Journal of Civil, Construction and Environmental Engineering
2025, Vol. 10, No. 2, pp. 60-74
https://doi.org/10.11648/j.jccee.20251002.12

Review Article

A Review on Aerospace-AI, with Ethics and Implications

Derrick Mirindi[1, *], David Sinkhonde[2], Frederic Mirindi[3], Tajebe Bezabih[2]

[1]School of Architecture and Planning, Morgan State University, Baltimore, USA

[2]Department of Civil and Construction Engineering, Pan African University Institute for Basic Sciences, Technology and Innovation, Nairobi, Kenya

[3]Department of Economics, University of Manitoba, Winnipeg, Canada

Abstract

The rapid advancement of aerospace technology, coupled with the exponential growth in available data, has catalyzed the integration of artificial intelligence (AI) across the aerospace sector. This comprehensive review examines the state-of-the-art applications of AI, machine learning (ML), deep learning (DL), and generative artificial intelligence (GenAI) in aerospace. Our analysis reveals that ML algorithms demonstrate remarkable capabilities: Random forest (RF) algorithm achieves precision within 10 meters for trajectory prediction, while support vector machines (SVMs) algorithms show 99.89% accuracy in aircraft fault detection. Decision trees (DTs) algorithms excel in aircraft system diagnostics with adaptive learning capabilities. In the realm of deep learning, convolutional neural networks (CNNs) algorithms achieve 79% accuracy in satellite component detection and structural inspection, while recurrent neural networks (RNNs) algorithms and Long Short-Term Memory (LSTM) networks demonstrate superior performance in 4D trajectory prediction and engine health monitoring. GenAI, particularly through Generative adversarial networks (GANs), has revolutionized airfoil design optimization, achieving less than 1% error in profile fitting and 10% error in aerodynamic stealth characteristics. However, these algorithms face scalability challenges when processing large-scale datasets in real-time applications, particularly in mission-critical scenarios. Our research also identifies four ethical considerations, including bias prevention in automated systems, transparency in decision-making processes, privacy protection in data handling, and the implementation of important safety protocols. This study provides a foundation for understanding the current landscape of aerospace-AI integration while highlighting the importance of addressing ethical implications in future developments. The successful implementation of these technologies will require continuous innovation in validation methodologies, establish universal ethical considerations standard, and enhanced community engagement through citizen science initiatives to involve stakeholders.

Keywords

Random Forest, Decision Tree, Convolutional Neural Network, Recurrent Neural Network, Long Short-Term Memory

1. Introduction

The rapid progress of artificial intelligence (AI) in the aerospace sector has led to major changes and transformations within the industry. Indeed, this has impacted areas such as aviation, including equipment maintenance and safety sys-

*Corresponding author: demir1@morgan.edu (Derrick Mirindi)

Received: 31 December 2024; **Accepted:** 3 February 2025; **Published:** 11 March 2025

tems [1, 2]. The advent of AI is not a recent phenomenon but has its origins in the early 1950s. Researchers such as John McCarthy and Marvin Minsky initiated AI [3, 4]. Subsequently, researchers employed machine learning (ML) tools such as support vector machines (SVM) [5, 6], gradient boosting machine (GBM) [7], k-nearest neighbors (kNN) [8], random forest (RF) [9], and decision trees (DT) [10] to enhance aerospace. Additionally, researchers have developed and utilized deep learning (DL) tools, including artificial neural network (ANN) [11], convolutional neural network (CNN) [12], long short-term memory (LSTM) [13], deep reinforcement learning (DRL) [14], and recurrent neural network (RNN) [15]. Furthermore, aerospace has recently utilized generative artificial intelligence (GenAI) [16] to improve the design and optimization of aircraft components. This allows engineers to simulate various scenarios and improve performance efficiencies.

Aerospace is defined as the branch of technology and industry concerned with both aviation and space flight. This field encompasses the design, development, and production of aircraft, spacecraft, satellites, and related systems and equipment. The aerospace field began its development in the early 20th century, marked by significant milestones such as the Wright brothers' first powered flight in 1993 [17]. NASA, one of the popular industries, has been at the forefront of aerospace advancements, investing in numerous missions that have expanded the understanding of space. From the Apollo moon landings to the Mars Rover explorations, NASA's in-novations have also inspired generations to pursue careers in science, technology, engineering, and mathematics (STEM) [18, 19]. The evolution of AI has significantly influenced the sustainable development of the aerospace sector. However, this evolution raised various ethical considerations that must be addressed to ensure responsible use of technology.

Indeed, the ethical implications of integrating AI in aero-space present challenges that require rigorous examination and innovative solutions. For instance, the fundamental question of accountability in AI-driven decision-making processes has become increasingly critical, particularly in scenarios where autonomous systems must make decisions affecting human lives. The industry must address complex ethical dilemmas, including the delicate balance between human control and machine autonomy, ensuring transparency in AI decision-making processes, and establishing important protocols for life-critical situations. Also, it was observed that there is not currently any study that combines the use of AI, DL, ML, and GenAI in aerospace and the implication of their ethical considerations.

Therefore, this research aims to investigate the potential use of AI in aerospace and the implication of ethical concerns. The ML algorithms involved in this study are the decision tree (DT), the random forest (RF), the support vector machine (SVM), the k-nearest neighbor (kNN), and the gradient boosting machine (GBM). On the other hand, the DL algo-rithms included in this research are the convolutional neural network (CNN), recurrent neural network (RNN), time series analysis, transformer neural network, deep reinforcement learning (DRL), and long short-term memory (LSTM). In addition, the generative algorithms involved in this study are the generative adversarial network (GAN), deep generative model (DGM), evolutionary algorithms, physics-informed neural network (PINN), and hybrid AI. Lastly, we also investigated the most important ethical considerations, including bias and discrimination, transparency, privacy and compliance, and safety protocols.

2. Data and Methods

This review utilized a systematic methodology to identify, analyze, and synthesize the existing research on AI, ML, and DL algorithms within aerospace contexts by examining five ethical considerations. Indeed, this systematic review methodology provides a thorough and transparent approach for the selection and assessment of pertinent literature while reducing bias. The review process started with the establishment of a precise research question and the delineation of inclusion and exclusion criteria for study selection. The research question directing this review was: "What are the current algorithms employed, applications, methodologies, and ethical considerations of AI, ML, and DL techniques within the aerospace context?" The inclusion criteria consist of peer-reviewed journal articles, conference proceedings, and book chapters published in English within the past decade (2010-2024). We carried out a comprehensive search strategy to identify pertinent literature by utilizing various electronic databases such as Web of Science, Scopus, IEEE Xplore, and Google Scholar. The search query combines keywords relevant to AI, ML, DL, ethical considerations, and aerospace contexts, including "artificial intelligence," "machine learning," "deep learning," "aerospace," "safety," "bias," "discrimination," "transparency," and "ethical consideration." Moreover, we conducted both forward and backward citation analyses of additional relevant studies. We used data extraction on selected studies, which included essential information such as research objectives, data sources, methodologies, results, and ethical considerations. This methodology allowed us to identify themes, trends, and challenges within this literature. The review research showed the potential of aerospace-AI and the ethical considerations of incorporating AI into aerospace decision-making processes.

3. Artificial Intelligence

AI has emerged as a transformative force in aerospace. It revolutionizes traditional approaches to flight operations, maintenance protocols, and safety systems. The evolution of AI encompasses several distinct but interconnected domains that are reshaping aerospace capabilities. At its core, AI represents the broader field of computer systems capable of performing tasks that typically require human intelligence. In aerospace applications, AI systems demonstrate decision-making capabilities across flight control, navigation, and system monitoring, transcending simple rule-based programming to achieve adaptive and intelligent responses to complex scenarios.

ML algorithms, as a subset of AI, introduce autonomous learning capabilities where systems evolve through experience without explicit programming. In aerospace contexts, ML algorithms analyze vast datasets from aircraft sensors, flight operations, and maintenance records to identify patterns and optimize performance. These systems continuously refine their models through iterative learning. They enhance prediction accuracy and decision-making reliability [20].

DL algorithms represent a branch of ML, employing neural networks that mirror human cognitive processes. In aerospace applications for instance, DL systems process complex, multidimensional data through hierarchical layers of abstraction, enabling advanced capabilities in image recognition, sensor data interpretation, and autonomous navigation. These networks excel at handling the intricate relationships present in aerospace systems, from aerodynamic modeling to structural health monitoring [21].

Generative Artificial Intelligence (GenAI) marks the latest evolution in AI, capable of creating novel content and solutions from existing data patterns. In aerospace, GenAI applications extend beyond traditional analytical tasks to generate design alternatives, simulate flight scenarios, and create synthetic training data. This technology, particularly through Large Language Models (LLMs), enables human-machine interactions and knowledge synthesis across aerospace domains [22].

As depicted in Figure 1, this hierarchical progression from AI (1950's) to GenAI (2020's) represents not just technological advancement but a fundamental shift in how aerospace systems learn, adapt, and create solutions to complex challenges in aviation and space exploration.

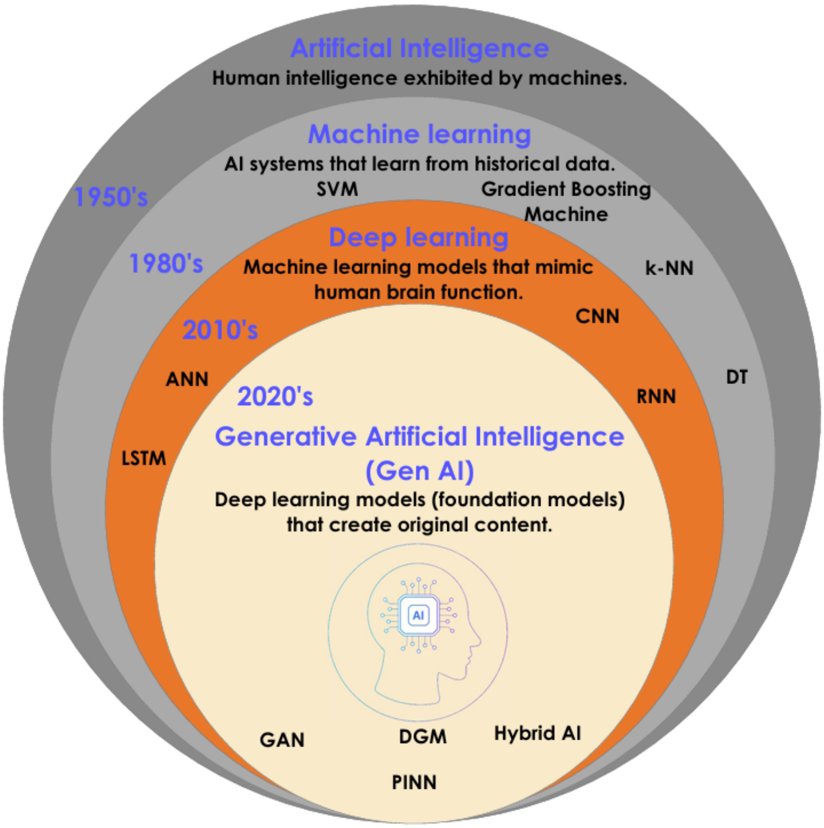

Figure 1. Schematic representation of AI, ML, DL, and GenAI in aerospace context.

3.1. Machine Learning Algorithm

3.1.1. Random Forest Random forest (RF) algorithm is a powerful unsupervised learning method that was introduced by Leo Breiman in 2001 [23, 24]. RF algorithm has become one of the most widely used ML techniques due to its versatility, robustness, and excellent predictive performance across various domains. In the field of aerospace, it can be applied for predictive maintenance of aircraft engines, trajectory prediction with errors under 10 meters (Moradi et al., 2024), fault detection in unmanned aerial vehicles (Lee et al., 2014), and magnetic navigation in GPS-denied environments (Moradi et al., 2024). [25]

RF algorithm operates on the principle of ensemble learning, specifically utilizing the bagging (Bootstrap Aggregating) technique. The algorithm creates multiple decision trees (e.g., 1000 trees for aircraft engine fault detection) using randomly sampled subsets of the training data (e.g., sensor readings from aircraft components) and features (e.g., vibration patterns, temperature measurements, pressure readings). Each tree in the forest is trained independently on a bootstrap sample drawn with replacement from the original dataset. The random feature selection process, known as feature bagging, occurs at each split of the tree, where only a subset of features is considered for splitting. For classification tasks such as aircraft component fault detection, the typical number of features considered at each split is approximately the square root of the total number of features, while for regression tasks including remaining useful life prediction of aircraft engines, it is typically one-third of the total features (Rizvi & Markani, 2023). This randomization helps to reduce correlation between individual trees, thereby improving the model's generalization capability and reducing overfitting.

The majority voting mechanism combines predictions from all trees in the ensemble. For classification problems such as aircraft system fault detection, each tree casts a vote for the predicted class, and the final prediction is determined by the majority vote [26]. For regression problems, for instance predicting aircraft engine degradation, the predictions are averaged across all trees to produce the final output.

The mathematical foundation of the RF algorithm can be expressed through several key formulas. For a given dataset D with n samples and m features, each tree T_i in the forest is trained on a bootstrap sample B_i where:

$$B_i = Bootstrap(D, n) \tag{1}$$

The final prediction for a new input x in classification is:

$$f(x) = m\{C_b(x)\}_1^B \tag{2}$$

where B represents the number of trees (typically 500-1000 for aerospace applications), $f_b(x)$ is the prediction of the b_{th} tree, and $C_b(x)$ is the class prediction of the b_{th} tree.

3.1.2. Support Vector Machine

The support vector machine (SVM) algorithm was raised as a ML algorithm, first developed by Vladimir Vapnik and his colleagues at AT&T Bell Laboratories in 1963 [27]. The algorithm has since evolved into one of the most robust and versatile classification methods in the field of aerospace. SVM can be used for aircraft fault detection achieving 99.89% accuracy [28], composite wing panel design optimization [29], aerodynamic data modeling [30], and flow separation classification in quantum computing applications [31].

Indeed, SVMs algorithms operate on the fundamental principle of finding the optimal hyperplane that maximally separates different classes in a feature space. This optimal hyperplane is determined by maximizing the margin, which is the distance between the hyperplane and the nearest data points from each class. These nearest points are called support vectors, as they literally "support" the boundary decision and are important in defining the classifier. In cases where data is not linearly separable in the original feature space (e.g., aircraft component fault detection), SVMs algorithms employ a technique known as the kernel trick, which implicitly maps the data into a higher-dimensional space where linear separation becomes possible [28]. The algorithm maintains efficiency by performing computations in the original space through kernel functions, without explicitly computing the high-dimensional transformation. Common kernel functions include linear (e.g., aircraft material classification), polynomial (e.g., aerodynamic data modeling), and radial basis function (RBF) kernels (e.g., flight control actuator fault detection), each suited for different types of classification problems [32].

Mathematically, SVM can be expressed through its optimization problem as depicted the Equation 3. For a binary classification task with training data (x_i, y_i) where $y_i \in -1, 1$, the primal optimization problem is:

$$min_{w,b} \frac{1}{2}\|\omega\|^2 + c\sum_{i=1}^{n}\xi_i \tag{3}$$

subject to $y_i(w^T x_i + b) \geq 1, \forall i$

where w represents the normal vector to the hyperplane, and b is the bias term. For non-linearly separable cases, the dual formulation incorporating kernel functions becomes:

$$max_{\alpha} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_i, x_j) \tag{4}$$

subject to $\sum_{i=1}^{n}\alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C, \forall_i$

where $K(x_i, x_j)$ is the kernel function, α_i are Lagrange multipliers, and C is the regularization parameter controlling the trade-off between margin maximization and classification error minimization [33]. The decision function for classifying new points becomes:

$$f(x) = sign(\sum_{i=1}^{n}\alpha_i y_i K(x_i, x) + b) \tag{5}$$

This mathematical equation allows SVMs to handle both linear and non-linear classification tasks efficiently while maintaining robust generalization capabilities.

3.1.3. k-Nearest Neighbors

The k-Nearest Neighbors (k-NNs) algorithms fundamen-tally rely on one critical component namely the distance metrics . The distance metrics form the backbone of the algorithm, with Euclidean distance being the most employed measure, expressed as:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (6)$$

where x and y represent feature vectors of aerospace components or flight parameters.

The alternative distance metrics include Manhattan distance: $d(x, y) = \sum_{i=1}^{n}|x_i - y_i|$ and Minkowski distance: $d(x, y) = (\sum_{i=1}^{n}|x_i - y_i|^p)^{\frac{1}{p}}$. The selection of k, representing the number of neighbors to consider, profoundly impacts the algorithm's performance. A smaller k value increases the model's sensitivity to noise, while a larger k value smooths decision boundaries but risks overgeneralization. The optimal k value is typically determined through cross-validation, considering the specific characteristics of the dataset and the desired balance between bias and variance.

In aircraft engine predictive maintenance, k-NN algorithm achieves 80% accuracy in fault detection by analyzing sensor data patterns [34]. The algorithm processes multiple parameters including vibration patterns, temperature fluctuations, and pressure readings to predict potential component failures before they occur. For instance, when monitoring turbofan engines, k-NN algorithm analyzes operational cycles to predict remaining useful life with high precision [34].

For aerospace component classification, the weighted k-NN algorithm variant employs the following prediction formula:

$$f(x) = \frac{\sum_{i=1}^{k} w_i y_i}{\sum_{i=1}^{k} w_i} \quad (7)$$

where w_i typically equals *1/di* to give more weight to closer neighbors in the feature space of aerospace parameters.

3.1.4. Gradient Boosting Machine

Gradient Boosting Machines (GBMs) algorithms emerged as a powerful ML algorithm when Jerome H. Friedman introduced it in 2001 [35, 36]. This sequential ensemble method has revolutionized predictive modeling by combining multiple weak learners into a strong predictor through an iterative process of error correction. In aerospace for example, the algorithm allows to process flight data for trajectory optimization, fuel consumption prediction, and maintenance scheduling [37].

The fundamental concept of GBMs algorithms lies in its sequential learning approach, where each subsequent model attempts to correct the errors made by previous models. For instance, in aircraft engine health monitoring, GBMs algorithms analyze sensor data sequences to predict remaining useful life with mean absolute errors under 4% [37]. Unlike random forests, which build trees in parallel, GBM algorithm constructs trees sequentially, with each new tree focusing on the residuals of the previous predictions. The process begins with a simple initial model, typically a decision tree with limited depth, which makes predictions on the training data. The algorithm then calculates the residuals, representing the differences between the actual and predicted values. These residuals become the target variables for the next weak learner. Each subsequent model in the sequence specifically aims to predict these residuals, effectively reducing the overall prediction error. The final prediction is formed by combining all these weak learners through a weighted sum, where each model's contribution is scaled by a learning rate parameter. This learning rate, also known as the shrinkage parameter, helps prevent overfitting by controlling how much each tree contributes to the final model. The sequential nature of this process ensures that each new model focuses on the aspects of the data that previous models struggled with, leading to increasingly accurate predictions.

GBM algorithm can be computed using the principle of functional gradient descent as depicted Equation 8. The algorithm aims to minimize a loss function *L(y, F(x))*, where y represents the true values and F(x) represents the model's predictions. The optimization process can be expressed as:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \quad (8)$$

where $F_m(x)$ is the model at iteration m, γ_m is the step length and h_m (x) is the base learner. The optimization process minimizes the loss function:

$$L(y, F_m(x)) = L(y, F_{m-1}(x) + \gamma_m h_m(x)) \quad (9)$$

This approach has proven particularly effective in optimizing flight operations and predicting maintenance requirements.

3.1.5. Decision Tree

Decision trees (DTs) algorithms are one of the foundational algorithms in ML, first introduced by Leo Breiman and his colleagues in 1973 through their seminal work "Classification and Regression Trees" (CART) [38]. This algorithm revolutionized the field of ML by providing an intuitive and interpretable approach to both classification and regression problems [39, 40].

As depicted in Figure 2, DT algorithm operates as a hierarchical structure that resembles an inverted tree, where decisions flow from top to bottom. The topmost node, known as the root node, represents the entire dataset, while subsequent nodes represent subsets created by applying specific decision rules. Each internal node corresponds to a decision based on a particular feature, with branches representing the possible outcomes of that decision. The terminal nodes, also called leaf nodes, contain the final predictions or classifications. The process of building a tree involves recursively partitioning the data into smaller subsets based on the most informative features. Each split aims to increase the homogeneity of the resulting subsets, making the predictions more accurate. The depth of the tree, the minimum number of samples required for a split, and the maximum number of leaf nodes are crucial hyperparameters that control the tree's complexity and prevent overfitting.

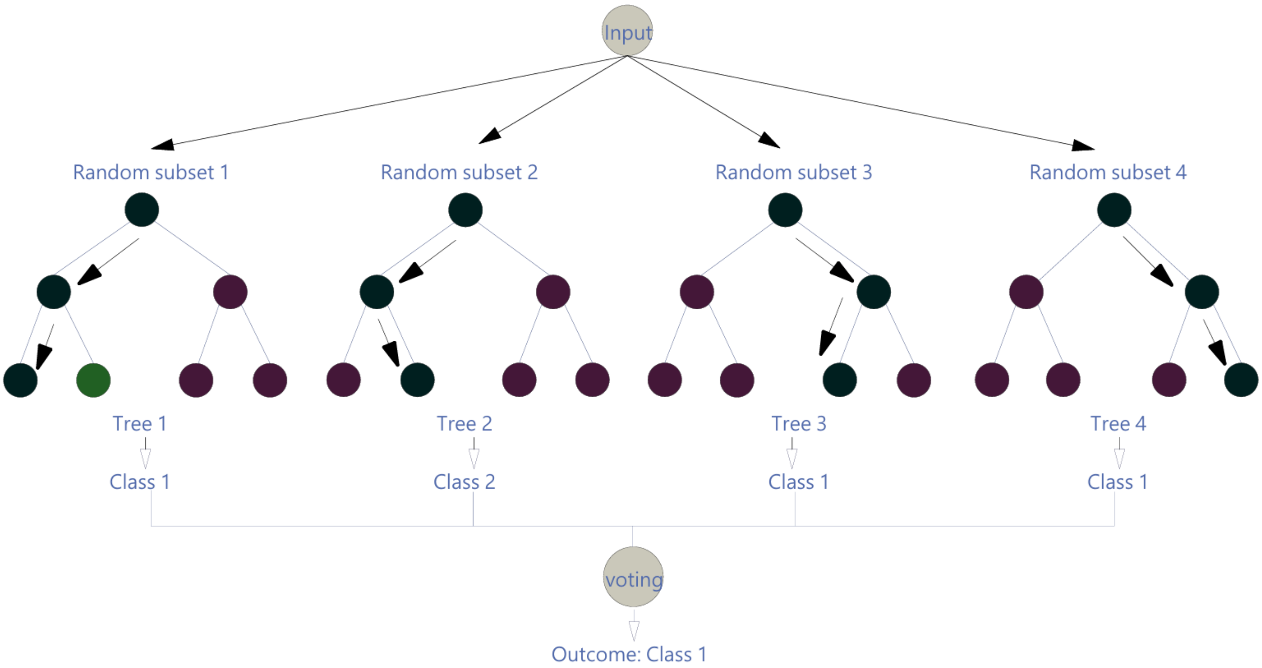

Figure 2. *Decision tree algorithm architecture.*

The mathematical foundation of DTs algorithms centers around measures of impurity and information gain. The three primary metrics used are the entropy, the Gini index, and the information gain (IG) [41]. The entropy, derived from information theory, measures the disorder or uncertainty in a dataset and is calculated as:

$$Entropy(S) = -\sum_{i=1}^{c} p_i \, log_2(p_i) \tag{10}$$

where p_i represents the probability of class *i* in dataset *S* (e.g., different fault types in aircraft systems), and *c* is the number of classes. Also, the Gini index, an alternative to entropy, measures the probability of incorrect classification and is computed as:

$$Gini(S) = 1 - \sum_i^c {p_i}^2$$

Furthermore, the Information Gain (IG) quantifies the reduction in entropy or Gini impurity after a split and is calculated as:

$$IG(S, A) = Impurity(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Impurity(S_v) \tag{11}$$

where *S* is the parent node dataset, *A* is the feature being split on, and S_v represents the subset of *S* for which feature *A* has value *v*. The algorithm selects splits that maximize information gain, ensuring optimal data separation at each node.

3.2. Deep Learning Algorithm

3.2.1. Convolutional Neural Network

Convolutional neural networks (CNNs) algorithms emerged as a development in AI, with their foundation laid by Yann LeCun and his colleagues at AT&T Bell Laboratories in 1989 [42]. The architecture was heavily inspired by biological research conducted by Hubel and Wiesel in the 1960s, who studied the cat's visual cortex and discovered that individual neuronal cells responded specifically to small regions of the visual field [43]. LeCun's pioneering work culminated in the development of LeNet-5 in 1998, which successfully demonstrated the practical application of CNNs algorithms in recognizing handwritten digits [44]. In aerospace applications, CNNs algorithms have demonstrated remarkable capabilities in satellite component detection, aircraft recognition, and structural inspection, achieving accuracy rates up to 79% in complex classification tasks [45].

The fundamental operation in CNNs algorithms is the convolution operation, which is mathematically expressed as:

$$(f * g)(t) = \int_{\infty}^{\infty} f(\tau) g(t - \tau) d\tau \tag{12}$$

In discrete form, for 2D images, this becomes:

$$(I * K)(i, j) = \sum_m \sum_n (m, n) K(i - m, j - n) \tag{13}$$

where I represent the input image and K is the kernel or filter. The stride parameter S determines the step size of the convolution operation, while padding P helps maintain spatial dimensions. The output feature map dimensions can be calculated using:

$$O = \frac{N-F+2P}{S} + 1 \tag{14}$$

where O is the output dimension, N is the input dimension, F is the filter size, P is padding, and S is stride.

In aerospace applications, CNNs algorithms have demonstrated performance in:

Satellite component detection with R-CNN algorithm architectures achieving precise identification of various spacecraft elements [46].

Real-time aircraft detection at airports with lightweight architectures reaching 13.6 fps processing speed (Li et al., 2022).

Structural fault detection in drone-like spacecraft, particularly for accelerometer and Inertial Measurement Unit (IMU) monitoring [47].

3.2.2. Recurrent Neural Networks and Time Series Analysis

Recurrent neural networks (RNNs) algorithms, first introduced by John Hopfield in 1982, represent a fundamental architecture in deep learning specifically designed to handle sequential data [48]. The core principle of RNNs lies in their ability to maintain an internal state (memory) that captures information about previous inputs in the sequence. The integration of RNNs in aerospace applications continues to evolve, with implementations now extending to autonomous navigation systems and real-time flight control optimization [49, 50]. Furthermore, the RNN architecture processes multiple sensor inputs simultaneously, including vibration patterns, temperature fluctuations, and pressure readings, to forecast potential component failures before they occur. The basic mathematical formulation of an RNN can be expressed through its hidden state equation:

$$h_t = \phi(w_{hh}h_{t-1} + w_{xh}x_t + b_h) \tag{15}$$

where h_t represents the hidden state at time step t (e.g., engine health status), $Ø$ is an activation function (typically tanh or ReLU) (e.g., tanh for normalized sensor readings), W_{hh} is the hidden-to-hidden weight matrix (e.g., historical sensor patterns), W_{xh} is the input-to-hidden weight matrix (e.g., current sensor readings), and b_h is the bias vector (it provides baseline adjustments). The output at each time step is computed as:

$$y_t = W_{hy}h_t + b_y \tag{16}$$

where W_{hy} is the hidden-to-output weight matrix and b_y is the output bias vector (Figure 3).

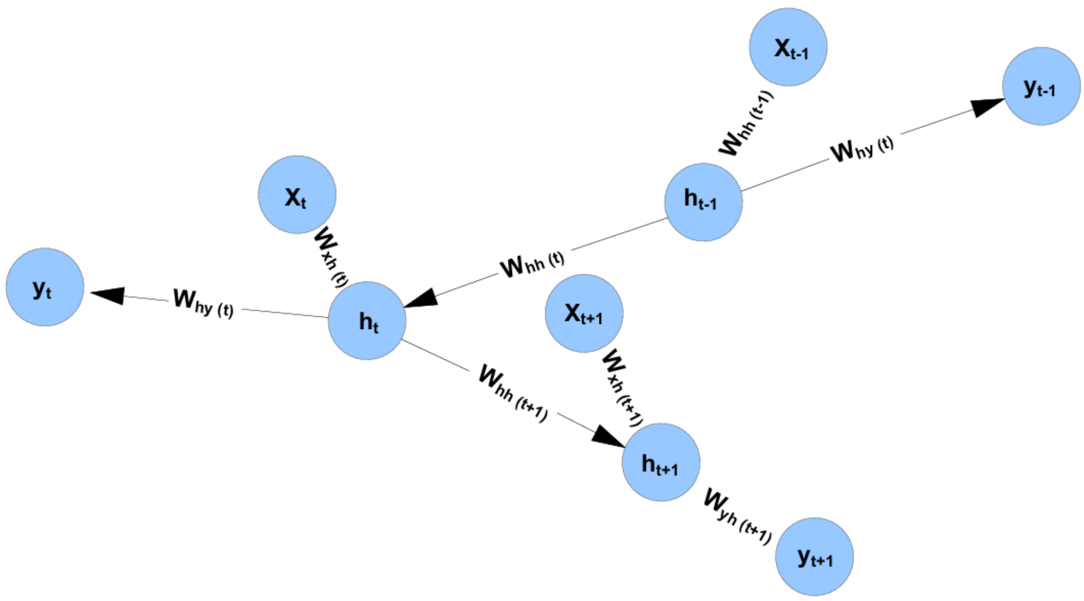

Figure 3. *RNN architecture connections description (with time steps t-1, t, and t+1, including hidden state h_t, input x_t, and output y_t, with weight matrices W_{hh}, W_{xh}, and W_{hy} labeled).*

On the other hand, the implementation of RNNs algorithms for time series analysis involves several techniques and considerations. Time series data, characterized by its sequential nature and temporal dependencies, requires a preprocessing and model architecture design [51, 52]. The input data is structured as sliding windows, where each window contains a fixed number of time steps used to predict the next value or sequence. The window size, denoted as τ, is an important parameter that determines how much historical information is considered for each prediction.

Feature engineering plays an important role in time series implementation. Common techniques include normalization using methods such as min-max scaling:

$$x_{normalized} = \frac{x - x_{min}}{x_{min} - x_{max}} \quad (17)$$

or standardization:

$$x_{s\tan dardized} = \frac{x - \mu}{\sigma} \quad (18)$$

where μ is the mean and σ is the standard deviation. Seasonal decomposition may also be applied to separate the time series into trend, seasonal, and residual components.

The architecture of RNNs algorithms for time series often incorporates multiple layers with varying numbers of hidden units. The choice of architecture depends on the complexity of the temporal patterns and the available computational resources. For multivariate time series, the input layer must accommodate multiple features, leading to modifications in the weight matrices and hidden state calculations. The output layer can be designed for different prediction tasks: single-step prediction produces one future value, while multi-step prediction generates a sequence of future values.

3.2.3. Transformer Neural Network

The transformer model architecture, introduced by Vaswani et al. in their seminal 2017 in the paper titled "Attention Is All You Need" [53], revolutionized natural language processing (NLP) and ML. The core innovation of transformers lies in their self-attention mechanism, which allows the model to weigh the importance of different parts of the input sequence dynamically. The mathematical foundation of self-attention begins with three learned matrices: Query (Q), Key (K), and Value (V). For an input sequence *X* such as flight data (e.g., position, velocity, attitude) and *W* matrices are learned parameters optimizing trajectory planning. These matrices are computed through linear transformations following these equations:

$$Q = XW_Q, K = XW_K, V = XW_V \quad (19)$$

where W_Q, W_K, and W_V are learnable parameter matrices. The attention scores are then calculated using the scaled dot-product attention formula:

$$A(Q, K, V) = soft\,max(\frac{QK^T}{\sqrt{d_k}})V \quad (20)$$

where d_k is the dimension of the key vectors, and the scaling factor $\sqrt{d_k}$ prevents the dot products from growing too large in magnitude. Multi-head attention extends this concept by allowing the model to attend to information from different representation subspaces simultaneously. Each attention head operates on different linear transformations of the input, and their outputs are concatenated and linearly transformed to produce the final output.

In air traffic management, Transformer-based trajectory prediction models show performance in demand forecasting and capacity balancing. The model processes multiple flight parameters simultaneously through multi-head attention with h heads is:

$$Multiead(Q, K, V) = Concat(head_1, \ldots, head_h)W^O \quad (21)$$

where each head is computed as:

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (22)$$

This self-attention mechanism allows transformers to capture complex dependencies between different positions in the input sequence, regardless of their distance. This is a significant advantage over recurrent architectures, which process sequences sequentially and can struggle with long-range dependencies.

The practical implementation of transformer networks involves several crucial technical components that ensure efficient and effective operation. The architecture consists of an encoder and decoder stack, each containing multiple identical layers. Each encoder layer comprises two main sublayers: a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. The feed-forward network applies two linear transformations with a ReLU activation in between:

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \quad (23)$$

A critical aspect of transformer implementation is positional encoding, which adds information about the position of tokens in the sequence. The standard implementation uses sine and cosine functions of different frequencies:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d\_model})$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d\_model})$$

where *pos* is the position and *i* is the dimension. Layer normalization is applied after each sublayer, following the formula:

$$LayerNorm(x) = \gamma \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \quad (24)$$

where *μ* and *σ* are the mean and standard deviation of the input, and γ and β are learnable parameters.

3.2.4. Deep reinforcement Learning

DRL represents a groundbreaking fusion of deep learning architectures with traditional reinforcement learning principles, first gaining prominence in 2013 with DeepMind's introduction of the Deep Q-Network (DQN) algorithm. This revolutionary approach enables agents to learn optimal behaviors through direct interaction with complex environments, utilizing deep neural networks to approximate value functions or policies.

The mathematical foundation of DRL centers on the Markov Decision Process (MDP), defined by the tuple *(S, A, P, R, γ)*, where *S* represents the state space, *A* the action space, *P* the transition probability function, *R* the reward function, and γ the discount factor. The primary objective is to maximize the expected cumulative reward, expressed using the Bellman Optimality Equation for Q-learning as:

$$Q * (s, a) = E[Rt + \gamma max_{a'} Q(s_{t+1} a') \mid s_t = s, a_t = a] \quad (25)$$

The deep Q-network (DQN) approximates this Q-function using a neural network parameterized by weights θ, minimizing the loss function:

$$L(\theta) = E[(R_t + \gamma max_{a'} Q(s_{t+1}, a'; \theta^-) - Q(s_t, a_t; \theta))^2 \quad (26)$$

where θ represents the parameters of a target network that stabilizes training. For flight control innovation, this loss function enables the fault-tolerant control achieving 99.9% reliability, the adaptation response to system failures, and the optimization fuel consumption through learned policies [54].

The practical implementation of DRL systems requires careful consideration of several critical components. The experience replay buffer, typically storing transitions (s_t, a_t, r_t, s_{t+1}), helps break temporal correlations and improves sample efficiency. Modern implementations often utilize prioritized experience replay, where transitions are sampled based on their temporal-difference error.

Furthermore, architecture selection plays an important role, with convolutional layers commonly used for visual inputs and fully connected layers for processing state representations. The epsilon-greedy exploration strategy, where θ gradually decreases over time, balances exploration and exploitation:

$$a_t = \{random\ action \quad with\ probability\ \epsilon \arg maxa\ Q(st, a; \theta) - with\ probability\ 1 - \epsilon \quad (27)$$

Advanced implementations incorporate techniques such as double DQN to reduce overestimation bias, dueling networks to separate state value and advantage estimation, and distributional RL to model the full distribution of returns. These enhancements have enabled DRL to achieve superhuman performance in various domains, from game playing to robotic control, marking a significant milestone in artificial intelligence development.

3.2.5. Long short-term Memory

LSTM algorithms, introduced by Sepp Hochreiter and Jürgen Schmidhuber in 1997, represent a significant advancement in the field of recurrent neural networks [55]. These specialized algorithms were designed to address the vanishing gradient problem that plagued traditional RNNs algorithms, enabling the capture and preservation of long-term dependencies in sequential data. LSTMs algorithms have revolutionized sequence learning tasks by maintaining a more internal memory state, allowing them to remember important information for extended periods while selectively forgetting irrelevant details.

The LSTM architecture incorporates three primary gate mechanisms: the forget gate, input gate, and output gate, working in concert with a memory cell. The forget gate determines which information should be discarded from the cell state, operating through the following mathematical formula:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (28)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (29)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (30)$$

with the final hidden state calculated as:

$$h_t = o_t * tanh(C_t) \quad (31)$$

where f_t, i_t, and o_t represent the forget, input, and output gates respectively.

For 4D trajectory prediction, attention-enhanced LSTM architectures have shown superior performance. The attention mechanism is defined as:

$$\alpha_t = soft\,max(W_\alpha\, tanh(W_h ht + W_s s_{t-1} + b_\alpha)) \quad (32)$$

where h_t represents the hidden state containing flight parameters and s_{t-1} is the previous decoder state.

3.3. Generative Algorithms

3.3.1. Generative Adversarial Network

Generative adversarial network (GAN) is a machine learning framework that pits a generator (data creator) against a discriminator (data evaluator). The generator aims to produce data so similar to real data that the discriminator cannot tell the difference (Figure 4). GANs in unsupervised machine learning are implemented by two competing neural networks. This setup captures or reproduces dataset variability and features [56]. GANs have revolutionized synthetic data generation in aerospace applications through their adversarial training architecture. The fundamental GAN equation for aerospace applications can be expressed as:

$$min_G max_D V(D, G) = E_{x \sim pdata(x)}[log D(x)] + E_{z \sim pz(z)}[log(1 - D(G(z)))] \quad (33)$$

where G represents the generator network and D the discriminator network. One of the applications of GANs is aircraft design optimization. Indeed, GANs achieve remarkable results in airfoil design, with conditional GANs (CGANs) demonstrating less than 1% error in fitting airfoil profile data and within 10% error for aerodynamic stealth characteristics [57]. For smooth air-foil generation, specialized architectures employ Bernstein polynomials:

$$y/c = (x/c)^{N1}(1 - x/c)^{N2} \cdot B_n(x/c) \qquad (34)$$

where *N1* and *N2* are airfoil shape parameters, and Bn represents the Bernstein polynomial.

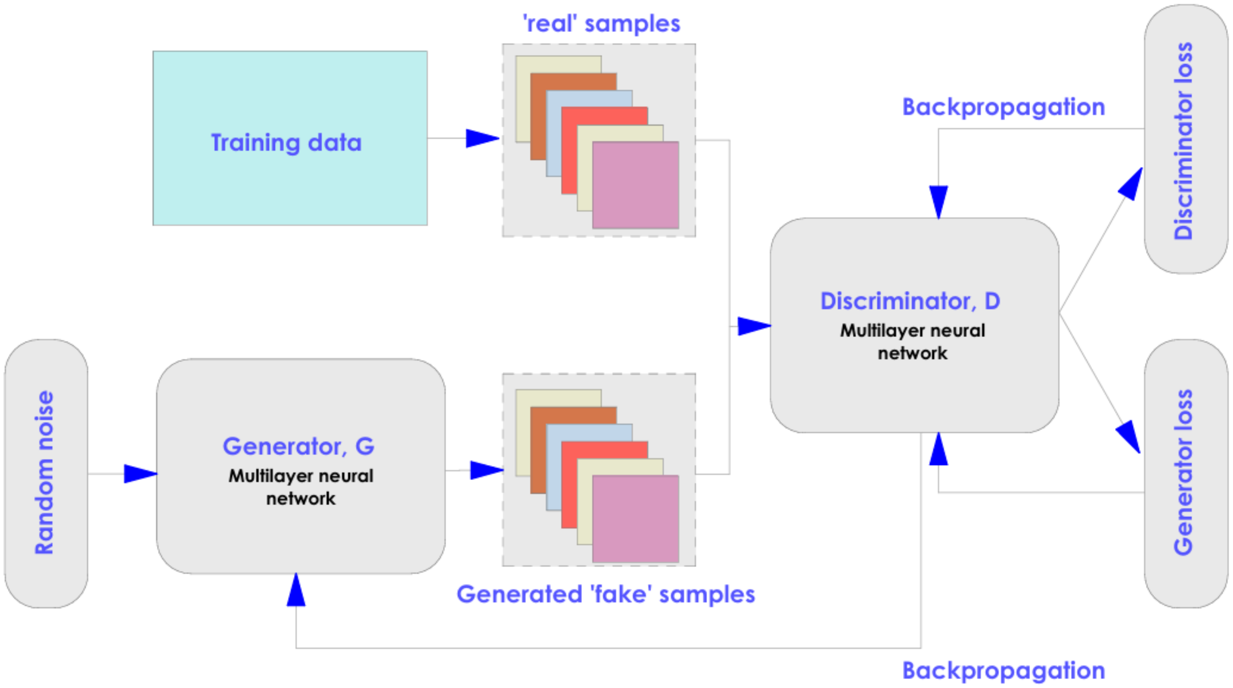

Figure 4. *Generative Adversarial Networks architecture.*

In flight path generation, GANs excel at producing synthetic 4D aircraft landing trajectories, enabling enhanced training data for air traffic management systems, for simulation of rare events and edge cases and for validation of safety protocols [58].

3.3.2. Deep generative Model

DGMs in aerospace focus on component design optimization through probabilistic modeling [59]. The core architecture employs variational autoencoders (VAEs) with the following loss function [60]:

$$\zeta = E_{q(z|x)}[log p(x \mid z)] - D_{KL}(q(z \mid x) \parallel p(z)) \qquad (35)$$

where *q(z|x)* represents the encoder network (e.g., mapping aircraft component geometries to latent space) and *p(x|z)* the decoder network (e.g., generating new aerospace designs). These models have proven particularly effective in generating aerodynamic shapes while maintaining physical constraints.

The models incorporate specific aerospace constraints through conditional probability distributions:

$$p(x \mid y) = \int p(x \mid z, y) p(z \mid y) dz \qquad (36)$$

where y represents design constraints such as lift coefficients or drag requirements, and structural integrity parameters [61].

3.3.3. Evolutionary Algorithm

Evolutionary algorithms in aerospace optimization utilize genetic operations and fitness functions specific to aerodynamic requirements [62]. The general form of the optimization problem can be expressed as:

$$min_x f(x) \text{ subject to } g_i(x) \leq 0, h(x) = 0 \qquad (37)$$

where *f(x)* represents the objective function (e.g., drag minimization), and $g_i(x)$ and $h_j(x)$ represent inequality and equality constraints (e.g., minimum lift requirements and structural limits) [63].

The covariance matrix adaptation evolution strategy (CMA-ES) has proven particularly effective:

$$x_{k+1} = x_k + \sigma_k N(0, C_k) \qquad (38)$$

where σ_k is the step size and C_k is the covariance matrix (e.g., adapting to the local fitness landscape) [64].

3.3.4. Physics-Informed Neural Network

PINNs integrate physical laws directly into neural network architectures [65]. The fundamental PINN loss function for aerospace applications combines data and physics-based terms:

$$\mathcal{L}_{total} = \mathcal{L}_{data} + \lambda \mathcal{L}_{physics}$$

where λ balances the contribution of physical constraints. For aerodynamic applications, the physics loss typically incorporates the Navier-Stokes equations:

$$\mathcal{L}_{physics} = \left\| \frac{\partial u}{\partial t} + (u \cdot \nabla)u + \frac{1}{\rho} \nabla_{p-\nu} \nabla^2 u \right\|^2 \quad (39)$$

where u represents velocity fields (for aircraft boundary layers) and p pressure distributions (around aerodynamic surfaces) [66].

3.3.5. Hybrid AI System

Hybrid AI systems combine multiple AI approaches to leverage their complementary strengths. The general framework integrates symbolic reasoning with neural networks through a hybrid loss function:

$$\mathcal{L}_{hybrid} = \alpha \mathcal{L}_{neural} + \beta \mathcal{L}_{symbolic} + \gamma \mathcal{L}_{constraint} \quad (40)$$

where α, β, and γ are weighting coefficients optimized for specific aerospace applications. These systems have demonstrated success in aerospace supply chain optimization and maintenance prediction.

The decision-making process incorporates both data-driven and expert knowledge components:

$$P(action \mid state) = f_{neural}(state) \cdot g_{symbolic}(rules) \quad (41)$$

where f_{neural} represents the neural network component (processing sensor data) and $g_{symbolic}$ represents the symbolic reasoning system (enforcing aerodynamic constraints) [67].

4. Ethical Considerations and Mitigation Strategies

In the rapidly evolving field of aerospace-AI integration, ethical considerations have become important for the use of responsible technological advancement. The aerospace industry's increasing reliance on AI necessitates a comprehensive framework addressing four important ethical dimensions while implementing effective mitigation strategies. These ethical considerations focus on ethical AI development practices in aerospace. The ethical framework encompasses the entire lifecycle of AI systems, from conception to deployment and maintenance. Furthermore, mitigation strategies extend beyond individual considerations to create a comprehensive ethical framework. In general, organizations have implemented cross-functional ethics committees that oversee AI development and deployment. These committees should regularly assess potential risks, update guidelines based on emerging challenges and ensure alignment with industry best practices. Continuous stakeholder engagement, including feedback from operators, maintenance personnel, and end-users, will help refine these strategies.

4.1. Transparency

Aerospace applications should maintain explainable AI architectures that allow stakeholders to understand decision-making processes. This is particularly important in safety-critical operations including flight control systems and navigation [68]. Organizations involved within the field should implement transparency through detailed documentation of AI models, regular audits, and the development of interpretable algorithms [69]. These measures enable engineers and operators to trace and validate AI decisions, ensuring accountability in both development and deployment phases.

4.2. Bias and Discrimination

Bias and discrimination prevention represents the second critical consideration. AI systems in aerospace applications should be designed to avoid perpetuating existing biases or creating new ones. The prevention of bias includes ensuring that training data represents diverse scenarios and populations, particularly in applications such as passenger screening systems or automated maintenance scheduling [70]. Discriminatory analytics can lead to unfair treatment and self-fulfilling prophecies that undermine individual autonomy and societal participation [71]. Mitigation strategies consist of implementing rigorous testing protocols for bias detection, utilizing diverse development teams, and regularly updating training datasets to maintain representational balance.

4.3. Privacy and Compliance

Privacy and compliance measures constitute the third ethical pillar. As aerospace-AI systems process vast amounts of sensitive data, comprising flight patterns, passenger information, and operational metrics, protecting privacy becomes crucial. Organizations part on the aerospace field should implement data protection frameworks that comply with international regulations, such as General Data Protection Regulation (GDPR), California Privacy Rights Act (CPRA), Personal Information Protection Law, and Brazil General Data Protection Law [72, 73], while maintaining operational efficiency. Privacy protection measures should consider encryption protocols, secure data storage systems, and strict access controls. Regular privacy impact assessments and updates to security measures will help to maintain the integrity of the data protection frameworks.

4.4. Safety Protocols

Safety protocols form the fourth ethical consideration. In aerospace applications, where AI systems often control important operations, specifically predictive maintenance, flight control systems, and autonomous navigation [74, 75], ensuring safety is crucial. The implementation encompasses redundant systems, notably multiple inertial measurement units and pitot tubes, fail-safe mechanisms comprising fault detection algorithms and automated fault isolation [76], and comprehensive testing procedures through validation and verification protocols. Organizations involved in aerospace domain should continue developing multi-layered safety frameworks that incorporate real-time monitoring for instance for engine performance and structural health, emergency override capabilities to return control to human operators, and regular system audits that will ensure compliance with safety standards and performance requirements. Safety protocols should balance automation benefits with human oversight to maintain optimal safety levels.

5. Conclusion

The integration of AI in aerospace represents a change in how we approach complex aerospace challenges. This comprehensive review has highlighted the synergy between traditional aerospace engineering and AI technologies. ML algorithms have demonstrated exceptional capabilities. For instance, the RF algorithm achieves sub-10-meter accuracy in trajectory prediction, while the SVM algorithm shows 99.89% precision in fault detection systems. DL algorithms, particularly CNN algorithms, have revolutionized structural inspection with 79% accuracy in satellite component detection. The emergence of generative AI has opened new frontiers in aerospace design optimization. GANs have achieved positive results in airfoil design. Its utilization demonstrates a less than 1% error in profile fitting and maintains aerodynamic stealth characteristics within 10% error margins. PINNs have successfully integrated fundamental physical laws into learning architectures. However, this technological advancement comes with significant ethical implications that demand immediate attention. The industry must address critical concerns regarding transparency in AI decision-making, bias prevention in automated systems, and implement safety protocols. Looking ahead, several key recommendations emerge. Some of these recommendations include the establishment of dedicated ethics boards for AI oversight, the development of international collaboration for universal safety standards within the aerospace field, and the implementation of regular system audits. Future research should focus on enhancing AI interpretability, developing more validation methodologies, and implementing various citizen science programs and the use of AI in the community. The successful integration of AI in aerospace depends on balancing technological innovation with ethical considerations and human oversight.

Abbreviations

AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
GenAI	Generative Artificial Intelligence
RF	Random Forest
SVM	Support Vector Machine
DT	Decision Tree
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
GBM	Gradient Boosting Machine
ANN	Artificial Neural Network
DRL	Deep Reinforcement Learning
RNN	Recurrent Neural Network
STEM	Science, Technology, Engineering, and Mathematics
k-NN	k-Nearest Neighbor
RBF	Radial Basis Function
LLM	Large Language Model
CART	Classification and Regression Tree
NLP	Natural Language Processing
DQN	Deep Q-Network
MDP	Markov Decision Process
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
GDPR	General Data Protection Regulation
CPRA	California Privacy Rights Act

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] Hassan, K., et al., Application of Artificial Intelligence in Aerospace Engineering and Its Future Directions: A Systematic Quantitative Literature Review. Archives of Computational Methods in Engineering, 2024: p. 1-56. https://doi.org/10.1007/s11831-024-10105-7

[2] Insaurralde, C.C. Artificial intelligence engineering for aerospace applications. in 2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC). 2020. IEEE. https://doi.org/10.1109/DASC50938.2020.9256770

[3] McCarthy, J., M.L. Minsky, N. Rochester, and C.E. Shannon, A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. AI magazine, 2006. 27(4): p. 12-12. https://doi.org/10.1609/aimag.v27i4.1904

[4] Rajaraman, V., JohnMcCarthy—Father of artificial intelligence. Resonance, 2014. 19: p. 198-207. https://doi.org/10.1007/s12045-014-0027-9

[5] De Cos Juez, F. J., P. G. Nieto, J. M. Torres, and J. T. Castro, Analysis of lead times of metallic components in the aerospace industry through a supported vector machine model. Mathematical and computer modelling, 2010. 52(7-8): p. 1177-1184. https://doi.org/10.1016/j.mcm.2010.03.017

[6] Chen, Z., S. Cao, and Z. Mao, Remaining useful life estimation of aircraft engines using a modified similarity and supporting vector machine (SVM) approach. Energies, 2017. 11(1): p. 28. https://doi.org/10.3390/en11010028

[7] Li, F., et al. A light gradient boosting machine for remainning useful life estimation of aircraft engines. in 2018 21st International Conference on Intelligent Transportation Systems (ITSC). 2018. IEEE. https://doi.org/10.1109/ITSC.2018.8569801

[8] Viale, L., A. P. Daga, A. Fasana, and L. Garibaldi, Least squares smoothed k-nearest neighbors online prediction of the remaining useful life of a NASA turbofan. Mechanical Systems and Signal Processing, 2023. 190: p. 110154. https://doi.org/10.1016/j.ymssp.2023.110154

[9] Dong, X., C. J. Taylor, and T. F. Cootes, A random forest-based automatic inspection system for aerospace welds in X-ray images. IEEE Transactions on Automation Science and Engineering, 2020. 18(4): p. 2128-2141. https://doi.org/10.1109/TASE.2020.3039115

[10] Qin, L., et al. Decision Tree-Based Resource Recommendation System for Aerospace Manufacturing. in 2024 7th International Conference on Computer Information Science and Application Technology (CISAT). 2024. IEEE. https://doi.org/10.1109/CISAT62382.2024.10695264

[11] Berke, L., S. Patnaik, and P. Murthy, Optimum design of aerospace structural components using neural networks. Computers & structures, 1993. 48(6): p. 1001-1010. https://doi.org/10.1016/0045-7949(93)90435-G

[12] Dai, W., K. Liang, and B. Wang, State monitoring method for tool wear in aerospace manufacturing processes based on a convolutional neural network (CNN). Aerospace, 2021. 8(11): p. 335. https://doi.org/10.3390/aerospace8110335

[13] Xu, Z., W. Zeng, X. Chu, and P. Cao, Multi-aircraft trajectory collaborative prediction based on social long short-term memory network. Aerospace, 2021. 8(4): p. 115. https://doi.org/10.3390/aerospace8040115

[14] Hovell, K. C., Deep Reinforcement Learning as Guidance for Aerospace Robotics. 2022, Carleton University. https://doi.org/10.22215/etd/2022-14922

[15] ElDali, M. and K. D. Kumar. Fault diagnosis and prognosis of aerospace systems using growing recurrent neural networks and LSTM. in 2021 IEEE aerospace conference (50100). 2021. IEEE. https://doi.org/10.1109/AERO50100.2021.9438432

[16] Teng, D., et al., Generative adversarial surrogate modeling framework for aerospace engineering structural system reliability design. Aerospace Science and Technology, 2024. 144: p. 108781. https://doi.org/10.1016/j.ast.2023.108781

[17] Allen, J., Aeronautics-1903; aerospace-2003. 2103. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 2005. 219(3): p. 235-260. https://doi.org/10.1243/095441005X30252

[18] Neal, C. R., The Moon 35 years after Apollo: What's left to learn? Geochemistry, 2009. 69(1): p. 3-43. https://doi.org/10.1016/j.chemer.2008.07.002

[19] Cortright, E. M., Apollo Expeditions to the Moon: The NASA History 50th Anniversary Edition. 2019: Courier Dover Publications.

[20] Mahesh, B., Machine learning algorithms-a review. International Journal of Science and Research (IJSR). [Internet], 2020. 9(1): p. 381-386.

[21] LeCun, Y., Y. Bengio, and G. Hinton, Deep learning. nature, 2015. 521(7553): p. 436-444. https://doi.org/10.1038/nature14539

[22] Banh, L. and G. Strobel, Generative artificial intelligence. Electronic Markets, 2023. 33(1): p. 63. https://doi.org/10.1007/s12525-023-00680-1

[23] Genuer, R., J.-M. Poggi, and C. Tuleau, Random Forests: some methodological insights. arXiv preprint arXiv: 0811.3619, 2008. https://doi.org/10.48550/arXiv.0811.3619

[24] Field, L. J., J. W. Kern, and L. B. Rosman, Re-visiting projections of PCBs in Lower Hudson River fish using model emulation. Science of the Total Environment, 2016. 557: p. 489-501. https://doi.org/10.1016/j.scitotenv.2016.02.072

[25] Azhar, S., Building information modeling (BIM): Trends, benefits, risks, and challenges for the AEC industry. Leadership and management in engineering, 2011. 11(3): p. 241-252. https://doi.org/10.1061/(ASCE)LM.1943-5630.000012

[26] Lee, S., W. Park, and S. Jung, Fault detection of aircraft system with random forest algorithm and similarity measure. The Scientific World Journal, 2014. 2014(1): p. 727359. https://doi.org/10.1155/2014/727359

[27] Joshi, A.V., Support vector machines, in Machine learning and artificial intelligence. 2022, Springer. p. 89-99. https://doi.org/10.1007/978-3-031-12282-8

[28] Grehan, J., D. Ignatyev, and A. Zolotas, Fault detection in aircraft flight control actuators using support vector machines. Machines, 2023. 11(2): p. 211. https://doi.org/10.3390/machines11020211

[29] dos Santos, R. R., T. G. d. P. Machado, and S. G. P. Castro, Support vector machine applied to the optimal design of composite wing panels. Aerospace, 2021. 8(11): p. 328. https://doi.org/10.3390/aerospace8110328

[30] Fan, H.-Y., G. S. Dulikravich, and Z.-X. Han, Aerodynamic data modeling using support vector machines. Inverse Problems in Science and Engineering, 2005. 13(3): p. 261-278. https://doi.org/10.1080/10682760412331330177

[31] Yuan, X.-J., et al., Quantum support vector machines for aerodynamic classification. Intelligent Computing, 2023. 2: p. 0057. https://doi.org/10.34133/icomputing.0057

[32] Cai, H., T. Yu, and C. Xia, Quality - Oriented Classification of Aircraft Material Based on SVM. Mathematical Problems in Engineering, 2014. 2014(1): p. 273906. https://doi.org/10.1155/2014/273906

[33] Ibm (2023). IBM. Retrieved 27 December 2023, from https://www.ibm.com/think/topics/support-vector-machine

[34] Maragatharaj, K. Dinesh, R. Vimal, and Santhiyakumari; A K-NN Algorithm-Based ML Model for Predictive Maintenance in Aircraft Engines. Journal of Propulsion Technology, 2024. 45(1): p. 3083-3089.

[35] Beatty, J. and S. Sundkvist, Comparative Analysis of Tree-Based Methods for Predictive Modeling. 2024.

[36] Dubossarsky, E., J. H. Friedman, J. T. Ormerod, and M. P. Wand, Wavelet-based gradient boosting. Statistics and Computing, 2016. 26: p. 93-105. https://doi.org/10.1007/s11222-014-9474-0

[37] Azo, R. (2024). AZO ROBOTICS. Retrieved 20 November 2024, from https://www.azorobotics.com/News.aspx?newsID=15492

[38] Breiman, L., Classification and regression trees. 2017: Routledge.

[39] Marudi, M., I. Ben-Gal, and G. Singer, A decision tree-based method for ordinal classification problems. IISE Transactions, 2024. 56(9): p. 960-974. https://doi.org/10.1080/24725854.2022.2081745

[40] Sun, Z., et al., An improved random forest based on the classification accuracy and correlation measurement of decision trees. Expert Systems with Applications, 2024. 237: p. 121549. https://doi.org/10.1016/j.eswa.2023.121549

[41] Ignatenko, V., A. Surkov, and S. Koltcov, Random forests with parametric entropy-based information gains for classification and regression problems. PeerJ Computer Science, 2024. 10: p. e1775. https://doi.org/10.7717/peerj-cs.1775

[42] Segev, I., J. Martin, and R. Knight, Math That Changed the World. 2024: Frontiers Media SA.

[43] Rojas, R. V. B., Artificial Intelligence: Genesis, Development, and Future, in Revolutionizing Communication. 2024, CRC Press. p. 1-15.

[44] Lu, M. and N. Ke. LeNet-5 handwritten digit recognition based on deep learning. in International Conference on Optics, Electronics, and Communication Engineering (OECE 2024). 2024. SPIE. https://doi.org/10.1117/12.3049259

[45] Zhang, H., Y. Zhang, Q. Feng, and K. Zhang, Review of Machine-Learning Approaches for Object and Component Detection in Space Electro-optical Satellites. International Journal of Aeronautical and Space Sciences, 2024. 25(1): p. 277-292. https://doi.org/10.1007/s42405-023-00653-w

[46] Chen, Y., J. Gao, and K. Zhang, R-CNN-Based Satellite Components Detection in Optical Images. International Journal of Aerospace Engineering, 2020. 2020(1): p. 8816187. https://doi.org/10.1155/2020/8816187

[47] Gallon, R., F. Schiemenz, A. Menicucci, and E. Gill, Convolutional Neural Network Design and Evaluation for Real-Time Multivariate Time Series Fault Detection in Spacecraft Attitude Sensors. arXiv preprint arXiv:2410.09126, 2024. https://doi.org/10.48550/arXiv.2410.09126

[48] Katte, T., Recurrent neural network and its various architecture types. International Journal of Research and Scientific Innovation (IJRSI), 2018. 5(3): p. 124-129.

[49] Wu, J., Y. Ye, and J. Du, Autonomous Drones in Urban Navigation: Autoencoder Learning Fusion for Aerodynamics. Journal of Construction Engineering and Management, 2024. 150(7): p. 04024067. https://doi.org/10.1061/JCEMD4.COENG-14787

[50] Hu, X., H. Wang, M. Gong, and T. Wang, Guidance Design for Escape Flight Vehicle against Multiple Pursuit Flight Vehicles Using the RNN-Based Proximal Policy Optimization Algorithm. Aerospace, 2024. 11(5): p. 361. https://doi.org/10.3390/aerospace11050361

[51] Middlehurst, M., P. Schäfer, and A. Bagnall, Bake off redux: a review and experimental evaluation of recent time series classification algorithms. Data Mining and Knowledge Discovery, 2024: p. 1-74. https://doi.org/10.1007/s10618-024-01022-1

[52] Sha, X., Time Series Stock Price Forecasting Based on Genetic Algorithm (GA)-Long Short-Term Memory Network (LSTM) Optimization. arXiv preprint arXiv: 2405.03151, 2024. https://doi.org/10.48550/arXiv.2405.03151

[53] Vaswani, A., et al., Attention Is All You Need. (Nips), 2017. arXiv preprint arXiv: 1706.03762, 2017. 10: p. S0140525X16001837.

[54] Zhu, L., et al., DRL-RNP: Deep Reinforcement Learning-Based Optimized RNP Flight Procedure Execution. Sensors, 2022. 22(17): p. 6475. https://doi.org/10.3390/s22176475

[55] Hochreiter, S., Long Short-term Memory. Neural Computation MIT-Press, 1997.

[56] Presidio (2024). Presidio. Retrieved 27 May 2024 https://www.presidio.com/exploring-the-power-of-generative-adversarial-networks-gans-with-azure/

[57] Jin, S.-Y., et al., Airfoil aerodynamic/stealth design based on conditional generative adversarial networks. Physics of Fluids, 2024. 36(7). https://doi.org/10.1063/5.0220671

[58] Repository, T. (2024). Repository TUdelft. Retrieved 01 January 2024. https://repository.tudelft.nl/record/uuid:3beee043-9afc-41a2-8933-e2bc75893b24#abstract

[59] Fanfarillo, A., B. Roozitalab, W. Hu, and G. Cervone, Probabilistic forecasting using deep generative models. GeoInformatica, 2021. 25: p. 127-147. https://doi.org/10.1007/s10707-020-00425-8

[60] Lopez-Alvis, J., E. Laloy, F. Nguyen, and T. Hermans, Deep generative models in inversion: The impact of the generator's nonlinearity and development of a new approach based on a variational autoencoder. Computers & Geosciences, 2021. 152: p. 104762. https://doi.org/10.1016/j.cageo.2021.104762

[61] Chen, W., K. Chiu, and M. Fuge. Aerodynamic design optimization and shape exploration using generative adversarial networks. in AIAA Scitech 2019 forum. 2019. https://doi.org/10.2514/6.2019-2351

[62] Arias-Montano, A., C. A. C. Coello, and E. Mezura-Montes, Multiobjective evolutionary algorithms in aeronautical and aerospace engineering. IEEE transactions on evolutionary computation, 2012. 16(5): p. 662-694. https://doi.org/10.1109/TEVC.2011.2169968

[63] Anderson, M. B. (2002, May). Genetic algorithms in aerospace design: substantial progress, tremendous potential. In Brussels: NATO/Von Karmen Institute Workshop on Intelligent System.

[64] Holst, T. L., Genetic algorithms applied to multi-objective aerodynamic shape optimization. 2005.

[65] Furfaro, R., A. D'Ambrosio, E. Schiassi, and A. Scorsoglio. Physics-informed neural networks for closed-loop guidance and control in aerospace systems. in AIAA SCITECH 2022 Forum. 2022. https://doi.org/10.2514/6.2022-0361

[66] Eivazi, H., M. Tahani, P. Schlatter, and R. Vinuesa, Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations. Physics of Fluids, 2022. 34(7). https://doi.org/10.1063/5.0095270

[67] Hagos, D. H. and D. B. Rawat, Neuro-Symbolic AI for Military Applications. IEEE Transactions on Artificial Intelligence, 2024. https://doi.org/10.1109/TAI.2024.3444746

[68] Raz, A. K., E. P. Blasch, C. Guariniello, and Z. T. Mian. An overview of systems engineering challenges for designing AI-enabled aerospace systems in AIAA Scitech 2021 Forum. 2021. https://doi.org/10.2514/6.2021-0564

[69] Protopapadakis, G., A. Apostolidis, and A. I. Kalfas. Explainable and interpretable AI-assisted remaining useful life estimation for aeroengines. in Turbo Expo: Power for Land, Sea, and Air. 2022. American Society of Mechanical Engineers. https://doi.org/10.1115/GT2022-80777

[70] Sharma, S. and M. S. Graydon, Social bias in AI and its implications. 2021.

[71] Tortora, L., Beyond Discrimination: Generative AI Applications and Ethical Challenges in Forensic Psychiatry. Frontiers in Psychiatry, 2024. 15: p. 1346059. https://doi.org/10.3389/fpsyt.2024.1346059

[72] Iata (2024). IATA. Retrieved 01 May 2024. https://www.iata.org/contentassets/da67b41b565c4bd88d5944b136cc8d15/data-protection-white-paper.pdf

[73] Enzuzo.(2024). Enzuzo. Retrieved 04 November 2024. https://www.enzuzo.com/blog/data-privacy-laws

[74] Qoco (2024). Ooco. Retrived 22 May 2024. https://www.qoco.aero/blog/ai-in-aviation-maintenance-how-its-changing-the-industry

[75] Aiola (2024). Aiola. Retrieved 24 July 2024. https://aiola.com/blog/ai-in-aircraft-maintenance/

[76] Eubank, R., E. Atkins, and S. Ogura. Fault detection and fail-safe operation with a multiple-redundancy air-data system. in AIAA Guidance, Navigation, and Control Conference. 2010. https://doi.org/10.2514/6.2010-7855