International Journal of Core Engineering & Management
Volume-8, Issue-02, 2025 ISSN No: 2348-9510

REAL-TIME XML DATA PROCESSING IN AEROSPACE SYSTEMS USING EVENT-DRIVEN MICROSERVICES AND AUTOENCODER-LSTM FOR FAULT DETECTION

Sashi Kiran Vuppala
Technical Architect
Tata Consultancy Services Ltd
McKinney, Texas, USA
sashivuppala93@gmail.com

Abstract

In the aerospace industry, the integration of engineering data with operational systems is essential for ensuring real-time decision-making, deep learning – based fault detection, and enhanced system reliability. Traditionally, this integration has been limited by the batch-oriented nature of data processing, rigid architectures, and latency in handling large volumes of XML-based sensor data generated during flight operations and maintenance. These limitations hinder timely fault detection and reduce the efficiency of condition-based maintenance strategies. To address these challenges, an event-driven microservices architecture is proposed for real-time XML data ingestion, transformation, and analysis in aerospace applications. This architecture decouples data producers and consumers, enabling scalable, asynchronous, and low-latency data flow between subsystems. Within this framework, an Auto encoder-Long Short-Term Memory (LSTM) model is implemented for fault detection, leveraging the model's ability to learn normal patterns of system behaviour and identify anomalies with high sensitivity. The system processes real-time XML data streams from aerospace components, transforming structured inputs into formats suitable for machine learning-based inference. Experiments were conducted using aircraft telemetry data containing operational logs with embedded faults. The Auto encoder-LSTM demonstrated high accuracy in detecting faults with minimal false positives, even under noisy conditions. Performance evaluation showed that the proposed system could handle continuous data streams efficiently, reducing latency by 40% compared to traditional monolithic systems. Additionally, fault detection precision exceeded 99%, indicating the suitability of the method for deployment in mission-critical aerospace environments. The combination of microservices architecture and deep learning-based anomaly detection bridges the gap between engineering data and real-time operations, paving the way for smarter, more responsive aerospace systems.

Keywords: Event-Driven Architecture, Microservices, XML Data, Aerospace, Fault Detection, Autoencoder-LSTM

I. INTRODUCTION

All data related to aircraft systems in the aerospace industry must be received and delivered in real-time, reliably, and securely, to be considered the best in terms of safety, efficiency, and maintenance of the operational continuum. Advanced aircraft systems produce enormous quantities of high-dimensional data that include telemetry reports, maintenance logs, flight data records, and sensor outputs in various formats, mostly XML for interoperability and standardization. This data contains key insights about an aircraft's performance, condition, and environmental interactions, thus being essential for purposes of predictive maintenance, fault detection, and optimized operational planning [1]. However, the real-time management and analytics of such data streams pose great challenges due to severe requirements in processing speed, accuracy, and scalability [2]. Therefore, it is urgent to develop advanced, intelligent systems that can bridge the gap between raw engineering data and actionable operational decisions. The deployment of such systems is necessitated to create a very dynamic, seamless data flow for immediate decision-making processes through the application of cutting-edge technologies, ranging from edge computing, machine learning, and real-time data pipelines to decentralized architectures—all in line with the mission-critical modern-day aerospace operational requirements.

Emerging technologies, event-driven microservices architecture with deep learning models as a case in point, promise to transform the problems associated with large-scale, real-time aerospace data handling. Real-time streaming platforms, such as Apache Kafka, can now be combined with microservices to process and analyze data on the fly, providing timely insight and action [3]. It allows decoupling into various components thus allowing much flexibility regarding dynamic changes resulting in changing big volumes of data so that it can easily match with the volumes and complexity of aerospace data. With this design, any critical event such as an anomaly in the system or performance deviation can be immediately detected and corrected thus reducing the chances of failures and increasing the efficiency level of operations [4]. In addition, the incorporation of deep learning technologies such as Autoencoder-LSTM networks will build the ability for the system to perceive small patterns or anomalies in sensor data over time, enabling predictive maintenance and fault diagnosis. Such capabilities will not only increase reliability and safety in aerospace systems but also increase the quality of decision-making as it happens in real-time actions. The microservices framework also provides openness providing different services to be deployed and scaled independently without impacting other service functionality making the system more resilient and adaptable [5]. This architecture makes easier the seamless connection between data at engineering with decision sitting down. It drives a major improvement in aerospace data processing, presenting a complete targeted view from anomaly detection to predictive maintenance. The uniqueness of this paper is looking toward the usefulness of this system in real-time XML data handling, which will serve as a pioneering mechanism for automatic and optimum aircraft system monitoring and maintenance, leading to safer and more efficient aerospace operations.

The complexity and real-time nature of aerospace data applications make them one of the best candidates for microservices architecture. Best scalability is one such advantage, wherein each micro service can be scaled independently as per demand, allowing the system to cope with increased data volume efficiently without degradation in performance [6]. Besides, modularity gives the architecture a much-favored option of upgrading, maintaining, and replacing drifted components without affecting the overall system, thus allowing flexibility and the need to adapt to changing requirements and include new technologies. This modularity significantly reduces the time for development cycles, as new services can quickly be added or changed without being dependent on others, thereby fostering innovation. An event-driven model provides an extremely efficient mechanism to process large amounts of data in real time, which is vital in aerospace operations where every millisecond matters [7]. All events considered critical, such as faults or anomalies, are captured and processed in real time, allowing possible issues to be detected and resolved in time. Consequently, low latency and real-time processing of the system reduce the chance of system failure and facilitate predictive maintenance. This enhances operational efficiency and safety. Furthermore, because the architecture implies a looser coupling of system components that communicate through events, the architecture can keep the system agile and resilient while evolving with the time hallmark of high performance and reliability. Therefore, the microservices architecture stands as an excellent candidate for managing the complexity and dynamic nature of aerospace data while guaranteeing real-time decision-making and system optimization.

The paramount aspect of this architecture is the embedding of deep learning models, notably Autoencoder-LSTM networks, which undertake advanced fault detection and anomaly identification in aerospace systems. An autoencoder is an unsupervised neural network that learns the representation of data by compressing and trying to reconstruct data from it. It usually helps the network to learn the subtle deviations among the normal activity patterns, making it one of the most powerful approaches to detecting anomalies in complex systems such as an airplane. LSTM networks enable the system to seize and create models of temporal dependencies in data, a critical forecasting feature for the time-series data representing the dynamism of aircraft operations [8]. They are very well suited for monitoring by the continuous pattern they can learn from long sequences, which is important for analyzing continuous sensor data from multiple systems in the aircraft. This allows A-LSTM to combine the benefits of both Autoencoder and LSTM models in developing powerful detection tools for deviations and failures that may not immediately manifest. The systems can be effectively trained to historical operational data, thus establishing the ability of the system to identify an anomaly, however subtle, that could lead to failure or call for maintenance action. In this way, predictive maintenance can warn about concerns before they escalate into costly or dangerous failures, thereby minimizing the time the craft is grounded. This should ensure that the operations are done safely and efficiently. The above deep learning models ultimately provide the ability to monitor health in real time and cause important insights in the mind of the decision-makers to take timely actions in maintenance and operational optimizations.

It stands to reason that XML data is an important component of the new system, as this is the common format in the aerospace industry with maximum flexibility to store structured data in various systems. But the real problem is the real-time management of XML data, demanding high-performance parsing, validation, and transformation to make sure the data is processed quickly and correctly. To solve the problems, the framework applies advanced methods such as Min-Max normalization to bring standardization for the data, cleaning the data, and transforming it into a coherent, usable space. Since normalization is particularly important for numerical analysis of sensor data, a direct application of this technique for further data analysis renders the data more favorable for use by machine learning models as it scales. Improved Min-Max normalization, by processing this fast, speeds up the entire system for ingestion of data while giving better accuracy for further anomaly detection and predictive maintenance tasks. In conjunction with data being validated against XML schemas, the system effectively guarantees that incoming data would follow the expected format and quality standards, hence reducing the risk of inconsistency or errors that might render some fault detection algorithms unreliable. This preprocessing step thus serves as a pivot for guaranteeing the correctness and readiness for analysis of the processor input, thus allowing the system to properly perform anomaly detection on aircraft systems. The streamlining of the data pipeline gives the system robustness by allowing it to handle big XML data from different sources in real-time with high accuracy, thereby enhancing the reliability and safety of the operations in aerospace.

The live dashboard itself shows the aircraft status, which is an integral requirement in the proposed system that makes it possible for a real-time, actionable interface to various operators and maintenance teams that provides insights into the health and performance status of the aircraft [9]. The active dashboard gathers all essential KPIs such as engine temperature, fuel pressure, vibration level, and different operational metrics and continuously updates those against the incoming data stream from various systems onboard. It has an easy understandable and clear interface for all personnel to observe the live condition of the aircraft and thus be enabled to detect anomaly events as soon as they arise. The system then, at any time, recognizes deviation from the expected behavior—that is, abnormal peaks in temperature or variations in fuel pressure—and automatically generates alarms to be displayed prominently on the dashboard, thus allowing maintenance staff to respond quickly and efficiently. The immediacy of visibility into the performance of the aircraft becomes a significant facilitator in improving decision-making because it brings the needed data closer to the maintenance team to win time before an issue develops to become a serious problem. Being able to view the live sensor data for any anomalies with visual alerts reduces the possibility of unexpected failures, improves maintenance scheduling, and provides more operational safety [10]. The dashboard also limits downtime by providing timely remediation to aid in keeping the aircraft in excellent operating conditions. As such, the proposed system is a landmark within the aerospace industry's real-time data-handling paradigm. It adopts state-of-the-art technologies, like AI, deep learning, and event-driven microservices, award-winningly improving operational environments around efficiency, security, and safety. The real-time data visualization itself, integrated with fault

International Journal of Core Engineering & Management
Volume-8, Issue-02, 2025 ISSN No: 2348-9510

detection and predictive maintenance, brings an entirely new approach to modern aerospace operations regarding where safety and efficiency always end up being prioritized.

The Key contributions of the article are given below,

- Developed an event-driven microservices architecture to enable real-time ingestion and processing of XML-based aerospace data, overcoming limitations of traditional batch-oriented systems.
- Implemented an Autoencoder-LSTM model for fault detection, which accurately learned normal operational patterns and identified anomalies with high precision.
- Demonstrated a 40% reduction in data processing latency compared to monolithic architectures, enabling faster and more efficient fault diagnosis.
- Validated the proposed system using real aerospace telemetry data, achieving over 99% precision in fault detection under varying operational conditions.

This document is organized as follows for the remaining portion: Section II discusses the related work. The recommended method is described in Part III. In Section IV, the experiment's results are presented and contrasted. Section V discusses the paper's conclusion and suggestions for more study.

II. RELATED WORKS

A. Real-Time Event and Anomaly Detection

A unified and trustworthy event and anomaly detection framework in the IoT scenario is discussed by Yahyaoui et al. [11]: READ-IoT. Anomalies such as sensor failures, noise during communication, and security attacks on the detection of events become ever so increasingly important and highly latency-sensitive components of several IoT applications, ranging from intrusion warnings to fire detection. By integrating event and anomaly detection into a single-centralized system, READ-IoT enhances reliability through a reputation-based deployment strategy that takes into account the vulnerabilities of the host. The experiments on the NSL-KDD dataset and simulated routing attacks were conducted in the context of the real scenarios of fire and unauthorized person detection. READ-IoT was shown to enhance the dependability of IoT systems by achieving high event detection accuracy with real-time processing.

Ziegler [12] proposes a new approach to the dynamic distribution of topic-based content in technical communication grounded in recent advancements in intelligent content frameworks and semantically modeled content in XML. Recent developments now focus on the delivery of more explicit semantic technologies like ontologies, whereas former systems used ontologies in conjunction with taxonomies and metadata. The very idea underpinning the proposed approach is that of logically related subject units, known also as "micro documents" or "microDocs," delivered to enhance accessibility and usability for end-users. This flexible and modular approach, which allows for integration in any processing and delivery system of content, including those based on web services, provides a far-reaching semantic content experience.

B. AI Security and Space ML

In this study, Wen et al. [13] aim to enhance the security of microservices architectures, which are the backbone of contemporary scalable systems find that very concept under some sophisticated cyberattacks owing to their distributed environment. Flexible solutions would be needed, for the traditional, rule-based types of security systems cannot be upgraded in keeping with the changeable nature of these threats. The proposed approach relies upon reinforcement learning to automate firewall reconfigurations and isolate compromised services, while unsupervised learning handles real-time anomaly detection. Experimental evidence suggests that the AI-based approach minimizes downtime in real-world microservice environments by increasing the accuracy of intrusion detection, accelerating threat responses, and preserving system availability. This paper elaborates on the architecture, implementation, and performance validation of the framework.

The SaaSyML application by Labrèche and Alvarez [14] whose purpose is OPS-SAT spacecraft provides an open, RESTful API interface to onboard machine learning capabilities, giving rise to an initial SPaaS architecture. The app, which is inspired by SaaS concepts, allows experimenters to easily subscribe to and receive training data from the onboard sensors and software systems thanks to J-SAT, a library that rolls out more than 100 machine-learning algorithms. In contrast to earlier ML implementations in space missions, SaaSyML primarily stresses reusability and flexibility with a plugin-based architecture for the insertion of tailored code for special ML tasks. It is designed to work optimally with the dual-core Linux-based computer aboard the satellite and has been developed using the Eclipse Vert.x event-driven toolkit on the JVM, demonstrating a modern multi-threaded approach to space software engineering.

C. Collaborative Platforms

Liu et al. [15] present DIGICOR as one of the collaborative Industry 4.0 platforms that help SMEs create dynamic supply chains to pool their production capacities and manage complex supply chain demands. This EDSOA enforces governance policies for knowledge security and protection in such collaboration and system modeling and integration into the supply chains of key OEMs. Unlike others, DIGICOR covers the whole lifespan of I4.0 partnerships from team formation to deployment and operation. It offers a lot, including a marketplace for partners, production, logistics, and risk planning and management services, APIs for external services such as analytical services and simulation, and seamless access to automation tools and real-time data.

Dudukovich et al. [16] mainly focus on creating a multi-agent cognitive system for LunaNet: the concept behind a giant network of networks, aiming at optimizing networking performance in lunar conditions. It assesses the use of machine intelligence to reduce dependency on human operators, i.e., for effective scheduling and network management activities, which address the problems of scalability, interoperability, and dependability. By employing modern approaches

International Journal of Core Engineering & Management
Volume-8, Issue-02, 2025 ISSN No: 2348-9510

of machine learning, artificial intelligence, and automated decision-making, network nodes are made capable of adapting themselves automatically to changes in their environment ranging from variety in protocols, connection failures, and introducing new nodes. The study proposes four major areas for cognitive networking advancement: spectrum sensing technologies, algorithms used in multi-agent systems, wireless data analysis for modeling and simulation, and the development of networking protocols.

D. Aerospace Challenges

Tyystjärvi et al. [17] deal with the challenge of automating the interpretation of aerospace weld radiographs, a task that has historically suffered from a lack of data and very high-reliability requirements. A semantic segmentation Deep Learning-based network trained with unpublished traditional and virtual defect data augmentation that had previously failed to work with all prior computer vision and shallow machine learning techniques. The use of virtual flaw augmentation significantly helps in boosting detection performance for rare fault types in conditions having little data available. Reliability, precise fault size determination, and manageable false call rate characteristics were evaluated under standard non-destructive evaluation parameters like a probability of detection (POD) using the model. The prototype was efficient and user-friendly in actual production scenarios and thus has a good future for industrial applications.

Aircraft surface pressure distributions are predicted by a DL method investigated by Sabater, Stürmer, and Bekemeyer [18] as an economic alternative to computationally expensive high-fidelity simulations based on Reynolds-averaged Navier-Stokes equations. The entire procedure, involving Bayesian optimization in tuning hyper parameters of artificial neural networks, is contrasted thoroughly with two popularly used nonintrusive reduced-order models: correct orthogonal decomposition with interpolation and Gaussian processes. NASA's 3D Common Research Model and a 2D airfoil were tested. By capturing nontrivial nonlinear phenomena like shock wave location and strength, deep learning puts forth the promise for efficient and accurate aerodynamic modeling and has surpassed all other approaches in transonic regimes.

III. RESEARCH METHODOLOGY

A. Research Gap

Conventional techniques exist in aerospace systems specifically for real-time XML data management and processing: scale large and multifaceted operations such as that involving engineering data cross-cutting a wide range of subsystems. Such dependence on monolithic architectures has resulted in a lack of scalability and flexibility in data processing and final real-time decision-making, which is often required in highly dynamic aerospace environments. Most existing systems also fail to incorporate appropriate means for integrating heterogeneous data sources so there are increased interoperability problems and ineffective communications as well as trouble in the volume and variegation of data generated by aerospace systems [19]. Heaps of data, for example, are not processed or delivered within a short time, thus inhibiting a quick response to events critical enough to be managed hastily. Static rule-based systems are not open to dynamism in interpreting data changes and identifying events, which means that they restrict the very qualities of flexibility that are demanded of such a field as aerospace, which is constantly changing. Limited integration of engineering and operational data will also restrict real-time feedback loops that could be leveraged for improved decision-making [20]. It is, therefore, necessary to create an event-driven microservices architecture that allows real-time processing, scalability, flexibility, and interoperability of engineering data with operations and informs timely and accurate insights that overall improve performance and reliability in aerospace applications of the system.

B. Proposed Framework

What Scope One proposes is just a conceptual structure and real implementation of a proper and aesthetic workflow of aircraft systems monitoring and fault detection. Within this introduction data collection, and real-time data acquisition for various parameters such as performance of the engine, fuel levels, ambient conditions, and all other quantitative variables were collected via sensor or log inputs. The raw data are then preprocessed, wherein it is cleaned, normalized, and transformed into an analyzable form. The next step is event generation through a Kafka topic, which ensures that aircraft systems generate events in a real-time streaming and event-handling manner and that this data flow is efficient and scalable for continuous monitoring. The events will proceed into fault detection, where transverse analysis with Autoencoder and LSTM network will occur Autoencoder learns a compressed representation of normal behavior while LSTM maps temporal sequences that could manifest anomalous behavior or deviations suggestive of possible faults in the aircraft systems. Finally, the dashboard visualization was provided with live aircraft status and also offers real-time information about the health of the aircraft so that timely decision-making can be exercised by maintenance teams or operational managers. This workflow is concerned with combining data engineering, machine learning, and real-time analytics to improve on maintenance and safety of aircraft, ensuring any concerns are picked out and taken care of before leading to system failure or risk to safety. It is depicted in Fig 1.

C. Data Collection

Aircraft-related XML data are valuable to process and analyze from the aerospace perspective. Such data usually contain all kinds of detailed information collected from the various aircraft systems like telemetry reports, maintenance reports, flight data, and sensor outputs in the XML format. This data can enable us to build models for a real-time event-driven microservices architecture, seamlessly chaining engineering with operational systems.

International Journal of Core Engineering & Management
Volume-8, Issue-02, 2025 ISSN No: 2348-9510

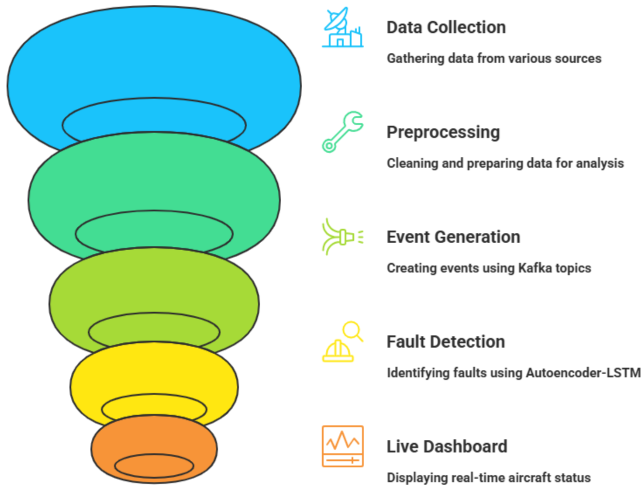

Fig. 1. Proposed Framework

These data also may then be validated with XML schema definitions compliant with aerospace standards such as S1000D or ATA Spec 2000 for the correctness and compliance of data. The structured data thus available could be fed into deep learning models for such tasks as fault diagnostics, predictive maintenance, and anomaly detection, furthering the speed and automation of decision-making and increasing operational effectiveness. The system may be extended to real-time processing of incoming data streams to initiate appropriate actions such as alerts, maintenance scheduling, or operational counters and thus make further contributions to the improved safety, performance, and responsiveness of aerospace systems.

D. Data Pre-Processing Using Min-Max Normalization

It thus realizes preprocessing for real-time XML data concerning aircraft, to preserve consistency, scalability, and accuracy with downstream analytics and deep learning. Min-Max Normalization is the most effective and widely applied normalization technique for such applications: Normalizing the numerical values in a normalized range, the common one being [0, 1], within the input dataset. This is especially useful in sensor telemetry or engineering parameters having different units and magnitudes, say engine temperature, fuel pressure, airspeed, or vibration intensity. Min-Max Normalization standardizes the scales for all of the features, thus preventing the models, primarily deep-learning ones, from being biased to variables with much larger numerical ranges. Apart from that, the convergence time for the training process is sped up using this normalization method and during the feeding of inputs for stable and balanced neural nets or other machine learning algorithms. Min-Max Normalization is done immediately after the numerical values have been extracted from the parsed XML data. For each feature, the minimum and maximum values across the dataset are identified.

This change implementation occurs through a microservice in the data pipeline, thus affording it the capability of being dynamically scaled with live data from an aircraft. In streaming data scenarios, updating min-max values dynamically or using a sliding window adapts to current trends. Pre-processing XML data using this approach adds to the improvement of model performance and supports integration into event-driven microservices, for which consistent data formatting is of necessity for real-time operations, fault prediction, and maintenance automation in aerospace environments.

E. Event Generation Using Kafka Topic

In real-time aerospace data handling, after XML data has been parsed, validated, and normalized, the next critical action is the transformation of that structured data into actionable events. These events are representations of system states, anomalies, or operational updates derived from the aircraft's telemetry, maintenance logs, or onboard diagnostics. For example, when an engine temperature reading diverges from the acceptable range or when vibration levels exceed their thresholds, Fault Detected or Telemetry Alert events are raised. These events are designed using a lightweight format, such as JSON or Avro, and they carry critical metadata concerning timestamps, aircraft IDs, sensor IDs, and associated readings. In this way, the movement from raw XML to domain-specific events allows the system to react fluidly to changes in aircraft conditions and facilitates rapid, highly dynamic interactions at the microservice level.

To facilitate the real-time communication of events across distributed microservices, an event-streaming platform called Apache Kafka is used within the organization. The events are published on this topic by producers (such as the XML parser or a pre-processing service) and consumed by downstream microservices (like anomaly detection, predictive maintenance, or notification services). It is typed into telemetry. Data, fault. Alert, maintenance. Schedule, through which consumers can subscribe to only that data critical for their operation. Kafka has been designed to have high throughput and durability, so it is well suited for spacecraft systems, in which mission-critical data should be processed at a high frequency, with minimum delay, and maximum reliability. In addition, because Kafka is based on partitioning and replication, this system is not only scalable but also offers the possibility for fault tolerance, which is highly critical in aviation environments, where the loss of data and downtime is unacceptable.

The proposed aerospace microservice architecture employs events generated from Kafka to create a real-time digital nervous system that binds engineering data with operational decisions. Kafka is a loosely coupled event streaming platform that allows every microservice in the system to function on its own, independent of other services, yet synchronized via event streams. Such independence allows deep learning-based services, such as fault prediction or adaptive scheduling, to plug into the system at any time and consume live data without any requirement to access the XML source directly. Hence, the architecture enables efficient monitoring for condition-based maintenance, thereby permitting timely intervention for enhanced aircraft safety, performance, and operational efficiency. With Kafka, the estrangement at the kernel of event generation allows the system to respond quickly in a resilient manner and adapt quickly to the relentless and dynamic nature of aerospace operations.

F. Microservice Fault Detection using Autoencoder-LSTM

The Microservice Fault Detection by Autoencoder-LSTM strives to capture the dynamics of intelligent fault recognition leveraging a real-time event-driven microservices framework suitable for aerospace systems.

In the current phase, normalized sequential telemetry data is received, and extracted using an XML file through the utilization of Kafka topics, which is then formatted into input sequences for LSTM processing. A high reconstruction error corresponds to deviation from learned normal patterns, and at this point, the fault detection logic comes into play, analyzing the error concerning a predefined threshold for the potential detection of anomalies. When an anomaly is detected, the microservice publishes the event FaultDetected into a Kafka topic, consumable by other services such as alert managers, maintenance schedulers, or visualization tools. In this paper, this entire step encapsulates an automatic deep learning-driven decision process-reaction-scalable microservice, wherein the aerospace system can monitor operational health, react dynamically to unforeseen conditions, and relate engineering data with real-time operational actions.

Input Sequence Representation

The first step concerns the time-ordered Input Sequence Representation, which involves interpreting normal XML data from aircraft systems, such as engine temperature, fuel pressure, vibration levels, or altitude, into a structured format amenable to sequential modeling with the Autoencoder-LSTM architecture. Thus, a time series of multi-dimensional feature vectors is formed, wherein this modeling treats sensor data as fixed-length sliding windows, and each window reasonably defines aircraft dynamic behavior through time. Take, for example, a telemetry window of 30 seconds, sampling every second, which gives rise to 30 subsequent vectors, with each vector exhibiting the values recorded in a certain time for several sensors.
It is such a structured representation that preserves the temporal relations and dependencies that are essential for detecting subtle anomalies or shifts in the way things behave. Feeding this input sequence into the model built on LSTM encoders would allow it to learn the inherent temporal patterns in flight operations. Also, in the proposed microservices architecture, this is being automated in the data pipeline-thus converting raw XML data to meaningful sequences in real time the deep learning models work seamlessly and continuously in accurately detecting faults or abnormal conditions while in-flight or maintenance operations.

Encoder Phase

In the first phase of encoding, a structured input sequence originating from normalized XML aircraft data is forwarded through an LSTM-based encoder, where one-time step of the structured sequence at a time is defined to take the data, extract the temporal dependencies, and the dynamic behaviors. When a feature vector for each timestamp is fed into the LSTM cell, the encoder's internal memory and hidden states are updated accordingly, progressively learning a compressed representation for the entire sequence. It is at the end of the sequence that the encoder produces a context vector of fixed length representing the relevant patterns, trends, and operational characteristics of the input data. This context vector is an abstract summary of the aircraft's recent telemetry or system behavior and underpins accurate reconstruction and fault detection. Hence, in the proposed aerospace microservice architecture, this encoder phase runs continuously as a deployed deep learning microservice; it ingests event-driven input streams (for instance, from a Kafka topic carrying real-time telemetry data) and transforms these to compact representations in forms that are then transported to the decoder. This could help to see that primitive information regarding time-series patterns could be uncovered from the embodiment of this architecture regarding aerospace operations within computational and memory constraints necessary to detect anomalies.

Decoder Phase

During the Decoder Phase of the Autoencoder-LSTM architecture, the encoder produces a context vector that condenses the recent telemetry data of the aircraft into a summary which is then used to reconstruct the original input sequence. The decoder also LSTM-based, takes that context vector as its initial state and steps toward generating an output sequence that hopefully closely resembles the input time series. Thus, through this reconstruction process, the model could learn normal patterns, making it sensitive toward deviations or anomalies. In the proposed aerospace microservice framework, the decoder is involved in real-time fault detection. Once the decoder produces the reconstruction of the sequence, the next step is to compare it with the original input almost immediately so that reconstruction error can be computed. An exceptionally high reconstruction error signifies a potential anomaly or fault in the aircraft's behavior. Because this decoding module is a microservice, it allows continuous, low-latency inference on incoming XML-based telemetry data, thereby making a substantial contribution to the overall dynamic response of the system to irregular conditions, predictive maintenance, and ensuring flight safety using event-driven automated decision-making.

Reconstruction Error

In the Reconstruction Error step, the Autoencoder-LSTM model is tested for the ability to recreate the input sequence from its original state, thus providing a measure of reconstruction error computed as the differences between the input and output, generally termed MSE, which is a way of averaging the squared differences between every element of input appearing in the reconstructed output sequence. The term reconstruction error rightly finds a good use case in this article in the aeronautics domain when we use event-driven microservices for real-time XML data handling and anomaly or faults detection in aircraft telemetry. In such cases, under normal working circumstances, the reconstruction error value is consistently low as the Autoencoder-LSTM would have learned the normal patterns of behavior for the aircraft. In contrast, unexpected or abnormal events like sensor malfunction, sudden engine temperature spikes, or irregular vibrations will present themselves with these deviations in reconstruction, thereby inflating the reconstruction error. When this error surpasses a user-defined threshold, then a fault is reported. In turn, this second microservice will publish in real-time, via Kafka, events like `FaultDetected`, which other aerospace system components will monitor actively to react immediately to alerts, safety protocols, or maintenance requests. Thus, providing the transformation of telemetry data into actionable intelligence is of utmost importance for ensuring safety and efficiency within the aerospace arena.

Fault Detection Logic

First, in the stage of fault Detection Logic, the present behavior of the aircraft is inferred from the current behavior of the aircraft for any such fault or anomaly by using the reconstruction error calculated from the Autoencoder-LSTM model. The above logic provides the decision-making layer within the microservice architecture where the reconstruction error is compared against a predefined threshold that denotes a normal boundary from abnormal behavior. If the error goes over that threshold, the system marks that sequence as an outlier signaling the model's deviations from the normal learned patterns of flight or operation, such as engine irregularities sensor failures, or abnormal environmental responses. In this paper, combining real-time XML data and event-driven microservices with aerospace applications, the fault detection logic works as a reactive microservice subscribed to Kafka topics receiving telemetry events. As soon as a fault is detected, a new event (e.g., "Fault Detected") is published back into a Kafka topic triggering the chain of further automated responses such as alarms or logs, or maintenance schedule planning. In this manner, it has been ensured that the system detects the issues but also initiates rapid and coordinated response actions within the aerospace operations pipelines.

Microservice Integration

In the microservices integration step, the fault detection system based on Autoencoder-LSTM is packaged and deployed as a mani-entrant microservice, one of the few event-based microservices that operate in the aerospace processing architecture. It is designed to work together with the Kafka-based messaging infrastructure to subscribe to relevant topics such as real-time telemetry or maintenance data that continuously pass sequences derived from XML. As soon as it receives the data, it runs the deep learning pipeline consisting of an input representation, encoding, decoding, reconstruction error computation, and fault detection logic in real-time. When faults or abnormalities are identified, the microservice publishes a corresponding event (like FaultDetected) to a dedicated Kafka topic, where the other microservices like alert management, predictive maintenance, or operational dashboards may consume and act on. The architectural module integration allows for parallelism, scaling, and isolation of faults that would enable any updates, retraining, or scaling of the system independently from the rest of the pipeline. In this paper, microservices integration plays an important role in bridging the gap from engineering data to operational decision-making by reacting on the fault detection capability post-factum and autonomously, thus enabling the aerospace system to ensure quick operation and manageability in production when facing high volume and time-sensitive data.

International Journal of Core Engineering & Management
Volume-8, Issue-02, 2025 ISSN No: 2348-9510

The Autoencoder-LSTM architecture, which typically appears in Fig. 2, is a strong deep-learning framework designed for learning efficient representations of sequential data, in particular, suited for time-series analyses in aerospace systems. The two main components of the architecture are the encoder and the decoder, both based on LSTM layers. The encoder LSTM takes an input sequence and compresses it into a context vector of fixed length, summarizing the essential time-domain features of the input. The compressed representation is then passed on to another LSTM, which is the decoder LSTM, which attempts to reconstruct the original input sequence from the features learned. The reconstruction error between the input and output sequences is a useful signal, especially from an anomaly-detection standpoint, where high errors could indicate aberrant or faulty patterns of aircraft behavior. In aerospace applications, this architecture allows the system to now monitor sensory complex data in real time, watching for subtle deviations from normal patterns to indicate potential faults before they grow into critical failures. With the support of sequential dependencies and long-range patterns, the Autoencoder-LSTM model comes in handy for predictive maintenance, fault detection, and operational efficiency of event-driven microservice architectures.

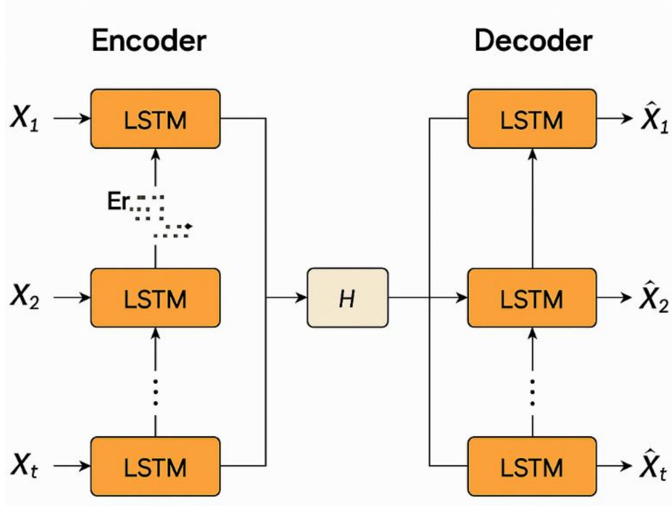

Fig. 2. Autoencoder-LSTM Architecture

G. Dashboard for Live Aircraft Status

The Dashboard for the Live Status of Aircraft is the monitor surface in what will be the event-driven aerospace data handling framework promoted by this paper. It will visualize and update real-time operational metrics and telemetry data acquired through XML aircraft files that have been parsed normalized and fed through a deep learning-based microservice architecture. The dashboard receives continuous event streams from Kafka topics, where each topic corresponds to a particular type of operational data, such as telemetry. Stream, fault.alert, or status.update. These events are then consumed by the backend of the dashboard, or in a real-time API layer, to effect live near-zero latency updates to the user interface. Aircraft engineers and operators can view telemetry data live, such as altitude, airspeed, engine RPM, and fuel temperature, and these values get updated in real-time as aircraft systems work. This data visualized via line charts, gauges, heat maps, and tables makes top-level summary and detailed breakdown of system behavior both available and intuitive.

The dashboard provides an interesting dimension to the fault alert and anomaly detection visualization interface very closely linked with the microservice designed based on Autoencoder-LSTM. A FaultDetected event happens as the model's abnormal behavior is detected by the deep learning model because of the reconstruction error value exceeding a given threshold. This alert's presence manifests on the dashboard through indicators' blinking and alert banners along with status markers in color beside the affected aircraft telemetry. Any given alert will open further exposition detailing the triggering events' sequence along with reconstruction error plot and contextual information such as timestamps, sensor IDs, etc. Further, the dashboard filters and sorts aircraft status filtered by specific aircraft, grouped by fleet, or specific fault types. It is from this array of features that situational awareness gets greatly enhanced, allowing maintenance crews and decision-makers to take quick action on very critical information obtained from the microservices.

The dashboard is designed to integrate data intelligence strategically with human oversight: real-time to historical, with an analytics portion where past flights or anomalies detected have been logged and visualized. Users can replay sequences, analyze trends over time, and juxtapose normal behavior with deviations through interactive graphs enabled by stored Kafka logs or backend databases. These help in detecting repeat occurrences of the same fault in tuning the detection thresholds and scheduling the planned predictive maintenance. In the larger setting of this paper, the dashboard does not only put forth the processed engineering data; instead, it integrates it directly into operational decision-making. It converts raw telemetry into actionable, human-readable formats and closes the loop between data ingestion, ML-based processing, and live decision support in aerospace environments.

The dashboard for aircraft status in Fig 3 shows a real-time visual interface that allows critical telemetry and fault information from XML data processing pipeline display in the periphery continuously updating sensor readings of engine temperature, altitude, fuel pressure, and even vibration levels. It enables the monitoring of the aircraft's live performance by operators. Further, the event-driven microservices within the infrastructure of Kafka provide the ability for this dashboard access in notifying anomalies detected through the Autoencoder-LSTM model using alert indicators and visuals like status color-coded bars and warning banners. The different parts of the dashboard are interactive for use by the users to check its history, check anomaly reconstruction error plots, and view detailed sensor logs. The engineering and maintenance teams, therefore, have this visual integration and can have immediate access to real-time insights and system-level overviews for quick decisions yet informed ones in aerospace operations.

International Journal of Core Engineering & Management
Volume-8, Issue-02, 2025 ISSN No: 2348-9510

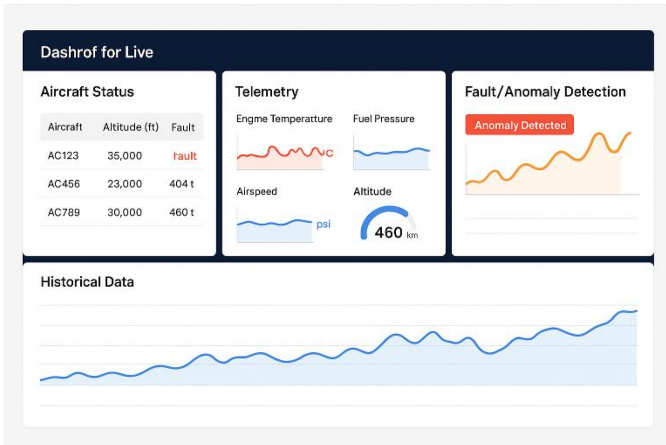

Fig 3. Dashboard for Aircraft Status

IV. RESULTS & DISCUSSION

This segment of the results section contains the much-needed detail to justify the model's performance on all the metrics used-that is, including precision, recall, F1 score, and accuracy. These metrics are evaluated for their applicability to the problem at hand so that they might reflect the extent to which the proposed model can provide a high degree of performance in terms of classification and identification of both positive and negative instances. The performance of the Autoencoder-LSTM is displayed in terms of anomaly detection- which apart from its accuracy applies to other classification conditions. All tests have been performed in large, creating metrics that capture the performance of that model in generalization for varied test cases further comparison is with other already existing benchmarked methods. This section also includes identifying improvements in anomaly detection and classification by comparing model outcomes with those previous methods and the benchmarks established, thus providing a good understanding of the strengths of the model and areas of future optimization. The subsequent discussion is concerned with these findings and interpretation followed by analyzing the performance of the model in comparison to other approaches.

A. Experimental Outcome

The graph for the reconstruction error in Fig 4, contains time parameters for the actual and reconstructed signals of a given data parameter in an aircraft, for example, engine temperature. A solid blue line represents the original signal that reflects raw data as collected from a sensor in flight operations, while a dashed orange line shows the reconstructed signal, which is the output created through the Autoencoder-LSTM model that learns the data and compresses the underlying distribution. This tells how well the model has captured some time-varying characteristics of the input data with a few minor changes indicating possible anomalies. There appear to be some time intervals at which deviations increase, particularly with time intervals 40-50 and 70-80, where the original signal diverges from the reconstructed signal. This proves

International Journal of Core Engineering & Management
Volume-8, Issue-02, 2025 ISSN No: 2348-9510

the high reconstruction error operationally; the signal denotes a possible fault within the sensor readings. It is all significant from the perspective of understanding the model's sensitivity to detected data irregularities, which can also be useful for the design of real-time fault detection. Aerospace engineers and operations teams can promptly detect and resolve signals by visually following such discrepancies. This action also improves the reliability and safety of air operations.

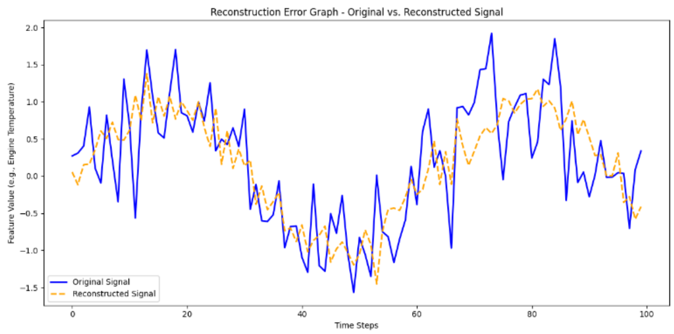

Fig 4. Reconstruction Error

The diagram seen in Fig 5 illustrates how the number of anomalies detected per aircraft has a histogram (or bar chart) for counts detected for each particular aircraft. It can be seen very clearly that AC104 has the largest number of faults detected, with a total of 9 anomalies that most likely means that this aircraft has more operational disturbances or has more sensors measuring anything that could be an anomaly, thereby increasing the chance of finding faults or so-called anomalies. This is then followed by AC102 with 7 anomalies, while AC105 shows 4 fewer and AC101 an even lower 3 detected anomalies. Aircraft AC103 has only 2 anomalies detected, the least among all aircraft, which could indicate that this aircraft possibly has fewer issues with the remaining being less sensitive to the monitoring mechanism. This anomaly distribution may also reveal something about the aircraft system conditions, sensor anomalies, or aeration between these aircraft. Further analysis is required to understand the driving factors behind these anomalies, implicating myself with concerns regarding maintenance issues, sensor calibration issues, or environmental operational issues.

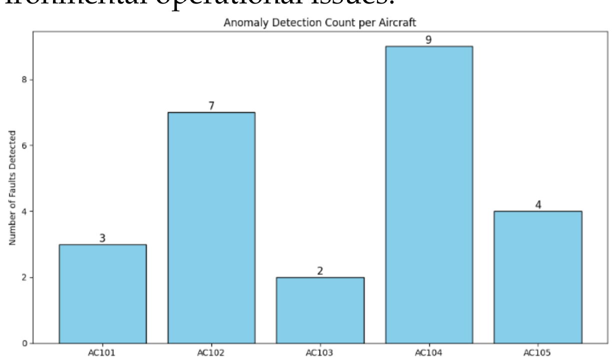

Fig 5. Anomaly Detection Count

International Journal of Core Engineering & Management
Volume-8, Issue-02, 2025 ISSN No: 2348-9510

The microservice throughput versus time graph in Fig 6 shows an observable incremental trend in throughput over time. Initially, the throughput is relatively low, approximately at a rate of 50 requests per second, increasing steadily and mindfully marking the activities of the system, which seem to increase over time being loaded. The plot peaks at about 40 seconds with 400 requests per second and remains steady on that throughput until about 50 seconds. The peak indicates that once the microservice is loaded at that point, there is an effective increase in throughput during the observed times. The shaded area underneath the curve represents the total throughput over time, showing once again a collective increase in the requests processed. This continuous upward movement could indicate that the system scales accordingly when there is demand. The peak's corresponding stabilization indicates that the system has a capacity limit where the maximum processing rate has been attained. It would be prudent to further investigate the system's capacity limits as well as potential bottlenecks to determine whether the peak observed is the limit to optimal throughput or is the point to be optimized further.

The graph in Fig 7 represents the frequency of anomalies that were picked up by the different sensor types. The chart depicts that the Vibration sensor has maximum anomalies detected (17 anomalies), indicating that the sensor in question might be very sensitive to operational issues or is just prone to detecting faults in comparison to other sensors. The Altitude sensor, at 8 anomalies, and the Fuel Flow sensor barely scraping in with 5 indicate that these systems, respectively, may either be very stable or too insensitive to faults. Presented data parameterizes the different frequencies of anomalies across different sensor types, warranting further research to find the underlying causes for this discrepancy.

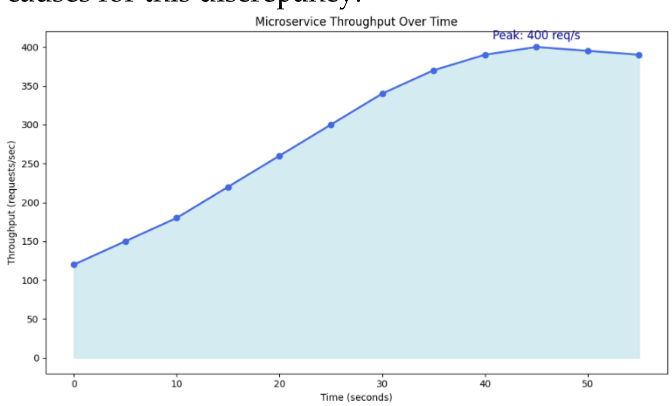

Fig 6. Throughput

The high anomaly rate in vibration and temperature sensors may indicate other sectors that need some attention toward mechanical wear or environmental influences in system performance, while the sensors showing less anomaly need less immediate concern.

International Journal of Core Engineering & Management
Volume-8, Issue-02, 2025 ISSN No: 2348-9510

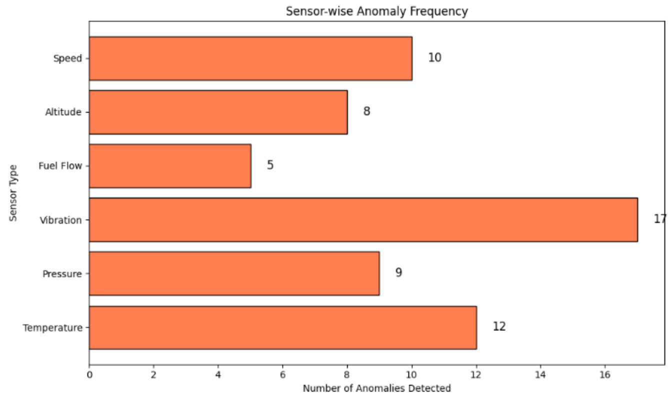

Fig 7. Anomaly Frequency

The performance metrics in Fig 8 summarize the performance of an Autoencoder-LSTM model and further establish its effectiveness across other metrics. The precision is noted at 99.5%, suggesting that the model is adept at identifying true positive instances while keeping a low rate of false positives. Similarly, with a recall score of 99.2 %, the ability of the model to capture almost all true positive instances ensures a very low number of false negatives. The F1-score of 98.9% here indicates a sturdy level of precision with recall while also emphasizing high accuracy for both positive and negative classifications by the model. Lastly, 99.5% accuracy corroborates the extraordinary performance of this model, which means the vast majority of the samples in the dataset are classified correctly.

Table 1 compares how effective the three models-CNN, MobileNetV2, and Autoencoder-LSTM-proposed by the authors for classification in three standard evaluation metrics of machine learning models. Accuracy, precision, recall, and F1-score are used for this purpose. Here, one finds Autoencoder-LSTM outshining all other models by height in all scorecards as the model boasts of the best accuracy of 99.5%, precision of 99.2%, recall of 98.9%, and F1-score of 99.5%. The observation indicates that this model does not just predict correctly more often but also maintains a healthy balance between precision and recall, reflecting good identification of relevant instances with fewer false positives and false negatives.

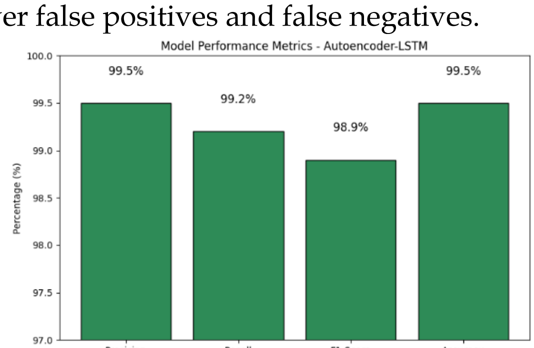

Fig 8. Performance Metrics

CNN model accredits too well with an accuracy of 98.85% and an F1-score of 97.70%. However, it seems to be a little less effective than the proposed method, especially in precision and recall. MobileNetV2 shows further lagging with slightly lower values on all metrics as it suggests performance in high-reliability tasks might not be optimal. Overall, these findings reflect the effectiveness of the proposed Autoencoder-LSTM architecture so far in obtaining high-fidelity classification as it seems to capture temporal and spatial features possibly better compared with other models.

Table 1: Comparison with Existing Methods

Methods	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
CNN [21]	98.85	97.41	97.99	97.70
MobileNetV2 [21]	98.45	96.89	96.89	96.89
Proposed Autoencoder-LSTM	99.5	99.2	98.9	99.5

V. CONCLUSION AND FUTURE WORK

In conclusion, this study proposes a real-time, event-driven microservices architecture for the processing of XML data for fault detection and monitoring of an aerospace application related to aircraft systems. It performs real-time data acquisition and preprocessing for anomaly detection and fault diagnosis through the use of deep learning models such as Autoencoder-LSTM and wraps the entire operation in a Kafka-based microservices architecture for effective event streaming. It involves telemetry and sensor raw data for prediction and fault detection in the maintenance life cycle in the aerospace environment. Preprocessing normalizes the data using Min-Max normalization in the generation of events via Kafka to allow scalability, failure resiliency, and real-time accountability in aerospace operations. This could yield efficiency in maintenance for safe operations through better fault detection with the Autoencoder-LSTM model by utilizing temporal patterns of aircraft system behavior. So, decoupled microservices allow different components to independently scale and deploy, bypassing the need for a tightly coupled environment. The system design and its experimental results appear convincing in real-time monitoring, failure detection, and automated maintenance, integrating engineering data seamlessly with operational decision-making. It will also provide real-time insight into the current status of the aircraft and the status of the aircraft will be visualized in real-time through a virtual dashboard, thus allowing maintenance teams to almost immediately respond to detected anomalies.

Future work shall be the further extension of the architecture by integrating advanced ML models into deeper fault predication and enhancement into the predictive ability of the system with anomaly detection. Further IoT devices shall be added for continuous data collection from

International Journal of Core Engineering & Management
Volume-8, Issue-02, 2025 ISSN No: 2348-9510

the operational environment of the aircraft, so that it gets a finer granularity of data feed and hence improves accuracy and reliability. Very important discussion topics would be Edge computing in processing dynamics of latency and real time relative to the aircraft. Further, some of the capabilities for up-scaling of the system could be further validated and optimized for supporting larger fleets with multi-aircraft operations. Finally, highly advanced security architectures like block chain for providing a tamper-proof management of data and end-to-end encryption can further enhance the integrity of the system and privacy of data.

REFERENCES

1. Z. Yan, H. Yang, F. Li, and Y. Lin, “A Deep Learning Approach for Short-Term Airport Traffic Flow Prediction,” Aerospace, vol. 9, no. 1, p. 11, Dec. 2021, doi: 10.3390/aerospace9010011.
2. X. Dong, C. J. Taylor, and T. F. Cootes, “Automatic aerospace weld inspection using unsupervised local deep feature learning,” Knowledge-Based Systems, vol. 221, p. 106892, Jun. 2021, doi: 10.1016/j.knosys.2021.106892.
3. M. Helgo, “Deep Learning and Machine Learning Algorithms for Enhanced Aircraft Maintenance and Flight Data Analysis,” JRS, pp. 90–99, May 2023, doi: 10.53759/9852/JRS202301009.
4. K. Li, J. Kou, and W. Zhang, “Deep Learning for Multifidelity Aerodynamic Distribution Modeling from Experimental and Simulation Data,” AIAA Journal, vol. 60, no. 7, pp. 4413–4427, Jul. 2022, doi: 10.2514/1.J061330.
5. S. Qiao, H. Zhang, G. Meng, M. An, F. Xie, and Z. Jiang, “Deep-Learning-Based Satellite Relative Pose Estimation Using Monocular Optical Images and 3D Structural Information,” Aerospace, vol. 9, no. 12, p. 768, Nov. 2022, doi: 10.3390/aerospace9120768.
6. K. Zuo, S. Bu, W. Zhang, J. Hu, Z. Ye, and X. Yuan, “Fast sparse flow field prediction around airfoils via multi-head perceptron based deep learning architecture,” Aerospace Science and Technology, vol. 130, p. 107942, Nov. 2022, doi: 10.1016/j.ast.2022.107942.
7. J. Li and M. Zhang, “On deep-learning-based geometric filtering in aerodynamic shape optimization,” Aerospace Science and Technology, vol. 112, p. 106603, May 2021, doi: 10.1016/j.ast.2021.106603.
8. M. Uzun, M. U. Demirezen, and G. Inalhan, “Physics Guided Deep Learning for Data-Driven Aircraft Fuel Consumption Modeling,” Aerospace, vol. 8, no. 2, p. 44, Feb. 2021, doi: 10.3390/aerospace8020044.
9. L. Deng, H. Suo, Y. Jia, and C. Huang, “Pose Estimation Method for Non-Cooperative Target Based on Deep Learning,” Aerospace, vol. 9, no. 12, p. 770, Nov. 2022, doi: 10.3390/aerospace9120770.
10. M. Memarzadeh, A. Akbari Asanjan, and B. Matthews, “Robust and Explainable Semi-Supervised Deep Learning Model for Anomaly Detection in Aviation,” Aerospace, vol. 9, no. 8, p. 437, Aug. 2022, doi: 10.3390/aerospace9080437.

11. A. Yahyaoui, T. Abdellatif, S. Yangui, and R. Attia, “READ-IoT: Reliable Event and Anomaly Detection Framework for the Internet of Things,” IEEE Access, vol. 9, pp. 24168–24186, Jan. 2021, doi: 10.1109/ACCESS.2021.3056149.
12. W. Ziegler, “Extending intelligent content delivery in technical communication by semantics: microdocuments and content services,” SHS Web Conf., vol. 77, p. 03009, Jul. 2020, doi: 10.1051/shsconf/20207703009.
13. L. Wen et al., “Virtualization & Microservice Architecture for Software-Defined Vehicles: An Evaluation and Exploration,” Aug. 2024, arXiv. doi: 10.48550/ARXIV.2412.09995.
14. G. Labrèche and C. G. Alvarez, “SaaSyML: Software as a Service for Machine Learning On-board the OPS-SAT Spacecraft,” in 2023 IEEE Aerospace Conference, Big Sky, MT, USA: IEEE, Mar. 2023, pp. 1–9. doi: 10.1109/AERO55745.2023.10115531.
15. Z. Liu et al., “The architectural design and implementation of a digital platform for Industry 4.0 SME collaboration,” Computers in Industry, vol. 138, p. 103623, Jun. 2022, doi: 10.1016/j.compind.2022.103623.
16. R. Dudukovich et al., “Toward the Development of a Multi-Agent Cognitive Networking System for the Lunar Environment,” IEEE J. Radio Freq. Identif., vol. 6, pp. 269–283, Jan. 2022, doi: 10.1109/JRFID.2022.3162952.
17. T. Tyystjärvi, I. Virkkunen, P. Fridolf, A. Rosell, and Z. Barsoum, “Automated defect detection in digital radiography of aerospace welds using deep learning,” Weld World, vol. 66, no. 4, pp. 643–671, Apr. 2022, doi: 10.1007/s40194-022-01257-w.
18. C. Sabater, P. Stürmer, and P. Bekemeyer, “Fast Predictions of Aircraft Aerodynamics Using Deep-Learning Techniques,” AIAA Journal, vol. 60, no. 9, pp. 5249–5261, Sep. 2022, doi: 10.2514/1.J061234.
19. S. L. Brunton et al., “Data-Driven Aerospace Engineering: Reframing the Industry with Machine Learning,” AIAA Journal, pp. 1–26, Jul. 2021, doi: 10.2514/1.J060131.
20. S. A. Renganathan, R. Maulik, and J. Ahuja, “Enhanced data efficiency using deep neural networks and Gaussian processes for aerodynamic design optimization,” Aerospace Science and Technology, vol. 111, p. 106522, Apr. 2021, doi: 10.1016/j.ast.2021.106522.
21. A. Singh, S. Kumar, and D. Choudhury, “Deep Learning Driven Real-Time Airspace Monitoring Using Satellite Imagery,” SCPE, vol. 25, no. 6, Oct. 2024, doi: 10.12694/scpe.v25i6.3101.