IJHSR
ijhighschoolresearch.org
RESEARCH ARTICLE

A Comparative Analysis of AI Model and Traditional Model in 4D Trajectory Prediction Using Different Error Metrics

Harshitha Vishnu

Fremont High School, 575 W Fremont Ave, Sunnyvale, California, 94087, USA; mailtoharshithavishnu@gmail.com
Mentor: Bahae Samhan

ABSTRACT: The development of advanced predictive models can help ensure safe and efficient operations within the complex area of the air traffic domain. Accurate 4-dimensional (latitude, longitude, altitude, and time) trajectory prediction plays a crucial role in enhancing the safety and efficiency of modern air traffic management systems. This study presents a comparative analysis of two trajectory prediction models, a linear regression model and a neural network, focusing on their ability to accurately predict the position of an aircraft given factors such as latitude, longitude, altitude, and time. Using data obtained from OpenSky, both models were evaluated for prediction accuracy on different error metrics like Euclidean error, MAE, MSE, RMSE, MAPE, and R^2. Results from the study indicate that while the neural network model performed better on the Euclidean error, the linear model had lower scores on all the other error metrics. This underperformance may be attributed to inadequate feature engineering, overfitting, or insufficient hyperparameter tuning, and implementing techniques like hyperparameter tuning methods, regularization methods, or additional features could enhance the model's accuracy. These findings provide critical insights into the strengths and limitations of each model and highlight the importance of balancing model complexity with performance requirements to refine predictive systems in next-generation ATM systems.

KEYWORDS: Robotics and Intelligent Machines, Machine Learning, 4D trajectory prediction, Aircraft Trajectory Modeling, Artificial Intelligence in Aviation.

Introduction

The need for accurate trajectory prediction models in civil aviation is essential for improving flight safety, enhancing operational efficiency, and minimizing delays.[1] As global air traffic continues to increase, precise models are required to optimize flight paths, allowing for the effective use of increasingly congested airspace.[2] One such advancement is the use of 4D trajectory prediction, which not only accounts for latitude, longitude, and altitude but also incorporates time, making it superior to traditional 3D predictions, which generally include latitude, longitude, and altitude.[3] In this context, latitude refers to the aircraft's north-south position while longitude refers to its east-west position relative to the Earth's equator, altitude measures the distance above a surface or sea level, and time tracks positional changes over a period.

The implementation of accurate trajectory prediction models can aid in making efficient use of limited airspace. This way, flights can follow the optimal flight paths while increasing the number of aircraft in an area, while the area is still safe. These optimal paths also reduce the time required for aircraft to reach their destinations while also ensuring the safety of the passengers and the flight crew.[2]

Traditional trajectory prediction methods, such as Linear Regression, have been widely used for identifying linear relationships in flight data in small datasets.[4] However, this model is not capable of predicting non-linear relationships often present in aviation data, so it might underfit the data. This would lead to a lot of outliers, resulting in a higher residual error, which means the model would make inaccurate predictions.[5]

Artificial Intelligence and Machine Learning are promising alternatives to aid in trajectory prediction. AI models, especially neural networks, excel at identifying non-linear relationships in data.[6] Neural networks have multiple layers with multiple nodes, and they produce an output through multiplication, summation, activation function application, and other such processes. These models can identify patterns in complex data that go unnoticed by simpler Regression models.[7]

This study addresses the following research question: How does a neural network-based AI model compare to traditional Linear Regression in predicting 4D aircraft trajectories in terms of accuracy and practical applicability for air traffic management? This research aims to compare the predictive accuracy of a traditional Linear Regression model with that of a Neural Network model, evaluating how AI can improve 4D trajectory modeling in civil aviation. Key metrics for comparison include Euclidean Error, Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and R-squared (R^2) scores. The data used for this analysis was obtained from OpenSky, a free platform offering real-time aircraft position data (including latitude, longitude, altitude, and time) collected through ADS-B transponders. The dataset includes variables such as call sign, origin country, time position, last contact, longitude, and latitude. For this study, the input variables were longitude, latitude, altitude, and time deltas. To compare the differences in models' predictive accuracy, a Linear regression model was employed as the traditional model, and a Neural Network model with TensorFlow and TFlite was integrated as the AI model.

This study contributes to civil aviation by emphasizing the benefits of comparing traditional and AI-based models for trajectory prediction and aims to provide insights into which methodologies best optimize flight paths and manage air traffic, improving the overall efficiency of air traffic management systems. By addressing the limitations of traditional models and demonstrating how enhanced prediction models can streamline aviation systems, the research supports more efficient use of airspace, reduces delays, and ensures safer skies for the future.

Background and Evolution of 4D Trajectory Modeling

4D trajectory modeling, which refers to the management of aircraft positions in three dimensions, namely latitude, longitude, and altitude, in addition to time, has evolved significantly over the years. This development is primarily driven by the rapid growth of the aviation industry and reduced flight intervals, which prioritize the need to optimize routes for safety, efficiency, and environmental impact.

The initial efforts in trajectory prediction for civil aviation were grounded in basic physical models, such as kinematic equations of motion. These early models primarily used deterministic methods to predict future aircraft positions based on historical data.[8] However, they were simplistic and did not account for various factors like weather, wind, or air traffic.

Research in 4D trajectory prediction began with relatively simpler models. One of the first techniques that emerged for trajectory prediction was linear regression, where trajectory data for aircraft was fitted to a linear model. This allowed for simple predictions based on past observations or an easily recognizable pattern. However, this method's accuracy was limited due to its inability to identify complex patterns in non-linear data, which meant it often underfitted the data and did not provide good predictions.[8]

Recently, with the development of the aviation industry, statistical models have become prominent. Among the most significant advancements was the introduction of the Kalman Filter in trajectory modeling.[9] The Kalman filter is an algorithm that can use data that is noisy or inaccurate and estimate an unknown variable with greater accuracy. Kalman filters surpassed linear regression in air trajectory prediction due to their ability to handle dynamic systems and uncertainties effectively.[10]

As computing power grew in the 1990s, more sophisticated nonlinear models were introduced.[11] Instead of merely predicting future positions, these models integrated the concept of 4D trajectory modeling, incorporating not only three spatial dimensions but also time. Methods such as the Extended Kalman Filters (EKF), Monte Carlo Simulations, and stochastic models gained popularity during this time in the air traffic management industry.[12]

Today, 4D trajectory modeling has become advanced. With the advancement of deterministic models, which provide a single, concise prediction of an aircraft's future trajectory based on initial data and well-defined equations of motion. Model Predictive Control (MPC), which predicts the future state of an aircraft based on data about its current state, is a great advancement in deterministic models because it continuously optimizes to surpass limitations like weather and air traffic. Advancements in deterministic kinematic models include more sophisticated aircraft dynamics and flight mechanics.

Research is increasingly focusing on models that can adjust predictions dynamically as real-time data, like weather updates or unexpected air traffic changes, becomes available, improving model responsiveness.[13,14] Additionally, combining deterministic and probabilistic approaches, such as deterministic models enhanced with neural networks, is a growing area, as it can offer more reliable predictions by balancing precision with flexibility for handling unpredictable factors.[15]

Probabilistic models are statistical models that use uncertainty as a factor in their predictions. By considering all the possibilities, these models can deal with otherwise unpredictable changes like weather, air traffic congestion, and unexpected delays. Some examples of probabilistic models include Monte Carlo simulations and Bayesian networks.[12] In Monte Carlo simulations, multiple trajectories are predicted when various factors like weather and wind speed are inputted into the model. Bayesian networks represent the relationships between different variables probabilistically, which is useful in incorporating both historical and real-time data.

Despite their statistical flexibility, traditional deterministic and probabilistic models still struggle with the dynamic, nonlinear complexities of real-world aviation. Deterministic approaches rely on fixed equations, limiting their adaptability to unforeseen disruptions like sudden weather shifts or air traffic congestion.[13] Probabilistic methods, while accounting for uncertainty, often require extensive computational resources to scale effectively and may lack the granularity to capture intricate spatial-temporal patterns.[12] These shortcomings highlight the need for more adaptive, data-driven solutions, prompting the exploration of AI-based techniques, which leverage machine learning to dynamically refine predictions using real-time data.

Traditional trajectory prediction models are useful in many situations, but also include a lot of limitations. These models are unable to handle relationships between non-linear data, like wind speed and turbulence.[16] If they are asked to find patterns in complex, non-linear data, they would simplify the output and therefore underfit the data. These models also have limited adaptability to constantly updating real-time data. These foundational models also perform poorly in complex or noisy data. They are not adapted to handle the complexity required for large-scale data.

Collectively, traditional trajectory prediction models—whether deterministic or probabilistic—face inherent constraints. Key limitations include reliance on simplified linear assumptions, leading to underfitting in nonlinear scenarios; poor scalability with high-dimensional, noisy data; and static architectures unable to assimilate real-time updates efficiently. These gaps necessitate a paradigm shift toward AI-driven models capable of learning complex patterns autonomously and adjusting predictions dynamically.

Neural networks, particularly deep learning architectures, have shown promising results in predicting 4D trajectories. One of the most common architectures for 4D trajectory prediction is the Long-Term short-memory network. LTSM networks are a type of recurrent neural network (RNN) that excels in mastering long-term dependencies.[17] LTSMs contain a memory cell that can hold information for a long period of time. LTSMs can also decide what information to include and what to remove. LTSMs are effective at learning from historical trajectory data and predicting future positions in both space and time, and this makes them ideal predictors of the aircraft's future trajectory. Graph Neural Networks (GNNs) are an emerging architecture for aviation trajectory prediction, especially in multi-aircraft scenarios. GNNs contain data points called nodes, which are linked by lines called edges. GNNs model the interactions between different aircraft as nodes in a graph, with edges representing potential interactions (such as proximity with other aircraft or potential conflicts).

This progression from traditional deterministic models to AI-driven 4D trajectory modeling reflects the industry's shift toward more adaptive and predictive solutions that are capable of addressing the complexities of modern aviation. By incorporating real-time data and handling nonlinear interactions, AI models offer unprecedented accuracy and flexibility that traditional models cannot achieve, setting a new standard in trajectory prediction.[8,18]

The advancements in trajectory modeling, from deterministic methods to AI-driven techniques, demonstrate the aviation industry's commitment to addressing the complexities of modern air traffic management. While traditional models like linear regression have laid the foundation for trajectory prediction, their limitations in handling nonlinear and dynamic data necessitate the exploration of more sophisticated approaches. This motivates this research to investigate how integrating a neural network-based AI model can improve the accuracy and efficiency of 4D trajectory predictions compared to traditional methods. The following sections outline the methodology, including data collection and modeling approaches, that underpin this comparative analysis.

■ Methodology

Data Collection and Description:

The dataset used for this research was obtained from OpenSky, a free receiver network that contains credible aircraft information received through ADS-B transponders. The attributes extracted from the dataset include Callsign, Origin country, time position, last contact, latitude, longitude, and altitude. As 4D trajectory prediction involves latitude, longitude, altitude, and time, the corresponding attributes will be significant to the research. In aviation, a callsign is a unique identifier that consists of the aircraft's name along with a combination of unique numbers or letters. This is used to distinguish an aircraft during communication with the control center or other aircraft, which avoids confusion when multiple aircraft are in the same space. These attributes contribute to accurate tracking of the aircraft's position, enabling precise trajectory prediction. For this research, the callsign will be used to distinguish the aircraft, which is critical in predicting its trajectory. Additionally, attributes like origin, country, and last contact also help identify a particular aircraft and distinguish it from others.

Model Selection:

To address the need for accurate trajectory prediction, two modeling approaches were introduced: a traditional linear regression model and a neural network-based AI model. Each approach offers unique strengths and leverages different aspects of the data. The linear regression model provides a straightforward approach to understanding simpler relationships among variables, while the AI model can handle the non-linear complexities in location data, which may enhance flexibility and prediction accuracy for real-time predictions.

Linear regression attempts to predict the relationship between an independent variable (Time delta, Latitude delta, and Longitude delta) and the target variables (Latitude and Longitude), assuming a linear relationship between these variables. This model also assumes constant variance, independence of errors, no multicollinearity, and performance.

The AI model's hyperparameters (e.g., layer size, activation functions, epochs) were selected based on empirical best practices for trajectory prediction tasks, balancing computational efficiency and predictive performance. While systematic hyperparameter optimization techniques like grid search or cross-validation were not employed due to resource constraints, key design choices were validated through iterative testing.

The AI model consists of an input layer, two hidden layers with 64 nodes and ReLU activation functions, and an output layer with two nodes to predict latitude and longitude. ReLU (Rectified Linear Unit) is used in the hidden layers because it is able to handle nonlinear data effectively, and the output layer remains linear to directly predict latitude and longitude. The model utilizes 64 nodes in each layer to achieve a balanced learning capacity without overfitting the data. The output layer contains two nodes corresponding to the latitude and longitude predictions. TensorFlow was chosen for training because of its ability to execute various tasks across many platforms and its general performance with neural networks after training. The model was converted to TensorFlow Lite (TFLite) for its compactness, efficiency, and ability to deploy the model on mobile or embedded systems, which allows for faster prediction time, ideal for real-time trajectory prediction.

For model training and evaluation, the training dataset is split into 80% training and 20% testing for both models using TrainTestSplit. 10% of the training data is further split to be used as validation data during neural network training to monitor and reduce overfitting. For the neural network, a default batch size was assigned following the framework's optimal settings for efficiency. However, adjustments can be made depending on hardware and computational power. The neural network is trained for 10 epochs, balancing training time with achieving sufficient accuracy in learning trajectory patterns.

DOI: 10.36838/v7i8.5

Evaluation metrics:

Euclidean Error measures the direct spatial distance between predicted and true positions, providing a clear indication of geographic accuracy, which is essential for trajectory applications. A smaller Euclidean error indicates that two points are closer together, while a larger error means they are further apart, and in this scenario, a smaller error would be optimal. This metric is very easy to interpret, and its calculations are straightforward. Euclidean Error is particularly suitable for trajectory prediction because it directly quantifies positional deviations in physical space—a critical requirement for aviation safety and navigation. Additionally, its simplicity ensures computational efficiency, making it practical for real-time systems where rapid error assessment is necessary. Euclidean Error is defined as:

$$Euclidean\,Error = \sqrt{\left(Latitude_{pred} - Latitude_{true}\right)^2 + \left(Longitude_{pred} - Longitude_{true}\right)^2}$$

Equation (1)

Mean Absolute Error (MAE) calculates the average absolute error in predictions, giving insight into general prediction accuracy and robustness against outliers. It gives an idea of how much, on average, the predictions deviate from the actual values. MAE gives the average magnitude of errors in the same units as the output. The lower the MAE, the better the model is at making accurate predictions. MAE can be represented as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \qquad \text{Equation (2)}$$

Where yi is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of data points.

Mean Squared Error (MSE) is the average of the squared differences between predicted and actual values. It penalizes larger errors more than smaller ones due to the squaring of the differences. The lower the MSE, the better the model is performing. A small MSE indicates the model is making mostly accurate predictions. It is more sensitive to outliers because it emphasizes larger errors. The formula for MSE is:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \qquad \text{Equation (3)}$$

The square root of MSE, Root Mean Squared Error (RMSE), has the same units as the target, making it easier to interpret in terms of geographical units. It provides insight into the overall prediction accuracy, emphasizing larger errors. Like MSE, the lower the RMSE, the better the model's performance. However, RMSE is easier to interpret since it is in the same units as the data. RMSE is especially useful when larger errors need to be penalized more. RMSE can be derived as shown:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \qquad \text{Equation (4)}$$

Mean Absolute Percentage Error (MAPE) measures error as a percentage, making it helpful for understanding the model's relative accuracy and comparing different models' performance. A lower MAPE means better predictive performance. It is particularly useful when comparing the accuracy of models across different datasets or scales. However, it can give extreme values when actual values are close to zero. It is also easily interpretable as a percentage. The formula that can be used to derive MAPE is:

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_1}\right| \qquad \text{Equation (5)}$$

R-squared (R^2) indicates the proportion of variance in the data explained by the model. Its values range from 0 to 1. Higher R^2 values show that the model explains more of the observed data variance, which is especially useful for assessing linear models' effectiveness. Negative values can occur if the model performs worse than a simple baseline model. R-squared can be represented as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2} \qquad \text{Equation (6)}$$

Where $\bar{y}$ is the mean of the actual values

Experimental Setup:

This project utilizes Python for implementation, with libraries including TensorFlow, TFLite, and Scikit-Learn. TensorFlow is used to build, train, and convert the model into a TFLite model for efficiency, and Scikit-Learn, which contains modules like StandardScaler and TrainTestSplit, is used for linear regression and data scaling. The TensorFlow model was initially trained in a typical Python runtime environment compatible with TensorFlow 2.x. The TFLite runtime interpreter is lightweight, making it suitable for resource-constrained environments, ensuring efficient model predictions even for systems with limited computational power, such as the home PC used to implement the model for this research. An 80/20 train-test split is applied, with 80% of the data used for model training and 20% for final evaluation to assess the model's overall performance. Within the training data, a 10% validation split is applied to monitor the neural network's performance during training. This approach helps track model convergence and identify any potential overfitting to the training data. Given the model's relatively simple architecture and the deterministic nature of trajectory data (e.g., temporal dependencies), targeted hyperparameter selection was prioritized over exhaustive search methods. This approach aligned with the goal of lightweight deployment for real-time applications. Finally, feature scaling was applied, using StandardScaler, to improve model generalization and performance.

Results

This section compares the performance of the traditional linear model with that of the AI model, using a range of key error metrics to assess prediction accuracy. The primary metric applied, Euclidean Error, measures the straight-line distance between predicted and actual values, providing an overall indicator of model accuracy. Additionally, several other metrics were utilized to offer a nuanced performance assessment. Mean Absolute Error (MAE) was calculated to provide the average magnitude of errors in the same units as the target variable, giving a straightforward measure of typical error size. Mean Squared Error (MSE), which represents the average squared differences between predicted and actual values, was also applied. This metric penalizes larger errors more heavily than smaller ones due to the squaring of the differences, making it sensitive to outliers and emphasizing significant deviations in prediction. Root Mean Squared Error (RMSE), similar to MSE, also accentuates larger errors but is easier to interpret because it is expressed in the same units as the target data. To provide insights into performance in relative terms, Mean Absolute Percentage Error (MAPE) was used, which expresses error as a percentage, making it particularly useful for understanding model accuracy across different scales. Lastly, the R^2 score, or coefficient of determination, was applied to evaluate how well each model explains the variance in the target data. This metric indicates the proportion of variance in the dependent variable that is predictable from the independent variables, thus providing an estimate of each model's explanatory power. Collectively, these metrics offer a comprehensive view of model performance, revealing strengths and limitations across various aspects of predictive accuracy and error sensitivity.

While the AI model achieved a lower Euclidean error, indicating better performance on this metric, it underperformed on other key metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and R^2.

The traditional linear model and the AI model displayed notable differences in predicting latitude. Although the AI model achieved a Euclidean Error of 36.073, which was substantially lower than the traditional model's Euclidean Error of 59.538, the AI model's performance was inconsistent across other metrics. For instance, the MAE for the AI model was 1835.694, which is considerably higher than the traditional model's MAE of 10.986, indicating a greater overall prediction error. Similar trends were observed in the MSE, RMSE, and MAPE values, with the AI model recording markedly higher errors (MSE of 3369773.141 and RMSE of 1835.694) compared to the traditional model (MSE of 223.239 and RMSE of 14.941). Additionally, while the traditional model had an R2 score of 0.458, the AI model's error could not be computed, represented by NaN, meaning "Not a Number," which signifies an undefined or unrepresented value. This shows how a valid error could not be provided for this metric. These discrepancies suggest that while the AI model achieves closer proximity to actual values in specific instances (as indicated by lower Euclidean Error), it may not consistently capture the underlying data structure as effectively as the traditional model across all error metrics. Table 1 and Figure 1 summarize these results for latitude prediction.

Table 1: Comparison in latitude prediction accuracy between the Traditional and the AI model using various error metrics. The results indicate that the traditional model performed better than the AI model on every metric except for Euclidean error, where the AI model performed better by a small margin.

	Traditional Model	AI Model
Euclidean Error	59.538	36.073
MAE	10.986	1835.694
MAP	223.239	3369773.141
RMSE	14.941	1835.694
MAPE	53.768	5861.486
R^2	0.458	NaN

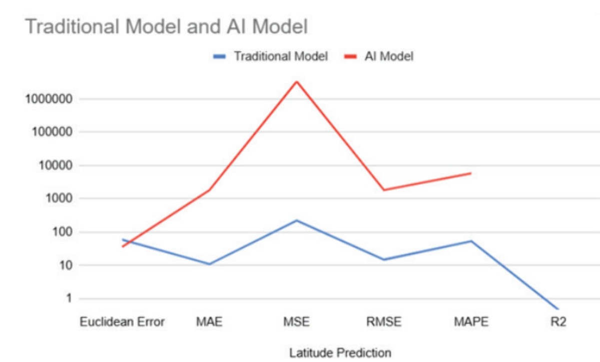

Figure 1: The visual representation of the models' performance in predicting the latitude indicates that the AI model performed worse than the traditional model at every error metric except Euclidean error (a lower error equals better accuracy). It can also be seen that there is a large margin of difference between the traditional model's and AI model's performance for each metric.

For longitude prediction, the AI model again demonstrated a lower Euclidean Error of 36.073 compared to the traditional model's 59.538. However, as with latitude prediction, the AI model underperformed significantly on other metrics. The MAE for the AI model was 5259.041, substantially higher than the traditional model's 57.724, and the MSE and RMSE values for the AI model were also significantly higher (MSE of 27657509.561 and RMSE of 5259.041) compared to the traditional model's MSE of 4560.720 and RMSE of 67.533. Furthermore, the MAPE value for the AI model was exceedingly high at 5428.508, contrasting sharply with the traditional model's MAPE of 102.830. The R^2 metric could not be computed for the AI model (NaN), whereas the traditional model achieved an R^2 of 0.472, suggesting some level of explanatory power in the traditional model. Table 2 and Figure 2 display these comparative results for longitude prediction.

Table 2: Comparison in longitude prediction accuracy between the Traditional and the AI model using various error metrics. The results indicate that the traditional model performed better than the AI model on every metric except for Euclidean error, where the AI model performed better by a small margin.

	Traditional Model	AI Model
Euclidean Error	59.538	36.073
MAE	57.724	5259.041
MAP	4560.720	27657509.561
RMSE	67.533	5259.041
MAPE	102.830	5428.508
R^2	0.472	NaN

DOI: 10.36838/v7i8.5

Figure 2: The visual representation of the models' performance in predicting the longitude indicates that the AI model performed worse than the traditional model at every error metric except Euclidean error (a lower error equals better accuracy). It can also be seen that there is a large margin of difference between the traditional model's and AI model's performance for each metric, suggesting the traditional model may be reliable for generalized local predictions.

The findings indicate that while the AI model excels in minimizing Euclidean Error, it struggles with other critical error metrics, particularly MAE, MSE, RMSE, and MAPE, for both latitude and longitude predictions. This discrepancy may stem from the AI model's sensitivity to data inconsistencies or its potential overfitting to specific patterns within the dataset. Conversely, the traditional linear model demonstrated more consistent performance across a wider range of error metrics, suggesting it may be more reliable for generalized location predictions despite a slightly higher Euclidean Error.

These results highlight the importance of selecting appropriate performance metrics when evaluating predictive models, as different metrics can highlight varying aspects of model accuracy and reliability. Future studies may explore ways to optimize the AI model's performance across all error metrics, potentially by refining model architecture or employing additional data preprocessing techniques to improve generalization.

Discussion

The performance analysis shows that although the AI model performed better than the Linear model at Euclidean Error, it performed worse in other errors. This may be because while Euclidean error can be effective at assessing overall spatial accuracy in predictions, it is not able to capture the nuances of individual coordinate predictions, such as latitude or longitude.[19] For instance, integrating spatial autocorrelation in machine learning models can enhance accuracy by accounting for geographical data. However, relying only on Euclidean distance can result in individual errors being masked by overall distance calculations, leading to potentially misleading performance assessments. Traditional models like spatial lag, which incorporate spatial features, can better capture the spatial dependency and reduce prediction errors.[18,20]

The AI model's poorer performance on MAE, MSE, and RMSE (despite its Euclidean error advantage) suggests it struggles with coordinate-specific precision. Unlike Euclidean distance, which aggregates errors into a single spatial value, these metrics penalize directional biases (e.g., consistent overestimation of altitude). The AI model's focus on holistic spatial accuracy may come at the cost of localized errors in individual dimensions (latitude/longitude/altitude), which are weighted equally in traditional metrics.[21]

Another reason the traditional model performed better than the AI model could be the fact that traditional models often perform better on simpler metrics like RMSE, MSE, and MAE.[22] This provides stable and interpretable results. If the model had a simpler structure, such as in linear regression or traditional random forest models, it delivered more consistent results due to fewer complexities in data relationships. While AI models may introduce noise or fail to generalize due to their complexity, traditional models, such as spatial lag or linear regression, prioritize interpretability, making them more reliable for simpler datasets or metrics like RMSE and MAE.[23] This stability is particularly evident when spatial dependencies are weak and the models are not overfitted. As seen in the comparison between spatial models and traditional machine learning models, simpler models tend to generalize better under specific conditions.[18] Furthermore, traditional models generally require fewer computational resources, enabling faster optimization and fewer errors caused by resource limitations during training.[24]

Additionally, AI models may exhibit significant errors due to several factors, such as overfitting training data, inadequate feature engineering, or insufficient hyperparameter tuning. Overfitting is particularly problematic when the model learns noise in the training data, leading to poor generalization on test data.[25] Especially on small or imbalanced datasets, AI models are prone to overfitting and memorizing instead of generalizing, while traditional models, which are less data-intensive, are less likely to overfit.[26] Additionally, the inclusion of irrelevant or insufficient features may increase the runtime and hinder the model's ability to capture complex relationships between inputs and outputs.[25] Furthermore, fine-tuning model architecture and parameters is crucial to optimizing performance. AI models often depend on well-designed features. Inadequate preprocessing or irrelevant features can degrade their performance compared to traditional models, which are more robust to such shortcomings.[27] For instance, AI models may overlook natural spatial relationships unless specifically programmed, unlike spatial regression models designed to address geographic dependencies explicitly.[28] The AI model's suboptimal performance on non-Euclidean metrics may also stem from its inability to prioritize coordinate-specific errors. Euclidean distance aggregates spatial deviations into a single value, potentially masking directional biases.[29] In contrast, traditional models optimize for individual coordinate errors directly, aligning better with metrics like MAE. Additionally, the AI model's fixed architecture (e.g., 64-node hidden layers) may lack the adaptability to capture localized spatial patterns, whereas traditional methods like spatial lag explicitly model geographic dependencies.[30] This limitation becomes pronounced when training data lacks sufficient variability in spatial-temporal features, further exacerbating directional errors. While the AI model's superior performance in Euclidean error suggests potential for spatial accuracy, its inconsistency across other metrics raises questions about its robustness. For example, probabilistic models like Bayesian networks16 or hybrid approaches[16] may better handle uncertainty in dynamic conditions, such as weather disruptions or air traffic variability, by combining deterministic predictions with probabilistic adjustments.

The AI model's higher MAE/RMSE scores may also stem from its inability to prioritize error types critical for aviation. For example, altitude errors (safety-critical) and time errors (delay-sensitive) are treated equally with lateral position errors in the loss function. Traditional models, by contrast, often optimize for domain-specific priorities (e.g., penalizing altitude deviations more heavily), aligning better with operational needs.

Fundamentally, the AI model's failures in non-Euclidean metrics reflect a misalignment between its training objective (minimizing bulk spatial error) and aviation's need for dimension-aware precision. While Euclidean optimization suits general spatial tasks, trajectory prediction requires balancing heterogeneous errors (e.g., time vs. altitude), necessitating custom loss functions or hybrid architectures that blend AI's nonlinear capacity with traditional models' interpretable constraints.

It is found that advanced AI techniques like deep learning, though powerful, require careful design choices to minimize these issues.[15,18] Advancing the performance of AI models requires the exploration of more sophisticated models or alternative architectures. Emerging techniques such as reinforcement learning, generative adversarial networks (GANs), transfer learning, and neuro evolution offer promising solutions to issues that traditional machine learning models struggle with. In addition, these models significantly enhance predictive accuracy and reduce errors, leading to better performance. For instance, GANs have been utilized in scenarios requiring creative data generation, such as vehicle trajectory prediction,[13] and show great potential for use in the trajectory prediction of aircraft. Similarly, reinforcement learning allows models to interact with their environments, developing accuracy over time, which could improve the performance of the model, leading to better trajectory predictions. Explainable AI (XAI), which refers to a set of processes and methods designed to make AI models more transparent and interpretable to humans, also offers frameworks that could not only improve performance but also make AI decisions more interpretable and transparent, aiding in debugging and model refinement. It takes accountability for its decisions while also mitigating bias.[31]

Moreover, emerging technologies, such as quantum AI, are beginning to demonstrate significant potential, offering massive computational power capable of addressing highly complex tasks with greater precision.[32] This diversity in AI techniques provides a range of alternatives that may better align with the specific requirements of a given problem domain and dataset characteristics. These advancements highlight the importance of exploring and adopting innovative architectures, particularly in scenarios where traditional models underperform.[14]

The findings of this study align with the potential of advanced techniques like reinforcement learning (RL) and GANs to address observed shortcomings. RL's iterative reward-based optimization could dynamically adjust predictions in response to real-time errors (e.g., wind shifts), while GANs could synthesize rare but critical scenarios (e.g., extreme turbulence) to improve generalization. Explainable AI (XAI) frameworks, such as SHAP or LIME, could further bridge the gap between the AI model's 'black-box' predictions and the interpretability of traditional models, enabling targeted debugging of coordinate-specific errors (e.g., latitude bias) and fostering trust in aviation applications.

The linear model's stability in MAE and RMSE shows us its reliability for scenarios where computational efficiency is prioritized, indicating the need to align model selection with specific operational requirements. For example, AI models for precision in controlled environments versus traditional models for generalizability. Therefore, we suggest that future work should test hybrid frameworks to mitigate the limitations observed in standalone AI or linear approaches.

Limitations and Future Directions

While the data obtained from the OpenSky network was accurate, it was restricted in quantity, potentially limiting the model's ability to generalize across diverse scenarios. The dataset also had a few inconsistencies, such as missing values, which impacted the predictive accuracy of the model. Inaccuracies in recorded flight parameters or limited temporal and spatial resolution may have introduced noise into the training process, which might have led the model to provide biased or inaccurate predictions. The complexity of the AI model used in this study might have contributed to its overfitting or underfitting of the data. Overfitting arises when the model captures noise in the training data, resulting in reduced generalization for unseen data. In contrast, underfitting occurs when a model is too simple to capture the complexities or the underlying patterns in the data. To address these issues, hyperparameter tuning methods like grid search, random search, or Bayesian optimization could have been implemented to find the best hyperparameter configurations. Additionally, regularization methods like weight decay, dropout, or early stopping could have been utilized to reduce overfitting. On the other hand, to minimize underfitting, the complexity of the model could have been increased using feature engineering and hyperparameter tuning. Moreover, the choice of features utilized might not have fully captured the dynamics of the system being modeled. Features like weather, turbulence, and air traffic at that specific time could have been incorporated to capture underlying patterns and increase predictive accuracy. While the data collected represents a diverse range of flights from many flight regions, the data was collected during a specific time frame, so it might not be representative of data obtained during other time frames, which limits the model's ability to accurately predict data from other time frames. The dataset may also contain inherent biases based on geographical coverage, aircraft types, or airline operators represented in the OpenSky network. Such

DOI: 10.36838/v7i8.5

sampling biases could skew the model's understanding of typical flight patterns. Additionally, missing values in critical parameters like altitude or velocity may reflect systemic gaps in ADS-B coverage rather than random noise. These data quality issues compound the challenges of training reliable predictive models, as they may cause the AI system to learn artifacts of data collection limitations rather than true trajectory patterns.

Several data preprocessing and augmentation approaches could mitigate these limitations. For missing values, multiple imputation techniques could estimate plausible values while accounting for uncertainty, rather than simple deletion or mean imputation. For temporal biases, implementing time-based stratification during train-test splits would ensure all time periods are represented. For spatial biases, geographic weighting could balance representation across flight regions. To reduce noise, Kalman filtering could be utilized to smooth erratic position reports while preserving true trajectory patterns.

For more robust testing, implementing k-fold cross-validation (e.g., 5- or 10-fold) is recommended, especially for the traditional linear regression model.[33] This technique provides a more comprehensive performance estimate across different data partitions. Additionally, using early stopping or monitoring validation loss can help prevent overfitting in the neural network model by stopping training when improvement plateaus.[34]

To enhance AI model performance, exploring advanced methodologies is crucial. Techniques such as reinforcement learning, GANs, and transfer learning offer promising avenues for improvement. For example, GANs have shown success in generating synthetic trajectory data to improve predictive models,[13] while reinforcement learning can iteratively optimize models by interacting with dynamic environments.

Explainable AI (XAI) frameworks also present significant opportunities, making AI predictions more transparent and interpretable. By improving model accountability and mitigating bias, XAI frameworks not only enhance performance but also increase trust in AI systems.[31]

Emerging technologies, such as quantum AI, offer unparalleled computational power, which could revolutionize trajectory prediction by addressing the complexities of large-scale, nonlinear datasets with higher precision.[32] Exploring these techniques alongside traditional models will help identify the most effective methodologies for optimizing 4D trajectory prediction.

These findings have direct implications for real-world air traffic management (ATM) systems. The comparative performance analysis suggests that hybrid systems combining traditional models' reliability with AI's pattern recognition capabilities could optimize trajectory prediction in operational environments. For instance, linear models could serve as baseline predictors while AI components handle complex, nonlinear scenarios like weather disruptions or congested airspace, creating a more robust ensemble system. These results could inform the phased implementation of AI in ATM systems. The performance metrics established here provide concrete benchmarks for aviation authorities evaluating prediction systems, particularly in balancing accuracy requirements with computational constraints. Future work could test these models in simulation environments mirroring actual air traffic control workflows to validate operational feasibility.

In summary, this study shows that while traditional models excel in simplicity, efficiency, and interpretability, AI models show immense promise for handling complex, nonlinear data. However, their effectiveness depends on overcoming challenges such as overfitting and inadequate feature engineering. The integration of advanced techniques, including GANs, reinforcement learning, and XAI, highlights the potential for AI to set new standards in 4D trajectory modeling, improving air traffic management and operational efficiency.

Acknowledgments

The author would like to acknowledge Prof. Bahae Samhan, PhD, Associate Professor of Business Information Systems at Illinois State University, for his constant support and guidance throughout the research process.

References

1. Shi, Z., Xu, M., Pan, Q., Yan, B., & Zhang, H. (2018, July). LSTM-based flight trajectory prediction. In *2018 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.*
2. Rosenow, J., Lindner, M., & Scheiderer, J. (2021). Advanced flight planning and the benefit of in-flight aircraft trajectory optimization. Sustainability, 13(3), 1383.
3. Tian, Y., He, X., Xu, Y., Wan, L., & Ye, B. (2020). 4D trajectory optimization of commercial flight for green civil aviation. *IEEE Access*, 8, 62815-62829.
4. Shin, S., Lee, S., Son, C., Yee, K. (2022). Aircraft design parameter estimation using data-driven machine learning models. In *Proceedings of the 33rd Congress of the International Council of the Aeronautical Sciences* (ICAS 2022).
5. Li, C., & Braun, M. (2022). Analyzing aircraft turnaround processes using small dataset relationships.
6. Calvo Fernández, E., Cordero, J. M., Vouros, G., Pelekis, N., Kravaris, T., Georgiou, H., Fuchs, G., Andrienko, N., Andrienko, G., Casado, E., Scarlatti, D., & Costas, P. (2017). DART: A machine-learning approach to trajectory prediction and demand-capacity balancing
7. Zhao, L., Zhang, J., & Li, L. (2019). Trajectory forecasting with neural networks: An empirical evaluation and a new hybrid model. *IEEE Transactions on Intelligent Transportation Systems*, 20(6), 2095-2106. https://doi.org/10.1109/TITS.2018.2879827
8. Zeng, W., Chu, X., Xu, Z., Liu, Y., & Quan, Z. (2022). Aircraft 4D Trajectory Prediction in Civil Aviation: A Review. *Aerospace,* 9(2), 91.
9. Welch, G., & Bishop, G. (2006). An introduction to the Kalman filter
10. Prevost, C. G., Desbiens, A., & Gagnon, E. (2007). Extended Kalman filter for state estimation and trajectory prediction of a moving object detected by an unmanned aerial vehicle. In *Proceedings of the IEEE International Conference on Control Applications* (pp. 1058-1063). IEEE.
11. Bondarenko, D., & Baskin, K. (2017). Big History, Complexity Theory, and Life in a Non-Linear World. From Big Bang to Galactic Civilizations: *A Big History Anthology*, 3, 183-196.
12. Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT Press.
13. Xu, T., Zhang, Q., Liu, Y., & Wang, X. (2023). Adaptive real-time 4D trajectory prediction in the presence of dynamic weather and

air traffic conditions. *IEEE Transactions on Intelligent Transportation Systems.*
14. Pang, Y., Liu, X., Zhang, M., & Huang, Y. (2023). Real-time adaptability in 4D trajectory prediction under dynamic conditions. *Journal of Air Transport Management*, 102, 114798. https://doi.org/10.1016/j.jairtraman.2023.114798
15. Ma, J., & Tian, L. (2023). Predictive modeling for aircraft trajectories based on hybrid CNN-LSTM networks. *IEEE Transactions on Intelligent Transportation Systems*, 24(7), 1326–1341. https://doi.org/10.1109/TITS.2023.3034893
16. Zhang, Y., Zhang, D., & Jiang, H. (2023). Review of Challenges and Opportunities in Turbulence Modeling: A Comparative Analysis of Data-Driven Machine Learning Approaches. *Journal of Marine Science and Engineering*, 11(7), 1440. https://doi.org/10.3390/jmse11071440
17. Staudemeyer, R., & Morris, E. (2024). – Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks. Ar5iv. https://ar5iv.org/html/1909.09586
18. Ma, L., Meng, X., & Wu, Z. (2024). Data-Driven 4D Trajectory Prediction Model Using Attention-TCN-GRU. Aerospace, 11(4), 313.
19. O. E. Drummond, X. R. Li, and C. He, "Comparison of Various Static Multiple-Model Estimation Algorithms," in *Proc. 1998 SPIE Conf. on Signal and Data Processing of Small Targets*, vol. 3373, pp. 510–527, 1998.
20. Naser, M.Z., Alavi, A.H. Error Metrics and Performance Fitness Indicators for Artificial Intelligence and Machine Learning in Engineering and Sciences. Archit. Struct. Constr. 3, 499–517 (2023). https://doi.org/10.1007/s44150-021-00015-8
21. Alligier, R., Gianazza, D., & Durand, N. (2023). Machine Learning and Trajectory Prediction: Why Euclidean Metrics Mislead. Journal of Air Transportation, 31(2), 45-59.
22. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.
23. Bai, Y., Yu, W., & Li, H. (2023). Evaluating spatial data models: Metrics and their implications for geographic predictions. *Journal of Spatial Science*, 68(1), 12–29.
24. Zhang, L., & Han, P. (2023). Tuning hyperparameters for deep learning: A systematic approach to optimization. Computational Intelligence, 39(3), 345–360
25. Aliferis, C., & Simon, G. (2024). Overfitting, Underfitting, and General Model Overconfidence and Under-Performance Pitfalls and Best Practices in Machine Learning and AI. In *Artificial Intelligence and Machine Learning in Health Care and Medical Sciences: Best Practices and Pitfalls* (pp. 477-524). Cham: Springer International Publishing.
26. Wang, R., & Chen, Y. (2023). The role of feature engineering in addressing data dependency issues in machine learning. *Expert Systems with Applications*, 226, 120079
27. Liu, X., Zhang, Q., & Wang, J. (2023). Comparing machine learning and traditional statistical models: A metric-based performance analysis. *International Journal of Data Science*, 12(2), 145–162.
28. Lee, J., & Shin, K. (2023). Handling scalability and overfitting in deep learning for big data analysis. *Journal of Big Data*, 10(1), 45–61
29. Besse, P. C., Guillouet, B., Loubes, J. M., & François, R. (2016). "Review of nonlinear trajectory prediction methods for aircraft." *Journal of Aerospace Information Systems*, 13(3), 123-138
30. Nguyen, D. T., Le, L., & Sharma, R. (2021). "Limitations of deep learning for spatial data: A case study in aviation trajectory prediction." *IEEE Transactions on Intelligent Transportation Systems*, 22(5), 2765-2776
31. Letrache, K., & Ramdani, M. (2023). *Explainable Artificial Intelligence: A Review and Case Study on Model-Agnostic Methods*. IEEE. https://ieeexplore.ieee.org/abstract/document/10373722
32. Wichert, A. (2020). P*rinciples of quantum artificial intelligence: Quantum problem solving and machine learning*. World Scientific. https://doi.org/10.1142/9789811224317_0001
33. Wong, T. T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross-validation. Pattern Recognition, 48(9), 2839–2846. https://doi.org/10.1016/j.patcog.2015.03.009
34. Liu, X.; Kounadi, O.; Zurita-Milla, R. Incorporating Spatial Autocorrelation in Machine Learning Models Using Spatial Lag and Eigenvector Spatial Filtering Features. ISPRS Int. J. Geo-Inf. 2022, 11, 242. https://doi.org/10.3390/ijgi11040242

■ Author

Harshitha Vishnu is a freshman at Fremont High School in Sunnyvale, California. She has a deep passion for STEM, especially in the fields of Aerospace Engineering and Artificial Intelligence. She aims to pursue a future in AI and aerospace engineering, with a focus on solving real-world challenges through technology and innovation.

DOI: 10.36838/v7i8.5