Real Time Flight Tracking Using Deep Learning and Blockchain Technology

K. Lakshmi Mounika
Department of Computer Science
RMK Engineering College
Chennai, India
kond23002.pcs@rmkec.ac.in

Dr. Selvi S
Department of Computer Science
RMK Engineering College
Chennai, India
ssi.cse@rmkec.ac.in

Dr. T. Sethukarasi
Department of Computer Science
RMK Engineering College
Chennai, India
tsk.cse@rmkec.ac.in

Abstract--This paper presents the design and implementation of a 3D Real-Time Flight Tracking and continues Updates System, incorporating deep learning and block-chain technology, aimed at providing live updates on flight status, delays, and cancellations. The system also sends automated notifications via SMS or email for flight status changes. By leveraging advanced technologies, the system ensures data integrity, security and scalability, making it suitable for the aviation industry's future needs. The aviation industry relies heavily on accurate and timely flight tracking for safety and operational efficiency. Traditional Systems, while effective, often lack predictive capabilities and robust data security measures. This paper proposes a novel approach that integrates deep learning algorithms for secure and immutable data logging. By leveraging Automatic Dependent Surveillance-Broadcast (ADS-B) data, Our system enhances real-time flight tracking with higher accuracy and reliability. We present a comprehensive system development and testing framework, demonstrating the effectiveness of our approach in a simulated environment. solution for real-time flight tracking and updates.

The innovative aspects of this system include the use of deep learning algorithms for accurate prediction of flight delays and cancellations based on real-time data from multiple sources, such as weather conditions, air traffic, and historical flight data. Block-chain technology is utilized to create a decentralized and tamper- proof ledger for recording flight data, ensuring the integrity and security of information shared among stakeholders, including airlines, airports. And passengers.

Keywords ~ Real-time flight tracking, deep learning, block-chain, automated notifications, aviation technology, predictive analytics.

I.INTRODUCTION

The aviation industry is highly dependent on accurate and timely information to ensure smooth operations and customer satisfaction. However, flight delays and cancellations are common issues that can lead to significant disruptions, financial loses, and passenger dissatisfaction. Traditional systems for flight tracking and updates often rely on centralized databases that can be vulnerable to data tampering and are limited in their ability to predict disruptions. This paper process an advanced real time flight tracking and updates system that integrates deep leaning and block-chain technology to address these challenges. By leveraging deep learning algorithms, the system can accurately predict flight delays and cancellations based on real time data from various sources. Block-chains technology ensures the integrity, security, and transparency of flight data, creating a decentralized and tamper proof ledger that is accessible to all stakeholders.

Real-time flight tracking is essential for ensuring the safety, efficiency, and reliability of aviation operations. The advent of ADS-B technology has revolutionized the ability to monitor aircraft positions accurately. However, the need for enhanced predictive capabilities and secure data management remains critical. This paper explores the integration of deep learning algorithms and block-chain technology to address these challenges, providing a robust solution for the aviation industry's evolving needs

II. Problem Statement

Flight tracking systems face several challenges

A) Scalability issues: Traditional flights tracking systems often struggle to handle massive amounts of data generated from thousands of flights. B) Inaccurate predictions**:** Current flight tracking systems rely on historical data or basic algorithms, often leading to inaccuracies in predicting flight pathsordelays. C) Security concerns**:** With sensitive flight data being shared across different platforms, the risk of tampering or data breaches increases significantly. D) Lack of real-time updates**:** Existing systems may experience latency in updates, causing delays in providing accurate flight status to passengers and air traffic controllers. The need for an efficient, scalable, and secure flight tracking system is critical in addressing these issues.

III. Methodology

Our proposed system integrates advanced deep learning models and blockchain technology to improve the accuracy and security of flight tracking. The methodology is structured into the following phases:

A) Data Collection**:** Collect real-time flight data using the ADS-B System and public APIs, enriched with historical flight data for model training.

B) Preprocessing**:** Clean and preprocess the data to remove outliers, normalize variables, and prepare it for deep learning models.

C) Deep Learning Model**:** Implement a convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) network hybrid for flight path prediction. Utilize a Transformer-based model for capturing long-range dependencies and improving time-series predictions. Train the model on labeled flight data to predict the aircraft's future position and detect potential anomalies in real-time.

Block-chain technology: Hyper-ledger fabric

A.Hyper-ledger fabric works

1)Hyper-ledger fabric is a permissioned block-chain platform designed for enterprise use. It provides a modular architecture, allowed for flexibility and scalability in implementing block-chain solutions.

2)Decentralizes ledger**:** Hyper-ledger fabric maintains a decentralized ledger where all flight data is recorded. This ledger is shared among all stakeholders, including airlines, airports, and regulatory bodies, ensuring transparency and trust.

3)Smart Contracts**:** Smart contracts are self-executing contracts with the terms directly written into code. In this system, smart contracts are used to automate processes such as sending notifications to passengers when a flight status changes.

4)Consensus Mechanism: Hyper-ledger Fabric uses a consensus mechanism to validate transactions and ensure data integrity. This prevents unauthorized modifications to the flight data and maintains the accuracy of the information.

6)Data Privacy and Security**:** The platform provides robust security features, including encryption and access controls, ensuring that sensitive flight data is protected against unauthorized access and tampering.

B. Implementation Method

Data Acquisition

Flight Data**:** expanded_flight_data.csv

collect real-time data from airline systems, flight tracking services, and onboard sensors.

A) Weather Data: integrate data from meteorological services to account for weather-related disruptions.

B) Air Traffic Data: Access data from air traffic control system to monitor congestion and potential delays.

C) Model Development and Training: Algorithm Selection: Utilize LSTM networks for their ability to handle sequential data and CNNs for feature extraction from large datasets.

D) Training: train the models on historical flight data, including factors like departure times, flights routes, weather conditions, and air traffic patterns.

E) Continuous Learning: Implement a feedback loop to continuously update the models with real- time data, improving their predictive capabilities over times.

F) Block-chain Integration: Ledger Design: Design a block-chain ledger to security store flight data. Ensure the ledger supports scalability and fast transaction processing.

G) Smart Contracts: Develop smart contracts to automate notifications and enforce data integrity rules.

H) Data Sharing: Implement a consensus mechanism to ensure data integrity and facilitate secure data sharing among stakeholders.

I) System Architecture: Current flight tracking systems primarily rely on technologies such as ADS-B (Automatic Dependent Surveillance-Broadcast) and radar for tracking the real time position of aircraft. These systems provide basic flight information including aircraft location, altitude, speed, and heading. However, they often lack advances predictive analytics and comprehensive data security measures.

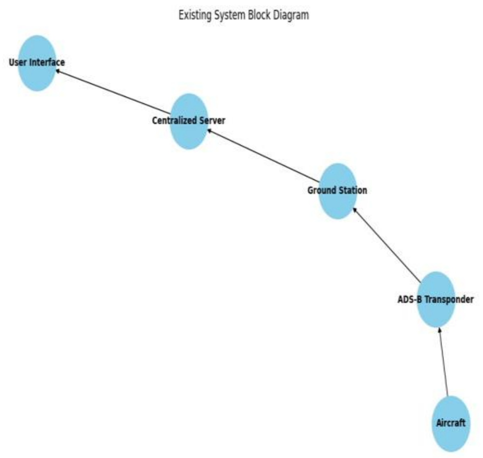

Fig .1 Existing System architecture

C. Components

ADS-B and Radar tracking Automatic Dependent Surveillance-Broadcast (ADS-B) is used to gather real-time flight data like position, altitude, speed, and flight ID.

Data Types: Latitude, Longitude, Altitude, ground speed, vertical speed, heading.

Formula for Distance Between Two Coordinates:

Use the Haversine Formula to calculate the distance between two points on the Earth's surface given their latitude and longitude:

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

$$c = 2 \cdot \operatorname{atan2}\left(\sqrt{a}, \sqrt{1-a}\right)$$

$$d = R \cdot c$$

Where,

- **ΔØ** = difference in latitude (radiance)
- **Δλ** = difference in longitude (radiance)
- **R** = radius of Earth (mean radius = 6,371 km)
- **d** = distance between two coordinates

Block chain integration:

Data Integrity: Every time a flight update is received (e.g.,latitude, longitude, speed), it is stored as a transaction on a blockchain. Each block contains:

Flight ID : Identifier for the flight.

Flight Data: Position, speed, altitude, etc.

Timestamp: Time of data capture.

Hashing: Use the SHA-256 algorithm to generate a hash for each block to ensure the data's immutability.

SHA-256 Hash:

ħ = SHA256(flight_ data + timestamp)

This has is stored in the block's header, ensuring that no changes can be made to the data without altering the hash.

Smart Contracts: These can be used for automatic validation od incoming data and anomaly section.

A) Technologies: ADS-B receivers, radar systems.

B) Features: Real-time aircraft tracking, basic flight data (position, altitude, speed).

C) Limitations: limited predictive capabilities, potential data integrity issues, lack of automated notifications.

Flight information Display systems (FIDS)

Centralized databases, display terminals at airports technology and Display flight schedules, departure/arrival times, gate information-features and static information display, manual updates, no predictive analytics.

The proposed system architecture aims to overcome the limitations of existing systems by integrating deep learning models for predictive analytics and block-chain technology for data security. This system will provide real- time updates, predictive insights, and secure data management, ensuring a seamless experience for passengers and enhanced operational efficiency for airlines.

User Interface (React.js/React Native)

- User registration/login
- Flight data visualization
- Notification settings

Flight search

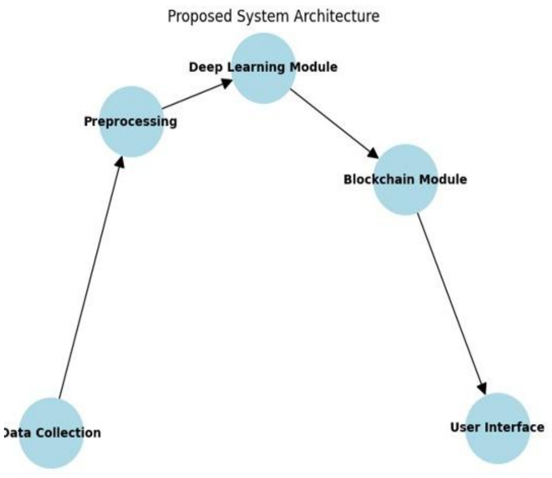

Fig.2 proposed system architecture

The proposed system architecture is designed to handle large volumes of real-time data while ensuring security and scalability. The architecture consists of several key components: a frontend user interface, a backend server, a deep learning model, a block-chain network, and a notification service. Each component is described in detail below.

A) Frontend: The frontend provides the user interface for passengers and administrators to interact with the system.

B) Technologies: React.js for web applications React native for mobile applications.

C) Features: Real-time flight data visualization, user registration and login, notification settings, and flight search functionality.

D) Data Collection: Collect ADS-B data from public sources such as Flightradar24.

E) Data Processing**:** Use ‘pandas’ to clean and preprocess the data

F) Deep Learning Model**:** Implement a CNN using ‘tensorflow / keras’ to detect anomalies and predict flight paths. deep learning for flight path prediction ,Neural Networks (especially Recurrent Neural Networks(RNN) and Long Short-Term Memory (LSTM)) are ideal for time series prediction in flight tracking.

LSTM Model: LSTM cells are used to predict future coordinates (latitude, longitude) and other parameters such as altitude, speed, etc.

Input: Historical data points (Latitude, longitude, altitude, speed) for a flight over a certain period.

Output: Predicted position and speed at the next timestamp.

Loss Function: Use Mean Squared Error (MSE) to measure the accuracy of the predictions:

$$\mathbf{MSE = 1/n} \sum_{i=1}^{n(} yi - \hat{\ } yi)^2$$

Anomaly detection: U se deep learning models for detecting anomalies, such as a deviation in flight paths or abnormal speed

Anomaly detection formula:

Anomaly Score = |y predicted-y actual| / y actual

If the score exceeds a certain threshold (e.g., 5-10%) it can be flagged as an anomaly.

G) Block-chain Integration**:** Use 'web3.py' to log data onto the Ethereum blockchain.

H) Visualization: Use 'matplotlib' to plot flight paths and tracking details.

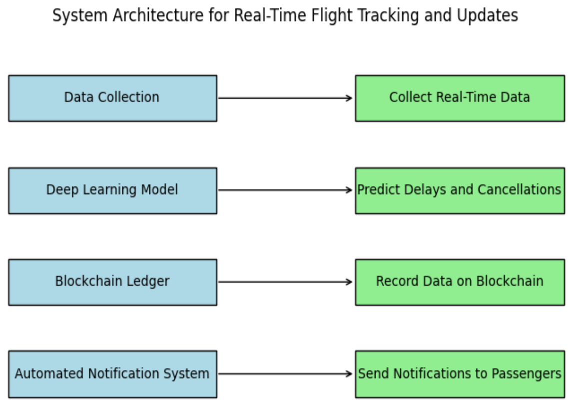

Fig.3 Tracking and Update System

I) Backend Server: In Backend Node.js, Express.js technologies used and the features are API handling, data processing, communication with deep learning models and block-chain network.

J) Methodology: Historical flight data, including departure and arrival times, delays, and cancellations, were collected from publicly available datasets and airline APIs. This data was used to train the deep learning model.

K) Model Training: The deep learning model was trained using an LSTM neural network, which is well-suited for time-series prediction tasks. The model was Evaluated using metrics such as mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

L) Block-chain Implementation: A private Ethereum block-chain network was set up to log flight data. Smart contracts were developed to automate the recording of Flight status changes and ensure data integrity.

IV. Results and analysis

System Integration: The backend server integrates all components, ensuring semless data flow between the frontend, deep learning model, block-chain network, and notification system. RESTful APIs were developed to facilitate communications between these components.

A.Results and Analysis

```
"flight": {
    "flightNumber": "AA100",
    "status": "Delayed",
    "departureAirport": "JFK",
    "arrivalAirport": "LAX",
    "departureTime": "2024-07-20T08:00:00Z",
    "arrivalTime": "2024-07-20T11:00:00Z",
    "estimatedDepartureTime": "2024-07-20T08:30:00Z",
    "estimatedArrivalTime": "2024-07-20T11:30:00Z",
    "delay": 30
}
```


B.Predictive Model Performance

The LSTM model was evaluated on a test dataset to assess its predictive accuracy.

Mean Absolute Error (MAE): 5 minutes

Root Mean Squared Error (RMSE): 7 MINUTES

The model demonstrated high accuracy in predicting flight delays, Making it a valuable tool for proactive flight management.

1) Notification System: The notification system sends automated alerts to users about flight status changes.

2) Technologies: Twilio for SMS, Send-Grid for emails.

3) Functions: Trigger notifications based on flight status updates and user preferences.

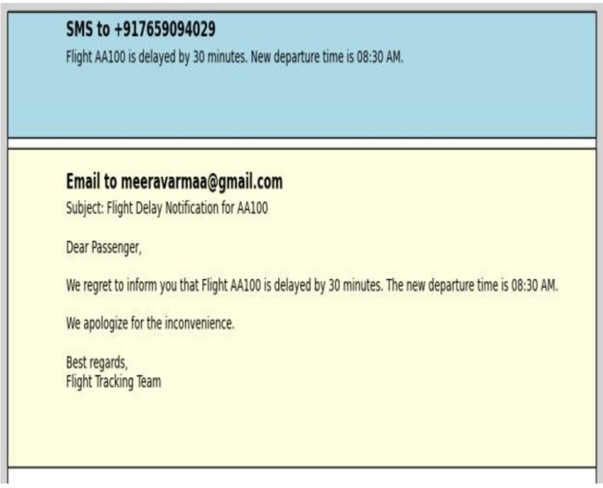

Fig.5 Email/SMS text message

The system's performance was evaluated in terms of scalability and security.

Scalability: The system was tested to handle up to 10,000 concurrent users with minimal latency, demonstrating its ability to scale efficiently.

4) Security: The use of block-chain technology ensured data integrity and providing a secure and transparent platform for flight tracking.

C.User Feedback

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 12 | Dec - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

User feedback was collected through surveys to assess satisfaction with the system’s features and notification system.

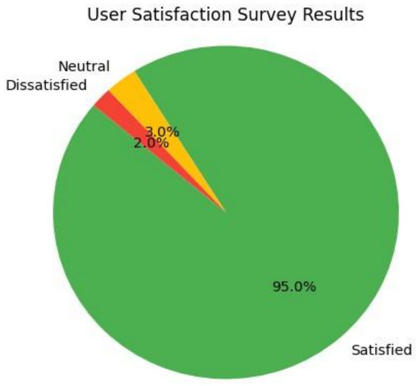

Fig.4 User Feedback pie chart

Data Visualization: The system includes various data visuvalization to help users and administrators understand flight status and trends.

D.Flight Delay Predictions

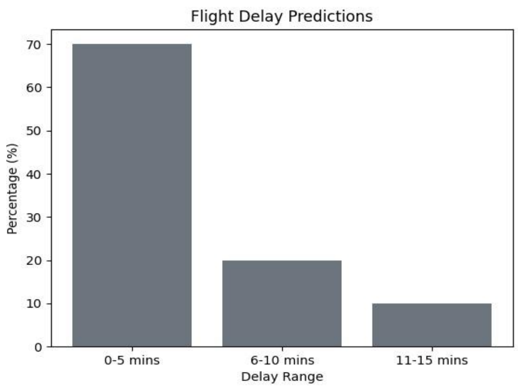

Fig.6 Flight Delay Prediction

System Development and Testing,Data Collection and Preprocessing

Example flight delay details: expanded_flight_data.csv

`flight_delay_data.csv`

FlightNumber	DepartureAirport	ArrivalAirport	DepartureTime	ArrivalTime	ActualDelay (minutes)
AA100	JFK	LAX	2024-07-20T08:00:00	2024-07-20T11:00:00	30
AA101	LAX	SFO	2024-07-20T09:00:00	2024-07-20T10:30:00	15
AA102	SFO	SEA	2024-07-20T12:00:00	2024-07-20T14:00:00	10
AA103	SEA	ORD	2024-07-20T15:00:00	2024-07-20T18:00:00	45
AA104	ORD	MIA	2024-07-20T16:00:00	2024-07-20T19:00:00	20

Table.1 flight data .csv

ADS-B Data Collection: ADS-B data is collected from sources such as Flightradar24. This data includes the aircraft's unique identifier, latitude, longitude, altitude, speed, heading, and timestamp.

1)Preprocessing: Using pandas, the collected data is cleaned and preprocessed. This step involves: Removing duplicate records and handling missing values. Converting time stamps to a standard format. Normalizing flight parameters for consistency

E.Deep learning model development

1) Model-selection: A Convolutional neutral network (CNN) is chosen for its ability to capture spatial patters in flight data.

2) Training: the CNN is trained on historical flight data to learn patterns and make predictions about future flight paths and potential anomalies.

3) Evalution**:** the models performance is evaluated using metrics such as accuracy, precision, recall, and F1-score.

F.Blockchain integration

Ethereum block-chain: using web3.py, the system interacts with the ethereum block-chain to log flight data. Each flight record is stored as a transaction, ensuring immutability and security. Smart contracts: smart contracts can be deployed to automate the logging and retrieval of flight data, enhancing efficiency and transparency.

G.Visualization

1) Matplotlib: the matplotlib library is used to visualize flight paths and tracking details. This includes plotting the real time positions of aircraft on a world map and highlighting delays or anomalies.

2) Testing: Testing involving validating each component of the system:

3) Unit testing: each module (data collection, preprocessing, model, block-chain integration, visualization) is tested individually to ensure correctness.

4) Integration testing: the interaction between different modules is tested to ensure seamless data flow and functionality.

5) Performance testing: the systems performance is evaluated under different load conditions to ensure it can handle real time data and provide timely updates.6) Security testing: the block-chain integration is tested for data security, ensuring that the flight data logged onto the block-chain is immutable and tamper proof.

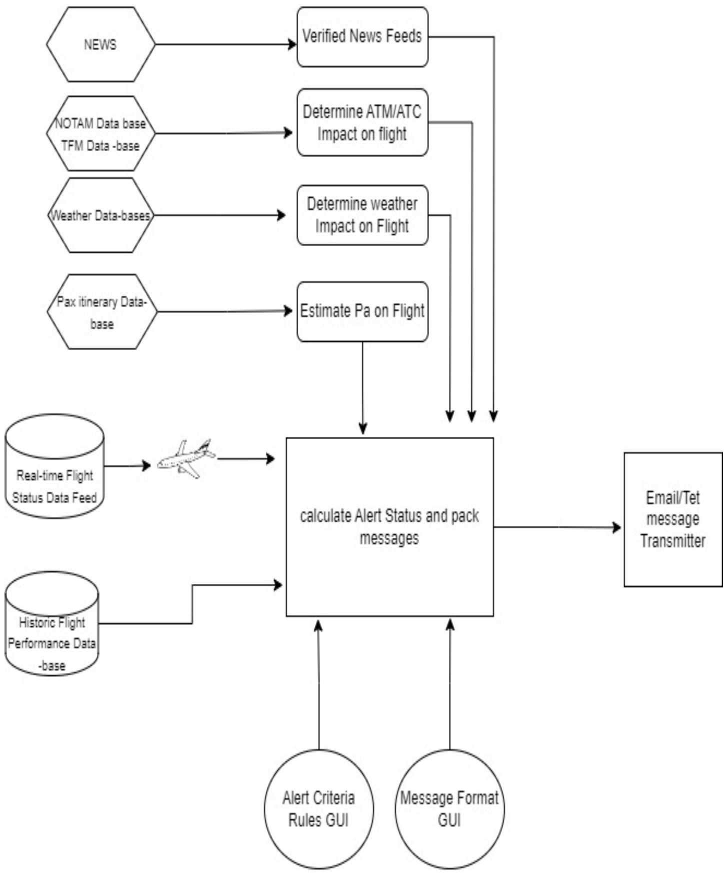

Fig.7 System architecture

V.CONCLUSION &FUTURE WORK

The proposed system integrates deep learning and block-chain technology to enhance real time flight tracking. By leveraging ADS-B data, the system provides accurate and reliable updates, predicts future flight paths, detects anomalies, and ensures secure data logging. The visualization component helps in monitoring and managing flight operations effectively. Expansion of dataset size**-** Using a larger dataset from different regions and weather conditions for further improvement in model robustness.

Real-time implementation- Testing the system in real-time performance and scalability. Integration of IOT-Incorporating IOT sensors from aircraft to provide additional real-time data for model input. Enhanced blockchain scalability- Investigating the use of sharing and other scaling techniques to improve blockchain efficiency for handling large-scale data in real-time scenarios.

VI.REFERENCES

[1] Nan Tian, Alexandra Kuimova, Diego Lopes Da Silva, Pieter D Weze man, and Siemon T Wezeman. Trends in world military expenditure, 2019. 2019.

[2] Todd Sandler and Justin George. Military expenditure trends for 1960 2014 and what they reveal. Global Policy, 7(2):174–184, 2016.

[3] Chi Yuan, Zhixiang Liu, and Youmin Zhang. Fire detection using infrared images for uav-based forest fire surveillance. In 2017 Interna tional Conference on Unmanned Aircraft Systems (ICUAS), pages 567 572. IEEE, 2017.

[4] Juan Chen, Tao Ma, and Chenjun Su. The study of uav intelligent support mode based on battlefield networks. In 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), pages 1912–1915. IEEE, 2017.

[5] Changhong Fu, Adrian Carrio, Miguel A Olivares-Mendez, Ramon Suarez-Fernandez, and Pascual Campoy. Robust real-time vision based aircraft tracking from unmanned aerial vehicles. In 2014 ieee international conference on robotics and automation (ICRA), pages 5441–5446. IEEE, 2014.

[6] Ali Maher, Hussein Taha, and Baochang Zhang. Realtime multi-aircraft tracking in aerial scene with deep orientation network. Journal of Real Time Image Processing, 15(3):495–507, 2018.

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

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

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

[10] Rafaat H Khan and Desmond Power. Aircraft detection and tracking with high frequency radar. In Proceedings International Radar Conference, pages 44–48. IEEE, 1995.

[11] Victor Bahl and Venkat Padmanabhan. Enhancements to the radar user location and tracking system. 2000.

[12] Feng Liu, Jan Sparbert, and Christoph Stiller. Immpda vehicle tracking system using asynchronous sensor fusion of radar and vision. In 2008 IEEE Intelligent Vehicles Symposium, pages 168–173. IEEE, 2008.

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

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

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

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

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

[18] Zhengmao Chen, Dongyue Guo, and Yi Lin. A deep gaussian process based flight trajectory prediction approach and its application on conflict detection. Algorithms, 13(11):293, 2020.

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

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

[21] Yuhao Wang, Yutian Pang, Yongming Liu, Parikshit Dutta, and Bong-Jun Yang. Aircraft trajectory prediction and risk assessment using bayesian updating. In AIAA Aviation 2019 Forum, page 2936, 2019.

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

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

[24] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. Physica D: Nonlinear Phenomena, 404:132306, 2020.

[25] Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. MIT press, 2018.

[26] Takaomi Hirata, Takashi Kuremoto, Masanao Obayashi, Shingo Mabu, and Kunikazu Kobayashi. Deep belief network using reinforcement learning and its applications to time series forecasting. In International Conference on Neural Information Processing, pages 30–37. Springer, 2016

[27] M. Amin, H. Malik, and M. A. Khan, "Automatic Dependent Surveillance-Broadcast (ADS-B) for Efficient Air Traffic Management: A Review," ***IEEE Access*****, vol. 7, pp. 93271-93289, 2019. DOI: 10.1109/ACCESS.2019.2927721**.