Comparative Analysis of Machine Learning and Statistical Methods for Aircraft Phase of Flight Prediction

Stephen Kovarik, Liam Doherty,
Kiran Korah, Brian Mulligan,
Ghulam Rasool, Yusuf Mehta
Henry M. Rowan College of Engineering
Rowan University
Glassboro, United States of America

Parth Bhavsar
Department of Civil and Construction Engineering
Kennesaw State University
Marietta, United States of America
pbhavsar@kennesaw.edu

Mike Paglione
Aviation Research Division
Federal Aviation Administration
Atlantic City, United States of America

Abstract— **Phase of flight (POF) prediction estimates the future state of aircraft along planned trajectories, allowing the prediction of potential conflicts as well as optimization of air space, controlled by the Federal Aviation Administration. In this paper, we present a study conducted to develop three different POF forecasting machine learning models and a statistical regression model using four-dimensional GPS and RADAR Track data from 57 flights provided by an En Route Computer System. The investigated machine learning models include Long Short-Term Memory Recurrent Neural Network (LSTM-RNN), Support Vector Machine (SVM), and Neural Ordinary Differential Equations (NODE). These were developed to forecast the horizontal and vertical POF of the current aircraft for the next time step. The results in this study indicate that LSTM-RNN models are more suitable for POF prediction than SVM and statistical regression models, with NODE being a promising model for future trajectory prediction research.**

Keywords-Trajectory Prediction; Machine Learning; Long Short-Term Memory; Support Vector Machine; Neural Ordinary Differential Equations; Regression; Phase of Flight

I. INTRODUCTION

The Federal Aviation Administration (FAA) and other related global organizations routinely use trajectory prediction to estimate the position of an aircraft when it deviates from its planned route. Lateral deviation is commonplace in aviation for a multitude of reasons; different conflicts arise between takeoff and touchdown that aircraft must maneuver around for safe and efficient travel. Inclement weather, turbulence, pilot behavior, and the presence of other aircraft nearby are some of the primary reasons that force the subject aircraft to alter the course of travel. The circumstances vary for each flight, so different avoidance tactics are warranted. According to Dupuy and Porretta [3], the current trajectory prediction methodology is broken down into four parts: (i) preparation, (ii) trajectory prediction, (iii) trajectory export, and (iv) trajectory update. Preparation involves generating a flight script that defines instructions for each segment of the trajectory. Trajectory Prediction uses computational methods and algorithms to turn the flight script into a trajectory using models such as aircraft performance and meteorological data.

The particular state of the aircraft is critical to the trajectory prediction process, such as position, speed, and phase of flight (POF). The goal of this study was to develop a model that utilizes artificial intelligence (AI) to predict vertical and horizontal POF, which is a critical part of trajectory prediction. This paper presents the development of three machine learning models for predicting aircraft vertical and horizontal POF as well as a statistical regression algorithm for developing ground truth datasets. These machine learning models are Support Vector Machine (SVM), Long Short-Term Memory Recurrent Neural Network (LSTM-RNN), and Neural Ordinary Differential Equations (NODE). The statistical algorithm that was used as a foundation for this paper and the data used for training and testing was provided by the FAA and is explained in detail in a research paper published by Paglione and Oaks in 2006 [17].

II. RELATED WORK

A. *Existing Trajectory Prediction Practices*

Every aircraft is equipped with radio transponders that RADAR systems can use to identify the flight. Most aircraft are equipped with a Global Positioning System (GPS), which provides the position of the aircraft during flight. RADAR systems and GPS provide the ground truth data about the position of the aircraft, which can be utilized to (1) develop trajectory prediction models as well as (2) evaluate the accuracy of these models during post-analysis. However, the standard data transmission interval of RADAR systems (typically 12 seconds) and the noise in the measurements (i.e., sensor data, specifically RADAR sensor) are primary limitations that impact estimation of actual flight path [2]. The current practice is to obtain an accurate tracking function to ensure efficient performance in air traffic control despite this shortcoming. Trajectory estimation is performed to create a better representation of the aircraft's

This project is a result of an active collaborative effort between undergraduate researchers supported by Rowan University CREATEs (Center for Research and Education in Advanced Transportation Engineering) with data and technical expertise provided by the FAA.

actual flight path, which can be achieved in two ways. The first method involves developing a model that utilizes an aircraft's position to estimate the trajectory of its remaining path. The second method includes determining the lateral deviation from the actual flight path and utilizing this information to estimate the remaining trajectory.

In addition to RADAR track and GPS positioning data, researchers have utilized various algorithms with additional parameters to analyze flight data. Some methods include geometric properties, for example, algorithms that consider the angle between the heading of the plane after deviation occurs compared to its original track [1]. Along with coordinate data for latitude, longitude, and altitude, trigonometric calculations can determine maneuvers for the plane to return to the optimal route of travel. Groundspeed can also be utilized for kinematic modeling or other physics-based models. Once thresholds for permissible values are established, algorithms can determine if a flight is following its intended path, and if not, it can determine how much difference in position lies between the actual and theoretical position [17]. This statistical approach is one of the methods used in this study. The outcomes of the statistical approach were compared with machine learning models.

B. Limitations

A fundamental limitation for current trajectory prediction methods is the accuracy of the GPS and RADAR track measurements. Accuracy is evaluated by running simulations to examine phases where the flight follows a straight trajectory and where it performs a maneuver. Paglione and Ryan, 2005, analyzed several flights for measurement accuracy and, for one flight sample, reported that "The radar track swung wide of the GPS positions, being offset by 0.33 nm, and lagged the GPS positions by several seconds" [2]. This study compares the forecasting performance of each model, using both GPS and RADAR track data. Other studies have identified potential room for error in the data, which are used to inform the trajectory prediction system. Dupuy and Porretta [3] argue that uncertainty arises from aircraft intent, current aircraft position, aircraft performance library data, and meteorological library data. This uncertainty is further compounded by the use of different mathematical models when predicting the trajectory, which may model the aircraft using either center of gravity and angular velocity or forces and moments. Uzun and Koyuncu [4] confirmed this observation by demonstrating how takeoff mass affects climb speed in a way that is not accounted for in current trajectory prediction systems. They also establish the framework for a system that would not only predict the trajectory of the aircraft and its overall flightpath but also actively adapt to changes in route to reduce the error, as mentioned above [4]. The desire to modify these existing systems with real-time aircraft positional data is the motivation for developing improved trajectory prediction, which considers these and other proposed parameters.

C. Phase of Flight Prediction

The National Transportation Safety Board (NTSB) defines the phases of flight as distinct, standardized characterizations of the different possible "period[s] within a flight" [5]. The NTSB establishes a taxonomy for the individual phases to allow for better reporting of incidents and clarity across multiple aviation industries. An increasing amount of research, spurred in part by the FAA's NextGen initiative [6], is drawing from this knowledge of the phases of flight to create an increasingly robust trajectory prediction system to improve air travel safety and decrease the burden on air traffic controllers. Knowing the active phase of an aircraft in real-time, as well as predicting when an aircraft is likely to transition between phases, is vital to modernizing trajectory prediction systems. The research in this paper is concerned with the determination of an aircraft's horizontal and vertical POF.

III. CURRENT AND PROPOSED TRAJECTORY & PHASE OF FLIGHT PREDICTION SYSTEMS

The current research demonstrates that there are multiple approaches to change or improve trajectory prediction. Table 1 is a matrix table that shows the parameters and methods of each study. This detailed review of the current and proposed prediction systems helped the research team to determine the direction of the research method outlined in this paper.

IV. METHOD

The research approach for this study involved the re-creation of a regression algorithm and the development of SVM, LSTM-RNN, and NODE machine learning models. The first step was to prepare the 'ground truth' dataset for training the machine learning models. 'Ground truth' refers to the dataset with the aircraft's true POF for each recorded timestep. The data, provided by the FAA, includes 57 flights recorded between January and February 2005 from the Salt Lake City Air Route Traffic Control Center. This is the same dataset used in [17]. The purpose of re-creating the regression algorithm was to verify the ground truth dataset and to create a baseline for comparison with the machine learning models. The results that were obtained are useful for further research and development for multi-step time series forecasting models with additional layers and dataset parameters.

TABLE I. OUTLINE OF CURRENT AND PROPOSED TRAJECTORY & PHASE OF FLIGHT PREDICTION SYSTEMS

Parameter	4D	Rate of climb or descent	TAS	Mass	Route	Intent Language	Machine Learning	Wind	Genetic
3D Position	[7] [8] [9] [10] [11] [12]	[13] [3] [14]	[15] [3] [16]		[1] [3] [17]	[18] [19] [20] [21]	[22] [23] [24] [25]	[16]	[26] [27]
Time	[7] [8] [9] [10] [11] [12]	[13] [3] [14]			[1] [3] [17]	[18] [19] [20] [21]			[26] [27]
Airspeed		[14] [3]	[15] [3] [16]	[4]	[1] [3] [17]	[18] [19] [20] [21]	[22] [23] [24] [25]	[16]	
Altitude		[13] [3] [14]	[3]	[28] [29] [4]	[1] [3] [17]	[18] [19] [20] [21]	[22] [23] [24] [25]	[16]	
Mass				[28] [29] [4]		[18] [19] [20] [21]	[22] [23] [24] [25]		[26] [27]
Other Physical Features				[28] [29] [4]		[18] [19] [20] [21]	[22] [23] [24] [25]		[26] [27]
Environment		[13] [3] [14]	[15] [3] [16]			[18] [19] [20] [21]	[22] [23] [24] [25]	[30] [31] [32] [16]	[26] [27]
Flight Plan					[1] [3] [17]	[18] [19] [20] [21]			[26] [27]
Control Input					[1] [3] [17]	[18] [19] [20] [21]	[22] [23] [24] [25]	[16]	[26] [27]
Intent/Language					[1] [3] [17]	[18] [19] [20] [21]	[22] [23] [24] [25]		[26] [27]
Angle					[1] [3] [17]	[18] [19] [20] [21]	[22] [23] [24] [25]	[16]	
Thrust						[18] [19] [20] [21]	[22] [23] [24] [25]		
Lift		[14]				[18] [19] [20] [21]	[22] [23] [24] [25]		
Drag		[14]		[28] [29] [4]		[18] [19] [20] [21]	[22] [23] [24] [25]	[30] [31] [32] [16]	
Performance		[14]		[28] [29] [4]		[18] [19] [20] [21]	[22] [23] [24] [25]		[26] [27]
Energy rate		[14]	[15] [3] [16]	[28] [29] [4]		[18] [19] [20] [21]			

V. DATA

The algorithms were trained and tested with 4D aircraft positional data provided by an En Route host Computer System (HCS). For each aircraft (labeled *acid_cid*, unique aircraft identifier), the train and test datasets consisted of time (*time*), an x-position (*xCoord*), a y-position (*yCoord*), and an altitude (*altitude*) associated with each timestep. Horizontal POF was available for training, labeled *pofHorz* with outputs straight (STR), and turn (TRN). For machine learning purposes, *pofHorz* was converted to binary values, 0 and 1, for TRN and STR, respectively. Similarly, vertical POF was labeled *pofVert* with outputs descending (DSC), ascending (ASC), and level (LVL). *PofVert* was converted to -1, 1, and 0, respectively.

The only ground truth vertical POF data that was available for this study was GPS data. GPS data consisted of 103070 timesteps for 57 flights and RADAR track data at 11307 unique data points for 57 flights. Both GPS and RADAR track datasets were then separated by *acid_cid* for training and testing, giving 57 unique datasets. Training and testing datasets were created with a 2:1 ratio. There was an average class imbalance of 87.6/12.4 percent. For the machine learning models, class weights were implemented to place more emphasis on the minority classes for the classifiers to learn equally from all classes. The RADAR track data has more noise than the GPS data. The method of data transmission has a significant impact on how noisy a dataset is. The RADAR track data used in this study was interpolated to have timesteps of ten seconds, whereas the GPS data has one-second timesteps. The performance of the models using GPS and RADAR track data is analyzed by taking into account this noise in the results section.

VI. MODELS

A. Regression

The Python script that was developed for the regression model used the algorithm outlined in [17]. To verify the accuracy of the Python script developed for this study, the set of flight data and accuracy metrics from [17], were used. The development of this statistical approach also helped verify that the datasets being used to train and test the machine learning models presented in this paper were the same datasets used in [17].

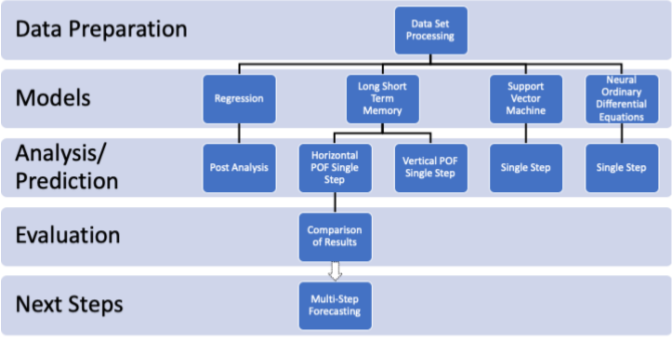

Figure I. Research Approach

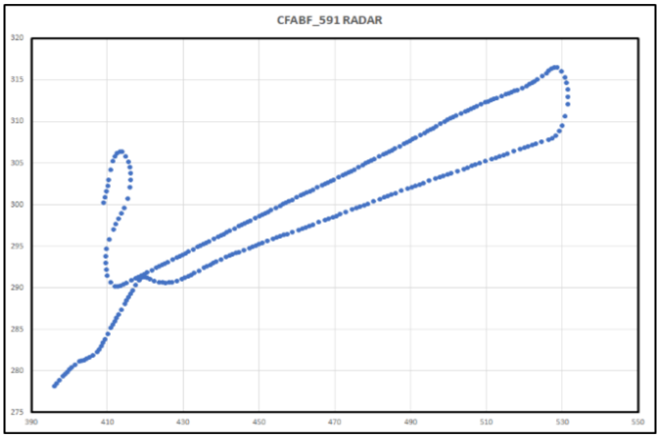

Figure II. 2D Trajectory – RADAR Track Data

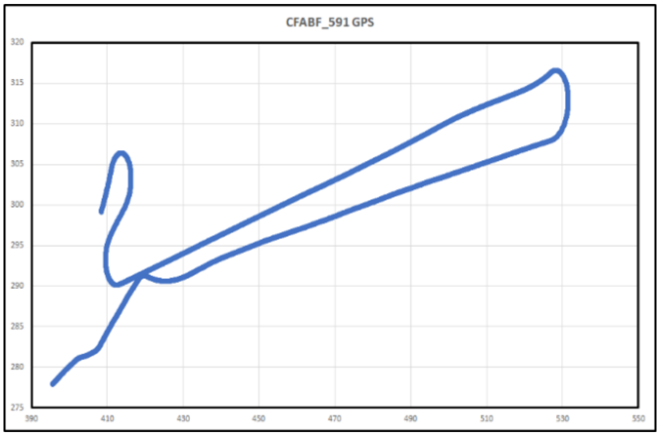

Figure III. 2D Trajectory - GPS Data

1) *Horizontal POF*

The algorithm uses two tiers of testing to determine whether the aircraft is considered to be flying straight or turning. These tiers of testing are inherently conditional statements containing thresholds that determine the POF of an aircraft at a given timestep.

If the first tier isn't conclusive enough to determine the POF that the aircraft is in, then the second tier of testing is used. The first tier uses the Pearson correlation coefficient (Pearson's r) for the XY coordinates in a time window surrounding the current row of flight data. This coefficient has a value between -1 and +1, which represents the linearity between different variables. The two thresholds that make up the first tier of testing are *horzPearsonThreshHi* and *horzPearsonThreshLo* [17]. If the Pearson's r value was greater than the *horzPearsonThreshHi*, the aircraft is determined to be flying straight. If the value is less than the *horzPearsonThreshLo*, the aircraft is determined to be turning [17]. The SciPy Python library was used for calculating Pearson's r and for deriving the linear regression equation [44].

The second tier of testing uses quadratic regression analysis and a flatness metric to provide the final POF determination. This tier can be seen as a filter to help reduce noise in the dataset [17]. Numpy was used for the rotation of data, the creation of the polynomial, and the flatness metric [47].

TABLE II. HORIZONTAL AND VERTICAL THRESHOLDS

Threshold*	Value
Vertical	
vertSlopeThresh	0.5, 1.5, 2.5
vertTimeWindow	25, 55, 120
Horizontal	
horzPearsonThreshHi	0.998, 0.997, 0.995
horzPearsonThreshLo	0.6, 0.1
horzRSqrThresh	0.92, 0.82, 0.4
horzFlatnessThresh	0.25, 0.1
horzTimeWindow	25, 40, 50, 60, 70, 80

**Threshold definitions are provided in Appendix A.*

2) *Vertical POF*

A detailed algorithm is given in [17], which calculates the slope of altitude data for a given time window to determine vertical POF. The SciPy Stats library was used to calculate the slope for the time window surrounding each timestep [44]. The parameters that can be changed in this program to lower the probability of error are the vertical slope threshold, vertSlopeThresh, and the vertical time window, vertTimeWindow. A design of experiments study was performed by [17] to determine the best set of thresholds for determining vertical POF.

The regression model was run with vertical POF ground truth data for all 57 flights using the combination of thresholds given in Table 3. The error probabilities for the model were compared with the error probabilities from the study performed in [17]. The results are described in detail in the 'Results' section of this paper.

3) *Preparing the Program for Combined Flight Data*

An *acid_cid* needs to be associated with each timestep, to run the program with combined flight data. The program uses a filter to split the dataset by flight to create an array of flight datasets. A loop structure is used to handle each flight separately and apply the tiers of testing to the changing time window. The output consists of a text file containing the total combined probability of error for all combined flights, as well as individual flight results.

B. Support Vector Machine

The preprocessing steps that were taken to shape the problem into a supervised learning problem included splitting the flights by their *acid_cid* and turning a sequence of five observations into a single vector with an output $(n-1)$ times for n observations.

The way that the new vector is developed for supervised learning is as follows. Suppose that there are two consecutive observations with *xCoord*, *yCoord*, *altitude*, and the POF. The corresponding vector is formed by taking the magnitude of the difference between consecutive *xCoords*, *yCoords,* and *altitudes* as the three components of the input X and the classification of the second observation as the output Y. This process is done for each pair of consecutive observations in the flight record to obtain the input file for the supervised learning algorithm. The supervised learning algorithm used is the Scikit-learn SVC module with a linear kernel and 1/(number of features) for gamma γ [48].

In attempts to improve the accuracy of our model, we used multiple methods of preprocessing our data. One such method was computing the angles between time steps, aiming for a connection between larger angles and turning trajectory, but this was inaccurate because the time steps were too close together to produce any meaningful differentiation between turning and non-turning states, as the angles were too small. We attempted to remedy this by taking angles between larger time steps, which also was unable to produce good results. We also tried placing the angles on a logarithmic scale to magnify differences between turning and not turning states. The best SVC model with our preprocessing methods obtained a 73% average accuracy over 10 runs, using the difference in latitude and longitude as input to the algorithm. In addition, we found that adding a vertical component (i.e., keeping the altitude variable) did not improve results.

C. Long Short-Term Memory (LSTM)

The Long Short-Term Memory (LSTM) model was utilized next to improve results. LSTM networks are specialized machine learning models for handling sequential data. These are based on recurrent neural networks (RNNs); however, they do not suffer from gradient vanishing or exploding problems. An LSTM captures correlation in timesteps of a sequential dataset and is an ideal candidate model for the aircraft trajectory prediction [38].

There have been a few studies that use LSTM models for aircraft trajectory prediction [45] [46]. Whereas these studies tend to focus on overall trajectory prediction in four dimensions which include inclement weather, human behavior, and other uncertainties, this paper is focused on using an LSTM network to significantly reduce the amount of error in a small area of trajectory prediction which is the determination of an aircraft's POF.

The Keras Python Library was used to design a multivariate time series one-step forecast model [43]. The first task was to prepare the dataset for the model, which involved converting the time series data into a format for supervised learning. Formatting the problem for supervised learning included normalizing the data, defining an input sequence, and defining an output sequence [41][42].

Since horizontal POF is concerned with binary classification (1 and 0) and vertical POF is concerned with multi-class classification (1, 0, and -1), two separate models were developed to tackle each problem separately (Figure IV). The sequential model network for horizontal POF consists of a hidden LSTM layer with 50 neurons and an activation layer with a sigmoid function. The network was trained using 8 epochs and a batch size of 72. The only change for the vertical POF model was that a $tanh$ activation function was used in the activation layer, and the model was trained using a batch size of 150. The loss functions used to calculate the accuracy of the models were 'binary cross-entropy' for horizontal POF and 'hinge' for vertical POF.

The hyperparameters used for these models were determined using a simple search. Throughout development, if overfitting or underfitting occurred, the hyperparameters were changed accordingly. The final set of hyperparameters produced the highest accuracy.

For determining the final forecasted POF, filters were applied to the outputs of the activation functions. These filters rounded the outputs to 0 and 1 for horizontal POF, and 0, 1, -1 for vertical POF. These filters, when combined with the proper activation functions, are shown to prove accurate results, as shown in the results section.

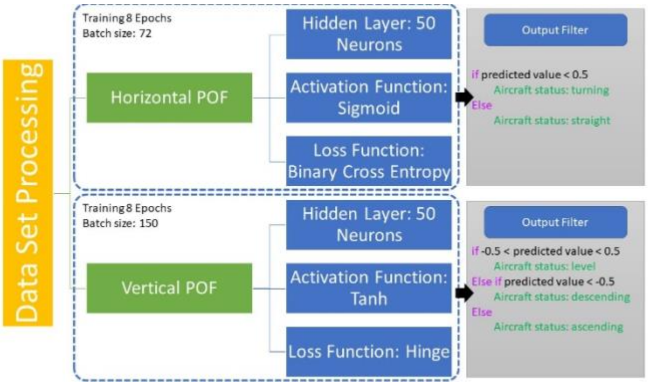

Figure IV LSTM Models

D. Neural Ordinary Differential Equations (NODE)

We also utilized a Neural Ordinary Differential Equations (NODE) model [39]. NODE is a type of machine learning model formulated from the popular Residual Network (ResNet) architecture [40]. The residual connection blocks formed in ResNets are considered as steps in Euler's Method for solving an Ordinary Differential Equation (ODE). Once this realization is made, the solution method can be arbitrarily modified, opening this class of models to a vast range of mathematical tools that have been in development for, in some cases, centuries. Many of these solution methods offer significant advantages over the standard residual network. In particular, the adjoint method for solving the ODE backward in time offers fixed memory cost, a remarkably valuable asset for large models [39].

The primary advantage of NODE in our problem is the ability to be defined continuously, with no fixed number of layers as would be in the classical models. This advantage means that a flight trajectory, which is a continuous physical process, can be modeled more closely. Our model defines the dynamics of the network through a simple convolutional network and solves the resulting problem using the adjoint method. As a first step, our model must have some function representing the dynamics for the model, allowing the ODE solver to integrate through time and produce a prediction. The network used for this purpose is a convolutional network with 2 sequential blocks. In each block, a 1x1 convolution with a length 3 filter is applied to a sample, and the output is passed through the ReLU activation function.

The central part of the model is the ODE solver. We used the solver proposed in [39]. Using PyTorch and the adjoint method described in detail in [39], we developed a model that propagates from the initial state to the final state. Our solver can produce features of the data that are, (1) activated using a ReLU function, (2) flattened for input to a linear classifier, and (3) finally run through the classifier. The model does not have a fixed number of layers as in the LSTM model and can run with a constant memory cost. Optimization of the network is performed using stochastic gradient descent with momentum (learning rate of 0.1) and mean square error as the loss function.

The NODE model is currently able to classify one variable, meaning that only horizontal POF or vertical POF may be determined in a single model. However, we look to extend this so that the model can perform multiple classifications, i.e., allowing the model to classify both horizontal and vertical POF. Our model does well with classifying horizontal and vertical POF individually, so we believe that adding the ability to perform multiple classifications will make this model quite useful for an all-in-one trajectory modeler.

VII. Results

The events used to calculate the probability of false calls and missed calls of each model [17] can be seen in Table 3.

$$P(MC) = \frac{MC}{(MC + VC)} \tag{1}$$

The probability of missed calls "is the estimated probability of falsely detecting a turn/vertical transition that actually does not occur" [17].

$$P(FC) = \frac{FC}{(FC + NC)} \tag{2}$$

The probability of false calls is where the algorithm determines the incorrect aircraft POF. Although the loss function accuracy was calculated for each machine learning model, these probabilities are the most useful way to compare performance.

TABLE III. Horizontal and Vertical Event Calls [17]

		Algorithm - Detected Event	
		Turn	*No Turn*
		Ascending or descending	*Level*
Actual Event	*Turn* *Ascending or descending*	Valid Call (VC)	Missed Call (MC)
	No Turn *Level*	False Call (FC)	Valid No Call (NC)

When compared to the results obtained by the program in [17], it was seen that the Python regression program developed in this study had slightly lower probabilities of error. For horizontal POF, the probability of false calls was reduced by 3.45%, and the probability of missed calls was reduced by 6.01%. For vertical POF, the probability of false calls was reduced by 4.39%, and the probability of missed calls was reduced by 3.76%. Note that the results for the regression model varied as the thresholds were changed and most often performed closer to the error probabilities of the legacy program created in [17].

The LSTM-RNN model performed the best out of all the models. Since this model was created for single-step forecasts, it can also be used as a replacement to the classic regression algorithms that have been used up to this point in the post-flight analysis. When this LSTM-RNN model is further developed for multi-step prediction, it will become useful for real-time trajectory prediction. Our results prove that LSTM-RNN models are suitable for aircraft POF forecasting and provide a solid foundation for further research using these networks in practice.

TABLE IV. POF WITH GPS DATA

Phase of Flight with GPS Data		**Type of Model**				
		Legacy	*Regression*	*SVM*	*LSTM*	*NODE*
Probability False Calls	*Horz*	0.079	0.0445	0.08	0.00	0.06
	Vert	0.050	0.0061	0.14	0.01	0.06
Probability Missed Calls	*Horz*	0.069	0.0089	0.22	0.00	0.07
	Vert	0.048	0.0104	0.02	0.00	0.05
Accuracy (Loss Function)	*Horz*	X	X	0.70	1.00	0.87
	Vert	X	X	0.84	0.99	0.89

TABLE V. HORIZONTAL POF WITH RADAR TRACK DATA

Horizontal Phase of Flight with RADAR Track Data	**Type of Model**			
	Regression	*SVM*	*LSTM*	*NODE*
Probability False Calls	0.0445	0.11	0.00	0.06
Probability Missed Calls	0.0089	0.16	0.00	0.03
Accuracy (Loss Function)	X	0.73	1.00	0.91

We found the SVM model not to be suitable for this problem. We were able to achieve only at best mediocre results with significant preprocessing of data, whereas the other models performed better with less human intervention. Our tests show that the SVM struggles significantly to classify turning states, and this is likely due to a significant imbalance of data in the training set. Due to this imbalance, the SVM overfits the dataset and performs poorly on new examples, leading to a high error in testing.

NODE seems to be a promising step forward for flight trajectory prediction. With their previously mentioned benefits, including naturally continuous dynamics, extendibility to modeling a path entirely, and computational benefits, we believe that NODE will provide a useful framework for a more general-purpose analysis. Although less specialized in POF prediction than the LSTM, the other advantages of NODE make them an appealing option for producing flight trajectory curves using limited data.

VIII. Conclusion

As technology reduces the burden on ground control, a higher number of aircraft can occupy the airspace, making air travel both safer and more accessible. POF prediction and real-time aircraft data acquisition, processing, and analytics, in particular, are emerging technologies that will modernize traditional trajectory prediction systems. This paper presented a regression model to create a baseline for evaluating three AI models, LSTM-RNN, NODE, and SVM.

Future Recommendation

The research presented in this paper gives a strong foundation for future optimization of the machine learning models by adding more layers and extending the proposed models, specifically LSTM and NODE for multi-step forecasting. If aircraft POF can be predicted in advance with a low probability of error, the performance of complete trajectory prediction systems will improve. The developed models treated horizontal and vertical POF as two separate problems for simplicity.

Limitations of Research

There are a few limitations of this research. One limitation is that the results only show the probability of error for single-step forecasts. While the LSTM-RNN model performed better for the available data, further evaluation with additional data set and k-fold cross-validation is required to assess the robustness of the model.

Contributions

Undergraduate researchers, Stephen Kovarik, Liam Doherty, Kiran Korah, and Brian Mulligan, were responsible for the research and development of the project. Dr. Bhavsar, assistant professor at Kennesaw State University, was the leader of the project and supervised all technical research and development. Rowan University Professors, Dr. Yusuf Mehta, director at CREATEs (Center for Research and Education in Advanced Transportation Engineering), and Dr. Ghulam Rasool, assistant professor, supported the project as required. Mike Paglione, Manager of the FAA Aviation Research Division, provided the GPS and RADAR Track Data that was used for training and testing the machine learning models and provided professional feedback on the research. Dr. Kirti Yenki and Dr. Nguyen Thanh supported undergraduate researchers through their technical expertise.

Acknowledgments

This research effort was made possible with the resources and continuous support provided by the FAA. Paul Caron, Program Manager of the Big Data and Analytics Working Group of the FAA Software and Systems Branch, assisted Mike Paglione in providing the student team with professional advice and continuous support as required by the research team. Ph.D. Fellow, Christopher Angelini, assisted in the technical review of the machine learning models. Issouf Kindo, an undergraduate student of computer science, assisted in the literature review. Ghulam Rasool was partly funded by NSF awards OAC-2008690 and ECCS-EPSRC-1903466. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Federal Aviation Administration and National Science Foundation.

References

[1] N. Nelson and M. Paglione, “Determination of Lateral and Vertical Adherence to Route,” presented at the AIAA Guidance, Navigation, and Control (GNC) Conference, Boston, MA, 2013, doi: 10.2514/6.2013-5131.

[2] M. M. Paglione and H. F. Ryan, “Comparison of Host Radar Positions to Global Positioning Satellite Positions,” presented at the 24th Digital Avionics Systems Conference, Washington, DC, USA, 2005, doi: 10.1109/DASC.2005.1563441.

[3] M.-D. Dupuy and M. Porretta, “Preliminary results for a robust trajectory prediction method using advanced flight data,” presented at the AIAA 26th Digital Avionics Systems Conference, Dallas, TX, USA, 2007, doi: 10.1109/DASC.2007.4391831.

[4] M. Uzun and E. Koyuncu, “Data-Driven Trajectory Uncertainty Quantification For Climbing Aircraft To Improve Ground-Based Trajectory Prediction,” *Anadolu University Journal of Science and Technology*, vol. 18, no. 2, pp. 323–345, Mar. 2017, doi: 10.18038/aubtda.270074.

[5] Common Taxonomy Team, “Phase of Flight Definitions and Usage Notes.” National Transportation Safety Board, Apr-2013.

[6] “What is NextGen?,” *Federal Aviation Administration*, 08-Aug-2019. [Online]. Available: https://www.faa.gov/nextgen/what_is_nextgen/.

[7] X. Zhang and Y. Han, “Research on Climb Performance and 4D Trajectory Prediction of Aircraft,” presented at the 2013 International Conference on Computer Sciences and Applications, Wuhan, China, 2013, doi: 10.1109/CSA.2013.98.

[8] “4D Trajectory Concept,” *Skybrary*, 15-Dec-2017. [Online]. Available: https://www.skybrary.aero/index.php/4D_Trajectory_Concept.

[9] J. Zhang, T. Ge, Q. Chen, and F. Wang, “4D trajectory prediction and uncertainty analysis for departure aircraft,” *Journal of SouthWest JiaoTong*, vol. 54, no. 4, pp. 800–806, Jul. 2016, doi: 10.3969/j.issn.0258-2724.2016.04.027.

[10] P. Hagelauer and F. Mora-Camino, “A soft dynamic programming approach for on-line aircraft 4D-trajectory optimization,” *European Journal of Operational Research*, vol. 107, no. 1, pp. 87–95, May 1998, doi: 10.1016/S0377-2217(97)00221-X.

[11] D. I. Sotiriou and S. D. Fassois, “Non-stationary adaptive time series modeling for aircraft 4D trajectory conformance monitoring,” presented at the 2014 6th International Symposium on Communications, Control and Signal Processing, Athens, Greece, 2014, doi: 10.1109/ISCCSP.2014.6877807.

[12] L. Zheng, L. Shuang-Hong, and W. Xue-Li, “General aircraft 4D flight trajectory prediction method based on data fusion,” presented at the 2015 International Conference on Machine Learning and Cybernetics, Guangzhou, China, 2015, doi: 10.1109/ICMLC.2015.7340940.

[13] N. M. Grevtsov, “Synthesis of control algorithms for aircraft trajectories in time-optimal climb and descent,” *Journal of Computer and Systems Sciences International*, vol. 47, pp. 129–138, Feb. 2011, doi: 10.1134/S1064230708010164.

[14] D. Accardo *et al.*, “Fast vertical trajectory prediction method for air traffic management, and relevant atm system,” US20120158220A1, 21-Jun-2012.

[15] W. Gracey, “Measurement of Aircraft Speed and Altitude,” NASA Langley Research Center, Hampton, VA, USA, Technical Report NASA-RP-1046, L-12610, May 1980.

[16] D. Delahaye and S. Puechmorel, “TAS and wind estimation from radar data,” presented at the 2009 IEEE/AIAA 28th Digital Avionics Systems Conference, Orlando, FL, USA, 2009, doi: 10.1109/DASC.2009.5347547.

[17] M. Paglione and R. Oaks, “Determination of Horizontal and Vertical Phase of Flight in Recorded Air Traffic Data,” presented at the AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, CO, USA, 2006, doi: 10.2514/6.2006-6772.

[18] J. A. Besada, G. Frontera, J. Crespo, E. Casado, and J. Lopez-Leones, “Automated Aircraft Trajectory Prediction Based on Formal Intent-Related Language Processing,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1067–1082, Jun. 2013, doi: 10.1109/TITS.2013.2252343.

[19] J. Lopez-Leones, M. A. Vilaplana, E. Gallo, F. A. Navarro, and C. Querejeta, “The Aircraft Intent Description Language: A key enabler for air-ground synchronization in Trajectory-Based Operations,” presented at the 2007 IEEE/AIAA 26th Digital Avionics Systems Conference, Dallas, TX, USA, 2007, doi: 10.1109/DASC.2007.4391836.

[20] M. A. Vilaplana, E. Gallo, and F. A. Navarro, “Towards a formal language for the common description of aircraft intent,” presented at the 24th Digital Avionics Systems Conference, Washington, DC, USA, 2005, doi: 10.1109/DASC.2005.1563351.

[21] E. J. C. Magaña, L. P. D’Alto, and M. V. Ruiz, “Determining a descent trajectory described by an aircraft intent description language (AIDL),” US9766630B2, 19-Sep-2017.

[22] A. de Leege, M. M. V. Paassen, and M. Mulder, “A Machine Learning Approach to Trajectory Prediction,” presented at the AIAA Guidance, Navigation, and Control (GNC) Conference, Boston, MA, 2013, doi: 10.2514/6.2013-4782.

[23] G. Atmeh and K. Subbarao, “A Dynamic Neural Network with Feedback for Trajectory Generation,” presented at the 4th IFAC Conference on Advances in Control and Optimization of Dynamical Systems, Tiruchirappalli, India, 2016, vol. 49, pp. 367–372, doi: 10.1016/j.ifacol.2016.03.081.

[24] Y.-H. Kim, “Aircraft Climb Trajectory Prediction Using Neural Network,” *Applied Mechanics and Materials*, vol. 373–375, pp. 1247–1250, Aug. 2013, doi: 10.4028/www.scientific.net/AMM.373-375.1247.

[25] R. Kaidi, M. Lazaar, and M. Ettaouil, “Neural network apply to predict aircraft trajectory for conflict resolution,” presented at the 2014 9th International Conference on Intelligent Systems: Theories and Applications (SITA-14), Rabat, Morocco, 2014, doi: 10.1109/SITA.2014.6847309.

[26] R. D. Oaks, “A Study on the Feasibility of Using a Genetic Algorithm to Generate Realistic Air Traffic Scenarios Based on Recorded Field Data,” presented at the AIAA Guidance, Navigation, and Control Conference and Exhibit, Monterey, California, USA, 2002, doi: 10.2514/6.2002-4767.

[27] S. Conway and S. Mondoloni, “An airborne conflict resolution approach using a genetic algorithm,” presented at the AIAA Guidance, Navigation, and Control Conference and Exhibit, Montreal, Canada, 2001, doi: 10.2514/6.2001-4054.

[28] C. A. Schultz, D. Thipphavong, and H. Erzberger, “Adaptive trajectory prediction algorithm for climbing flights,” presented at the AIAA Guidance, Navigation, and Control Conference, Minneapolis, Minnesota, USA, 2012, doi: 10.2514/6.2012-4931.

[29] D. P. Thipphavong, C. A. Schultz, A. G. Lee, and S. H. Chan, “Adaptive Algorithm to Improve Trajectory Prediction Accuracy of Climbing Aircraft,” *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 1, pp. 15–24, Jan. 2013, doi: 10.2514/1.58508.

[30] R. Cole, S. Green, M. Jardin, B. E. Schwartz, and S. G. Benjamin, “Wind prediction accuracy for air traffic management decision support tools,” presented at the rd 3 USA/Europe Air Traffic Management R&D Seminar, Napoli, Italy, 2000.

[31] T. I. Baumgartner and U. Maeder, “Trajectory prediction for low-cost collision avoidance systems,” presented at the 2009 IEEE/AIAA 28th Digital Avionics Systems Conference, Orlando, FL, USA, 2009, doi: 10.1109/DASC.2009.5347567.

[32] A. Franco, D. Rivas, and A. Valenzuela, “Probabilistic aircraft trajectory prediction in cruise flight considering ensemble wind forecasts,” *Aerospace Science and Technology*, vol. 82–83, pp. 350–362, doi: 10.1016/j.ast.2018.09.020.

[33] J. Zhang, J. Liu, R. Hu, and H. Zhu, “Online four dimensional trajectory prediction method based on aircraft intent updating,” *Aerospace Science and Technology*, vol. 77, pp. 774–787, Jun. 2018, doi: 10.1016/j.ast.2018.03.037.

[34] S. M. Stigler, “Francis Galton’s Account of the Invention of Correlation,” *Statistical Science*, vol. 4, no. 2, pp. 73–79, 1989, doi: 10.1214/ss/1177012580.

[35] C. Gong and D. McNally, “A methodology for automated trajectory prediction analysis,” presented at the AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, Rhode Island, USA, 2004, doi: 10.2514/6.2004-4788.

[36] J. Berard, “Method and device for generating a flight plan for a tactical flight of an aircraft,” US7272491B1, 18-Sep-2007.

[37] S. Barber, “Event-based flight management system, device, and method,” US9620021B1, 11-Apr-2017.

[38] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Dec. 1997, doi: 10.1162/neco.1997.9.8.1735.

[39] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, “Neural Ordinary Differential Equations,” *arXiv:1806.07366 [cs, stat]*, Dec. 2019.

[40] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *arXiv:1512.03385 [cs]*, Dec. 2015.

[41] “Sequence Classification with LSTM Recurrent Neural Networks in Python with Keras.” [Online]. Available: https://machinelearningmastery.com/sequence-classification-lstm-recurrent-neural-networks-python-keras/. [Accessed: 14-Feb-2020].

[42] “Multivariate Time Series Forecasting with LSTMs in Keras.” [Online]. Available: https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/. [Accessed: 14-Feb-2020].

[43] “Keras.” [Online]. Available: https://keras.io/. [Accessed: 14-Feb-2020].

[44] “SciPy.org — SciPy.org.” [Online]. Available: https://www.scipy.org/. [Accessed: 14-Feb-2020].

[45] Y. Liu and M. Hansen, “Predicting Aircraft Trajectories: A Deep Generative Convolutional Recurrent Neural Networks Approach,” *arXiv:1812.11670 [cs, stat]*, Dec. 2018.

[46] Z. Shi, M. Xu, Q. Pan, B. Yan, and H. Zhang, “LSTM-based Flight Trajectory Prediction,” in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8, doi: 10.1109/IJCNN.2018.8489734.

[47] “NumPy — NumPy.” [Online]. Available: https://numpy.org/. [Accessed: 14-Feb-2020].

[48] “1.4. Support Vector Machines — scikit-learn 0.22.1 documentation.” [Online]. Available: https://scikit-learn.org/stable/modules/svm.html. [Accessed: 14-Feb-2020].

Appendix A. Definitions of Data Parameters And Thresholds

Name	**Definition**
acid_cid	The unique aircraft identifier, used to filter the data by flight.
vertSlopeThresh	Threshold which determines the bound where an aircraft is determined to be ascending or descending [17].
vertTimeWindow	The window of timestamps surrounding the current timestamp in the vertical regression algorithm [17].
horzPearsonThreshHi	The upper Pearson’s r bound which determines whether an aircraft is straight [17].
horzPearsonThreshLo	The lower Pearson’s r bound which determines whether an aircraft is turning [17].
horzRSqrThresh	Determines whether an aircraft is turning in the second level of testing using the R squared metric [17].
horzFlatnessThresh	Determines whether an aircraft is turning in the second level of testing using the flatness metric explained in detail in [17].
horzTimeWindow	The window of timestamps surrounding the current timestamp in the horizontal regression algorithm [17].