Advances in Aircraft and Spacecraft Science, *Vol. 11, No. 1 (2024) 33-53*
https://doi.org/10.12989/aas.2024.11.1.033

Enhancing air traffic management efficiency through edge computing and image-aided navigation

Pradum Behl[a] and S. Charulatha*

School of Aeronautical Sciences, Hindustan Institute of Technology and Sciences, Rajiv Gandhi Salai (OMR), Padur, Kelambakkam, Tamil Nadu 603103, India

(Received December 20, 2023, Revised February 19, 2024, Accepted May 10, 2024)

Abstract. This paper presents a comprehensive investigation into the optimization of Flight Management Systems (FMS) with a particular emphasis on data processing efficiency by conducting a comparative study with conventional methods to edge-computing technology. The objective of this research is twofold. Firstly, it evaluates the performance of FMS navigation systems using conventional and edge computing methodologies. Secondly, it aims to extend the boundaries of knowledge in edge-computing technology by conducting a rigorous analysis of terrain data and its implications on flight path optimization along with communication with ground stations. The study employs a combination of simulation-based experimentation and algorithmic computations. Through strategic intervals along the flight path, critical parameters such as distance, altitude profiles, and flight path angles are dynamically assessed. Additionally, edge computing techniques enhance data processing speeds, ensuring adaptability to various scenarios. This paper challenges existing paradigms in flight management and opens avenues for further research in integrating edge computing within aviation technology. The findings presented herein carry significant implications for the aviation industry, ranging from improved operational efficiency to heightened safety measures.

Keywords: air traffic management; data transmission; edge computing; flight management system; image processing; MATLAB; terrain analysis

1. Introduction

Edge computing empowers handling information nearer to the source, diminishing inertness and empowering constant investigation of huge measures of information produced via air traffic management frameworks. Edge computing can dissect authentic and continuous information to upgrade flight courses in light of elements like atmospheric conditions, airspace congestion, and eco-friendliness (Asif *et al.* 2023). According to Fazeldehkord *et al.* (2022), edge computing is a novel approach to meet the growing needs of the aviation sector. It is a decentralized computing architecture that places processing power closer to the data source. This in-depth study explores edge computing’s critical role in aviation, explaining its benefits, uses, and revolutionary influence on air traffic control.

Edge computing has become a game-changing paradigm as the aviation industry struggles with

*Corresponding author, Ph.D., E-mail: scharul@hindustanuniv.ac.in
[a]Ph.D. Student, E-mail: pradum.behl.15@gmail.com

http://www.techno-press.org/?journal=aas&subpage=7 ISSN: 2287-528X (Print), 2287-5271 (Online)

the demands of real-time data processing, low-latency communication, and adaptive solutions. Edge computing uses computer resources closer to end users or devices than typical cloud computing models, which process data in distant data centers. This results in faster processing, lower latency, and improved performance (Liu *et al.* 2019).

There are numerous advantages of integrating edge computing into avionics and aviation systems, which overcome the drawbacks of traditional methods. Edge computing in aviation systems reduces latency essential for real-time applications like autonomous vehicles by processing near data source, overcoming drawbacks of traditional methods. (Chen *et al.* 2021). Furthermore, Edge computing maximizes network capacity by minimizing continuous data transmission to distant cloud servers. This efficient utilization of bandwidth resources improves scalability and lessens network congestion (Koubaa *et al.* 2020).

A paradigm shift in improving privacy and security safeguards in the aviation industry is brought about by edge computing. Edge computing enhances aviation privacy and security by locally processing/storing sensitive data, mitigating transmission risks, providing better data control and reducing cyber threat exposure (Bekkouche *et al.* 2018). Moreover, edge computing enables apps to operate seamlessly in limited/intermittent network conditions, ensuring continuous functionality in challenging environments. (McEnroe *et al.* 2022).

The unparalleled expansion of the aviation sector worldwide has amplified strain on air traffic management systems. According to (Zhou *et al.* 2020), traditional approaches that depend on centralized processing find it difficult to meet growing demands, which causes delays, inefficiencies, and safety issues. A game-changing answer to these problems is edge computing, which provides dispersed processing powers that perfectly mesh with the dynamic nature of contemporary flight operations (Zheng *et al.* 2021).

Edge computing provides numerous advantages over traditional cloud models (Jianzhong, 2020). Firstly, there is reduced latency by processing near data sources, critical for real-time applications like AR, automation, and autonomous vehicles (Mijuskovic *et al.* 2020). Secondly, the network bandwidth is maximized by reducing the need for constant data transmission to the cloud, sending only relevant data or processed results minimizes network congestion and optimizes the utilization of limited bandwidth resources (Byonggon 2018). Additionally, edge computing enhances security and privacy by enabling local processing and storage of sensitive data, mitigating transmission risks (Dogea *et al.* 2023). This localized approach provides better data control and reduced cyber risk exposure. Edge computing also enables app functionality in limited/disconnected network conditions. (Hayat *et al.* 2021). Edge devices can function and make crucial choices by processing data locally, which guarantees continuous functioning even in difficult network circumstances (Subburaj *et al.* 2023).

Air traffic management systems are under increased strain as a result of the exponential growth of the global aviation industry in recent years. As demonstrated by Satyanarayanan (2017), traditional approaches that rely on centralized processing are finding it difficult to meet the growing needs. Among the difficulties mentioned by Naveen and Kounte (2019) are the widespread problems of delays, inefficiencies, and safety concerns that call for a review of current procedures.

To provide a dynamic and flexible solution for the current aviation landscape, this study takes advantage of the decentralized processing capabilities of edge computing and the accuracy of image processing algorithms. However, while edge computing offers significant advantages in terms of latency reduction and bandwidth conservation, it also presents certain challenges. These include issues related to its reliability and its limited use in specific sectors such as aviation.

Increased air travel due to population/economic growth poses major challenges for current air traffic control (Shrestha *et al.* 2021). Traditional centralized cloud-based systems cannot cope with surging, dynamic flight operations volume and complexity, causing delays and inefficiencies (Wang *et al.* 2022). This study recognizes the urgency to address current flight operation issues exacerbated by climate change, aiming to develop an adaptable system meeting present and future industry needs.

The driving force behind the use of edge computing in aviation over more conventional cloud-based methods is its flexibility and potential for a wide range of flying circumstances. Although edge computing is widely used in drone technology, its use in manned aircraft adds another level of complexity to the optimization of air traffic control (Dubey *et al.* 2019). Edge computing increases the effectiveness, safety, and adaptability of air traffic management systems by optimizing data transfer speeds and enabling real-time terrain-informed flight path recommendations (Liu *et al.* 2023). While edge computing shows promise for aviation applications, reliability concerns pose adoption challenges that motivate further research like this study.

Donald and John (2016) present a novel approach to address the challenges of Inertial Measurement Unit (IMU) drift and Global Navigation Satellite Systems (GNSS) unavailability in airborne platforms. The proposed algorithm combines Content-Based Image Retrieval and nonparametric Bayesian estimation strategies to provide a rough position estimate for an aircraft. By leveraging aerial and satellite imagery, it aims to determine the aircraft's location within a rectangular search area of tens of kilometers. Designed for scenarios with large position uncertainty, such as extended GNSS outages, the algorithm has effectively reduced navigation system uncertainty by several orders of magnitude in flight tests. The paper outlines the feature database creation process, navigation algorithm, and potential applications for image-aided navigation in challenging environments.

Image-aided navigation leverages aerial/satellite imagery to enhance navigation accuracy and reliability, especially when traditional IMU/GNSS methods are unavailable or unreliable, which is critical for airborne platforms. It compares real-time aircraft images with a pre-stored database using techniques like Content Based Image Retrieval and Bayesian estimation to estimate position. This approach can significantly reduce navigation uncertainty, even during extended GNSS outages with large position uncertainties, ensuring safe and efficient aircraft operation.

In the context of the paper, we explore how edge computing can further enhance the performance of image-aided navigation systems. For instance, edge computing could enable faster processing of the image data, thereby reducing latency and improving the real-time responsiveness of the navigation system. Furthermore, edge computing could facilitate more efficient use of bandwidth, as only the most relevant data would need to be transmitted over the network.

In this research, they attempt to establish a novel framework for the seamless integration of image-aided aircraft navigation systems with edge computing (Jérémy 2013). This work provides real-time terrain-aware flight path recommendations and optimizes data transmission speeds by focusing on critical factors like flight-to-ground station distance, altitude profiles, and flight path angles. 'Ground stations' here refer to virtual observation/communication points along the simulated flight path at 25%, 50%, and 75% distances. Through rigorous analysis and validation, it aims to significantly enhance the efficiency, security, and adaptability of air traffic management systems.

2. Methodology

The following segment presents an initial analysis, succeeded by comprehensive research undertaken to evaluate the feasibility of edge computing technology in the field of aviation navigation and communication. The first case study compares the effectiveness of edge computing with traditional methods for improving flight characteristics, with an emphasis on avionics equipment. This preliminary study provides a basis for a more thorough analysis of the integrated edge computing-based image-assisted flight navigation system, which is covered later in this work.

In the domain of aviation safety and efficiency, navigation frameworks stand as stalwart guardians, directing flights through the tremendous field of the skies. In the past, these systems relied on traditional techniques such as ground-based navigational aids (like VORs and NDBs), radar surveillance, and procedural control to identify the ideal altitudes and flight paths. However, these methods have limitations in terms of coverage, accuracy, and capacity. But the emergence of edge computing has opened a new chapter that holds the potential to revolutionize navigation methods. A pioneer of innovation, edge computing brings real-time data processing and analysis capabilities. This technical wonder enables dynamic optimization of flight parameters, enabling navigation systems to achieve previously unheard-of levels of flexibility and agility through IoT-enabled systems and real-time data transmission. Standardized development and testing frameworks tailored to aviation edge computing systems are lacking at present. Methodologies will need to evolve with further research.

2.1 Preliminary methodology (Comparison between edge computing and conventional cloud-based computing)

This first study aims to compare the effectiveness of edge computing vs traditional approaches in aviation navigation, specifically focusing on the avionics area. This initial analysis prepares the foundation for a more thorough investigation that will explore the details of an integrated edge computing-enabled image-assisted flight navigation system. The initial research methodology involves the use of a simulated Flight Management System (FMS) to enable a comparative evaluation of edge computing and traditional approaches across a predetermined flight path. The traditional method involves figuring out the ideal height based on certain distance thresholds, which typically involve predefined waypoints and air traffic control instructions, in this work, the waypoints are given by the user. These thresholds, while effective, are static and do not account for real-time changes in flight conditions. On the other hand, the edge computing methodology dynamically optimizes altitude by integrating real-time data processing and analysis, offering a more flexible and responsive approach. The execution time of each technique is measured to evaluate its efficiency.

The following is a summary of the algorithmic framework used to perform the comparison:
The algorithm simulates an autonomous flight management system (FMS) that determines optimal altitudes during a flight based on the distance traveled. It incorporates edge computing techniques to detect and correct invalid flight data at the edge rather than transmitting it back to the core system.

The key steps are:

1. The starting aircraft position (latitude, longitude, altitude) and a series of waypoints defining the flight path are input by the user.

2. The conventional method iterates through each waypoint to:

• Calculate the current position using the waypoint data.
• Determine the distance travelled from the previous waypoint using the Haversine formula.
• Establish a targ*et al*titude based on the distance thresholds.
• Make a slight adjustment to the targ*et al*titude, considering the altitude at the start of this iteration (referred to as the 'initial altitude').
• Update the 'initial altitude' for the next iteration to the targ*et al*titude.
3. The edge computing method performs the same steps, but also:
• Checks if the distance is within the allowed range.
• Calls edge computing functions to correct any invalid flight data.
• Calculates a new optimal altitude based on the corrected flight data.
• Adjusts the aircraft's altitude to match this new optimal altitude.
• Passes this adjusted altitude back to the main loop for the next iteration.
4. The edge computing functions specifically:
• Detect outliers in the current altitude
• Fix the outliers by setting them to valid altitudes
• Return the fixed optimal altitude to the main loop
5. No actual trajectory corrections or manoeuvres are simulated. Only the optimal altitudes are calculated and set based on distance thresholds.
6. The waypoints represent the current aircraft position. The optimal altitudes are calculated separately based on the distance travelled from the starting position.

This algorithmic framework lays the foundation for the initial study by contrasting edge computing-based and conventional navigation systems (Cloud-based). It makes it easier to evaluate how well they performed in terms of timing accuracy and ideal altitude computation. The second study's methodology is a ground-breaking combination of edge computing and image-aided flight navigation systems that has the potential to completely transform air traffic control and navigation. By combining the strengths of both technologies harmoniously, important issues in contemporary aviation are addressed.

2.2 Edge computing-based system

The ability to dynamically compute real-time data transfer speeds between aerial vehicles and ground stations is the fundamental component of this system. The system automatically adapts to a variety of flying conditions by considering critical characteristics like flight-to-ground station distance, destination coordinates, altitude profiles, and flight path angles. Even in the face of fluctuating conditions, this dynamic computation guarantees the best possible communication throughput throughout the trip.

Concurrently, the integration of cultured image processing techniques constitutes a pivotal facet of this approach. Through the application of edge detection, color thresholding, and gradient analysis, the system meticulously refines user-selected images, furnishing a sharper and more precise representation. This enhancement procedure assumes a crucial role in subsequent terrain analysis, guaranteeing that the input image effectively delineates the terrain it depicts.

In addition, the method makes use of an advanced topography analysis algorithm that can discriminate between flat and mountainous areas. Gaussian curvature is used in this analysis to provide a more sophisticated curvature analysis by utilizing gradient and curvature calculations. The system can improve safety and operating efficiency by fine-tuning flying trajectories by understanding terrain characteristics. All things considered, the suggested strategy provides an allencompassing answer to the problems facing the aviation sector.

The algorithmic framework can be briefly summarized as:

- Data Transmission Speed Calculation module: This algorithm is at the heart of the system. The real-time data transmission speeds between aircraft and ground stations are dynamically computed. Crucial factors considered by the computation include flight path angles, altitude profiles, user-provided destination coordinates, and the separation between the aircraft and the ground station. This calculation is done strategically throughout the flight route to allow for flexibility in a variety of situations.
- Image Enhancement Module: To improve the user-selected images, the image processing algorithm uses a number of advanced approaches. This procedure includes gradient analysis, color thresholding, and edge identification using the Canny method. Together, these processes improve the input image and set it up for the terrain analysis that follows.
- Terrain Analysis Module: Differentiating between plain and hilly environments is largely dependent on the terrain analysis method. It depends on gradient and curvature computations; for curvature analysis, Gaussian curvature is used. This offers important information about the characteristics of the area. Slope and curvature threshold values are dynamically adjusted under certain contextual conditions to improve predicting accuracy.
- Flight Path Optimization module: Determining flight path angles to identify any departures from the planned trajectory is a crucial step in the process. When deviations are detected, the system recommends adjustments to keep the intended flight path.

Results are obtained using simulations run through MATLAB code created using the algorithm that is covered later in the paper, given the computational nature of this research

By integrating functions to calculate distances, flight route angles, and perform terrain analysis, the MATLAB code creates a simulated environment for thorough study. Plots displaying flight trajectories, altitude variations, speed profiles, and terrain analysis results provide a clear visual representation of the results.

This carefully thought-out approach provides a solid foundation for investigating how edge computing-enabled image-aided flight navigation systems can improve air traffic control and navigation effectiveness. Through the methodical application of the supplied code, the research aims to validate the effectiveness of this innovative method and clear the way for future developments in aviation technology.

3. Implementation details

This section explores how the suggested strategy is put into practice, including a thorough rundown of the system’s code structures, important algorithms, technological issues, and data flow. The first code segment begins by requesting some important information from the user, which includes the flight’s initial altitude, longitude, and latitude. It then requests data regarding the number of waypoints that define the flight route, as well as the associated longitude, latitude, and altitude for every waypoint.

After that, the software starts to simulate the Flight Management System (FMS) navigation using edge computing and conventional computing techniques. When using the traditional method, the software calculates the current altitude, current position, and distance to each waypoint as it iterates through them. It uses pre-established thresholds to calculate the ideal altitude and shows the altitude together with the present position. Every waypoint is subject to this iterative procedure, which changes the beginning altitude as needed.

The edge computing approach follows a similar iterative procedure. This time, the application uses advanced edge computing algorithms to compute the ideal altitude in addition to calculating the current position, distance, and height. When flight data deviates from permitted bounds, efforts are made to use cutting edge computing techniques to correct altitude. This two-method simulation allows for a thorough assessment of FMS navigation in a variety of scenarios.

Furthermore, the code's operation depends on a number of auxiliary functions, each of which carries out particular computations and logic to make the simulations easier. These tasks are essential to maintaining accurate and consistent results during the program's run.

In the field of aviation, the need for efficient and adaptable systems is paramount. Here is an innovative approach to address these needs through two key strategies: Edge Computing Integration and Modular Design. These strategies not only enhance the system's performance but also ensure its adaptability to future advancements, Edge Computing Integration allows the system to dynamically adapt to changing flight conditions and data loads. Modular Design ensures easy integration of future advancements by compartmentalizing functionalities, allowing for efficient updates and expansion of capabilities. Let's delve into these concepts in detail to understand their significance and implementation in our system.

Also, we explore the subtleties of integrating edge computing technology, for which the code offers a useful example of communication and landscape analysis. The simulations are carried out with great care, calculating the ideal altitudes and gathering copious amounts of data for analysis. Supporting functions use sophisticated algorithms to guarantee accurate calculations. The system gains the agility to dynamically adjust to changing flight conditions by distributing computational duties closer to the data source-a critical component in modern aviation.

- Edge Computing Integration: This makes it possible to calculate transmission speed and process data in real-time, simulating the effectiveness of real edge computing settings. In this implementation, edge computing integration plays a crucial role. The system mimics the features of edge computing settings by utilizing MATLAB as the computational platform.
- Modular Design: Discrete functions are assigned specified duties in an implementation that follows a modular design methodology. This makes the code easier to read and maintain and makes it easier to integrate new features in the future. The code determines the great-circle distance between two sites on the surface of the Earth given their latitude and longitude coordinates using the Haversine formula. It is used in this context to calculate the distance between the cities of departure and arrival, which is important for a number of flight planning and navigational tasks. The Haversine formula calculates the distance between two points on the surface of a sphere, typically the Earth. It is given by

$$hav(\theta) = sin^2\left(\frac{\theta}{2}\right) \tag{1}$$

The haversine formula for the central angle (*d/r*) between two points with latitude φ and longitude λ is

$$hav\left(\frac{d}{r}\right) = hav(\varphi_1 - \varphi_2) + cos(\varphi_1)\, cos(\varphi_2) hav(\lambda_2 - \lambda_1) \tag{2}$$

Where:

- d is the distance between the two points along a great circle of the sphere,
- r is the radius of the sphere,
- φ_1, φ_2 are the latitude of point 1 and latitude of point 2,

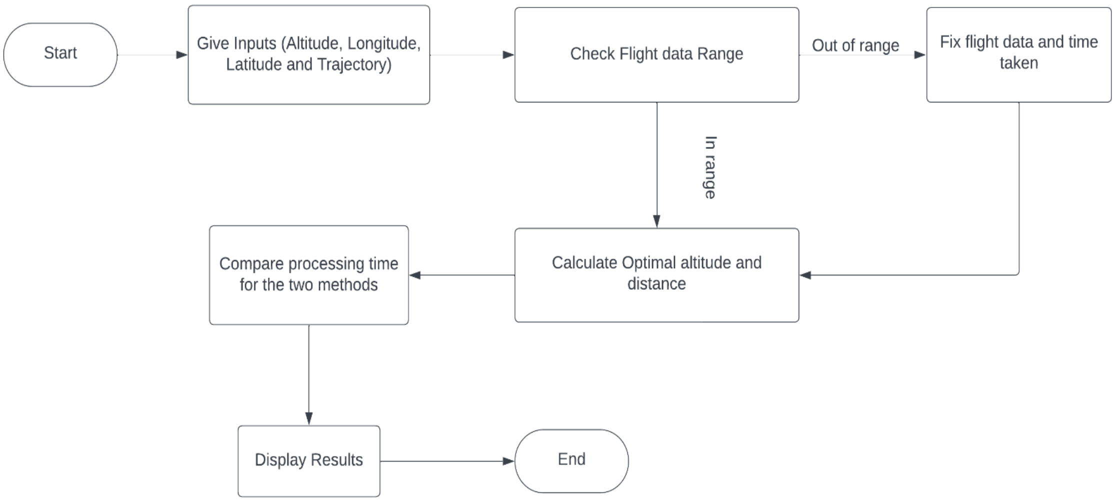

Fig. 1 Flow chart for preliminary code (Comparison of conventional and edge computing)

• λ_1, λ_2 are the longitude of point 1 and longitude of point 2.

To solve for the distance d, we can apply the inverse haversine (archaversine) or use the arcsine (inverse sine) function

$$d = r \cdot archav\left(hav\left(\frac{\mathrm{d}}{\mathrm{r}}\right)\right) = r \cdot arc\,sin\left(\sqrt{hav\left(\frac{\mathrm{d}}{\mathrm{r}}\right)}\right) \tag{3}$$

When working with a sphere, such as the Earth, this formula in Eqs. (1)-(3) assumes a perfect, spherical shape. In geographical and navigational calculations, it is utilized to determine the great circle distance between two places. The program can calculate the distance precisely by applying the Haversine formula, which also allows for accurate computations of other flight factors such as data transmission speed. In order to maximize flight paths, forecast transmission durations at particular distances, and guarantee effective communication between the aircraft and ground stations, this distance data is crucial.

Due to on-board aircraft computing restrictions, the edge processing functionality in this simulation is limited to relatively simple analytics algorithms. Redundancy mechanisms are incorporated in the system architecture to maximize reliability despite these limitations.

The proposed approach's MATLAB-based implementation is divided into a number of discrete functions, each of which is responsible for carrying out particular calculations and analyses, which are summarized in Figs. 1 and 2, respectively, for the preliminary and extended study:

The proposed approach's execution depends on a strong technical foundation in the effort to improve air traffic communication and navigation efficiency. Here, learn more about the crucial technical factors that serve as the cornerstone for the system's operation.

After reviewing the Code Structures, Key Algorithms, and Technical Aspects of the Implementation, it is necessary to have a better understanding of Data Flow and processing, which can be summed up as follows:

• User Input and Initialization: The implementation starts with inputs from the user, which include bearing, altitude updates, departure and arrival cities, and elevations. The starting parameters for the flight simulation are set by these inputs.

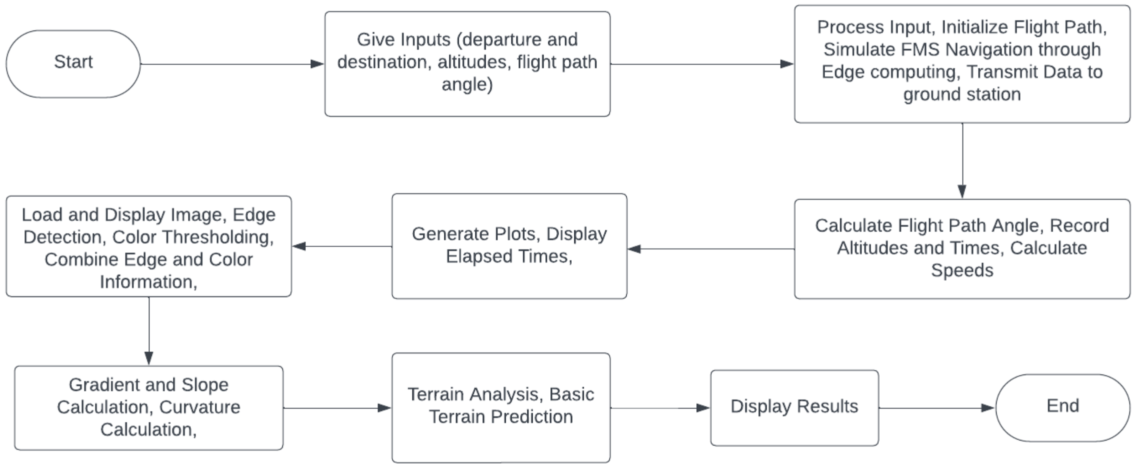

Fig. 2 Flow chart for main code (Communication and terrain analysis using edge computing)

• Data Transmission Speed Calculation: The system dynamically determines the best possible communication between the aircraft and ground stations at different locations along the flight path by calculating data transmission speeds based on user-defined criteria.
• Image Processing and Terrain Analysis: The user is prompted to choose an image via the terrain analysis tool. A number of processing operations are applied to the selected image, such as edge detection, color thresholding, gradient analysis, and curvature computation. This procedure results in the classification of the terrain type (hilly or plain).
• Optimization of the Flight Path: To identify any veering off course, the flight path angles are assessed. Corrective action is advised in the event of deviations to preserve the intended flight path.

The code's modular architecture makes sure that data and processing flow clearly, facilitating effective execution and simple integration of new features. This project shows how to improve air traffic communication and navigation efficiency by fusing edge computing with image processing techniques.

4. Simulation observations and result discussion

This section provides a detailed analysis of the findings and presents the outcomes of the applied approach. It contains graphical depictions of the data as well as a comparison of the outcomes of the simulations.

4.1 Preliminary work (Comparison of edge computing to conventional cloud based computing)

The first study compares the effectiveness of edge computing and the traditional method using a simulated Flight Management System (FMS). For entering flight data such starting altitude, longitude, latitude, and altitude, the FMS has an interface.

Ten-way stations for an aircraft traveling from Delhi Airport to Chennai Airport are taken into

Table 1 Waypoints used for the case study

Waypoint 1	Waypoint 2	Waypoint 3	Waypoint 4	Waypoint 5
DELHI (Delhi Airport) Latitude: 28.556162° N Longitude: 77.100647° E Altitude: 15000 ft.	SOLOT Latitude: 25.2680° N Longitude: 75.5969° E Altitude: 20000 ft.	BIKOP Latitude: 23.3395° N Longitude: 75.0432° E Altitude: 25000 ft.	AGELA Latitude: 21.3492° N Longitude: 77.8297° E Altitude: 30000 ft.	MUBAN Latitude: 19.5791° N Longitude: 79.7547° E Altitude: 35000 ft.
Waypoint 6	**Waypoint 7**	**Waypoint 8**	**Waypoint 9**	**Waypoint 10**
DOXOR Latitude: 17.6775° N Longitude: 81.6153° E Altitude: 30000 ft.	VURUG Latitude: 16.4763° N Longitude: 82.7792° E Altitude: 25000 ft.	KADUK Latitude: 14.4987° N Longitude: 83.8981° E Altitude: 20000 ft.	KATIB Latitude: 13.5295° N Longitude: 80.5017° E Altitude: 15000 ft.	CHENNAI (Chennai Airport) Latitude: 12.9900° N Longitude: 80.1692° E

Table 2 Time taken by the two systems (in milliseconds)

Conventional system	Edge computing system
107.53	9.3573
53.412	10.869
49.578	8.6853

consideration here. As shown in Table 1 below, the waypoints are chosen so that the airplane may fly straight forward depending on the latitudes and longitudes.

The time, expressed in milliseconds, for both the edge and conventional computing systems is shown in Table 2.

Conversely, different iterations of the Edge Computing System show different processing times. The system took roughly 9.3573 milliseconds for the first run, 10.869 milliseconds for the second, and 8.6853 milliseconds for the third to process the data.

These findings underscore the effectiveness of both systems and show how optimization can be achieved through iterative implementation; this is especially clear from the declining processing times that are seen for both systems over the course of several iterations.

The observed reduction in processing times across the iterations, even with the same input, can be attributed to the inherent efficiency of the algorithms employed in the code. These algorithms are designed for optimal performance and may exhibit improved efficiency as they are executed repeatedly due to various factors such as caching, just-in-time compilation, or other system-level optimizations. It’s important to note that these variations in processing times are expected and do not necessarily indicate a learning or optimization process within the system itself.

But in terms of processing speed, the edge computing method continuously beat the traditional system. At an average of 9.97 milliseconds, the edge computing system showed considerably faster processing times throughout the course of the three runs. The efficiency of the edge computing system is also seen in Fig. 3, which is provided below.

When compared to the traditional system, the edge computing approach’s constantly shorter processing times show how effective edge computing is in this particular scenario. This result is consistent with edge computing’s predicted advantages, which analyze data closer to the source—in this case, the aircraft—instead of depending on a centralized system. This close closeness speeds up decision-making by reducing the duration of data transit.

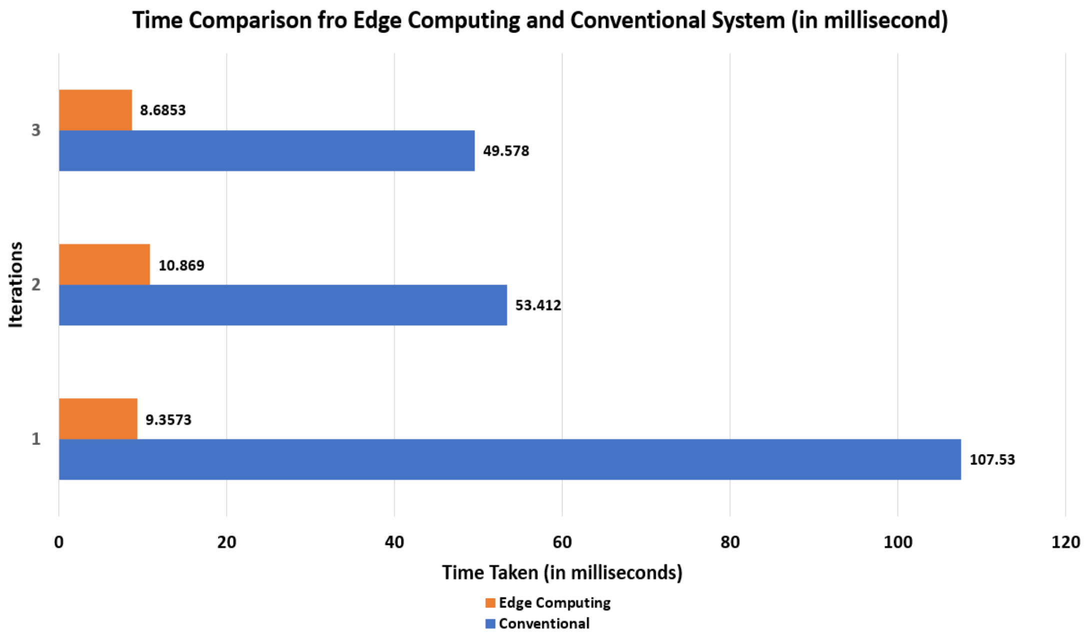

Fig. 3 Time comparison for the navigation systems used (in milliseconds)

4.2 Extended work (Evaluation of edge computing for data transmission)

According to the trend that has been seen, edge computing works well for real-time applications in flight management systems. Consequently, the edge computing strategy shows promise as a way to improve flight guidance, especially in situations when quick data processing is essential. After the initial code was run, the results demonstrated the effectiveness of edge computing technology. This effectiveness was then further optimized and utilized by running more MATLAB code, which produced informative findings that support the effectiveness of the suggested strategy. Five cities were included in the comparison analysis, which was done to provide a thorough evaluation of the system's performance: Chennai, Delhi, Bangalore, Hyderabad, and Kerala. The investigation concentrated on flight times from these cities at different elevations during transmission periods.

For additional study, the data was further tabulated. The generated charts provide a clear picture of how long it takes to transmit data at various altitudes. This comparison analysis provides insightful information about how well the suggested strategy functions in various operational and geographic contexts.

To make things easier to comprehend, this was done five times, each time giving the user the option to choose different cities for the departure and arrival. The code provided the coordinates for each city, and the user's choices determined which coordinates were used for additional calculations. Given that the aircraft is thought to be in cruise mode at the designated point, the input at 50% distance was maintained at 35,000 feet.

In the end, the findings were tallied and displayed graphically, which is seen in the Table 3, Figs. 4, and 5 below.

A thorough comparison of data transmission timings at significant waypoints throughout the

Table 3 Flight data used for the program

S. no	Flight Route	Alt at 25% of the distance	Alt at 50% of the distance	Alt at 75% of the distance	Time taken to transmit at 25% of the distance (in milliseconds)	Time taken to transmit at 50% of the distance (in milliseconds)	Time taken to transmit at 75% of the distance (in milliseconds)
1.	Delhi-Kerala	29000	35000	24000	7.5394	3.0468	2.8495
2.	Chennai- Hyderabad	31000	35000	27500	13.871	3.0683	2.6163
3.	Bangalore-Delhi	30000	35000	25000	17.227	3.7197	4.145
4.	Kerala-Hyderabad	28000	35000	31000	13.322	3.694	1.5614
5.	Bangalore-Delhi	27500	35000	20000	8.9564	3.8609	1.7156

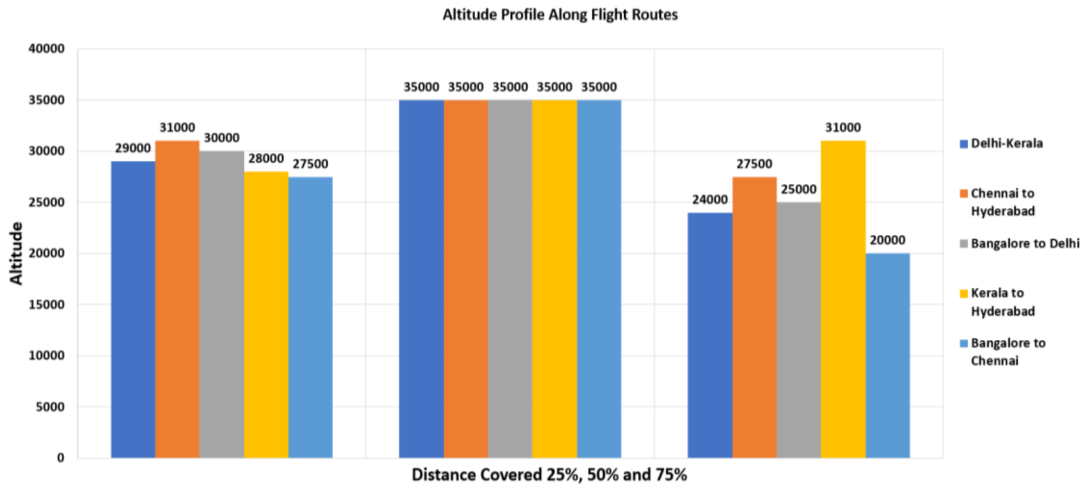

Fig. 4 Altitude profiles

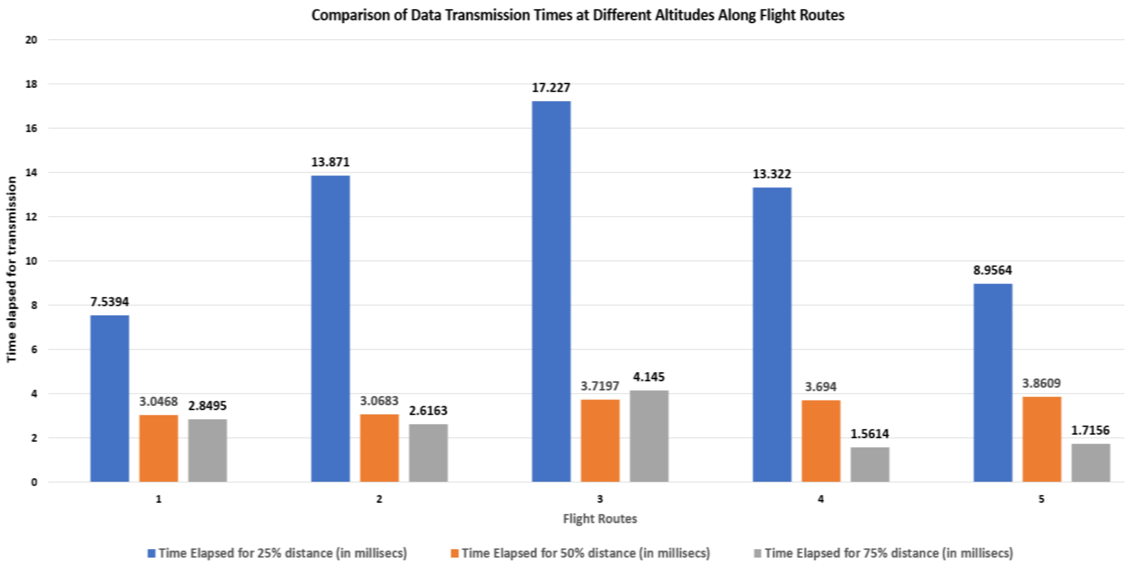

Fig. 5 Data transmission time

flight paths is provided by the Altitude vs. Distance plot in Fig. 4 and the data transmission plot in Fig. 5, which are based on the entered data. Three key lengths are shown by a set of bars for each flight route: 25%, 50%, and 75% of the total distance for the route. They can note the following details and derive some important inferences:

4.2.1 Case 1: Delhi to Kerala

For this approximately 3000 km north-to-south route, the user-defined altitude profile indicates an initial ascent from 29,000 feet at the 25% distance mark to 35,000 feet cruising altitude reaching around 50% into the journey. This is followed by a descent to 24,000 feet by the 75% mark as the aircraft approaches Kerala. The gradual altitude increase allows for efficient climb performance, while the subsequent reduction in altitude is calibrated for comfort and stability as the aircraft nears its destination. The data transmission times start higher at 7.5394 ms during the climbs phase but reduce to 2.8495 ms when nearing Kerala, reflecting the edge computing system's improving efficiency in transmitting critical flight data to core networks in real-time as the aircraft progresses on its descent.

4.2.2 Case 2: Chennai to Hyderabad

For this approximately 500 km route between Chennai and Hyderabad, the user-defined altitude profile consists of an initial climb to 31,000 feet at 25% distance, cruising at 35,000 feet around mid-journey, and a final descent to 27,500 feet by the 75% distance mark. While the simulation does not specify the underlying contextual factors driving this altitude profile selection, the edge computing enabled communication system demonstrates robust performance regardless of the trajectory variations which may be due to, Adherence to standard arrival/departure procedures at the origin and destination airports, Gradual ascent and descent rates calibrated for passenger comfort, Cruising altitude optimized for aircraft fuel efficiency, Adjustments to maintain separation from other traffic on this busy air corridor or any other factor that may be responsible as margin of error and possibilities are uncertain.

Specifically, the data transmission times remain consistently low across all phases of flight, with the fastest transmission time of 3.0683 ms recorded around the midpoint. This highlights the system's resilience and ability to maintain real-time data transfer even during altitude changes, without any degradation in speed or reliability. The consistent performance underscores the benefits of edge computing in ensuring seamless connectivity throughout an entire flight's progression.

4.2.3 Case 3: Bangalore to Delhi

For this approximately 1900 km southern route from Bangalore to Delhi, the user-defined altitude profile consists of a gradual automated climb from 30,000 feet at the 25% distance mark to 35,000 feet by the midpoint, followed by a descent to 25,000 feet around the 75% mark as the aircraft approaches Delhi. While the specific factors, like shifting air traffic and meteorological circumstances or any unforeseen circumstance, driving the selection of this altitude profile are not elaborated in the simulation details, the edge computing communication system demonstrates consistent performance in transmitting flight data despite the variations.

Specifically, the data transmission times remain reliably low throughout the different phases of the journey, with a maximum recorded time of 4.145 ms at the 75% distance point, even during the changes in altitude. This highlights the system's effectiveness in maintaining real-time connectivity and data transfer speeds without degradation, regardless of the aircraft's climbing, cruising, and descending trajectory shifts. The consistent transmission times underscore the adaptability and robustness of the edge computing architecture.

4.2.4 Case 4: Kerala to Hyderabad

For this approximately 1600 km route from Kerala to Hyderabad, the user-defined altitude profile specifies a climb from 28,000 feet at the 25% distance mark to 35,000 feet by the midpoint, holding at a cruising altitude of 35,000 feet for the majority of the journey. This is followed by a descent to 31,000 feet around the 75% mark as the aircraft approaches Hyderabad.

Significantly, despite the changes in altitude during the initial climb, cruising, and final descent phases, the edge computing-enabled data transmission system demonstrates extremely rapid speeds throughout. The fastest transmission time recorded is 1.5614 ms around the 75% distance threshold nearing the destination. This highlights the system's ability to reliably communicate critical flight data in real time between the aircraft and ground systems during all segments of the flight, including crucial take-off, cruise, and landing stages.

The rapid transmission times underscore the resilience and performance of the edge computing architecture in maintaining seamless connectivity regardless of variations in the aircraft's altitude profile.

4.2.5 Case 5: Bangalore to Chennai

For this approximately 325 km route from Bangalore to Chennai, the user-defined altitude profile consists of an initial altitude of 27,500 feet at the 25% distance mark, climbing to 35,000 feet by the midpoint, and then descending to 20,000 feet around the 75% mark as the aircraft approaches Chennai.

While the specific factors influencing this altitude selection are not elaborated in the simulation details, the edge computing communication system demonstrates consistent performance in transmitting flight data across all phases of the journey. Specifically, the transmission times remain low throughout, with the fastest time of 3.8609 ms recorded around the midpoint at 50% distance into the flight.

The consistent and efficient data transmission times highlight the robustness of the system architecture, despite variations in altitude across the take-off, cruising at midpoint, and landing descent segments. This underscores the benefits of edge computing in maintaining reliable real-time connectivity regardless of changes in flight trajectory profiles.

4.3 Terrain analysis and image aided navigation

After choosing the input image as instructed, a more detailed look at Figs. 6 and 7 illustrates the crucial functions that gradient (X) and (Y) components, slope, and curvature play in interpreting an image's topographical characteristics and enabling terrain analysis. Gradient components are used to identify boundaries and areas of significant contrast in a picture by measuring the rate of change in pixel intensity along the horizontal (X) and vertical (Y) axes. Making use of this data highlights possible borders between different terrains, providing a foundation for further research.

Derived from gradient data, the slope provides important information about the incline of the ground. Rough terrain is represented by steep slopes, and flatter terrain is represented by softer slopes. This measure is crucial for differentiating between plain and hilly terrain and for supplying information that is needed to optimize fly routes. Furthermore, by measuring the terrain's

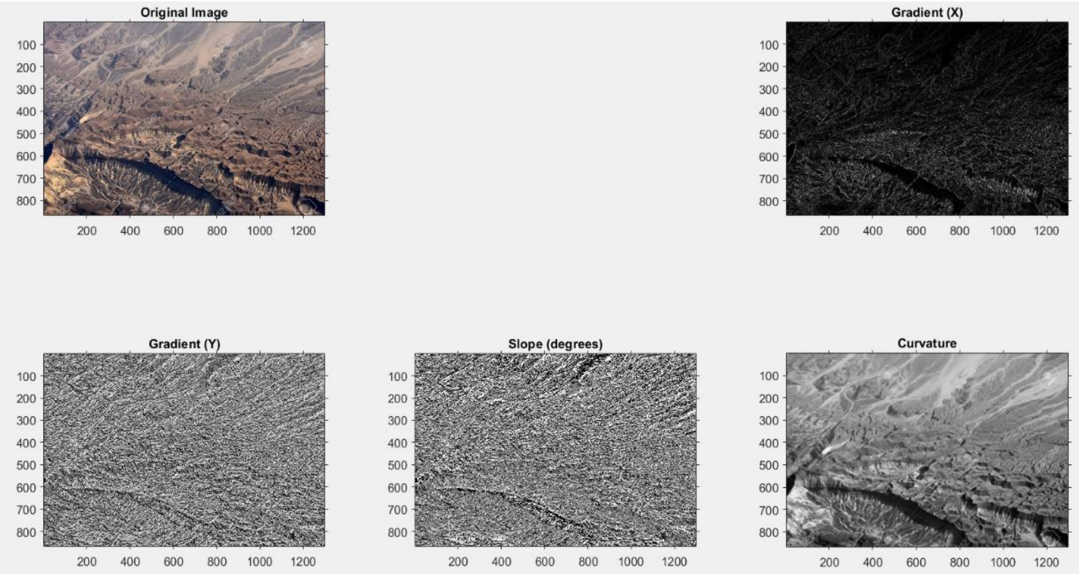

Fig. 6 Processed image to help in terrain analysis

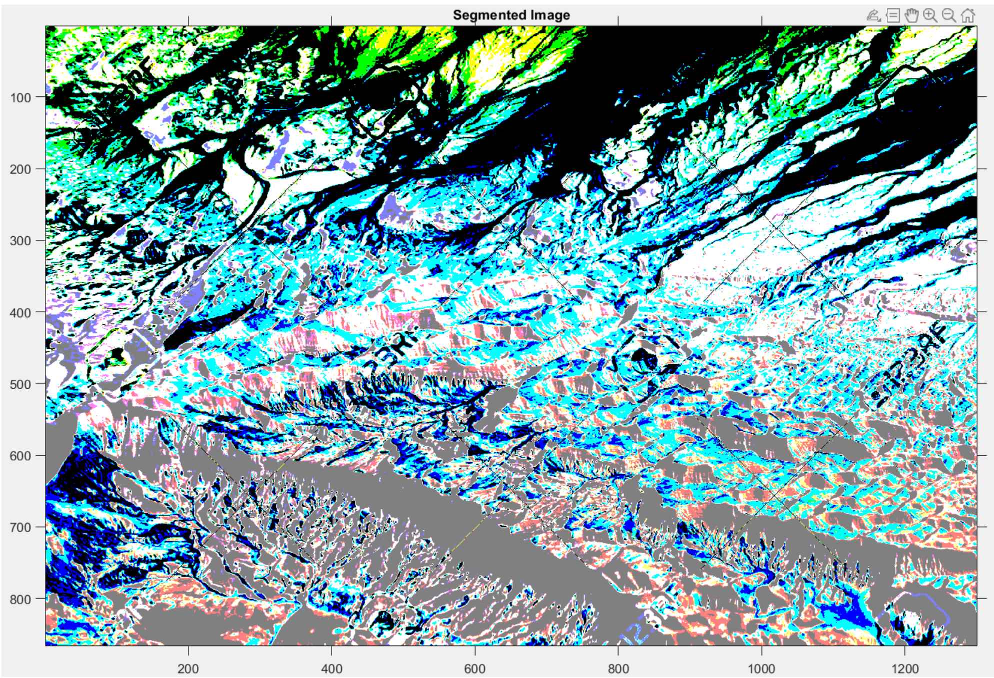

Fig. 7 Segmented image for final terrain analysis

deviation from a flat plane and detecting zones of convexity and concavity, curvature computation offers a sophisticated perspective of surface properties. This realization makes it easier to discern between plateaus and rolling hills.

Combining these measures allows the system to efficiently divide the picture and provide a finely detailed representation that distinguishes between different types of terrain. This improved image forms the basis for identifying different types of terrain, which helps pilots make well-informed decisions during fly navigation.

The code that has been implemented produces a segmented image that functions as a visual aid for the terrain study. Different hues are essential for distinguishing different landscape elements in Fig. 7. The combined use of edge detection and color thresholding algorithms produces the noticeable color differences. The demarcation of regions of interest offers a thorough understanding of the features of the landscape. The vivid hues highlight parts of the original image with notable texture diversity and contrast.

Furthermore, segmentation improves the visibility of terrain boundaries, which facilitates the detection of possible obstructions or difficult topographical features. An important result of this segmented image is that it serves as the foundation for further analysis to ascertain whether the chosen terrain is suitable for aircraft navigation.

It provides a visually clear depiction of the intricate features of the landscape, allowing decision-makers to make well-informed choices for maximizing flight paths and guaranteeing the effectiveness and safety of air traffic.

Overall, while results are promising, the simulation is scoped to initial feasibility. Large-scale validation in real-world conditions remains to be demonstrated. The results here establish the foundation for upcoming developments in aviation technology in addition to validating the efficacy of the integrated edge computing and image-aided navigation system.

5. Implication and discussion

The results of this study hold significant implications for the field of flight management systems and aviation technology. The consistent superiority of the edge computing approach in processing flight data underscores its potential for real-time applications. By minimizing processing time, edge computing enables quicker decision-making, which is paramount for ensuring the safety and efficiency of flights. This is especially crucial in scenarios where rapid adjustments to altitude are required, such as in response to adverse weather conditions or airspace congestion.

Moreover, the efficiency of edge computing can lead to enhanced fuel efficiency. The ability to promptly analyze and optimize flight altitudes reduces unnecessary fuel consumption, contributing to both cost savings for airlines and a reduction in environmental impact. This aligns with the broader industry trend towards greener aviation practices, a key consideration in an era of heightened environmental awareness.

The dynamic calculation of data transmission speeds reflects the system’s adaptability to changing flight conditions. This adaptability is essential in ensuring uninterrupted communication between aircraft and ground stations, even in scenarios where flight parameters may vary. The consistency observed in altitude profiles during cruising phases signifies a stable and efficient trajectory, which is fundamental for achieving optimal fuel efficiency and maintaining a safe cruising altitude. Additionally, the segmented terrain images provide a visual representation of the terrain’s characteristics, offering invaluable insights for flight path selection. By effectively categorizing terrain types, the system equips operators with crucial information for making informed decisions about flight routes, particularly in regions with varying topographical features.

By enhancing communication efficiency, this technology contributes to safer and more streamlined air travel operations. Moreover, the potential reduction in fuel consumption and emissions stands as a significant step towards achieving sustainability goals in the aviation sector. This has far-reaching implications for both environmental conservation and cost savings, aligning with global efforts to mitigate the environmental impact of air travel. Additionally, the system's ability to differentiate between plain and hilly terrains holds implications for emergency procedures and route planning, particularly in regions with challenging topographical conditions.

The efficiency and performance of the integrated system are the cornerstone of its impact. The real-time data transmission capabilities are pivotal in ensuring timely and reliable communication between aircraft and ground stations. This not only enhances operational efficiency but also contributes to the safety and security of air travel. The consistent altitude profiles observed along flight routes indicate well-optimized trajectories, crucial for achieving fuel efficiency and ensuring a safe cruising altitude. Furthermore, the terrain analysis algorithm's proficiency in distinguishing between plain and hilly terrains is a testament to its effectiveness in providing actionable insights for flight path planning. This capability significantly enhances the decision-making process for flight operators, particularly in regions with diverse topographies.

Wider adoption of aviation edge computing faces challenges like costs, legacy system integration, safety certifications, and institutional inertia. A hybrid model likely needs to leverage strengths of both edge and centralized clouds. Significant research is still required to develop resilient edge architectures purpose-built for aviation's demands.

In summary, the discussions presented in this chapter affirm that the integrated approach of using edge computing-based algorithms for navigation, and communication alongside image processing and terrain analysis has the potential to significantly enhance air traffic communication and navigation efficiency. The implications of this technology encompass various dimensions of aviation operations, from safety and efficiency to environmental sustainability and emergency preparedness. This research represents a significant step forward in advancing the capabilities of air traffic management systems.

6. Conclusions

In summary, this initial simulation study demonstrates the potential of edge computing to deliver faster processing times compared to a conventional centralized architecture, as evidenced by the lower latency results across the simulated test scenarios. The observed average processing time of 9.6372 milliseconds for the edge computing model versus 70.1733 milliseconds for the baseline (conventional, cloud-based) model indicates the possibility of improved performance for computation-intensive air traffic management applications. However, extrapolating these preliminary findings to imply tangible improvements in real-time decision-making or flight safety is premature. At this stage, the simulated scenarios are limited in scope and do not deeply examine factors that directly impact time-critical manoeuvres and safety outcomes. Significant further research would be required to realize this potential through aviation-specific edge computing optimizations tailored to priorities like rapid in-flight trajectory adjustments. While promising, the efficiency gains observed through simulation should not be overstated. Rather, this study serves principally to establish initial feasibility and provide motivation for more extensive design and testing of edge-based architectures for enabling the most demanding real-time air traffic capabilities envisioned.

This research demonstrates the potential of integrating edge computing and image processing to address key challenges in air traffic management. The proposed system architecture provides a solution to improve efficiency, reduce congestion, and enhance operational accuracy for flights. Specifically, by enabling accelerated on-board data processing, the edge computing element helps mitigate bottlenecks caused by transmitting all data to ground stations. Meanwhile, the image processing techniques allow for more sophisticated tracking and analysis of aircraft behaviour using visual feeds.

While the scale of this initial simulation study is limited, it establishes the feasibility of the integrated edge computing-image processing approach. With further development, this solution could generalize to wider aviation settings and provide safer and more efficient coordination of global air traffic. However, real-world deployment would require extensive testing and validation to translate the conceptual architecture into robust large-scale systems. Additionally, factors such as cost, regulatory compliance, and pilot acceptance would need to be addressed.

In summary, this research proposes a novel architecture for air traffic management and demonstrates its potential via simulation. Significant further efforts in prototyping, validation, and stakeholder alignment would be required to translate these conceptual solutions into real-world impact. Nonetheless, the study provides a valuable proof of concept for modernizing air traffic coordination.

The analysis of simulated altitude profiles and terrain data provides initial validation of the integrated edge computing-image processing architecture for air traffic management. Specifically, the image processing algorithms enable enhanced identification and classification of ground terrain features from visual feeds. Meanwhile, the edge computing element allows accelerated processing of these image’s onboard aircraft, rather than transmitting all raw data to ground stations.

While this feasibility study is limited in scope, the architecture demonstrates the potential for improved air traffic situational awareness and safety through real-time visual data analysis. However, further research would be needed to substantiate the broader claims about minimizing fuel consumption and environmental impact. Follow-on efforts could involve incorporating aircraft performance models to simulate the effects of trajectory optimization and congestion avoidance. Additionally, a cost-benefit analysis could quantify potential sustainability gains in terms of reduced fuel usage and emissions.

In summary, this initial simulation focuses on establishing the viability of the integrated edge-image processing approach, rather than comprehensively evaluating sustainability benefits. Substantiating the wider claims around minimizing environmental impact would require expanded modeling and simulation incorporating aircraft performance, emissions data, and economic factors.

7. Future work

While this study has made significant strides in advancing air traffic communication and navigation efficiency through the integration of edge computing-enabled image-aided flight navigation systems, there exist nuanced considerations and opportunities for future exploration:

- Enhanced Weather Conditions Evaluation: Conducting a more nuanced and exhaustive evaluation under various weather conditions is imperative. The impact of adverse weather scenarios, such as storms, low visibility, or turbulence, on the efficiency and accuracy of the edge computing system needs thorough investigation. This will ensure the adaptability and reliability of the system under diverse environmental challenges.
- Scalability and Complex Airspace Analysis: The scalability of edge computing solutions for

larger aircraft fleets and intricate airspace configurations should be a focus for future research. Assessing the system's performance in complex airspace environments, including high-traffic zones and areas with varied terrain, will provide insights into its robustness and applicability in real-world scenarios.

- Integration of Advanced Machine Learning Techniques: Exploring advanced machine learning techniques for terrain analysis presents a promising avenue. Integrating sophisticated algorithms capable of dynamic learning and adaptation based on real-time data inputs could further enhance the precision of flight path optimization. This includes the development of algorithms that can self-adjust based on historical data and evolving flight patterns.
- Optimizing Resource Utilization in Resource-Constrained Environments: Addressing the computational resource requirements in resource-constrained environments is crucial for the practical deployment of edge computing solutions. Future work should delve into optimizing resource utilization, exploring efficient algorithms, and considering hardware advancements that can maximize the system's performance while respecting the constraints of the operational environment.
- Comprehensive Cost-Benefit Analysis: A thorough and comprehensive cost-benefit analysis is essential for the practical implementation of edge computing infrastructure in the aviation industry. This analysis should not only consider the initial implementation costs but also account for long-term maintenance, scalability, and potential upgrades. Understanding the economic viability and return on investment will play a pivotal role in encouraging widespread adoption.
- Fine-Tuning Image Resolution and Quality: Fine-tuning the image resolution and quality is vital to ensuring accurate terrain analysis. Future research should explore the optimal balance between image resolution and computational efficiency, considering factors like network bandwidth, storage requirements, and real-time processing capabilities.
- Real-World Field Testing and Industry Collaboration: Conducting real-world field tests in collaboration with industry stakeholders is a critical step toward validating and refining the technology for practical applications. Engaging with aviation authorities, airlines, and relevant regulatory bodies will provide valuable feedback, ensure compliance with industry standards, and pave the way for seamless integration into existing air traffic management systems.

Areas of further investigation include conducting security assessments, wide-scale pilot trials, developing edge computing standards tailored to aviation, evaluating sustainability benefits, and research to bolster reliability and resilience for this mission-critical application.

Acknowledgment

I would like to place on record our sincere thanks to all those who contributed to the successful completion of my paper. I would like to express my deep sense of gratitude to the Hindustan Institute of Technology and Science, Chennai for allowing me to do this paper. I would like to express our thanks to Dean (Aeronautical & Aerospace Engineering) Dr. R. Asokan, (Head of the Department of Aerospace Engineering) Dr. Parthasarathy Vasanthkumar for inspiring and motivating us to complete this paper. I would like to thank my guide and corresponding author, Dr. S. Charulatha for continually guiding and actively participating in the paper, and giving valuable suggestions to complete the same.

References

Bekkouche, O., Taleb, T. and Bagaa, M. (2018), “UAVs traffic control based on multi-access edge computing”, *IEEE Global Communications Conference (GLOBECOM)*, 1-6. https://doi.org/10.1109/GLOCOM.2018.8647421.

Chen, Q., Zhu, H., Yang, L., Chen, X., Pollin, S. and Vinogradov, E. (2021), “Edge computing assisted autonomous flight for UAV: Synergies between vision and communications”, *IEEE Commun. Mag.*, **59**(1), 28-33. https://doi.org/10.1109/MCOM.001.2000501.

Chun, B., Oh, B., Cho, C. and Lee, D. (2018), “Design and implementation of lightweight messaging middleware for edge computing”, *Proceedings of the 6th International Conference on Control, Mechatronics and Automation*, 170-174. https://doi.org/10.1145/3284516.3284535.

Dogea, R., Yan, X.T. and Millar, R. (2023), “Implementation of an edge-fog-cloud computing IoT architecture in aircraft components”, *MRS Commun.*, **13**, 416-424. https://doi.org/10.1557/s43579-023-00364-z.

Dubey, A., Emfinger, W., Gokhale, A., Kumar, P., McDermet, D., Bapty, T. and Karsai, G. (2019), “Enabling strong isolation for distributed real-time applications in edge computing scenarios”, *IEEE Aerosp. Electron. Syst. Mag.*, **34**(7), 32-45. https://doi.org/10.1109/MAES.2019.2905921.

Fazeldehkordi, E. and Grønli, T.M. (2022), “A survey of security architectures for edge computing-based IoT”, *IoT*, **3**(3), 332-365. https://doi.org/10.3390/iot3030019.

Hayat, S., Jung, R., Hellwagner, H., Bettstetter, C., Emini, D. and Schnieders, D. (2021), “Edge computing in 5G for drone navigation: What to offload?”, *IEEE Robot. Autom. Lett.*, **6**(2), 2571-2578. https://doi.org/10.1109/LRA.2021.3062319.

Kamyar, R. and Taheri, E. (2014), “Aircraft optimal terrain/threat-based trajectory planning and control”, *J. Guid., Control Dyn.*, **37**(2). 466-483. https://doi.org/10.2514/1.61339.

Khan, A., Khattak, K.S., Khan, Z.H., Gulliver, T.A. and Abdullah. (2023), “Edge computing for effective and efficient traffic characterization”, *Sensor. (Basel, Switzerland)*, **23**(23), 9385. https://doi.org/10.3390/s23239385.

Koubâa, A., Ammar, A., Alahdab, M., Kanhouch, A. and Azar, A.T. (2020), “DeepBrain: Experimental evaluation of cloud-based computation offloading and edge computing in the internet-of-drones for deep learning applications”, *Sensor.*, **20**, 5240. https://doi.org/10.3390/s20185240.

Liu, K., Chauhan, S., Devaraj, R., Shahi, S. and Sreekumar, U. (2019), “Enabling autonomous unmanned aerial systems via edge computing”, *IEEE International Conference on Service-Oriented System Engineering (SOSE)*, 374-3745. https://doi.org/10.1109/SOSE.2019.00063.

Liu, Y., Jiang, L., Qi, Q., Xie, K. and Xie, S. (2023), “Online computation offloading for collaborative Space/Aerial-aided edge computing toward 6G system”, *IEEE Trans. Vehicular Technol.*, **73**(2), 2495-2505. http://doi.org/10.1109/TVT.2023.3312676.

Lu, P. and Pierson, B.L. (1995). “Optimal aircraft terrain-following analysis and trajectory generation”, *J. Guid., Control Dyn.*, **18**(3), 555-560. https://doi.org/10.2514/3.21422.

McEnroe, P., Wang, S. and Liyanage, M. (2022), “A survey on the convergence of edge computing and AI for UAVs: Opportunities and challenges”, *IEEE Intern. Thing. J.*, **9**(17), 15435-15459. https://doi.org/10.1109/JIOT.2022.3176400.

Messous, M.A., Hellwagner, H., Senouci, S.M., Emini, D. and Schnieders, D. (2020), “Edge computing for visual navigation and mapping in a UAV network”, *IEEE International Conference on Communications (ICC)*, 1-6. http://dx.doi.org/10.1109/ICC40277.2020.9149087.

Mijuskovic, A., Bemthuis, R., Aldea, A. and Havinga, P. (2020), “An enterprise architecture based on cloud, fog and edge computing for an airfield lighting management system”, *IEEE 24th International Enterprise Distributed Object Computing Workshop (EDOCW)*, October. https://doi.org/10.1109/EDOCW49879.2020.00021.

Naveen, S. and Kounte, M.R. (2019), “Key technologies and challenges in IoT edge computing”, *Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 61-65.

https://doi.org/10.1109/I-SMAC47947.2019.9032541.
Qi, J., Song, Q. and Feng, J. (2020), “A novel edge computing based area navigation scheme”, *CMC-Comput. Mater. Continua*, **65**(3), 2385-2396. https://doi.org/10.32604/cmc.2020.011651.
Satyanarayanan, M. (2017), “The emergence of edge computing”, *Comput.*, **50**(1), 30-39. https://doi.org/10.1109/MC.2017.9.
Shrestha, R., Oh, I. and Kim, S. (2021), “A survey on operation concept, advancements, and challenging issues of urban air traffic management”, *Front. Future Transp*., **2**, 626935. https://doi.org/10.3389/ffutr.2021.626935.
Song, Z., Bian, H. and Zielinski, A. (2016), “Application of acoustic image processing in underwater terrain aided navigation”, *Ocean Eng.*, **121**, 279-290. https://doi.org/10.1016/j.oceaneng.2016.05.039.
Subburaj, B., Jayachandran, U.M., Arumugham, V. and Suthanthira Amalraj, M.J.A. (2023), “A self-adaptive trajectory optimization algorithm using fuzzy logic for mobile edge computing system assisted by unmanned aerial vehicle”, *Drone.*, **7**, 266. https://doi.org/10.3390/drones7040266.
Venable, D.T. and Raquet, J.F. (2016), “Large scale image aided navigation”, *IEEE Trans. Aerosp. Electron. Syst.*, **52**(6), 2849-2860. https://doi.org/10.1109/TAES.2016.150603.
Vezinet, J., Escher, A.C., Guillet, A. and Macabiau, C. (2013), “State of the art of image-aided navigation techniques for aircraft approach and landing”, *Proceedings of the 2013 International Technical Meeting of The Institute of Navigation*, 473-607.
Wang, T., Fang, Q., Liu, G., Chi, M., Luo, Y. and Shen, J. (2022), “TN-GTN: fault diagnosis of aircraft wiring network over edge computing”, *J. Wireless Commun. Network.*, 62. https://doi.org/10.1186/s13638-022-02148-w.
Zheng, J., Yang, T., Liu, H., Su, T. and Wan, L. (2021), “Accurate detection and localization of unmanned aerial vehicle swarms-enabled mobile edge computing system”, *IEEE Trans. Indus. Inform.*, **17**(7), 5059-5067. https://doi.org/10.1109/TII.2020.3015730.
Zhou, F., Hu, R. Q., Li, Z. and Wang, Y. (2020), “Mobile edge computing in unmanned aerial vehicle networks”, *IEEE Wireless Commun.*, **27**(1), 140-146. https://doi.org/10.1109/MWC.001.1800594.

AP