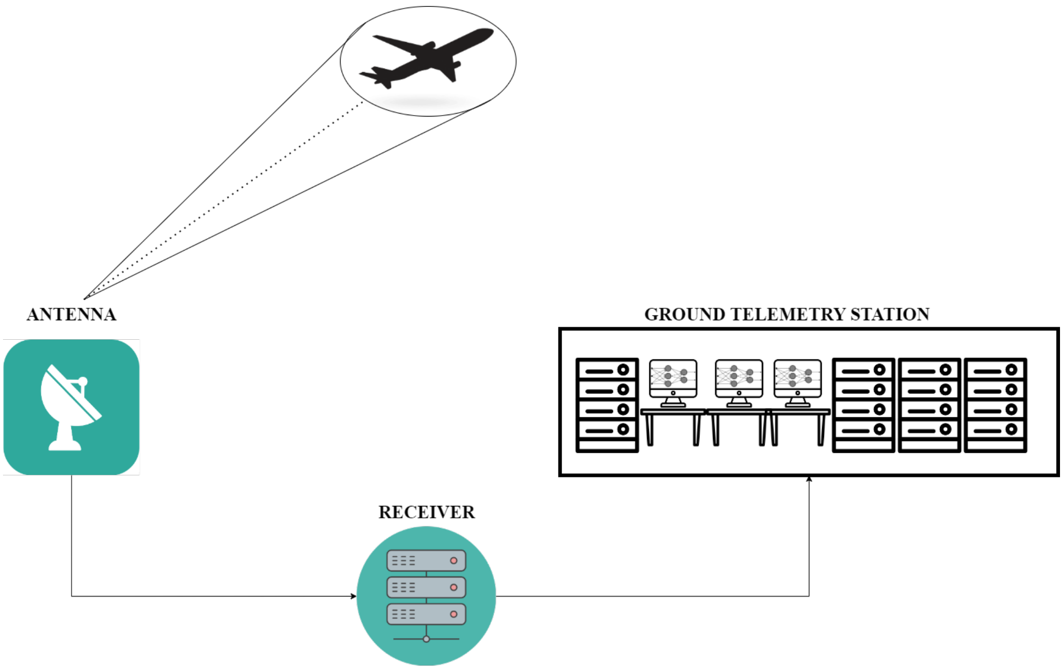

Figure 4.4 Current Telemetry System

4.1. Dataset

We test DeepAT on two different aircraft types. The first one is a fixed-wing high-maneuvering propeller aircraft and the other one is a UAV. As a result, we conduct our experimental analysis on two different types of datasets. We use real-time telemetry data that flows to the ground from aircraft. As can be seen in Fig. 4.4, the current telemetry system only gets the data from aircraft and sends it to a receiver and after doing the necessary engineering conversions on raw data, visualize the data and store it in the database. In a telemetry system, every parameter has its own sampling rate. These sampling rates can range from 1 sample for 1 second to 8000 samples for 1 second. In general, similar parameters like latitude, longitude, or altitude came at similar sampling rates. In our dataset, we used 10 samples for 1 second for propeller aircraft and we used 1 sample for 1 second for UAV. The fixed-wing aircraft has a high-maneuvering capability and can even successfully complete acrobatic maneuvers. In two different datasets, we have similar parameters but the behavior of the aircraft is completely different. By selecting two completely different aircraft, our goal is to show that our DeepAT model can effectively predict both low-maneuvering and high-maneuvering aircraft. We have about 20 parameters for each aircraft and 10 flights per aircraft. We trained our DeepAT model separately on each of these flights. Our parameters are basically Calibrated Air Speed, True Air Speed, Ground Speed, Vertical Speed, Lateral Acceleration, Longitudinal Acceleration, Normal Acceleration, Pitch Angle, Pitch Rate, Roll Angle, Roll Rate, Yaw Angle, Yaw Rate, Magnetic Heading, True Heading, Ground Track, Latitude, Longitude, GPS Altitude, East Velocity, North Velocity, and lastly Vertical Velocity.

Flight Number	**Flight Time**	**Total Number of Data**
F1	73 minute	43800
F2	113 minute	67800
F3	87 minute	52200
F4	56 minute	33600
F5	122 minute	73200
F6	72 minute	43200
F7	71 minute	42600
F8	90 minute	54000
F9	98 minute	58800
F10	75 minute	57000

Table 4.2 Data Specification of Propeller Aircraft

As can be seen in Table 4.2 each flight has its own size and separate time range. In general, the flight times of aircraft are close to each other but flight characteristics are changeable. On the other hand, Table 4.3 shows us that UAV has various numbers of different flight times.

Flight Number	Flight Time	Data Size
F1	261 minute	15660
F2	379 minute	22740
F3	137 minute	8220
F4	148 minute	8880
F5	255 minute	15300
F6	1933 minute	115980
F7	1659 minute	99540
F8	139 minute	8340
F9	116 minute	6960
F10	3000 minute	180000

Table 4.3 Data Specification of UAV

We can say that UAV have similar flight characteristics in each flight. Flight time does not have a great effect on the predictions for aircraft with repetitive movements in flights like this one.

4.2. Pre-processing

We used real-time telemetry data collected with the ground control system. We selected two different aircraft and 10 flights per aircraft, we had 20 different flights in total. In each flight, we have about 20 parameters. Most of these parameters have different sample rates. Firstly, we arrange these parameters at a fixed sampling rate of 10 values per second.

It is required to clean the data, after getting all parameters to a fixed sampling rate. In telemetry systems, unreliable data continues to flow even when the connection is interrupted. Such data needs to be removed. As seen in Fig. 4.5, there are peaks due to the loss of connection. These peaks mean there is a problem with this data. In order to remove this data, we limit the minimum and maximum positions in latitude, longitude, and altitude values. Even after these limitations, still unreliable data or anomalies can be observed. To overcome this problem, we detect and discard the anomalies and unreliable data with Interquartile Range (IQR) method [40].

$$Q1 = (1/4) * n \tag{11}$$

$$Q3 = (3/4) * n \tag{12}$$

$$IQR = Q3 - Q1 \tag{13}$$

$$UpperBound = Q3 + (1.5 * IQR) \tag{14}$$

$$LowerBound = Q1 - (1.5 * IQR) \tag{15}$$

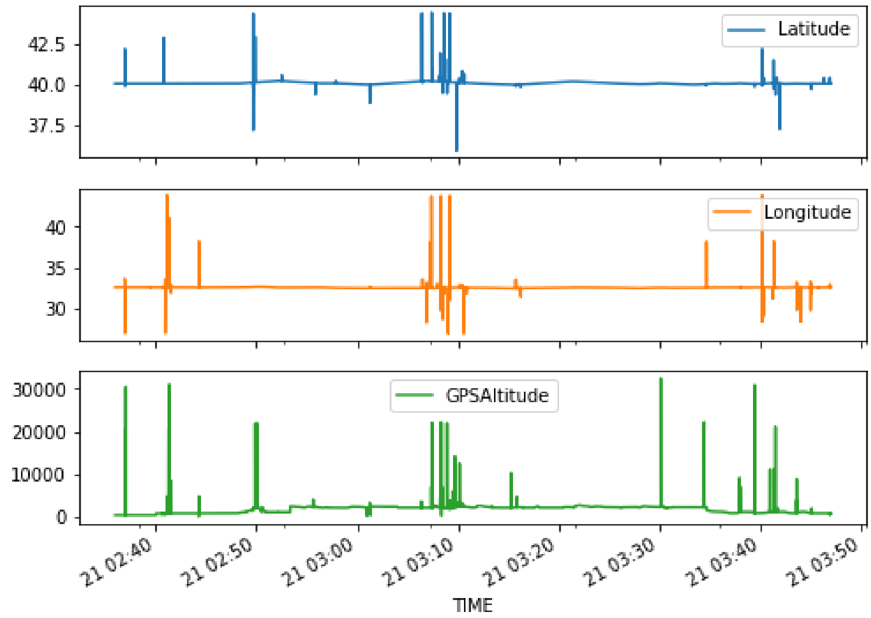

Figure 4.5 Data Sample Before Cleaning

However, just getting the data between Q1 and Q3 is not the way to eliminate outliers. To do this, a new range is defined in IQR. As in equation 15 and equation 14 we create a new bound to remove only the outliers After these calculations we can say that any data outside of these points can be called outliers. After detecting unreliable data, we normalize the data by extracting the mean and dividing by the standard deviation. We split the dataset into 70% for training, 20% for validation, and 10% for testing. We do not use cross validation. Since the time-series datasets are dependent on the previous data and are sequential within the dataset, the datasets cannot be divided into random folds as is done in cross validation. The cross validation method used for time-series is not suitable for real-time prediction. The cross validation method applied for the time-series is done by starting the dataset from a small subset and enlarging the dataset to use larger datasets.

5. EXPERIMENTAL ANALYSIS

5.1. Experimental Setup

The experiments were conducted on a laptop with 64GB RAM, 3.3GHz Intel core I9 processor, and NVIDIA GeForce RTX 3080 Graphics Card. Also, we utilize Keras, TensorFlow, Matplotlib, seaborn, and SciPy libraries. To analyze the superiority of our proposed DeepAT model, we compared our model with LSTM, GRU, Encoder-Decoder LSTM, Encoder-Decoder GRU, and Stacked Encoder-Decoder LSTM models. These models were chosen for comparison because they are commonly used in the literature, particularly in aircraft systems.

5.2. Experimental Results and Discussion

In this thesis, the prediction of 3D next position of two different aircraft are analyzed in detail. Due to their different structures, the same proposed model is used separately for each aircraft flying datasets, one of them is a high-maneuvering propeller aircraft and the other one is a UAV. We use MAE and MSE values as evaluation metrics to measure the deviation of real and predicted values of the latitude, longitude, and altitude of the aircraft. We utilized MAE and MSE to evaluate the performance of our model's predictions in terms of both general results and highly biased results. MSE was applied to see the high-bias estimations. Because RMSE did not highlight large deviations as much as the MSE and was computed by using the square root of the MSE value, it was not employed. The MAE and MSE metrics are computed with the below equations.

$$MAE = \sum_{i=1}^{D} |x_i - y_i| \tag{16}$$

$$MSE = \sum_{i=1}^{D} (x_i - y_i)^2 \tag{17}$$

We compared our proposed DeepAT model with state-of-the-art. The results show that our proposed model has higher accuracy than state-of-the-art. As seen in Table 5.1, our proposed DeepAT shows high performance in predicting the next location of the propeller aircraft. As seen in Table 5.10, on the other hand, other models also has good results for UAV due to the aircraft's mostly repetitive path and low-maneuvering capability. Such complex structures do not perform as well as expected in aircraft with low maneuverability and repetitive movements.

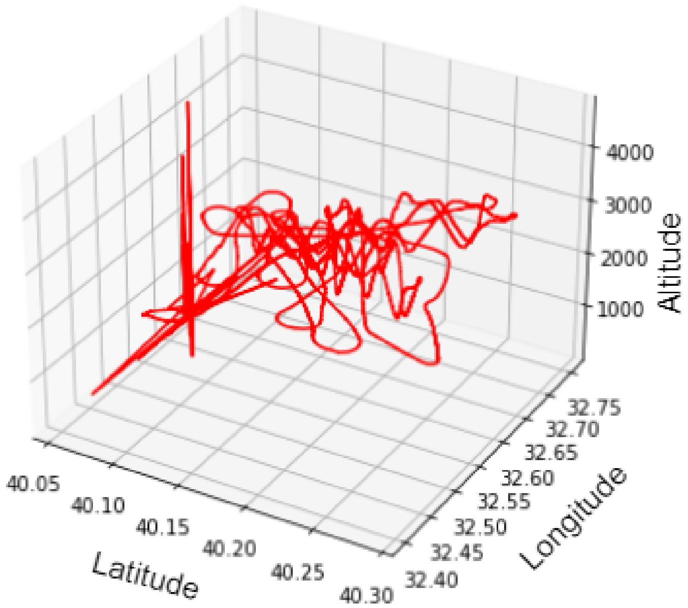

Figure 5.1 Example of 3D Path of Propeller Aircraft

As seen in Fig. 5.1 and Fig. 5.2, there is a noticeable difference in capability between the propeller aircraft and the UAV mission routes.

The proposed model is examined in detail in two sub-headings, one of which is a fixed-wing propeller aircraft and the other is a UAV. The benefits of each are examined in depth, and the advantages of the proposed model are clearly emphasized.

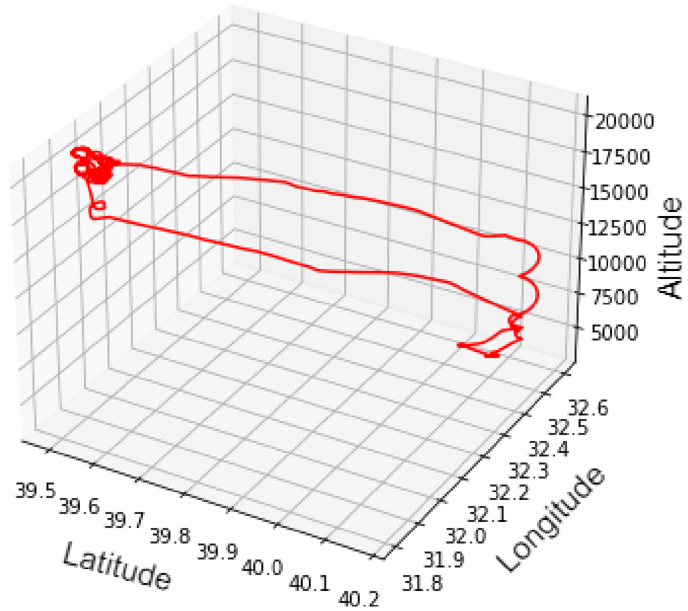

Figure 5.2 Example of 3D Path of UAV

5.2.1. Fixed-Wing Propeller Aircraft

The DeepAT model is trained separately for each flight. Average MAE and MSE results are given in Table 5.1. MAE and MSE metrics are evaluated individually for the latitude, longitude and altitude predictions. The obtained results show that DeepAT model performs better in average MAE and MSE scores when it is compared with the other approaches. The Encoder-Decoder GRU model has the closest average MAE score to that of the DeepAT model. For example, the Encoder-Decoder GRU average MAE score is 0.1512 where it is 0.1493 for DeepAT. Similarly, Altitude MAE average result of DeepAT is 0.0983 and altitude MAE result of Encoder-Decoder GRU is 0.1006. The Longitude prediction has similar results. The LSTM model had the lowest prediction results for all metrics of MAE and MSE, whereas the other approaches had similar results among themselves. The results show that DeepAT model has better results when compared to benchmark models. Furthermore, experimental results are almost similar when each flight is inspected separately for models. Accordingly, the DeepAT model has consistency in its prediction.

The predictions made by the model are quite accurate and are close to the actual values. The small variations observed in the predicted values can be attributed to the limitations of the radar antenna range. It can be concluded that the sequence prediction made by the model has a smooth transition and the predicted values gradually converge towards the real values as the sequence progresses. The model's performance is consistent and the predictions are reliable. Overall, we can say that the model is able to effectively capture the underlying patterns in the data and predict the values with a high degree of accuracy.

	MAE			**MSE**		
	Latitude	**Longitude**	**Altitude**	**Latitude**	**Longitude**	**Altitude**
DeepAT	**0.1493**	**0.1553**	**0.0983**	**0.0585**	**0.0593**	**0.0356**
StEncDecLSTM	0.1515	0.1604	0.1285	0.0632	0.0683	0.0562
EncDecGRU	0.1512	0.1585	0.1006	0.0655	0.0629	0.0398
EncDecLSTM	0.1652	0.1661	0.1109	0.0714	0.0750	0.0492
GRU	0.1563	0.1642	0.1105	0.0768	0.0718	0.0511
LSTM	0.1829	0.1828	0.1365	0.0877	0.0864	0.0659

Table 5.1 Average MAE and MSE Results of Propeller Aircraft

Upon a thorough examination, it can be inferred that the altitude prediction results are superior to those of latitude and longitude prediction. This is due to the penalties incurred by even minimal errors during the normalization stages. Despite the changes in latitude and longitude values being relatively small, it is crucial to estimate these changes as accurately as possible in order to prevent aircraft loss. As demonstrated in the Table 5.2, Table 5.3 and Table 5.4 the DeepAT model does not produce optimal results in the third flight, albeit it yields results that are comparable to those of other models in the estimation of latitude, longitude, and altitude. However, it appears that all benchmark models performed worse in this particular flight compared to other flights. This may be attributed to the inability to effectively extract the specific erroneous data associated with the flight, as well as the unique characteristics of certain flights. Additionally, it is observed that in the 10th flight, the performance of all models is significantly inferior compared to other flights. Upon a detailed examination of this flight, it is apparent that the instantaneous data losses are substantial, and this data cannot be classified as outliers as it falls within certain constraints. In the 5th flight, the performance of each model is commendable and the results are comparable. A more specific analysis of this flight reveals that the data loss is minimal, and the flight path is relatively routine in comparison to other flights.

Flights	**DeepAT**	**StEncDecLSTM**	**EncDecGRU**	**EncDecLSTM**	**GRU**	**LSTM**
1	0.1178	0.1110	0.0697	0.1886	0.2525	0.1664
2	0.1651	0.1487	0.1645	0.1661	0.1746	0.1988
3	0.2475	0.2392	0.2616	0.2793	0.2619	0.2710
4	0.1983	0.2146	0.2195	0.2287	0.1997	0.2314
5	0.0299	0.0233	0.0317	0.0100	0.0273	0.0487
6	0.1144	0.1190	0.1171	0.1165	0.1172	0.1168
7	0.0742	0.0893	0.0956	0.0943	0.1334	0.0938
8	0.0680	0.0820	0.0745	0.0605	0.0797	0.1472
9	0.1564	0.1676	0.1628	0.1701	0.1607	0.1635
10	0.2914	0.3506	0.3152	0.3382	0.3875	0.3916

Table 5.2 Latitude MAE Results of Propeller Aircraft

Flights	**DeepAT**	**StEncDecLSTM**	**EncDecGRU**	**EncDecLSTM**	**GRU**	**LSTM**
1	0.1792	0.1862	0.1327	0.1806	0.2321	0.2491
2	0.2281	0.2180	0.2236	0.2204	0.2219	0.2385
3	0.1428	0.1516	0.1279	0.1257	0.1390	0.1426
4	0.1815	0.2005	0.1998	0.1964	0.1889	0.1961
5	0.0224	0.0432	0.0327	0.0213	0.0079	0.1013
6	0.1673	0.1705	0.1775	0.1930	0.2330	0.1818
7	0.1223	0.1456	0.1599	0.1616	0.1630	0.1650
8	0.1692	0.1408	0.1584	0.1524	0.1627	0.1719
9	0.1379	0.1323	0.1475	0.1454	0.1296	0.1332
10	0.2026	0.2160	0.2253	0.2647	0.2340	0.2494

Table 5.3 Longitude MAE Results of Propeller Aircraft

Flights	**DeepAT**	**StEncDecLSTM**	**EncDecGRU**	**EncDecLSTM**	**GRU**	**LSTM**
1	0.0598	0.0610	0.1071	0.0544	0.1558	0.1130
2	0.1674	0.1754	0.1656	0.1848	0.2155	0.2112
3	0.1311	0.1250	0.1093	0.1281	0.1568	0.1538
4	0.0981	0.1219	0.1053	0.1208	0.1004	0.1065
5	0.0038	0.0123	0.0078	0.0080	0.0327	0.1109
6	0.0914	0.1072	0.1055	0.1089	0.1098	0.1229
7	0.0492	0.0893	0.0654	0.0728	0.0762	0.1114
8	0.0338	0.0642	0.0161	0.0449	0.0424	0.0456
9	0.1082	0.1520	0.1042	0.1047	0.1050	0.1487
10	0.2411	0.2969	0.2206	0.2824	0.2342	0.2416

Table 5.4 Altitude MAE Results of Propeller Aircraft

An in-depth examination of the mean squared error (MSE) values reveals the remarkable success of the DeepAT model. The analysis takes into account both the compression of data during preprocessing and the estimation errors of each model. Upon examining the Latitude, Longitude, and Altitude values over Table 5.5, Table 5.6 and Table 5.7, it was found that the DeepAT model demonstrated a commendably narrow margin of error for each parameter, thereby yielding low MSE values that effectively penalize significant deviations in the margin of error. The results of this analysis highlight the robustness and accuracy of the DeepAT model in approximating and predicting the Latitude, Longitude, and Altitude values. Upon examining the 10th flight in detail, it becomes apparent that all models perform comparatively poorly in comparison to their results from other flights. Nevertheless, the DeepAT model still exhibits a remarkably low MSE for each parameter compared to other models. A closer look at the MSE values for this flight reveals that the DeepAT model avoids producing results with significant deviations, despite producing some erroneous results. This highlights the ability of the DeepAT model to maintain a certain degree of accuracy, even in challenging conditions. With the exception of the 2nd flight, the DeepAT model consistently produced favorable results when compared to benchmark models across all flights. In the case of the 2nd flight, the results generated by the DeepAT model were notably similar to those produced by other models. This demonstrates the overall reliability and effectiveness of the DeepAT model in comparison to benchmark models, with only a minor deviation in performance for the 2nd flight.

Flights	**DeepAT**	**StEncDecLSTM**	**EncDecGRU**	**EncDecLSTM**	**GRU**	**LSTM**
1	0.0271	0.0389	0.0387	0.0611	0.0835	0.0423
2	0.0430	0.0418	0.0495	0.0558	0.0684	0.0790
3	0.9801	0.0997	0.1121	0.1358	0.1149	0.1297
4	0.1463	0.1660	0.1992	0.1597	0.1302	0.1739
5	0.0003	0.0005	0.0008	0.0008	0.0010	0.0023
6	0.0257	0.0261	0.0279	0.0283	0.0289	0.0281
7	0.0132	0.0171	0.0246	0.0202	0.0349	0.0193
8	0.0070	0.0110	0.0080	0.0083	0.0164	0.0295
9	0.0324	0.0352	0.0355	0.0385	0.0355	0.0405
10	0.1614	0.2292	0.1992	0.2094	0.2554	0.3326

Table 5.5 Latitude MSE Results of Propeller Aircraft

Flights	DeepAT	StEncDecLSTM	EncDecGRU	EncDecLSTM	GRU	LSTM
1	0.0524	0.0599	0.0314	0.0615	0.0803	0.1062
2	0.0919	0.0838	0.0967	0.0867	0.0866	0.1083
3	0.0351	0.0371	0.0275	0.0425	0.0384	0.0400
4	0.1013	0.1454	0.1246	0.1693	0.1065	0.1089
5	0.0005	0.0018	0.0005	0.0004	0.0006	0.0102
6	0.0751	0.0643	0.0770	0.0913	0.1309	0.0539
7	0.0771	0.1350	0.1036	0.1086	0.0944	0.0727
8	0.0477	0.0327	0.0419	0.0380	0.0487	0.0518
9	0.0274	0.0283	0.0303	0.0321	0.0284	0.0317
10	0.0763	0.0950	0.0956	0.1298	0.1037	0.2810

Table 5.6 Longitude MSE Results of Propeller Aircraft

Flights	DeepAT	StEncDecLSTM	EncDecGRU	EncDecLSTM	GRU	LSTM
1	0.0148	0.0224	0.0273	0.0263	0.0419	0.0343
2	0.0972	0.0953	0.1058	0.1129	0.1239	0.1240
3	0.0327	0.0921	0.0345	0.0360	0.0473	0.0436
4	0.0251	0.0741	0.0648	0.0637	0.0629	0.0689
5	0.0001	0.0001	0.0006	0.0006	0.0010	0.0123
6	0.0131	0.0459	0.0091	0.0168	0.0283	0.0292
7	0.0083	0.0150	0.0160	0.0366	0.0207	0.0272
8	0.0029	0.0082	0.0016	0.0051	0.0082	0.0084
9	0.0161	0.0275	0.0166	0.0187	0.0171	0.0287
10	0.1462	0.1916	0.1320	0.1915	0.1602	0.2830

Table 5.7 Altitude MSE Results of Propeller Aircraft

The tables presented above demonstrate the superiority of our proposed DeepAT method in terms of mean absolute error (MAE) and mean squared error (MSE) values for propeller aircraft. Upon examining each flight individually, it becomes evident that the DeepAT model delivers satisfactory results in predicting the next position in 3D for propeller aircraft. This supports the efficacy of our proposed DeepAT method in providing accurate predictions for propeller aircraft.

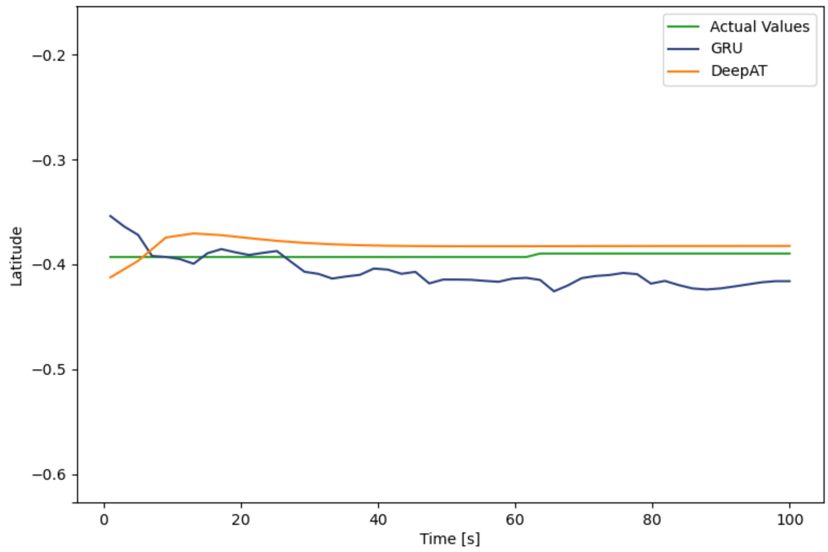

Figure 5.3 DeepAT and GRU Latitude Result for Flight 4 in Propeller Aircraft

As seen in Fig. 5.3 DeepAT model and the GRU model both aim to predict the latitude value based on input data, and both models produce predicted values that converge towards the actual latitude value as time progresses. The proposed DeepAT method demonstrates remarkable proximity to actual values in comparison to GRU, as the deviation between the predicted and actual latitude values for the DeepAT model remains within a relatively small margin, while GRU exhibits significantly higher deviations. This suggests that the DeepAT method is more accurate in predicting latitude values for this particular flight. It is possible to highlight that DeepAT model takes into account the temporal relationship between the latitude values, which can help it make better predictions compared to a model that only considers each value independently.

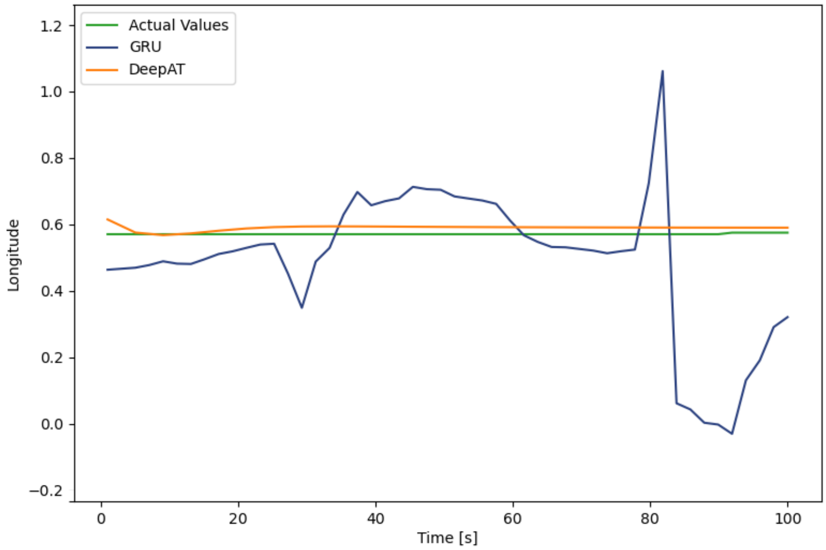

Figure 5.4 DeepAT and GRU Longitude Result for Flight 4 in Propeller Aircraft

DeepAT model predicts the longitude sequence better than GRU as seen in Fig. 5.4. In addition, it appears that DeepAT outperforms GRU in terms of predicting longitude. The values predicted by DeepAT are closer to the actual values compared to GRU, as evidenced by the smaller difference between the actual and predicted values and the sequential structure. This suggests that DeepAT is more effective at capturing the patterns and relationships in the data, leading to more accurate predictions. Based on figure, it appears that the GRU model is giving biased results for longitude prediction, as the values it predicts deviate significantly from the actual values. The model's predictions are consistently off from the actual values, either being higher or lower than the actual values. This bias in the predictions can lead to inaccuracies in the results and may cause problems in practical applications where the accuracy of the predictions is important. As can be seen in Fig 5.5, the proposed DeepAT

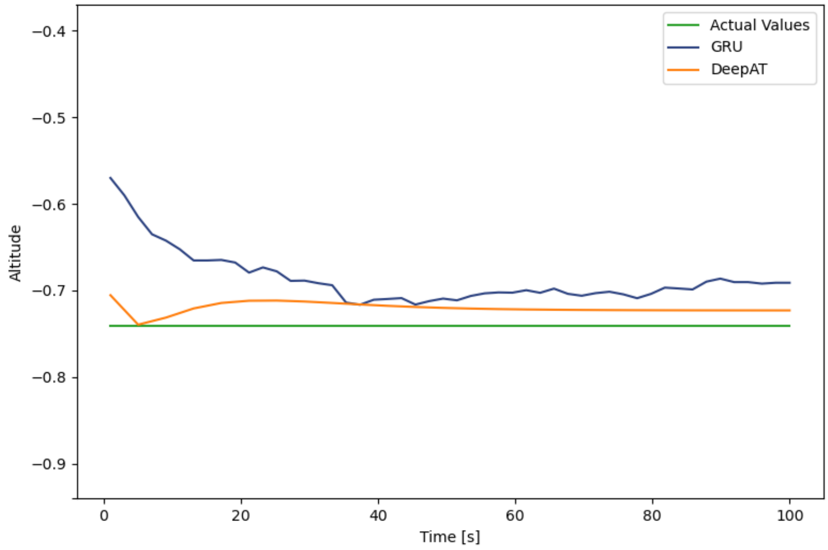

Figure 5.5 DeepAT and GRU Altitude Result for Flight 4 in Propeller Aircraft

model gives a closer prediction to the actual values compared to GRU in this scenario. This can be seen from the differences between the actual and predicted values, where the differences between the actual and predicted values by DeepAT are smaller than those by GRU. The actual altitude values seem to be close to the values predicted by the DeepAT model, with deviations in the range of a few percent. On the other hand, the GRU model appears to deviate more from the actual values and the deviation increases as the flight progresses. DeepAT model has a sequential structure which means that the information from the previous time step is taken into consideration in making predictions at each time step. This can lead to better prediction results compared to the GRU model that evaluates each prediction independently. Additionally, the results show that the DeepAT model has a closer range to the actual values compared to the GRU model, which suggests that the DeepAT model is more accurate in predicting altitude in this case.

The Table 5.8 illustrates the maximum deviation between the actual and predicted points for flight 4. The results in the table demonstrate that the DeepAT model outperforms other models in terms of accuracy in predicting latitude, longitude, and altitude values. Furthermore, the results from Table 5.8 highlight that the Encoder-Decoder structure of

	Latitude	Longitude	Altitude
DeepAT	0.5593	0.6522	0.5525
StEncDecLSTM	1.0541	0.9837	0.8571
EncDecGRU	1.2359	0.8388	0.9890
EncDecLSTM	1.3579	1.3967	1.3911
LSTM	1.1719	1.8655	1.7152
GRU	1.8021	0.9637	1.4790

Table 5.8 Maximum Deviation Between Actual and Predicted Values in Flight 4

the DeepAT model effectively reduces the range of maximum deviation between the actual and predicted values. The results in Table 5.8 reveal that the implementation of the Encoder-Decoder structure in GRU and LSTM models leads to a significant reduction in the deviation between actual and predicted values. Unlike the models without this structure, which have a relatively high deviation margin, the models that utilize the Encoder-Decoder structure exhibit a gradual decrease in deviation. This highlights the effectiveness of the Encoder-Decoder structure in improving the accuracy of prediction results.

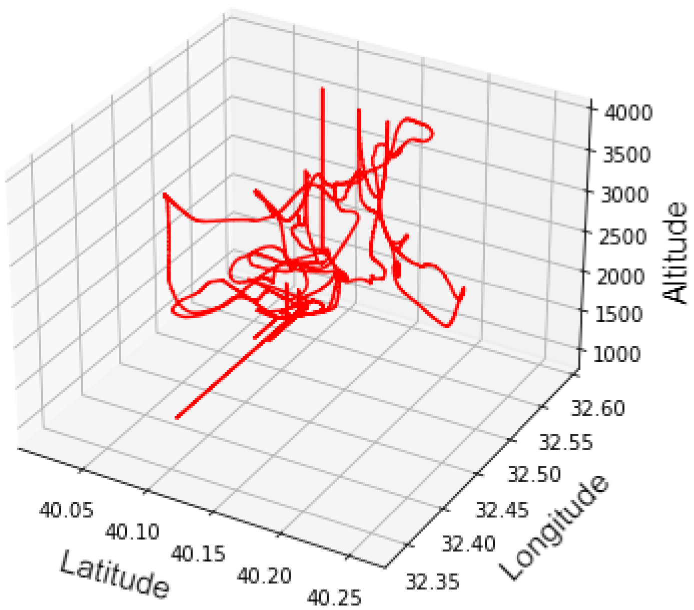

Figure 5.6 3D Flight Route Visualization for Flight 4 in Propeller Aircraft

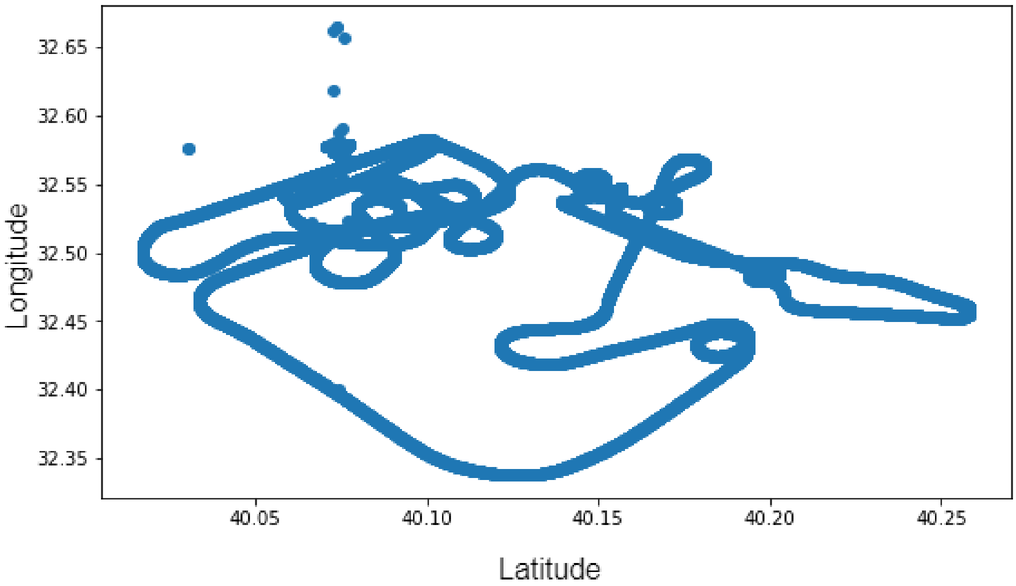

Figure 5.7 2D Flight Route Visualization for Flight 4 in Propeller Aircraft

The unreliable data from the aircraft can be observed in the form of straight lines in Figure 5.6 and as dots in Figure 5.7. Despite this, our DeepAT model has demonstrated exceptional accuracy in predicting the aircraft's exact or close locations in 3D. As depicted in Fig. 5.6, the aircraft can execute sharp maneuvers, such as a 1500ft change in altitude within a short period of time with a spin maneuver. The comparison of the actual latitude, longitude and altitude values to the values predicted by DeepAT and GRU shows that DeepAT gives results that are closer to the actual values. This suggests that the DeepAT model is more accurate in predicting these values than GRU.

However, the GRU model still has some biases in its predictions, as seen in the deviations from the actual values. This may be due to the structure of the GRU model, which evaluates each prediction within itself, rather than taking into account the sequential information of the data.

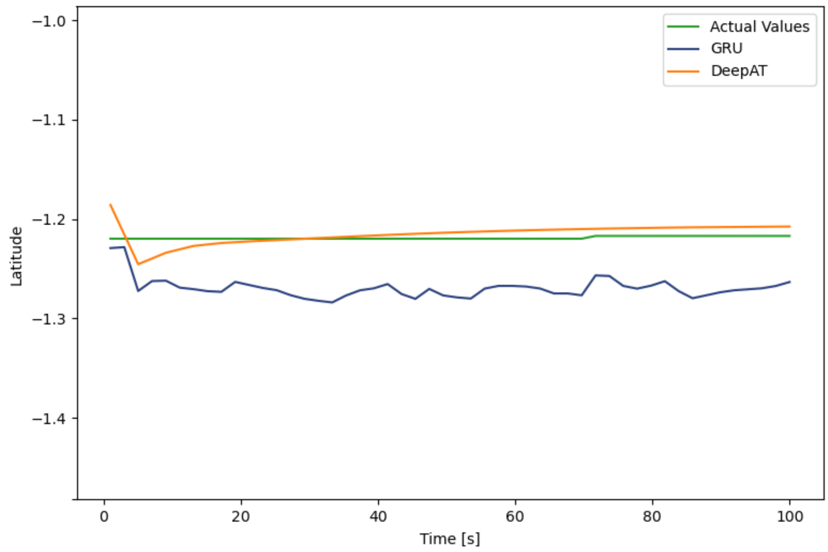

Figure 5.8 DeepAT and GRU Latitude Result for Flight 10 in Propeller Aircraft

As can be seen in Fig. 5.8 the DeepAT model appears to perform better than the GRU model in terms of accuracy for predicting Latitude values, as can be seen from the lower error for the DeepAT compared to the GRU in figure. Additionally, the DeepAT model's predictions are generally closer to the actual Latitude values compared to the GRU model's predictions, indicating that the DeepAT model has a higher level of accuracy in this case. This suggests that the DeepAT model may be a more favorable choice for predicting Latitude values in a similar scenario.

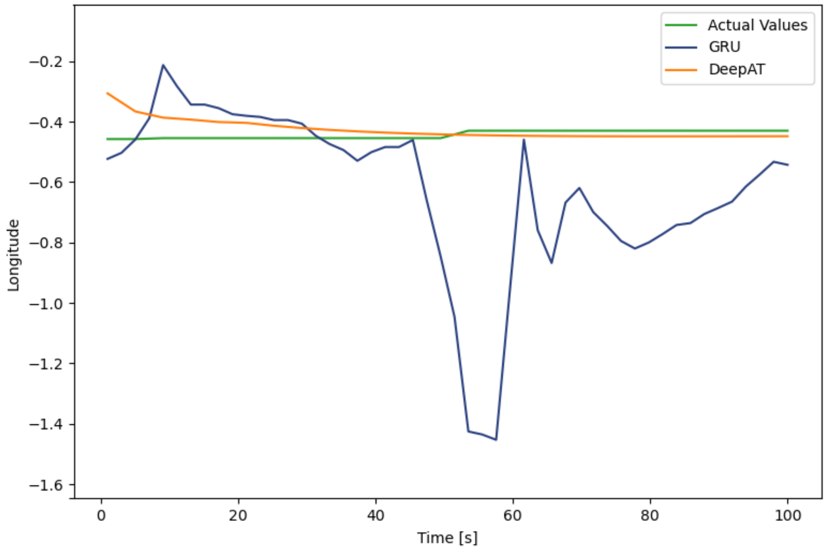

Figure 5.9 DeepAT and GRU Longitude Result for Flight 10 in Propeller Aircraft

Fig. 5.9 demonstrates the comparison of the longitude prediction results of the DeepAT and GRU models with the actual values. The results depicted in the figure indicate that the GRU model has a large deviation in its predictions, while the DeepAT model, due to its sequential structure, is able to better predict the changes in longitude. These results suggest that the DeepAT model provides significantly better longitude prediction results compared to the GRU model.

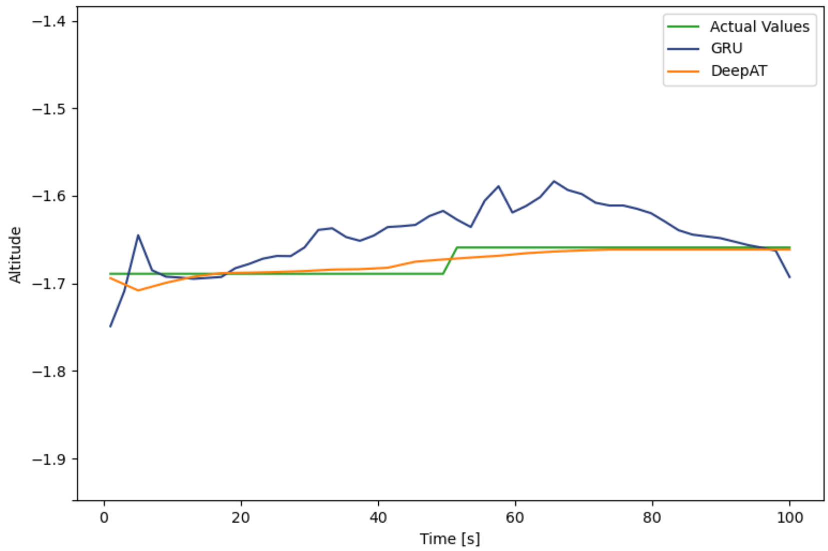

Figure 5.10 DeepAT and GRU Altitude Result for Flight 10 in Propeller Aircraft

These Fig. 5.10 represent the predicted altitude values from DeepAT and GRU and the actual altitude value for various points in a flight path. As can be seen in Fig. 5.10, the DeepAT model is capable of predicting the altitude values with a small margin of error compared to the actual values, but with small deviations in some instances. The GRU model also predicts the altitude values with a margin of error, but with higher deviations than the DeepAT model. It is possible that the sequencing capacity of the DeepAT model allows it to better capture the temporal dependencies in the flight data, leading to improved predictions.

The above figures compare the results of our proposed DeepAT model and GRU model with 3D predictions at the flight of a fixed-wing propeller aircraft. This comparison consists of latitude, longitude, and altitude values. The results show that DeepAT's predictions have significantly better predictions than GRU model. If we examine it more closely, in Fig. 5.9 and , it is observed that both models contain errors in their predictions of longitude. However, unlike GRU, DeepAT model converges to the real values despite the errors. While DeepAT model can predict 3D within the antenna angle, GRU model does not support the sequential prediction structure.

	Latitude	**Longitude**	**Altitude**
DeepAt	0.3492	0.4872	0.2416
StEncDecLSTM	0.6487	0.5231	0.4416
EncDecGRU	0.5144	0.5951	0.3518
EncDecLSTM	0.5426	0.5378	0.6153
LSTM	1.4130	1.3315	2.4275
GRU	1.4041	1.4716	2.2391

Table 5.9 Maximum Deviation Between Actual and Predicted Values in Flight 10

The Table 5.9 illustrates the maximum deviation between the actual and predicted points for flight 10. The analysis of the results from this specific flight shows that the DeepAT model performed well in terms of latitude, longitude, and altitude values, even at the point of maximum deviation. Table 5.9 indicates that the approaches based on the Encoder-Decoder structure are effective in reducing the maximum deviation between the actual and predicted values, thereby demonstrating the efficacy of the Encoder-Decoder-based methodologies in sequence prediction tasks. The performance of the GRU and LSTM models in predicting altitude values with high variations is observed to be unsatisfactory, as a significant deviation is observed in their results. Conversely, the DeepAT and the StEncDecLSTM models, which incorporate an Encoder-Decoder structure, demonstrate promising outcomes in their prediction of altitude values with high variations, exhibiting a substantial reduction in deviation. These results highlight the efficacy of Encoder-Decoder based approaches in improving the accuracy of sequence predictions.

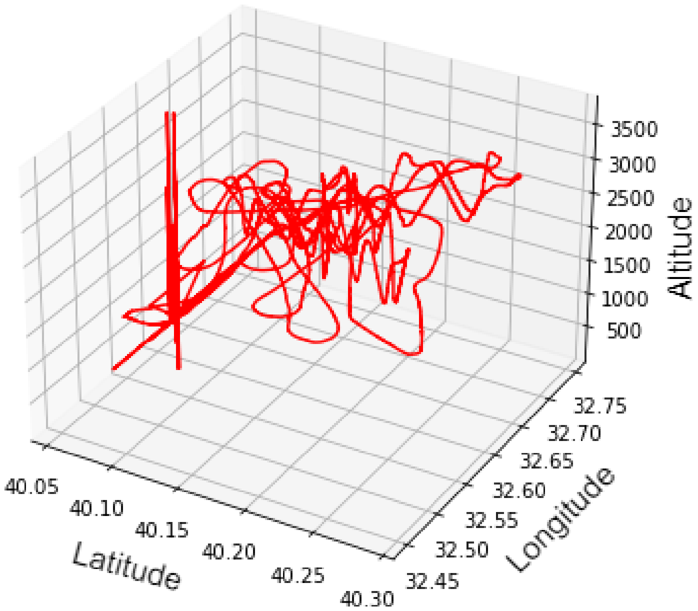

Figure 5.11 3D Flight Route Visualization for Flight 10 in Propeller Aircraft

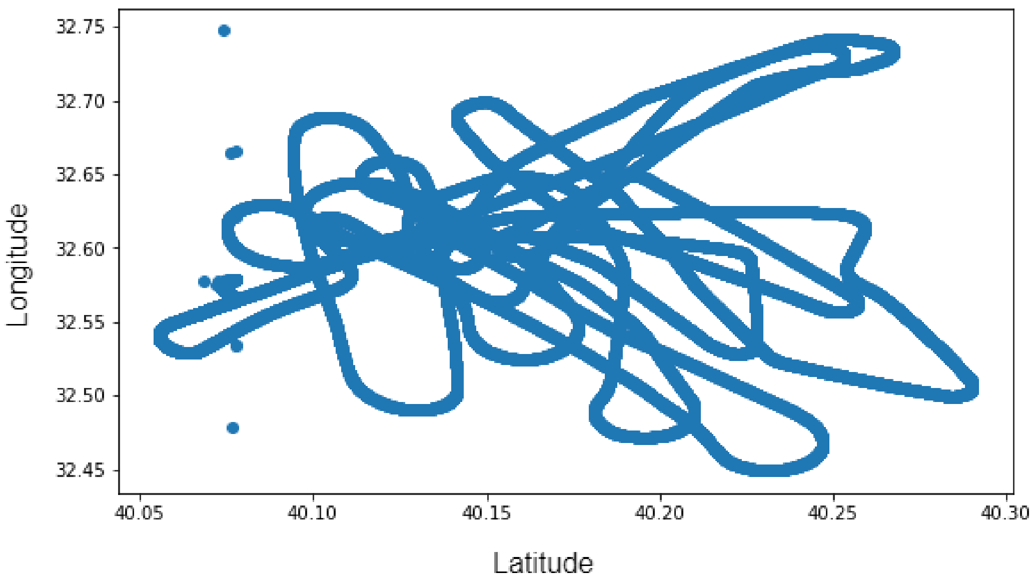

Figure 5.12 2D Flight Route Visualization for Flight 10 in Propeller Aircraft

As can be seen from the dots in Fig. 5.12 and the spikes in Fig. 5.11 unreliable data still flows from aircraft to ground. But even with this kinds of unreliable data DeepAT model can successfully predict the latitude, longitude and altitude values in small margins.

5.2.2. UAV

The DeepAT model is trained separately for each flight, and the average MAE and MSE results for the latitude, longitude, and altitude values are provided in Table 5.10. Unlike the propeller aircraft, it is observed that the predictions of each model for the UAV are similar to each other. As depicted in Fig. 5.1 and Fig. 5.2, these two aircraft have distinct structures, and due to the UAV's low capacity, simple models such as GRU also achieve satisfactory results. The results presented in Table 5.10 show that the proposed DeepAT method has competitive MAE and MSE values in all three dimensions and demonstrates its competitiveness in UAV aircraft. The DeepAT model has the best MAE results for latitude and altitude values, while the Encoder-Decoder GRU model slightly outperforms DeepAT in longitude prediction. It is noted that each model has similar predictions in all three dimensions, with close MAE values, for example, DeepAT has a Latitude MAE value of 0.2228, whereas GRU and StEncDecLSTM models have MAE values of 0.2261 and 0.2252, respectively. When examining the MSE values, it is apparent that the results are consistent with those obtained from the MAE values. It can be noted that the results are similar among each model, however, the StEncDecLSTM model stands out with a slight difference in terms of MSE values. Overall, it can be concluded that the proposed DeepAT method and the StEncDecLSTM model provide competitive results in terms of latitude, longitude, and altitude predictions in UAV aircraft.

	MAE			**MSE**		
	Latitude	**Longitude**	**Altitude**	**Latitude**	**Longitude**	**Altitude**
DeepAT	**0.2228**	0.2333	**0.1654**	0.2976	0.3054	0.1365
StEncDecLSTM	0.2252	0.2309	0.1661	0.2964	**0.3001**	**0.1355**
EncDecGRU	0.2283	**0.2296**	0.1662	0.2965	0.3114	0.1361
EncDecLSTM	0.2287	0.2306	0.1663	0.2953	0.3083	0.1356
GRU	0.2261	0.2318	0.1677	**0.2930**	0.3090	0.1376
LSTM	0.2314	0.2297	0.1665	0.3012	0.3116	0.1361

Table 5.10 Average MAE and MSE Results of UAV

It can be deduced from the results presented in the Table 5.11, Table 5.12 and Table 5.13 that the performance of each model varies depending on the flight characteristics. The prediction of longitude poses a more significant challenge compared to latitude and altitude estimations. The normalization phase compresses the data into a narrow area, leading to small errors being penalized. The results of each flight indicate that the relative success of each model is dependent on the flight characteristics. For instance, the results for the 1st flight showed that the latitude MAE values were relatively higher, whereas the longitude MAE values were predicted more accurately. On the other hand, the 9th flight showed that the latitude MAE values were more successful, while the longitude MAE values were relatively higher. These results emphasize the importance of considering the flight characteristics when evaluating the performance of prediction models.

Flights	**DeepAT**	**StEncDecLSTM**	**EncDecGRU**	**EncDecLSTM**	**GRU**	**LSTM**
1	0.2974	0.2984	0.3005	0.3000	0.2912	0.2878
2	0.2272	0.2258	0.2331	0.2268	0.2120	0.2184
3	0.1722	0.1733	0.1814	0.1801	0.1750	0.2108
4	0.3020	0.3008	0.3004	0.3001	0.2962	0.3000
5	0.1992	0.2029	0.1999	0.2070	0.2055	0.1987
6	0.2405	0.2492	0.2453	0.2474	0.2507	0.2616
7	0.1679	0.1646	0.1649	0.1647	0.1688	0.1704
8	0.1948	0.1984	0.2021	0.2086	0.1976	0.2018
9	0.2460	0.2553	0.2703	0.2612	0.2724	0.2686
10	0.1809	0.1842	0.1858	0.1912	0.1918	0.1967

Table 5.11 Latitude MAE Results of UAV

Flights	**DeepAT**	**StEncDecLSTM**	**EncDecGRU**	**EncDecLSTM**	**GRU**	**LSTM**
1	0.2083	0.2069	0.2096	0.2095	0.2139	0.2115
2	0.2401	0.2421	0.2407	0.2421	0.2502	0.2500
3	0.2445	0.2423	0.2494	0.2453	0.2343	0.2512
4	0.2540	0.2465	0.2536	0.2478	0.2494	0.2521
5	0.2464	0.2439	0.2447	0.2444	0.2519	0.2460
6	0.2471	0.2444	0.2469	0.2466	0.2457	0.2607
7	0.2334	0.2345	0.2353	0.2348	0.2308	0.2417
8	0.2566	0.2583	0.2652	0.2671	0.2573	0.2355
9	0.1654	0.1560	0.1131	0.1326	0.1392	0.1024
10	0.2378	0.2342	0.238	0.2362	0.2455	0.2466

Table 5.12 Longitude MAE Results of UAV

Flights	DeepAT	StEncDecLSTM	EncDecGRU	EncDecLSTM	GRU	LSTM
1	0.2149	0.2115	0.2120	0.2117	0.2132	0.2126
2	0.2005	0.2003	0.2008	0.2017	0.1998	0.2011
3	0.1417	0.1432	0.1432	0.1384	0.1540	0.1426
4	0.2102	0.2103	0.2102	0.2102	0.2100	0.2099
5	0.1102	0.1142	0.1156	0.1170	0.1183	0.1167
6	0.1229	0.1226	0.1226	0.1229	0.1229	0.1229
7	0.1531	0.1569	0.1555	0.1558	0.1579	0.1578
8	0.2087	0.2083	0.2086	0.2071	0.2082	0.2082
9	0.1362	0.1340	0.1360	0.1353	0.134	0.134
10	0.1564	0.1597	0.1582	0.1635	0.1592	0.1599

Table 5.13 Altitude MAE Results of UAV

When examining the tables presenting the latitude, longitude, and altitude MSE values of each model for each flight, it is evident that the results are contingent on the specific characteristics of each flight. Upon closer examination, it becomes apparent that the altitude values demonstrate higher accuracy when compared to the latitude and longitude values. Although the MSE values for each model are relatively similar, it can not be concluded that one model is superior to the others. However, it can be observed that simple models such as GRU and LSTM perform admirably on large aircraft like UAVs, despite their less complex structure. In particular, when analyzing the results of the 1st flight, it can be noted that there are deviations in the latitude estimations of all models, which could be attributed to the flight's characteristics or residual anomalies. On the other hand, all models demonstrate remarkable results in the 9th and 10th flights.In conclusion, while the performance of each model may differ depending on the flight characteristics, it can be said that all models provide acceptable results. When we delve deeper into each flight, we see that the models' performance changes depending on the flight characteristics. For example, the 1st flight showed relatively higher Latitude MAE values, while the Longitude MAE values were predicted more accurately. The 9th flight showed better results for Latitude MAE values, while Longitude MAE values showed relatively higher results. The model we proposed, DeepAT, has competitive MAE and MSE values for latitude, longitude, and altitude values and is able to produce good results for the UAV aircraft. However, the GRU model, which has a simpler structure, also performed well, thanks to the repetitive nature of the UAV's mission and its low maneuverability. Although the GRU model gives good results for individual predictions, it has the potential to quickly go off the antenna angle at some points as it lacks a sequential structure.

Flights	**DeepAT**	**StEncDecLSTM**	**EncDecGRU**	**EncDecLSTM**	**GRU**	**LSTM**
1	0.5543	0.5564	0.5642	0.5604	0.5369	0.5232
2	0.3546	0.3532	0.3474	0.3503	0.3297	0.3312
3	0.0746	0.0822	0.0853	0.0685	0.0771	0.1065
4	0.5851	0.5794	0.5852	0.5796	0.5706	0.5847
5	0.1197	0.1228	0.1199	0.1269	0.1264	0.1176
6	0.4573	0.4574	0.4420	0.4478	0.4634	0.5050
7	0.1642	0.1580	0.1590	0.1592	0.1647	0.1693
8	0.2001	0.2121	0.1897	0.2003	0.1781	0.1885
9	0.2854	0.2627	0.2947	0.2754	0.2984	0.2900
10	0.1810	0.1802	0.1781	0.1849	0.185	0.1968

Table 5.14 Latitude MSE Results of UAV

Flights	**DeepAT**	**StEncDecLSTM**	**EncDecGRU**	**EncDecLSTM**	**GRU**	**LSTM**
1	0.3012	0.2975	0.3065	0.3058	0.3145	0.3106
2	0.4149	0.4235	0.4170	0.4243	0.4503	0.4436
3	0.2804	0.2786	0.2907	0.2818	0.2659	0.2943
4	0.4240	0.4022	0.4218	0.4058	0.4113	0.4193
5	0.2893	0.2815	0.2859	0.2846	0.3010	0.2858
6	0.3268	0.3172	0.3241	0.3234	0.3190	0.3559
7	0.3206	0.3230	0.3249	0.3258	0.3137	0.3404
8	0.3309	0.3337	0.3506	0.3578	0.3245	0.2900
9	0.0513	0.0364	0.0758	0.0610	0.0587	0.0421
10	0.3153	0.3082	0.317	0.3133	0.3315	0.3342

Table 5.15 Longitude MSE Results of UAV

Flights	DeepAT	StEncDecLSTM	EncDecGRU	EncDecLSTM	GRU	LSTM
1	0.2942	0.2868	0.2879	0.2874	0.2913	0.2891
2	0.2331	0.2327	0.2339	0.2357	0.2320	0.2345
3	0.0804	0.0827	0.0821	0.0767	0.0963	0.0850
4	0.2477	0.2477	0.2478	0.2484	0.2470	0.2445
5	0.0533	0.0522	0.0535	0.0548	0.0561	0.0545
6	0.0715	0.0711	0.0712	0.0716	0.0715	0.0715
7	0.0247	0.0246	0.0242	0.0243	0.0250	0.0249
8	0.2618	0.2607	0.2615	0.2578	0.2603	0.2603
9	0.0742	0.0718	0.0740	0.0733	0.0718	0.0718
10	0.0244	0.0255	0.0250	0.0267	0.0253	0.0255

Table 5.16 Altitude MSE Results of UAV

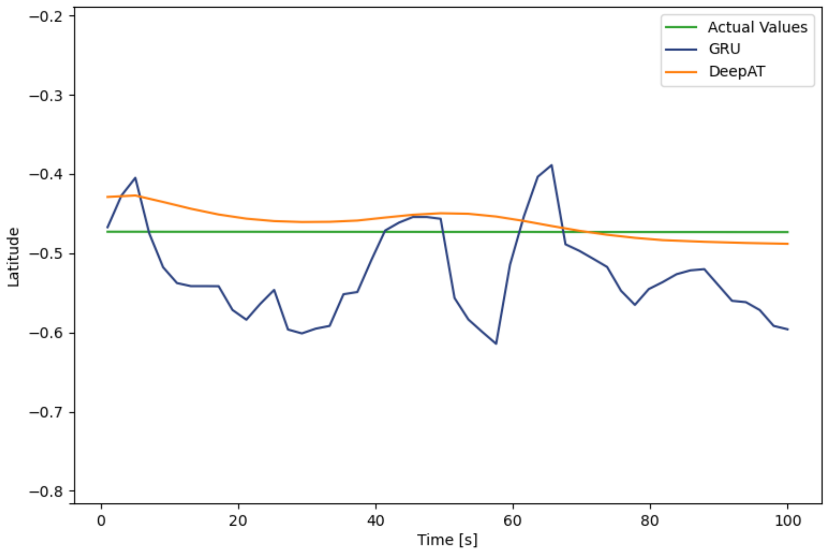

Figure 5.13 DeepAT's Latitude Result for Flight 8 in UAV

A comparative latitude prediction results between DeepAt and GRU are given in Fig. 5.13. Fig. 5.13, when analyzed, shows that the DeepAT model offers a more accurate prediction of the latitude value compared to the GRU model. The DeepAT model can be seen to converge towards the actual value with gradual and small fluctuations, while the GRU model is plagued with sudden and sharp fluctuations in its predictions. This disparity in performance can be attributed to the effectiveness of the encoder-decoder structure in DeepAT, which enables it to handle sequence predictions effectively. Despite the fluctuations in the predictions, the wide field of view of the antennas used for telemetry data collection helps mitigate the impact of small changes such as these.

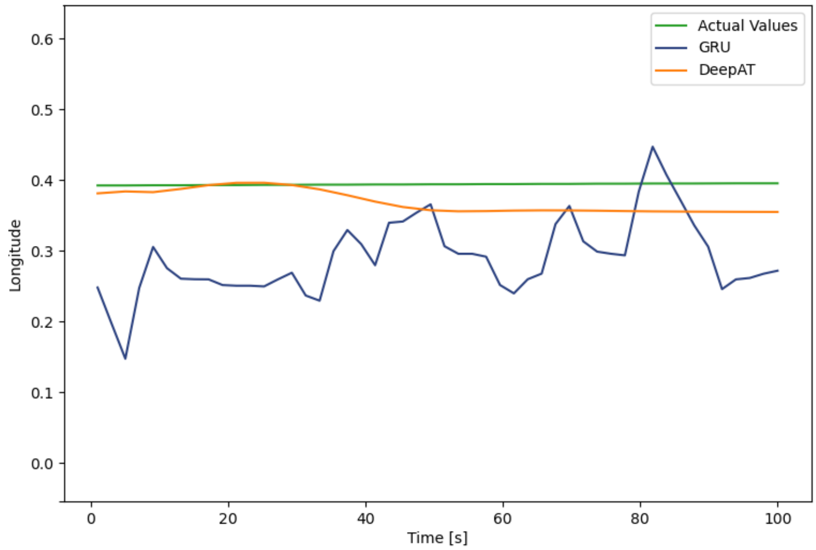

Figure 5.14 DeepAT's Longitude Result for Flight 8 in UAV

A comparative longitude prediction results between GRU and our proposed method are given in 5.14. When analyzing Fig. 5.14, it can be seen that the predictions of the GRU model deviate significantly from the actual longitude values. The GRU model's estimates have sharp increases and fluctuations, which can cause significant inaccuracies in the tracking of the aircraft. On the other hand, although the DeepAT model also has some fluctuations, it can be seen that it has a much better structure compared to the GRU model. The DeepAT model's predictions converge towards the actual values, and even in the moments where it deviates from the actual values, it remains within a close range. This indicates the success of the encoder-decoder architecture in sequence prediction. A comparative altitude prediction

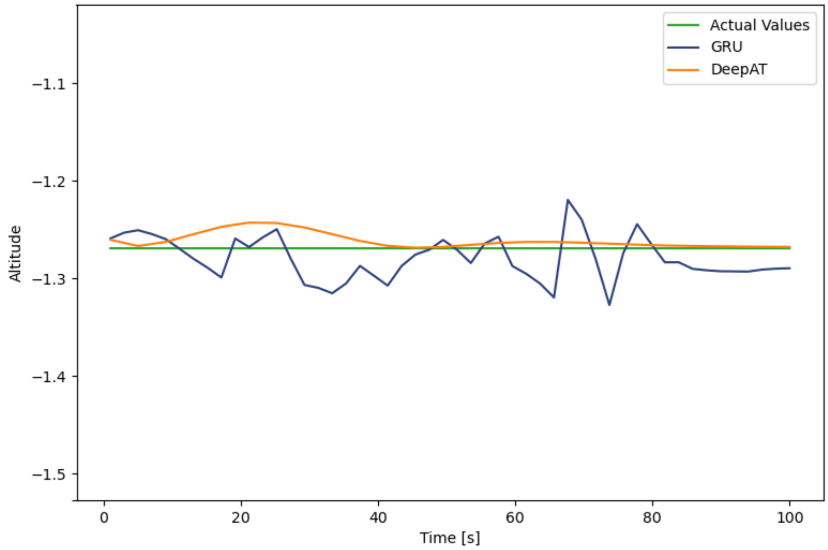

Figure 5.15 DeepAT and GRU Altitude Result for Flight 8 in UAV

results between DeepAT and GRU are given in Fig. 5.15. By examining Fig. 5.15, it is clear that the DeepAT model is able to estimate the altitude of the UAV with remarkable accuracy, despite some small deviations. This is due to the sophisticated architecture of the DeepAT model, which is specifically designed to handle complex and dynamic flight characteristics. The GRU model, on the other hand, lacks a sequential structure and although it provides good estimates at certain intervals, it falls short in accurately tracking the altitude of the aircraft. This is due to the fact that the predictions made by the GRU model are independent of each other and any large, instantaneous deviations will result in significant problems in the tracking of the aircraft. This highlights the importance of having a sequential structure in the predictive model in order to accurately track the UAV and ensure safe flight operations.

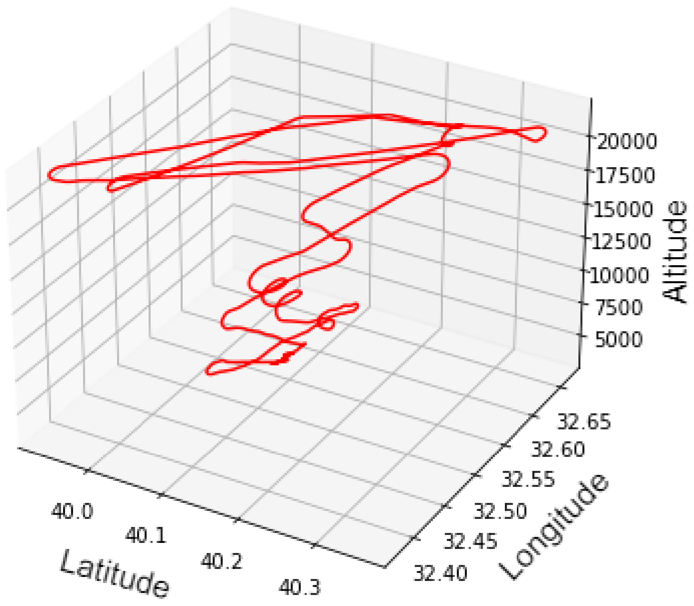

Figure 5.16 3D Flight Route Visualization for Flight 8 in UAV

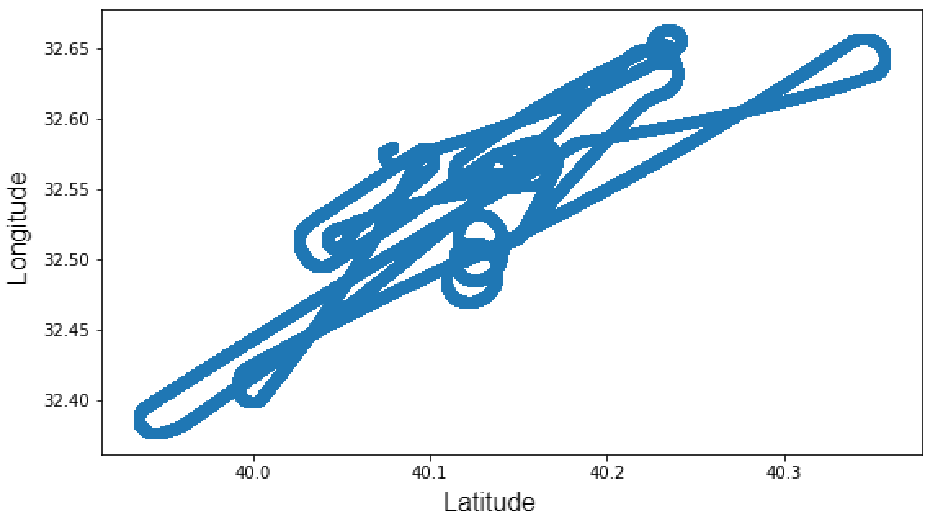

Figure 5.17 2D Flight Route Visualization for Flight 8 in UAV

Figures from Fig. 5.15 to Fig. 5.14 compare the results of our proposed DeepAT model and GRU model with 3D predictions at the flight of a UAV. Considering the results, as seen in the following Fig. 5.15, DeepAT model makes the altitude prediction better. Although it has a damped fluctuation, it does not disturb the sequential structure. On the other hand, although GRU is good at single-point predictions, it cannot provide a sequential structure.

Similar results are observed for latitude and longitude predictions. The UAV continuously draws circles during this flight, as seen in Fig. 5.16 and 5.17, which means that data that is substantially similar to the preceding data is repeated.

	Latitude	**Longitude**	**Altitude**
DeepAT	0.4096	0.7518	0.3159
StEncDecLSTM	0.4336	0.8351	0.3319
EncDecGRU	0.4017	1.1851	0.343
EncDecLSTM	0.5491	1.2729	0.3646
GRU	0.3977	1.2715	0.4697
LSTM	0.5198	1.3904	0.4697

Table 5.17 Maximum Deviation Between Actual and Predicted Values in Flight 8

Table 5.17 illustrates the maximum deviation between the actual and predicted points for flight 8. When looking at the maximum deviation in the predictions, it becomes clear that the encoder-decoder architecture has a significant impact on the estimations of longitude and altitude. It can be observed that models such as GRU and LSTM tend to have high deviations, particularly in the longitude predictions, while models based on encoder-decoder architecture demonstrate much more precise predictions within a relatively narrow range of deviation. This highlights the superiority of encoder-decoder based models when it comes to making accurate predictions in sequential data.

Fig. 5.18 displays the results of the latitude predictions made by the DeepAT and GRU models in a sequence where the latitude value is highly variable. It can be seen that the GRU model has made a substantial improvement compared to its initial predictions at the start of the sequence, and has been able to get closer to the actual latitude values. However, the fluctuations in its predictions and the large margin of errors demonstrate that it still has some difficulty in making consistent estimations. On the other hand, the DeepAT model has demonstrated remarkable consistency and accuracy in its predictions, due to its effective use of sequential structure. It starts off with an estimation close to the true latitude value and continues to produce predictions that are in close proximity to the actual data. In this sequence, although the GRU model might perform better in single-point estimations, the DeepAT model's performance is more robust and reliable due to its sequential approach.

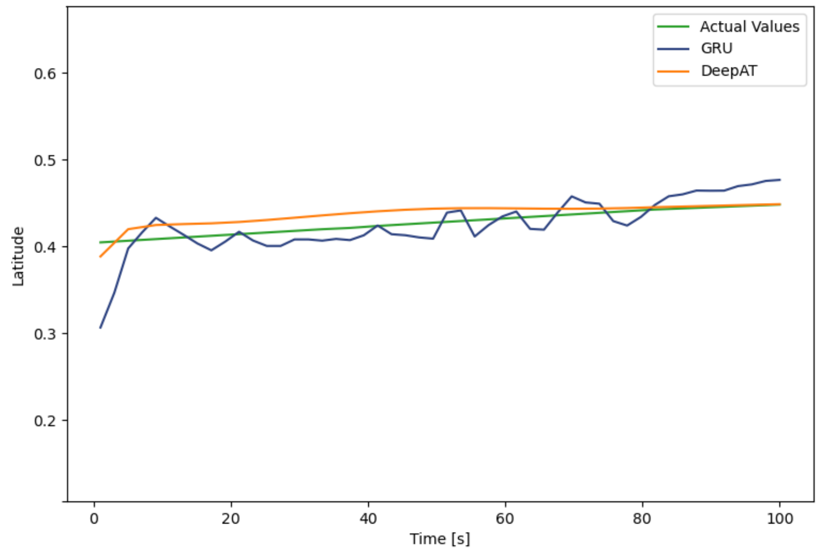

Figure 5.18 DeepAT and GRU Latitude Result for Flight 10 in UAV

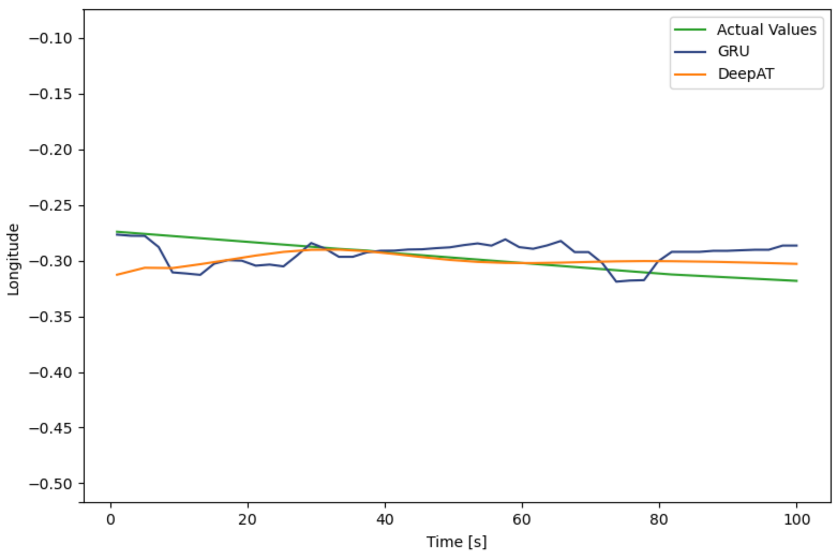

Figure 5.19 DeepAT and GRU Longitude Result for Flight 10 in UAV

Fig. 5.19 demonstrates the performance of the DeepAT and GRU models in predicting longitude values in a sequence where the longitude is constantly changing. The DeepAT model shows remarkable resilience in overcoming its initial estimation error, quickly adjusting its predictions to align closely with the actual longitude values. Meanwhile, although the GRU model appears to have a degree of sequential structure, its predictions are plagued by persistent fluctuations and deviations from the true longitude values. These findings suggest that the DeepAT model may be more effective in accurately predicting longitude values in dynamic environments, due to its ability to rapidly adapt to changes in the data.

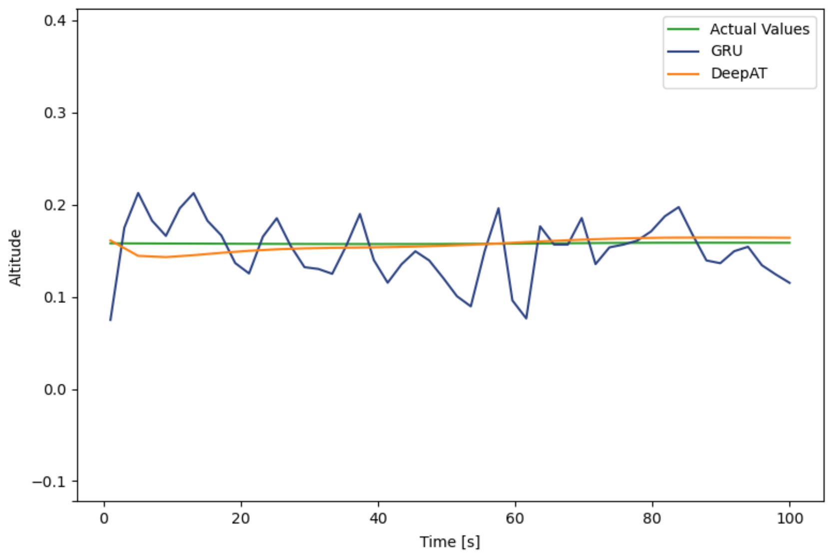

Figure 5.20 DeepAT and GRU Altitude Result for Flight 10 in UAV

As can be seen from Fig. 5.20, the results of the DeepAT and GRU models in estimating the altitude of the aircraft are quite different from each other. The GRU model, which struggles to provide continuous and consistent predictions in this sequence, tends to fluctuate greatly and deviate from the actual data points. In contrast, the DeepAT model demonstrates its superiority by starting with predictions that are close to the actual data points, and by leveraging its sequential structure, it produces reliable and accurate results throughout the flight. These predictions are essential in ensuring the safe and controlled navigation of the aircraft, and the performance of the DeepAT model in this regard is particularly noteworthy. This highlights the ability of the DeepAT model's sequential structure to make accurate predictions in a new and different flight scenario, demonstrating its robustness and reliability in estimating altitude values.

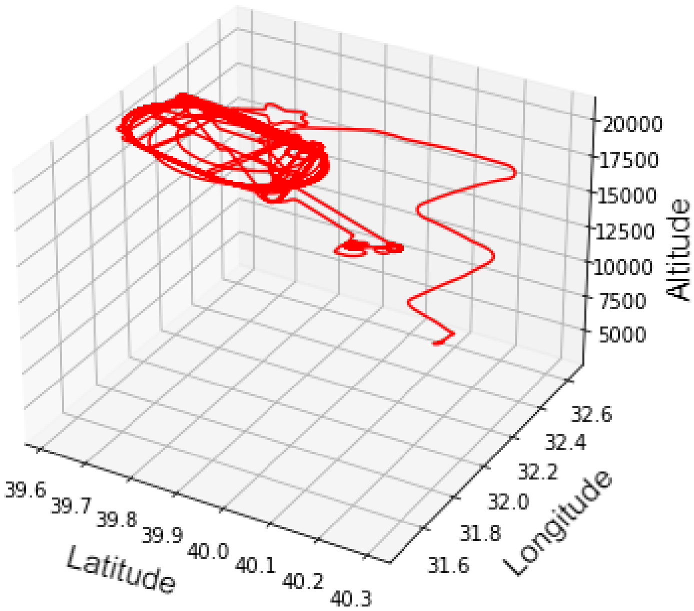

Figure 5.21 3D Flight Route Visualization for Flight 10 in UAV

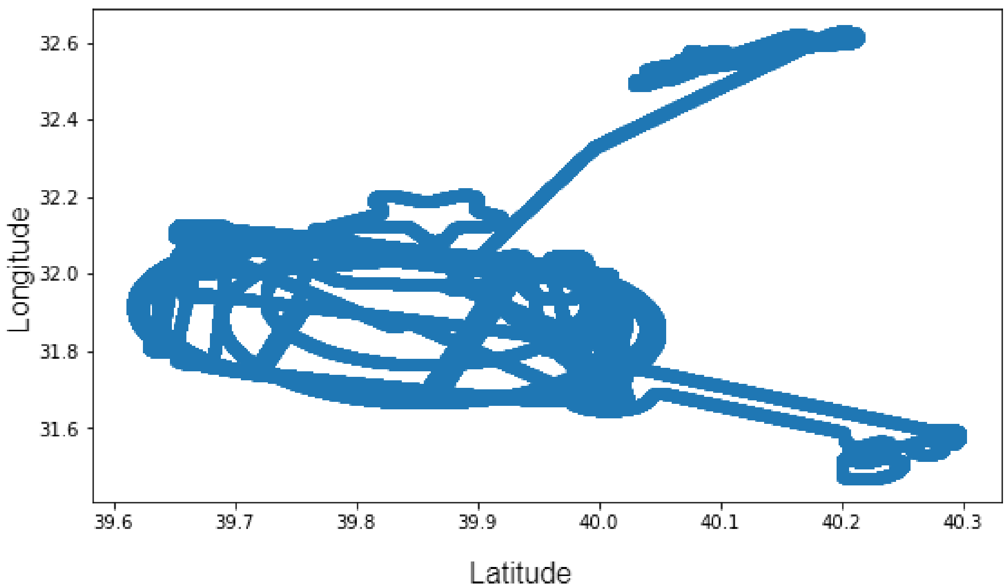

Figure 5.22 2D Flight Route Visualization for Flight 10 in UAV

Figures from Fig. 5.20 to Fig. 5.19 compare the results of our proposed DeepAT model and GRU model with 3D predictions at a specific type of a flight of a UAV. This flight is unique because the UAV does quick maneuvers that it hasn't done before and is not used to executing. As seen in Fig. 5.21 and Fig. 5.22, in this flight, the UAV draws the Turkish flag in the sky with the route it follows. This flight was made during the 19 May Commemoration of Atatürk, Youth and Sports Day, which is a national holiday. We are interested in this flight because the UAV makes sharp maneuvers in an attempt to depict stars and crescents. The MAE and MSE values provided by DeepAT model and GRU model we suggest for this flight are extremely similar. DeepAT model, which we propose using for this flight, has an MAE value of 0.1065, compared to GRU model's MAE value of 0.0955. We may conclude that DeepAT model we provide performs better in sharp maneuvering flights or during predictions of the next position of aircraft with sharp maneuverability after looking at flight examples of fixed-wing propeller aircraft beside this example.

	Latitude	**Longitude**	**Altitude**
DeepAT	0.7491	0.8609	0.1669
StEncDecLSTM	0.7845	0.8928	0.1649
EncDecGRU	1.0835	0.8577	0.2264
EncDecLSTM	1.1364	0.8729	0.2287
GRU	1.4778	1.3624	0.2627
LSTM	1.4535	1.2431	0.2644

Table 5.18 Maximum Deviation Between Actual and Predicted Values in Flight 10

Table 5.18 illustrates the maximum deviation between the actual and predicted points for flight 10. When analyzing the maximum deviation in detail, it becomes apparent that the encoder-decoder based models have a significant impact on the prediction results for both longitude, latitude, and altitude values. While GRU and LSTM models tend to produce high deviations, particularly in longitude predictions, encoder-decoder based models are capable of making predictions with much smaller deviation margins. Although each model performs well in terms of altitude predictions, it is evident that the encoder-decoder based methods produce outstanding results in terms of the maximum deviation metric. This highlights the effectiveness and reliability of these models in ensuring accurate and consistent predictions.

Model	MAE Latitude	Longitude	Altitude	**Metrics**	**Dataset**
DeepAT	0.1493	0.1553	0.0983	MAE/MSE	Telemetry
DeepAT	0.2228	0.2333	0.1654	MAE/MSE	Telemetry
LSTM [5]	0.0725	0.0552	77.9472	MAE/RMSE	ADS-B
WMM [5]	0.0788	0.0910	141.3293	MAE/RMSE	ADS-B
MM [5]	0.1065	0.1045	159.9611	MAE/RMSE	ADS-B
CNN-LSTM [26]	0.0170	0.0710	33.8320	MAE/RMSE	ADS-B
LSTM [26]	0.0170	0.0890	44.3250	MAE/RMSE	ADS-B
BP [26]	0.0460	0.1740	77.2540	MAE/RMSE	ADS-B
D-GRU [36]	4.4000	5.1700	1.5000	MAE/RMSE	Quadrotor
GRU [36]	4.9800	5.5200	1.5900	MAE/RMSE	Quadrotor

Table 5.19 Literature Comparison of DeepAT

Table 5.19 shows the comparison of the DeepAT model with different studies in the literature. Each model created in the literature has its own fundamental differences, so we cannot call this comparison an equal comparison. However, studies that used methods similar to the ones we used in the literature are shown in Table 5.19. In the studies, unlike our thesis, the RMSE value was chosen as the evaluation metric. At the same time, our thesis makes predictions in an area where there are no exact studies in the literature, as the aircraft type it is interested in. The studies in the literature are mostly concentrated on small quadrotors and commercial aircraft. DeepAT, on the other hand, makes its predictions on two very different aircraft than the studies. One of these aircraft is a highly maneuverable, propeller and very fast aircraft, and the other is a large wingspan UAV that can carry loads at high weights. On the other hand, the datasets and preprocessing steps used are also different in each study. When a rough comparison is made, the DeepAT model gives outstanding performance in the prediction of fixed-wing propeller aircraft than the studies in the literature. Since there is no similar study with similar metrics and aircraft type, comparison of UAV with the studies in the literature could not be made.

6. CONCLUSION

This thesis proposes a hybrid Stacked Encoder-Decoder GRU model named DeepAT for 3D next-location prediction. This model is implemented on a real flight dataset that is collected from the ground telemetry system in flight test phases. Two different types of aircraft are analyzed differently and extensively: fixed-winged propeller aircraft and UAV. This thesis aims to solve the problem of data loss in flight test phases. Knowing the next location of the aircraft gives us an upper hand in tracking the aircraft in real-time. This model is compared with state-of-the-art techniques (LSTM, GRU, Encoder-Decoder LSTM, Encoder-Decoder GRU) for sequence prediction. MAE and MSE metrics are used as evaluation metrics. According to the experimental results, DeepAT shows superior performance and predictions in propeller aircraft and gives similar results with GRU for UAVs. In addition, when compared to GRU model, our proposed DeepAT model achieves better performance for sequential structure. The experimental results show that using GRU-based Stacked Encoder-Decoder architecture is good at solving time-series sequencing problems in real-time. We can state that DeepAT model performs significantly better against sharp maneuvers than GRU and that DeepAT model performs similarly to GRU even during flights of the rather sluggish and cumbersome UAV. For two different aircraft, the proposed DeepAT model is contrasted with five other state-of-the-art methods. For the fixed-wing propeller aircraft, we acquire the values of 0.1493 for latitude MAE, 0.1553 for longitude MAE and 0.0983 for altitude MAE values. At the same time we acquire 0.0585 for latitude MSE, 0.0593 for longitude MSE and 0.0356 for altitude MSE values. Also, for the UAV, we obtain the values of 0.2228 for latitude MAE, 0.0233 for longitude MAE and 0.1654 for altitude MAE values. At the same time we acquire 0.2976 for latitude MSE, 0.3054 for longitude MSE and 0.1365 altitude MSE. While DeepAT model achieves superior results to all other methods for fixed-wing aircraft, it produces competitive results to each model for UAV flights.

REFERENCES

[1] Kate M Pavlock. Aerospace engineering handbook chapter 2 (v): Flight test engineering. Technical report, NASA Technical Reports Server.

[2] Yuhao Wang, Yutian Pang, Yongming Liu, Parikshit Dutta, and Bong-Jun Yang. Aircraft trajectory prediction and risk assessment using bayesian updating. In *Aiaa aviation 2019 forum*, page 2936. **2019**.

[3] Kwangyul Baek and Hyochoong Bang. Ads-b based trajectory prediction and conflict detection for air traffic management. *International Journal Aeronautical and Space Sciences*, 13(3):377–385, **2012**.

[4] Shane T Barratt, Mykel J Kochenderfer, and Stephen P Boyd. Learning probabilistic trajectory models of aircraft in terminal airspace from position data. *IEEE Transactions on Intelligent Transportation Systems*, 20(9):3536–3545, **2018**.

[5] Zhiyuan Shi, Min Xu, Quan Pan, Bing Yan, and Haimin Zhang. Lstm-based flight trajectory prediction. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, **2018**.

[6] Risto Miikkulainen, Jason Liang, Elliot Meyerson, Aditya Rawal, Daniel Fink, Olivier Francon, Bala Raju, Hormoz Shahrzad, Arshak Navruzyan, Nigel Duffy, et al. Evolving deep neural networks. In *Artificial intelligence in the age of neural networks and brain computing*, pages 293–312. Elsevier, **2019**.

[7] Tingwu Yang, Yufeng Yang, and Ting Zhang. *Telemetry Theory and Methods in Flight Test*. Springer, **2021**.

[8] Muhammed Emir çakıcı, Feyza Yıldırım Okay, and Suat Özdemir. Real-time aircraft tracking system: A survey and a deep learning based model. In *2021 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–6. **2021**. doi:10.1109/ISNCC52172.2021.9615681.

[9] Jun Zhang and Kim-Fung Man. Time series prediction using rnn in multi-dimension embedding phase space. In *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218)*, volume 2, pages 1868–1873. IEEE, **1998**.

[10] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, **2015**.

[11] Loïc Bontemps, Van Loi Cao, James McDermott, and Nhien-An Le-Khac. Collective anomaly detection based on long short-term memory recurrent neural networks. In *International conference on future data and security engineering*, pages 141–152. Springer, **2016**.

[12] Mohamed Abdel-Nasser and Karar Mahmoud. Accurate photovoltaic power forecasting models using deep lstm-rnn. *Neural Computing and Applications*, 31(7):2727–2740, **2019**.

[13] Mohammad J Hamayel and Amani Yousef Owda. A novel cryptocurrency price prediction model using gru, lstm and bi-lstm machine learning algorithms. *AI*, 2(4):477–496, **2021**.

[14] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, **2014**.

[15] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, **2014**.

[16] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, **2014**.

[17] Samet Ayhan and Hanan Samet. Aircraft trajectory prediction made easy with predictive analytics. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 21–30. **2016**.

[18] Portia Banerjee and Matteo Corbetta. In-time uav flight-trajectory estimation and tracking using bayesian filters. In *2020 IEEE Aerospace Conference*, pages 1–9. IEEE, **2020**.

[19] Arjen De Leege, Marinus van Paassen, and Max Mulder. A machine learning approach to trajectory prediction. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, page 4782. **2013**.

[20] Esther Calvo Fernández, José Manuel Cordero, George Vouros, Nikos Pelekis, Theocharis Kravaris, Harris Georgiou, Georg Fuchs, Natalya Andrienko, Gennady Andrienko, Enrique Casado, et al. Dart: a machine-learning approach to trajectory prediction and demand-capacity balancing. *SESAR Innovation Days, Belgrade*, pages 28–30, **2017**.

[21] Nikolas Julio, Ricardo Giesen, and Pedro Lizana. Real-time prediction of bus travel speeds using traffic shockwaves and machine learning algorithms. *Research in Transportation Economics*, 59:250–257, **2016**.

[22] Zhengyi Wang, Man Liang, and Daniel Delahaye. Short-term 4d trajectory prediction using machine learning methods. In *Proc. SID*, pages 1–10. **2017**.

[23] Hesam Shafienya and Amelia C Regan. 4d flight trajectory prediction using a hybrid deep learning prediction method based on ads-b technology: A case study of hartsfield–jackson atlanta international airport (atl). *Transportation Research Part C: Emerging Technologies*, 144:103878, **2022**.

[24] Lan Ma, Shan Tian, and Zhi-Jun Wu. A 4d trajectory prediction model based on the bp neural network. **2019**.

[25] Ping Han, Wenqing Wang, Qingyan Shi, and Jucai Yue. A combined online-learning model with k-means clustering and gru neural networks for trajectory prediction. *Ad Hoc Networks*, 117:102476, **2021**.

[26] Lan Ma and Shan Tian. A hybrid cnn-lstm model for aircraft 4d trajectory prediction. *IEEE access*, 8:134668–134680, **2020**.

[27] Ziyu Zhao, Weili Zeng, Zhibin Quan, Mengfei Chen, and Zhao Yang. Aircraft trajectory prediction using deep long short-term memory networks. *Proceedings of the CICTP*, **2019**.

[28] Xiaoge Zhang and Sankaran Mahadevan. Bayesian neural networks for flight trajectory prediction and safety assessment. *Decision Support Systems*, 131:113246, **2020**.

[29] Yutian Pang, Xinyu Zhao, Hao Yan, and Yongming Liu. Data-driven trajectory prediction with weather uncertainties: A bayesian deep learning approach. *Transportation Research Part C: Emerging Technologies*, 130:103326, **2021**.

[30] Cheng Cheng, Liang Guo, Tong Wu, Jinlong Sun, Guan Gui, Bamidele Adebisi, Haris Gacanin, and Hikmet Sari. Machine-learning-aided trajectory prediction and conflict detection for internet of aerial vehicles. *IEEE Internet of Things Journal*, 9(8):5882–5894, **2021**.

[31] Seyed Mohammad Hashemi, Ruxandra Mihaela Botez, and Teodor Lucian Grigorie. New reliability studies of data-driven aircraft trajectory prediction. *Aerospace*, 7(10):145, **2020**.

[32] Kai Zhang and Bowen Chen. Phased flight trajectory prediction with deep learning. *arXiv preprint arXiv:2203.09033*, **2022**.

[33] Yulin Liu and Mark Hansen. Predicting aircraft trajectories: a deep generative convolutional recurrent neural networks approach. *arXiv preprint arXiv:1812.11670*, **2018**.

[34] Ping Han, Wenqing Wang, Qingyan Shi, and Jun Yang. Real-time short-term trajectory prediction based on gru neural network. In *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, pages 1–8. IEEE, **2019**.

[35] Lan You, Siyu Xiao, Qingxi Peng, Christophe Claramunt, Xuewei Han, Zhengyi Guan, and Jiahe Zhang. St-seq2seq: a spatio-temporal feature-optimized seq2seq model for short-term vessel trajectory prediction. *IEEE Access*, 8:218565–218574, **2020**.

[36] Zhao Yang, Rong Tang, Jie Bao, Jiahuan Lu, and Zhijie Zhang. A real-time trajectory prediction method of small-scale quadrotors based on gps data and neural network. *Sensors*, 20(24):7061, **2020**.

[37] Zhizhou Zhang, Zhenglei Wei, Bowen Nie, and Yang Li. Discontinuous maneuver trajectory prediction based on hoa-gru method for the uavs. *Electronic Research Archive*, 30(8):3111–3129, **2022**.

[38] Lei Xie, Zhenglei Wei, Dali Ding, Zhuoran Zhang, and Andi Tang. Long and short term maneuver trajectory prediction of ucav based on deep learning. *IEEE Access*, 9:32321–32340, **2021**.

[39] Rong Tang, Zhao Yang, Jiahuan Lu, Hao Liu, and Honghai Zhang. Real-time trajectory prediction of unmanned aircraft vehicles based on gated recurrent unit. In *Green Connected Automated Transportation and Safety*, pages 585–596. Springer, **2022**.

[40] Songwon Seo. *A review and comparison of methods for detecting outliers in univariate data sets*. Ph.D. thesis, University of Pittsburgh, **2006**.