

Article

Fire Control Radar Fault Prediction with Real-Flight Data

Minyoung Kim ^{1,*}, Ikgyu Lee ¹ , Seon-Ho Jeong ¹, Dawn An ²  and Byoungserb Shim ¹¹ AI Development Team, Korea Aerospace Industries, Ltd., Seoul 06151, Republic of Korea;

ikgyu.lee@koreaaero.com (I.L.); seonho.jeong@koreaaero.com (S.-H.J.); tspace@koreaaero.com (B.S.)

² Advanced Mobility System Group, Korea Institute of Industrial Technology, Daegu 42994, Republic of Korea; dawnan@kitech.re.kr

* Correspondence: minyoung.kim@koreaaero.com; Tel.: +82-55-851-9535

Abstract

Unexpected failures of avionics equipment critically affect flight safety, operational availability, and maintenance costs. To address these issues, Condition-Based Maintenance Plus (CBM+) has emerged as a strategy to optimize maintenance timing based on equipment condition rather than fixed schedules. However, while aviation research has largely focused on engines and structures, studies on avionics systems remain limited, often relying on simulations. This study proposes a novel data-driven approach to predict avionics equipment failures using actual aircraft operational data. Maneuver-related sequences were analyzed to investigate correlations between flight patterns and equipment faults, and a two-stage framework was developed. In the feature extraction stage, a CNN-LSTM encoder compresses 10 s maneuver sequences into compact yet informative representations. In the fault prediction stage, AI models classify failures of the Fire Control Radar based on these features. Experiments with real flight data validated the effectiveness of the method, showing that the CNN-LSTM encoder preserved essential maneuver information, while the combination of Standard Scaling and Multi-Layer Perceptron achieved the best performance, with a maximum Fault Recall of 98%. These findings demonstrate the feasibility of practical CBM+ implementation for avionics equipment using only flight data, providing a promising solution to improve maintenance efficiency and aviation safety.

Keywords: Condition-Based Maintenance; avionics; fault prediction

Academic Editor: Yan (Rockee) Zhang

Received: 16 September 2025

Revised: 16 October 2025

Accepted: 17 October 2025

Published: 21 October 2025

Citation: Kim, M.; Lee, I.; Jeong, S.-H.; An, D.; Shim, B. Fire Control Radar Fault Prediction with Real-Flight Data. *Aerospace* **2025**, *12*, 945. <https://doi.org/10.3390/aerospace12100945>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Military aircraft availability directly affects operational capability, and extended downtime due to spare parts availability issues can cause serious problems. To address these issues, Condition-Based Maintenance Plus (CBM+) strategy, which performs maintenance based on real-time monitoring of equipment conditions, has emerged as critically important. CBM+ can utilize both physics-based and data-driven approaches, and with recent advances in AI technology, data-driven CBM+ research has been actively conducted. CBM+ has been recognized for its value by demonstrating maintenance efficiency improvements and cost reduction effects across various industrial sectors.

Aircraft Availability (AA) is a key indicator for evaluating the Air Force's operational readiness and is an important factor that forms the foundation of combat capability. Ritschel et al. [1] indicate that U.S. Air Force aircraft availability has continuously declined from approximately 75% in the early 1990s to approximately 62% in 2017, causing serious concerns among Air Force leadership. It was revealed that there is approximately a 0.05 h decrease in actual mission execution time per 100 h of operational time, proving that

decreased availability directly impacts actual operational capability. Furthermore, the same study confirmed that Not Mission Capable Maintenance (NMCM) accounts for the largest proportion of causes of availability decline and is simultaneously the most rapidly increasing sector. Notably, Not Mission Capable Maintenance Unscheduled (NMCMU) accounts for 72% of the total [1].

CBM+ technology can effectively resolve this availability decline problem caused by unscheduled maintenance. The handbook containing standards for U.S. military aircraft CBM systems defines the goals of CBM implementation as “reducing maintenance workload,” “increasing aircraft operational rates,” “improving aviation safety,” and “reducing maintenance costs” [2]. In essence, the core principle of CBM+ is “performing maintenance based on equipment condition”, which involves conducting maintenance proactively before failures occur, thereby minimizing unscheduled maintenance. The U.S. Department of Defense (DoD) has positioned CBM+ as a core strategy for weapon system sustainability and established a systematic implementation framework. Policy foundations were established through DoD Directive 4151.18, “Maintenance of Military Materiel” [3], and DoD Instruction 4151.22, “Condition-Based Maintenance Plus for Materiel Maintenance” [4], and specific implementation guidelines are provided through the DoD CBM+ Guidebook [5] updated in 2024. The U.S. military has moved beyond policy establishment to actively utilize actual CBM+ strategies in the field, and maintenance personnel also emphasize the importance of CBM+, saying that the maintenance paradigm has changed since the introduction of CBM+ maintenance strategies [6,7]. Indeed, the U.S. F-35 fighter case has been reported to reduce maintenance costs by approximately 50% through the introduction of advanced maintenance techniques, including CBM+ systems, proving its effectiveness [8].

Recent research on applying CBM+ to various aircraft systems has been vigorous; however, the majority of these studies focus on physical subsystems such as engines and mechanical components. Li et al. [9] observed that most CBM+ investigations are centered on mechanical degradation, while studies incorporating the combined effects of environmental factors and operational conditions remain relatively scarce. Consequently, CBM+ research addressing avionics remains insufficient. However, avionics involve complex interactions between hardware and software, which can lead to system-wide performance degradation or serious safety accidents when failures occur [10], and repair and replacement costs are also substantial [11], making the CBM+ application essential. There have been a few studies focusing on electrical systems; however, most of them have relied on simulation data, as it is difficult to obtain real data related to avionics. According to existing research, effective CBM+ application for electronic equipment requires the installation of dedicated sensors to acquire data such as temperature, voltage, and current [12–14]. However, installing sensors on aircraft requires various processes, including load analysis and stability verification of the equipment, making it nearly impossible to attach sensors for data acquisition on operational aircraft. Therefore, using data that can currently be acquired from aircraft is crucial for establishing CBM+ applicable to operational aircraft. Unfortunately, even currently available aircraft data has several limitations. The constraints of data collection, transmission, and storage capabilities of equipment installed on aircraft make it difficult to secure continuous and consistent data [15]. Since the CBM+ application requires continuous equipment monitoring through sustained data acquisition, it is important to consider sustainability from the algorithm development stage. This study used actual operational data rather than simulation data and proposes a new approach that utilizes data obtainable from systems already installed on aircraft without the process of attaching new sensors to target equipment. Furthermore, research was conducted to enable integration with Flight Data Recorder (FDR) data, which is the most stable obtainable data.

In other words, we conduct the study using only the maneuver-related data obtainable from the FDR, thereby proposing a sustainable methodology.

To develop a CBM+ algorithm for avionics systems utilizing maneuver data, the selection of target avionics should be prioritized. Since avionics systems vary widely in their functions and operational environments, developing a universal algorithm applicable to all systems is practically infeasible. Therefore, it is essential to select a representative system that shows a strong correlation with aircraft maneuvers and a high frequency of failures and to develop a model optimized for the fault characteristics of that system. Based on these criteria, the Fire Control Radar (FCR) was selected as the target avionics. According to previous research, the FCR system has the most frequent Line Replaceable Unit (LRU) repairs and replacements among avionics equipment, suggesting that CBM+ application would significantly impact AA improvement and maintenance cost reduction [16]. Additionally, the FCR is located at the aircraft's foremost section, receiving the most load from maneuvers, making maneuver-based fault prediction most likely to show meaningful results.

Specifically, real data from multiple aircraft of the same type was compressed using an autoencoder, and various prediction models were used to predict whether faults occurred in the FCR system. Here, parameters related to maneuvers were primarily used, since these data can be continuously acquired. We verified whether there is a significant correlation between aircraft maneuvers and faults. In the feature extraction stage, data from all aircraft were used to improve generalization, and in the fault prediction stage, models were developed for each aircraft to reflect individual aircraft characteristics. Various combinations of data scaling and models were experimented with to find the most efficient model for fault occurrence classification. The structure of this paper is as follows. Section 2 introduces research trends related to CBM+ and avionics fault prediction, and Section 3 provides detailed descriptions of the dataset used. Section 4 presents the autoencoder algorithm for data feature extraction, and Section 5 introduces fault prediction algorithms using extracted features. Finally, Section 6 discusses conclusions and future research directions based on the research results.

2. Related Works

2.1. Data Driven CBM+

CBM+ is an extension of the traditional CBM concept that integrates sensor data, diagnostic and prognostic technologies, and data analysis to evaluate the actual condition and performance of facilities or systems in real-time, supporting optimal maintenance decision-making based on this information [4]. Traditional CBM+ approaches have mainly relied on physics-based models and statistical methods, but these approaches have shown significant limitations in handling high-dimensional, nonlinear, and class-imbalanced industrial data [17]. Machine learning-based data-driven approaches can effectively overcome these constraints, and neural networks and ensemble methods particularly demonstrate superior performance in capturing complex nonlinear relationships compared to traditional physics-based models [18]. Industrial applications in semiconductors, aerospace, and other sectors have also shown that AI-based data analysis can successfully implement CBM+ even in areas where traditional physics-based approaches were difficult to apply, such as multi-dimensional, multi-parameter data [19].

2.2. CBM+ in Aircraft

Interest in CBM+ technology in the aviation industry has been driven by various factors, including increased aircraft complexity, rising operational costs, and strengthened safety requirements. Aircraft are highly complex systems composed of thousands of

components, making it essential to effectively monitor and manage the condition of each component for operational efficiency and safety assurance. Stanton et al. [20] indicate that aircraft have the capability to record vast amounts of sensor data across almost all components during flight, making a data-driven CBM+ application essential in this situation.

However, current aviation CBM+ research has mainly focused on engines, structural components, and flight control systems. This is because these systems have critical safety impacts when they fail, provide abundant sensor data, and have relatively high availability of public datasets [20]. In the engine field, data-driven predictive maintenance research using NASA's turbofan engine dataset is active, with studies constructing direct mappings between multi-sensor measurements and Remaining Useful Life (RUL) using Bi-LSTM networks [21]. Additionally, machine learning methodologies using autoencoder, LSTM, and Gaussian Process Regression are being applied to performance degradation indicator construction, RUL prediction, and maintenance timing optimization [22]. In structural health monitoring [23], research applying machine learning and deep learning methodologies such as CNN and LSTM to various aircraft structures is being conducted, and new approaches combining physics and machine learning, such as physics-informed neural networks, are also being explored [12]. Prognostic Health Management (PHM) systems in aircraft health management provide data-driven decision-making that offers insights into asset health and performance using data analysis and machine learning techniques [24].

While various aviation CBM+ applications are being implemented as described above, avionics CBM+ research is relatively insufficient. Unlike other mechanical components, electronic component failures often occur suddenly, and they frequently exhibit abrupt functional failures rather than gradual performance degradation, making failure prediction relatively difficult. Additionally, avionics equipment has very low fault occurrence rates by nature, resulting in data distributions heavily skewed toward normal conditions, making fault cases extremely rare and difficult to obtain [25] and often leading to biased learning results. Despite these difficulties, research to apply CBM+ to avionics equipment is ongoing. Bharadwaj et al. [26] proposed a model-based fault detection system using bond graph modeling, extended Kalman filter, and temporal causal graph to analyze fault propagation and signatures in power supply units, GPS, and INAV systems. Chen et al. [27] constructed a CNN-based diagnostic model targeting aircraft auxiliary power units and introduced a progressive structure to address data imbalance and fault confusion issues, improving diagnostic accuracy. However, both studies rely on simulation data, presenting limitations in reflecting various operational conditions and complex faults in actual flight environments. Our study addresses these limitations by proposing an approach to predict avionics equipment faults using actual operational data. Accordingly, this study seeks to contribute to a more practicable CBM+ framework by exploiting real flight data.

Nevertheless, several challenges must be addressed before such a CBM+ system can be deployed on operational aircraft. Contemporary CBM+ implementations on operational platforms adopt a variety of architectures that are dictated by the criticality of the subsystem and the associated computational requirements. For example, the F-35 fighter has historically relied on the Autonomic Logistics Information System (ALIS) and, more recently, on its successor, the Operational Data Integrated Network (ODIN) [28]. This architecture acquires and partially processes data in real time during flight, while the majority of diagnostic and prognostic analytics are executed in a ground-based cloud environment, thereby embodying a hybrid CBM+ approach [29]. In rotary-wing platforms, Health and Usage Monitoring Systems (HUMSs) are employed to assess the health of helicopter power-train components by means of vibration-based condition indicators (CIs) [30]. The HUMS continuously monitors critical parameters in real time and generates alerts whenever predefined thresholds are exceeded, whereas detailed data analysis is carried out

post-flight on the ground [30]. These systems illustrate that the practical implementation of CBM+ follows a staged approach. Initially, maintenance planning is based on post-flight analyses; subsequently, limited real-time monitoring is added to provide critical warnings; finally, full integration requires extensive certification and environmental validation.

3. Flight Data

3.1. About Flight Data

Among aircraft operational data, the most representative is FDR data. FDR data is mandatorily installed for aircraft safety assurance and accident investigation, and research utilizing this data for aircraft structural damage analysis has been conducted [15]. However, this data has limitations for use in this avionics fault prediction study because fault data related to avionics equipment is rarely acquired from FDR. The reasons for this are mainly two: the storage format of fault data and the storage capacity limitations of FDR equipment. FDR data consists of two types: discrete and continuous data, and fault data is recorded only in continuous data. Discrete data is recorded when specific events occur, requiring a small storage capacity, while continuous data records data at frequent regular intervals, requiring relatively large storage capacity. At this time, due to the storage capacity limitations of FDR equipment, continuous data from previous flights is frequently overwritten as new flights are conducted. Therefore, the acquisition rate of continuous data and fault data is very low, making it difficult to utilize. Since sustainable CBM+ algorithm development requires data that can be stably obtained, in subsequent sections of this paper, FDR data is defined and limited to discrete FDR data that can be obtained for all flights. FDR data mainly consists of maneuver-related parameters with irregular time steps and is not directly correlated with avionics equipment faults, making it difficult to use directly for model development.

Therefore, this study utilized MIL-STD-1553B Multiplex data (hereinafter, M1553B data) collected during actual aircraft operations. This data is real-time communication data between avionics equipment, containing fault data and various parameters recorded as continuous variables. So, this data is suitable for AI model learning for CBM+. However, unlike regularly available FDR data, the M1553B data acquisition procedure is not officially established, creating difficulties in data collection. For this reason, research was conducted to enable the CBM+ application even when M1553B data acquisition is not possible in the future. Specifically, this study explored the possibility of predicting FCR fault occurrence using FDR data in the future by linking maneuver-related parameters from FDR data with parameters from M1553B data. Figure 1 illustrates the limitations of FDR storage and the differences between FDR data and M1553B data through graphical examples.

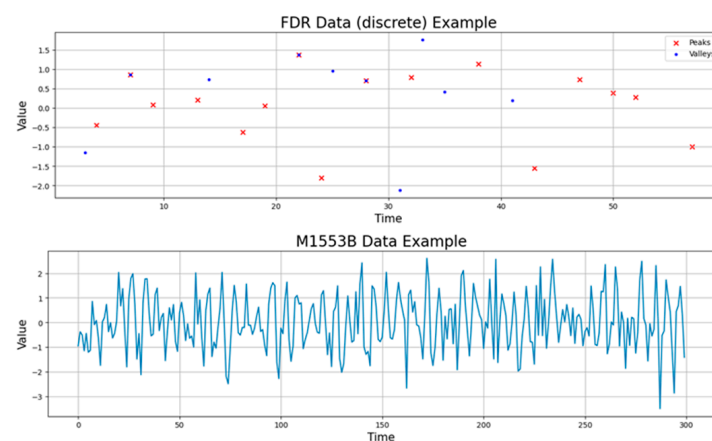


Figure 1. Example of Flight Data Recorder data and M1553B data (not real data).

3.2. Avionics Communication Data (M1553B)

3.2.1. MIL-STD-1553B

MIL-STD-1553B is a data bus standard for aircraft subsystem communication established by the U.S. DoD in 1978 and has been widely used ever since. This standard is based on a Command/Response protocol, where one Bus Controller (BC) can communicate with up to 31 Remote Terminals (RTs). This communication method enables the monitoring of all data exchanges between avionics equipment. The standard has been applied to military aircraft of various countries and has been adopted as NATO STANAG 3838 AVS, making it considered a cornerstone of digital avionics engineering. The M1553B data used in this study is a chronological record of all communication messages on such MIL-STD-1553B buses, containing status information, sensor data, and control commands of each avionics equipment.

3.2.2. Collected Data

This data was collected from several fixed-wing aircraft of the same type actually in operation over approximately 1 year and 6 months, and the greatest advantage of M1553B data is that it reflects actual operational environments. The number of flights per aircraft varies, but data acquired from approximately 1600 flights was utilized. To acquire data, after each flight, storage devices must be manually removed and M1553B data files transferred by hand, and the initial data, composed of raw bit data, must be parsed based on configuration rules. Through this process, data for each parameter can be extracted. Since the unit, frequency, and start/end times of data are all different for each parameter, how to perform preprocessing of this data is also very important. This study also underwent various preprocessing steps considering these points, and detailed preprocessing processes are covered in Section 4: Preprocessing.

The M1553B data obtained in this way contains various variables, extreme environments, and noise that occur during actual aircraft operations, unlike simulation data, enabling realistic CBM+ model development. Additionally, having continuous time-series characteristics makes it suitable for detecting system state changes and fault symptoms. Furthermore, simultaneous data collection from multiple avionics equipment enables analysis of interactions and correlations between systems. Since faults in specific system equipment can actually affect data from other system equipment, analyzing correlations between different systems is essential.

3.2.3. Used Parameters

M1553B data stores communication data between almost all systems, so the types of stored data parameters are very numerous. Since approximately 2500 parameters are acquired, it is difficult to use all parameters, making the decision of which parameters to use from the M1553B data very important. Our study aims to build a sustainable CBM+ system, and, for that purpose, we examined how the chosen parameters could be linked to the data that are most reliably available from the FDR. In other words, we wanted the fault prediction model for avionics to rely only on parameters that can also be obtained from the FDR. The continuously recorded FDR data includes about 50 parameters. We excluded values that are internally computed by the FDR itself (e.g., the System Integration values) and kept only those that can be acquired from M1553B data. This left us with 20 parameters. The selected variables are mainly related to aircraft maneuver—altitude, acceleration, angular rates, control-surface positions, etc. Detailed parameter names can be found in Table 1. Hereafter in this paper, “MUX data” is defined and used as data extracted from the following parameters from the M1553B raw data.

Table 1. Used parameters in MUX data.

Used Parameters	
Weight on Wheels	Yaw Rate
Pressure Altitude	Rudder Ram Position
Mach Number	Right Horizontal Tail Ram Position
Calibrated Airspeed	Left Horizontal Tail Ram Position
Longitudinal Acceleration	Right Flaperon Ram Position
Lateral Acceleration	Left Flaperon Ram Position
Normal Acceleration	Platform Azimuth
Angle of Attack	Core Speed
Angle of Sideslip	Freestream Air Temperature
Roll Rate	Pitch Rate

3.2.4. Preprocessing

MUX data has different scales, units, and frequencies for each data type, making it unsuitable for AI model training, requiring preprocessing. This study performed the following preprocessing steps:

1. Nearest Interpolation

Since the parameters used in this study are recorded at different frequencies, a process to integrate frequencies is needed to combine all data for learning. Nearest interpolation was applied to parameters with frequencies lower than the highest Hz to perform missing value interpolation. This method maintains the continuity of time-series data and has the advantage of preventing abrupt value changes.

2. Noise Smoothing

MUX data collected in actual operational environments contains various noise data. Therefore, the Savitzky–Golay filter was applied to improve data quality and extract meaningful time-series features [31]. The Savitzky–Golay filter applies N-th order polynomials to adjacent data sections using the least squares method to smooth signals. Unlike a simple moving average application, it can effectively remove noise while maintaining important morphological characteristics such as data peak height and width, making it suitable for time-series data analysis where shape preservation is important. In this study, the window size was set to 11 and polyorder to 3 to prevent over-smoothing. Through graphs, it was confirmed that the data distribution did not change significantly, and peaks were preserved. A detailed comparative analysis is presented in Appendix A.

3. Scaling

Normalization was performed to integrate various parameters with different ranges and units for learning. This study experimented with both Standard Scaling and MinMax Scaling methods to compare performance. Standard Scaling transforms each feature to have a mean of 0 and a standard deviation of 1, effective when features follow a normal distribution and less sensitive to outliers. MinMax Scaling linearly transforms all values to between 0 and 1, not forcing data distribution, but it is very sensitive to outliers. If scaling were performed for each flight individually, unique maneuver characteristics such as altitude or speed for each flight would be lost. Therefore, global scaling was performed for both scaling methods to extract generalized features considering maneuvers of all flights. That is, global mean, std, or global min/max values were calculated by integrating all data to perform scaling for each flight.

4. Padding

For batch learning of deep learning models, all input data lengths must be equal. Since actual data can have different lengths for each session or flight section, the zero-padding technique was used to unify data lengths.

3.3. Maintenance Fault List

Maintenance Fault List (MFL) is flag data representing various aircraft faults, obtained from M1553B data. As mentioned earlier, some flight MFLs are available from the FDR, but to collect MFLs for all flights, extraction from M1553B data is necessary.

This study extracted MFLs from the acquired M1553B data and used the MFL that occurred in each flight as fault labels for the corresponding MUX data segments.

3.3.1. Target Subsystem

MFL shows which system and equipment generated and the fault description. Most avionics equipment faults can be identified, and since there are many types, which equipment's fault to use is also an important issue.

This study conducted research targeting faults in the FCR system's avionics equipment. The FCR is a core avionics system that detects and tracks air and ground targets and guides weapons. The FCR operates by transmitting electromagnetic waves and analyzing reflected signals to determine target range, velocity, and position. The FCR is typically located in the forward nose cone of the aircraft, where it serves both as a radome protecting the radar and fulfills aerodynamic and structural purposes of the airframe [32]. This location exposes the FCR to a unique operational environment. Fighter aircraft require high maneuverability, and significant structural loads are generated throughout the airframe during maneuvers [33]. In particular, the forward-mounted FCR directly experiences maneuver-induced loads as stresses generated by changes in aircraft attitude, acceleration, and angular velocity are directly transmitted [34], leading to the assessment that the probability of maneuver-related failures would be high. Furthermore, the FCR consists of various subordinate equipment, including antennas, transmitters, and receivers, and is the system with the most frequent LRU repairs and replacements, suggesting that CBM+ implementation would have significant effects on availability improvement and maintenance cost reduction [16]. Therefore, this study used faults occurring in the FCR, employing all faults from this subsystem without distinguishing between subordinate equipment faults. Faults occurring in this subsystem include communication channel errors between the antenna and processor, internal module errors in the radar processor, and others. It should be noted that specific fault descriptions cannot be disclosed in this paper due to military security classification.

3.3.2. Mapping with MUX

Since MUX data and MFL data are extracted separately from M1553B data for use, mapping between the used MUX data and MFL data is necessary. During the extraction process, data is parsed by communication packets, and each packet has different characteristics; not all data is recorded at the same time. In this study, MUX data was segmented into regular units to create sequences, and MFL data that occurred within the time period of those sequences was mapped and utilized. A detailed example can be found in Figure 2.

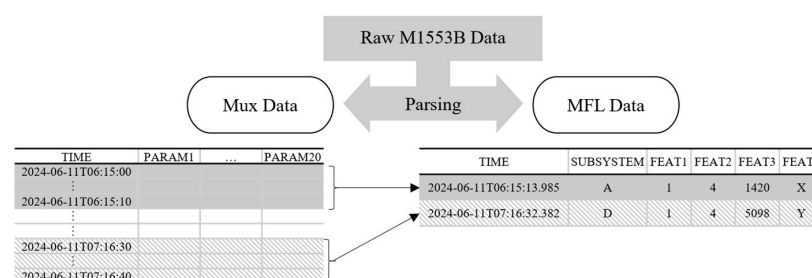


Figure 2. Mapping with MUX data and Maintenance Fault List data. Both datasets can be acquired from the M1553B data.

4. Feature Extraction

4.1. Model Architecture

Even after data preprocessing, using this data directly for AI model training presents difficulties. Meaningless data or noise can negatively affect AI model learning, and there are limitations in model architecture design and hardware resource utilization for learning vast amounts of data over long time periods. Therefore, extracting meaningful features from data must be performed first. Data feature extraction is one of the tasks of unsupervised learning, and the encoder–decoder structure is one of the most efficient strategies for this task. The encoder maps high-dimensional input data to low-dimensional latent representations, and the decoder reconstructs the original data from these compressed representations. CNN-LSTM encoders have been proven effective in CBM+ research across various industries, including wind power prediction research [35] and industrial process fault detection [36]. Therefore, this study employed the CNN-LSTM encoder approach to extract features from MUX data.

4.1.1. Encoder–Decoder Architecture

The learning method of the encoder–decoder structure involves compressing and then reconstructing actual data, with learning proceeding in the direction of minimizing the difference between the reconstructed data and the actual data. During training, the decoder is utilized for learning, but the encoder is only used for feature extraction after training completion. This study extracted 10 S maneuvers and represented them as single features. CNN extracts features from short-term data of approximately 1 s, and LSTM extracts features of approximately 10 s that reflect the time-series nature of the data. Figure 3 illustrates the overall encoder structure. Model training was conducted using Mean-Squared Error (MSE) Loss-based back-propagation learning. The loss function used is MSE Loss, which is most widely utilized in encoder–decoder structure model learning and helps stable convergence [37], and its equation is as follows:

$$\text{MSE Loss} = \frac{1}{N} \sum_{i=0}^{i=N} (y_i - \hat{y}_i)^2 \quad (1)$$

In Equation (1), N is the number of samples, y_i is the actual value of the i -th sample, and $\hat{y}_i$ is the predicted value that the decoder reconstructed by an extracted feature from y_i . The encoder–decoder-specific architecture was optimized through various experiments based on validation performance.

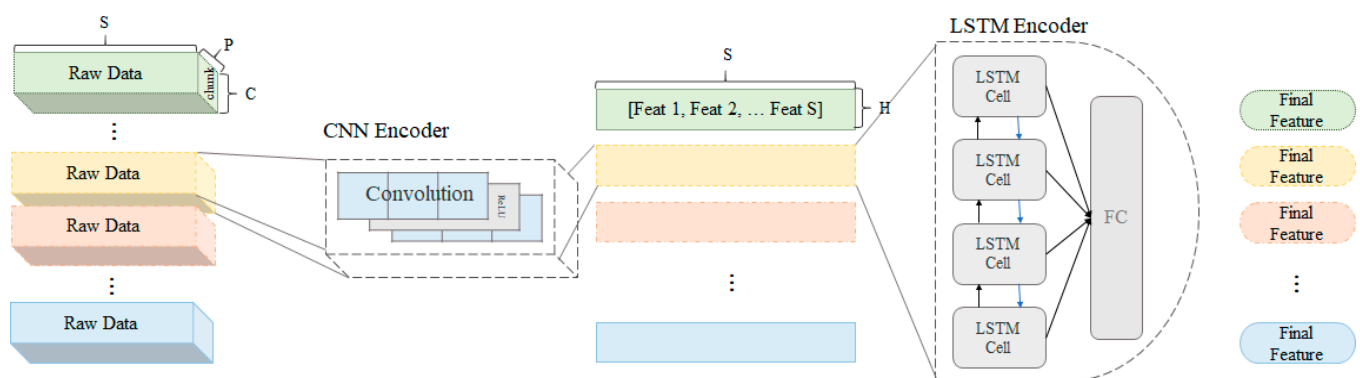


Figure 3. Overall feature extraction. Here, S = sequence length, C = chunk length, P = the number of parameters, and H = hidden dimension.

4.1.2. CNN

CNN is a model that effectively extracts local features by introducing convolution operations to the basic feed-forward neural network structure. This study utilized 1D CNN to capture short-term patterns or changes in data on the same time-series axis. Since kernel parameters are reused across all data, learning occurs faster due to fewer parameters compared to other model structures. The CNN encoder consists of two convolutional layers and ReLU layers between them. Convolutional layers extract local features of data. Since it is difficult to find time-series characteristics in approximately 1 s data, the model was configured so that time-series characteristics are not reflected in this layer. Additionally, to preserve data features as much as possible, pooling layers were not used, and ReLU activation functions were placed between convolutional layers to increase nonlinearity. The data input to CNN is configured in the shape $[N, S, T, P]$, where T is a parameter that determines how many rows the CNN encoder will compress in total. In this study, T was set to the max frequency of the used data to enable the extraction of approximately 1 s features from the data. P is the number of input parameters, and this study utilized a total of 20 parameters. S is a dimension created to extract features of longer sequences in the LSTM layer, and learning in this encoder proceeds regardless of the S dimension. That is, $[S, T, P]$ represents data for one section, and N indicates how many such data sections exist. The output data is converted to $[N, S, H]$, where H is the output dimension of the CNN encoder.

4.1.3. LSTM

Subsequently, an LSTM-based encoder extracts data features of longer sequences. LSTM can preserve important information over long periods by maintaining two internal states called 'cell state' and 'hidden state'. Each LSTM cell controls how information is stored, deleted, and output occurs through three gates (Forget, Input, and Output Gate). The Forget gate decides what information from the previous time step to discard, the Input gate selects how much new information from the current input to reflect, and the Output gate determines what information to pass to the next stage. This structure enables the LSTM model to effectively learn temporal relationships and patterns even in long and complex time-series data. Through this, the LSTM encoder identifies relationships and internal information within each sequence and compresses them into a single hidden state. A bidirectional technique was used that combines LSTM learning in the forward direction with LSTM learning in the reverse direction to learn relationships in both directions. The final hidden state output by the last cell becomes a feature representing the corresponding sequence through a fully connected layer. The data input to LSTM is configured in the shape $[N, S, H]$ that has passed through the CNN encoder, where S is the length of the sequence (i.e., how many seconds), and H is the output dimension of the CNN encoder. In this study, S was set to 10 to enable the extraction of features for approximately 10 s. The LSTM output becomes $[N, L]$, where L is the final output dimension of the LSTM encoder. This L = feature dim and is a feature representing the sequence and can be viewed as a vector containing meaningful features of 10 s of data.

4.2. Experiment

4.2.1. Data Integration

Since the goal of the feature extraction process is to extract meaningful features that well represent the data, learning was conducted using all aircraft to enable the extraction of general data features. If learning were conducted for each aircraft individually, it would be difficult to extract only general features because learning would include specific characteristics of each aircraft, such as missions performed and flight locations. If aircraft-

specific characteristics were reflected from the feature extraction stage, they would also affect the subsequent fault prediction stage, so we aimed to keep each process independent. Therefore, encoder learning was conducted after combining data from all aircraft. Both MinMax Scaling and Standard Scaling data were trained separately using encoders with the same architecture. To improve generalization performance, K-fold validation was conducted. Training was divided into 5-fold, and it was confirmed that the valid loss decreased in each fold.

4.2.2. Hyperparameters

The learning rate started at 0.001 and then decreased at regular intervals according to the step scheduler [38], with step = 3 and gamma = 0.9 set to enable smooth convergence of the model to optimal values. Early stopping was used to prevent model overfitting, and learning was terminated after approximately 30 epochs in all training sessions. AdamW was used as the optimizer. Other specific hyperparameters can be found in Table 2. These hyperparameters were determined through multiple experiments, with the learning rate found to be particularly critical. When the learning rate was set too high (0.01) or too low (0.0001), the loss failed to decrease, maintaining values around 0.02 for MinMax Scaling and 0.6–0.9 for Standard Scaling. All experiments were conducted using a single RTX A6000 GPU, with approximately 45–50 s per epoch, resulting in a total training time of about 30 min.

Table 2. Hyperparameter settings.

Hyperparameters		Value
batch		256
Learning Rate Scheduler	Init	0.001
	Step	3
	Gamma	0.9
Early Stopping	Threshold	0.001
	Patience	10

4.3. Result

The final MSE Loss of encoder learning showed results of MinMax Scale 4.18×10^{-5} and Standard Scale 0.021. Since all data exists in the [0, 1] range for MinMax Scale, the loss is very close to 0, and for Standard Scale, since data exists in approximately the [−3, 3] range following a normal distribution with mean 0 and standard deviation 1, a value of 0.02 appears to be quite a small loss. In the loss progression shown in Figure 4a, both training and validation losses decrease evenly, indicating that the model learned robustly without overfitting.

An interesting finding was that increasing model size by expanding CNN hidden dimensions or LSTM layers actually resulted in higher loss. As shown in Figure 4a, with a single LSTM layer, training proceeded well as described above, but when using three LSTM layers in a multi-layer LSTM configuration, the loss failed to decrease. This is attributed to insufficient feature diversity in the data and limited data quantity and distribution. This demonstrates that for reflecting temporal patterns in sequences of approximately 10 s, a single LSTM layer is appropriate, and increasing model size prevents proper learning. Based on these experiments, we selected an appropriate model size that achieved both computational efficiency and optimal MSE Loss. Various other values were tested, and the specific architecture that achieved the best MSE Loss can be found in Figure 4b.

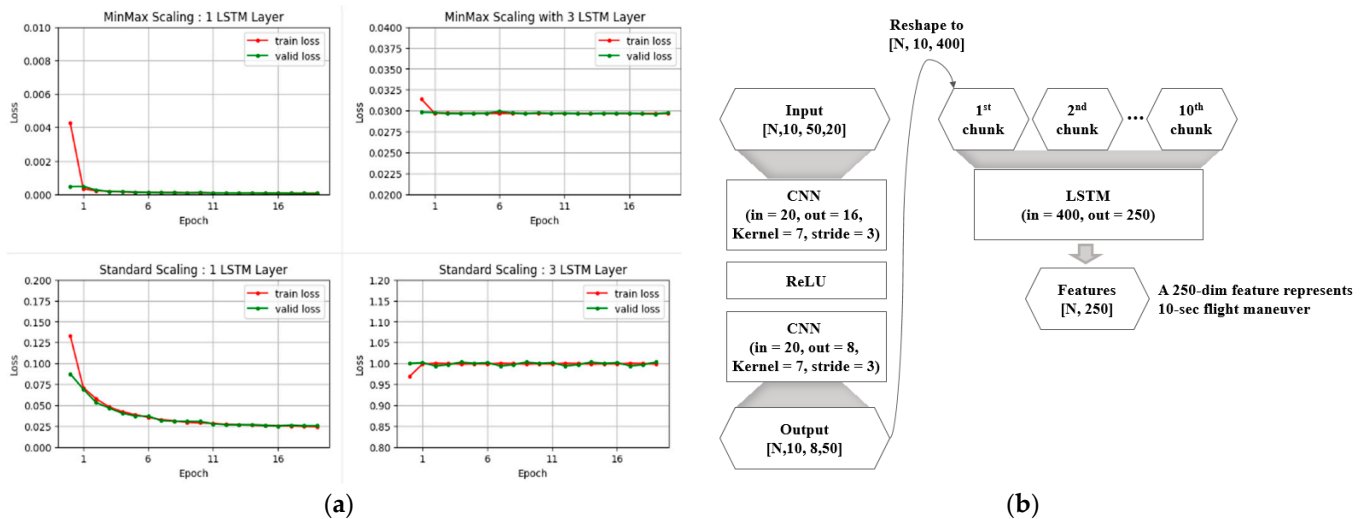


Figure 4. It shows the result of feature extraction. (a) shows a loss graph with different combinations. Only one LSTM layer shows the best performance. (b) presents the optimal CNN-LSTM encoder architecture, which was identified through extensive experimental evaluation.

5. Fault Prediction

The fault prediction stage of this study is a core process that predicts FCR faults using maneuver features extracted from the feature extraction stage. This stage analyzes the impact of aircraft maneuvers on FCR faults and verifies prediction performance in actual operational environments.

5.1. Data Preparation

We approached FCR fault prediction as a binary classification problem using data features extracted from the feature extraction stage. Specifically, this study analyzed the optimal maneuver duration that most influences fault occurrence with maneuver-related data features.

5.1.1. Window Strategy

First, this study used the window strategy to determine the optimal time window length for fault prediction by analyzing how different durations of historical maneuver data influence fault occurrence. At a given time point t , a maneuver feature sequence $\{f_1, f_2, \dots, f_w\}$ of window size length is input to predict the probability $P(\text{fault} | f_1, f_2, \dots, f_w)$ of fault occurrence in the last 10 s section. Here, each f_i is a feature vector compressing 10 s of maneuvers. For example, when the window size is 3, a total of 30 s ($10 \text{ s} \times 3$) of maneuver data is observed to predict whether a fault will occur in the last 10 s section. This approach is designed to capture changes in maneuver patterns before fault occurrence. The core of the window strategy is based on the assumption that faults do not occur immediately but appear as specific maneuver patterns accumulate.

5.1.2. Data Labeling

The label for binary classification was set as the fault occurrence status in the corresponding section through MUX and MFL mapping. Each flight is divided into fault-free flights and flights with faults.

In fault-free flights, all maneuvers were extracted and defined as normal maneuvers (labeled as 0). Conversely, flights with faults are divided into abnormal maneuvers (labeled as 1) and unlabeled maneuvers. Abnormal maneuvers are defined as maneuver sections including the time point when the fault occurred and preceded by the window size, while

other sections remain unlabeled. That is, all unlabeled maneuvers are not used for training and are discarded. This is because it is difficult to identify specifically what duration of maneuvers before a fault in a flight with faults should be considered normal maneuvers, making it difficult to determine what duration before the fault should be viewed as normal maneuvers. The specific labeling process is as follows:

- Normal label (0): When no fault occurs during the corresponding section.
- Abnormal label (1): When a fault occurs during the last 10 s of the corresponding section.

Figure 5 provides an easier understanding of the window and labeling process.

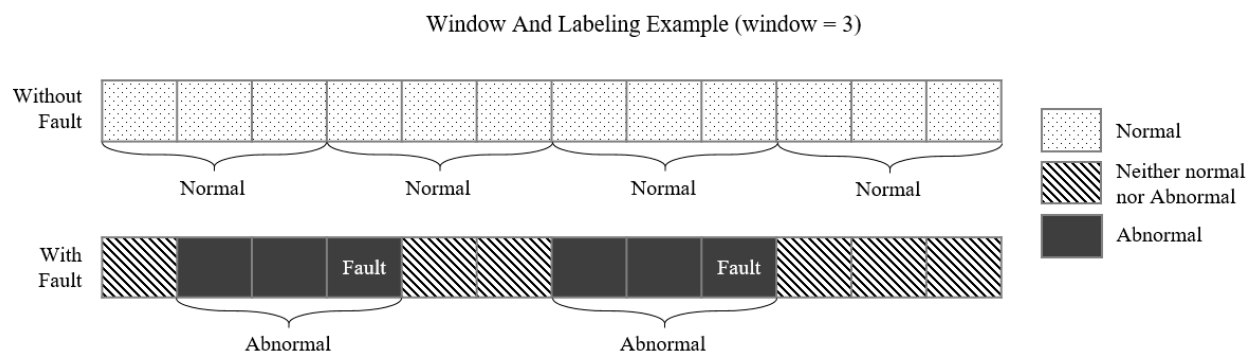


Figure 5. Window and labeling strategy: a normal maneuver can be extracted only from flight without fault. An abnormal maneuver must contain the sequence with the fault.

5.1.3. Data Imbalance

Since fault occurrence frequency during actual operations is not high, the data imbalance problem is severe. Among the aircraft from which data was collected, some had no faults at all, while others had up to 125 fault occurrences. Therefore, experiments were conducted only on aircraft with more than 45 fault occurrences. Aircraft that did not meet this criterion were excluded from predictive model construction due to the high possibility of causing biased results, such as overfitting due to insufficient abnormal training data. Experiments were performed on a total of four aircraft; flight counts and abnormal data counts per aircraft can be found in Table 3. Aircraft B and D had many flights, while Aircraft A and C had fewer flights. This demonstrates difficulties in data collection due to the cumbersome data acquisition process. Better results could be expected by automating the data collection process or securing longer-term data in the future. Nevertheless, due to severe data imbalance, the number of normal data used for training was randomly sampled to 500.

Table 3. The number of abnormal cases per craft (when the window size is 1).

Aircraft	Flight	Abnormal Case
A	15	60
B	82	90
C	12	59
D	62	125

5.2. Experiment

5.2.1. Prediction Model

Three basic classification models were selected for binary classification: Logistic Regression (LR), Multi-Layer Perceptron (MLP), and Random Forest (RF). Each model has different learning principles and characteristics, enabling analysis of maneuver-fault relationships from various perspectives. Logistic Regression provides interpretable results

through linear decision boundaries, while MLP has the capability to learn nonlinear patterns. Random Forest provides stable predictions through ensemble methods. Each model was implemented using the default parameters provided by the Scikit-learn libraries. The reason for not performing special hyperparameter tuning is that the purpose of this study is to demonstrate the correlation between aircraft maneuvers and faults, focusing on verifying the effectiveness of the proposed feature extraction method rather than model optimization.

5.2.2. Individual Aircraft Training

This study adopted a strategy of training independent models for each aircraft, as FCR fault characteristics can appear differently for each individual aircraft. Research on reliability analysis of systems operating in dynamic environments has reported that system degradation characteristics can appear differently for each individual depending on environmental and operational conditions [39]. Research on aircraft fleet management has also mentioned that even aircraft of the same model can have different maintenance characteristics and fault patterns for each individual aircraft [40].

5.2.3. Evaluation Metric

Experimental result evaluation was conducted based on Fault Recall. The model's output value is the probability of fault occurrence in the corresponding section, and if the probability is greater than 50%, it is classified as having a fault, i.e., an abnormal maneuver. Fault Recall represents the proportion of actual faults that were correctly identified by the model among all fault occurrences, and this metric was used as the primary evaluation criterion because it is important not to miss faults from an aircraft safety perspective. Fault Recall is expressed by (2), where TP (True Positive) is the case of correctly predicting a fault as a fault, and FN (False Negative) is the case of incorrectly predicting a fault as normal.

$$\text{Fault Recall} = \frac{TP}{TP + FN} \quad (2)$$

In aviation safety, Type II Error (missing faults) is much more dangerous than Type I Error (incorrectly judging normal as fault), so models were evaluated based on Fault Recall. Additionally, Normal Recall was also checked to verify that the model was not overfitting only to fault maneuvers. Three-fold cross-validation was applied within each aircraft to prevent overfitting and improve model generalization performance, and final performance evaluation was conducted by averaging the results of the three folds. While Fault Recall was the primary evaluation focus, to verify that the model was not overfitted to Fault Positives (cases where no fault occurred but was predicted as a fault), Accuracy, Precision, and F1 Score were additionally examined.

5.2.4. Undersampling Validation

To verify the possibility of information loss due to normal data undersampling, two validation experiments were designed. First, we compare the Fault Recall of models trained with 500 versus 1000 Normal Samples to evaluate the impact of sample size reduction on prediction performance. Second, we verify the distributional similarity between the complete normal dataset and the undersampled dataset using Kernel Density Estimation (KDE). Through these experiments, we aimed to confirm whether 500 Normal Samples sufficiently represent the characteristics of the original data.

5.3. Result

5.3.1. Scaling and Model Comparison Result

The Fault Recall performance results by aircraft and model are presented in Figure 6a for Standard Scaling data with a window size of 2, representing the optimal configuration

identified in this study. In that result, all classification models showed equally good performance, with the MLP model showing the best performance. Among these, the combination of Standard Scaling and MLP showed the best performance overall for most aircraft. However, the Logistic Regression model showed relatively lower performance compared to other models, which appears to be because it is difficult to find associations between maneuvers and faults in classification methods based on linear relationships. Particularly, since nonlinear relationships had already been extracted through architectures like CNN and LSTM in the maneuver feature extraction process, the performance of Logistic Regression was likely even worse. On the other hand, MLP demonstrated superior performance when combined with Standard Scaling, as the model's nonlinear learning capability synergized well with the normalization effects. Fault Recall achieved an average of 91% with a maximum of 98%, which is excellent performance considering that actual data was used.

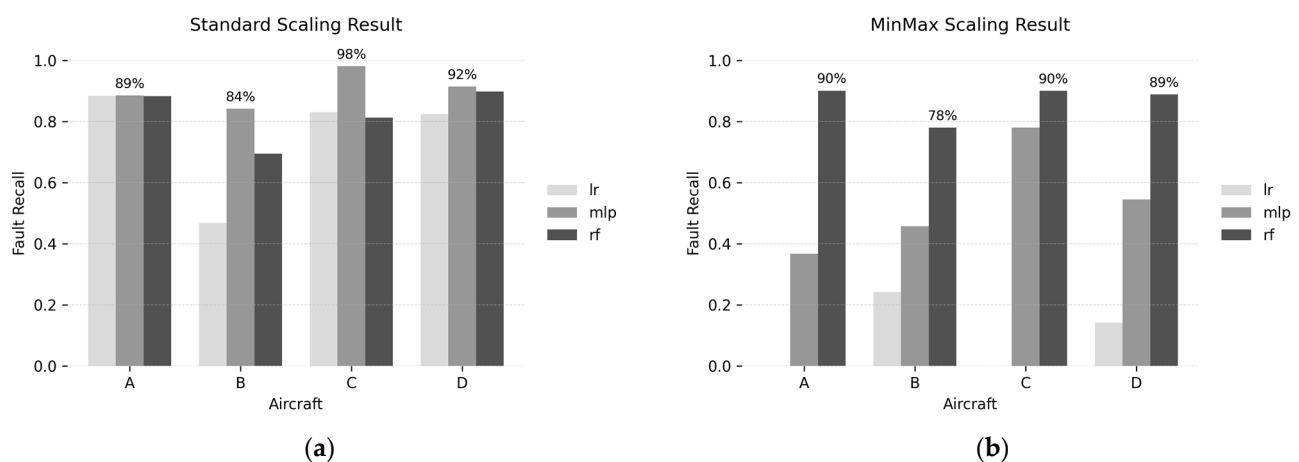


Figure 6. Fault Recall performance graph per scaling and the prediction model. (a,b) present the Fault Recall per aircraft, both obtained with a fixed window size of 2. (a) corresponds to Standard-scaled data, whereas (b) uses MinMax-scaled data. For Craft C, the Standard-scaled data achieves a maximum Fault Recall of 98%.

Conversely, for MinMax Scaling data, Random Forest models showed good performance, but the performance of MLP and Logistic Regression was relatively poor. This appears to be because MinMax Scaling compressed to the [0, 1] range, causing loss of feature variance information and negatively affecting neural network learning. Additionally, performing global scaling on all aircraft data likely caused specific outliers appearing only in certain flights to significantly affect scaling, greatly distorting the data distribution. Consequently, discontinuous-based models like Random Forest showed good performance, but Logistic Regression and MLP models showed significantly degraded performance due to the influence of such outliers. The results for MinMax Scaling data are presented in Figure 6b.

5.3.2. Window Size Effect Result

Analyzing the results by window size, there were no significant performance differences by window size overall, but window 2 showed the best performance. Specific performance by window size is shown in Figure 7a. It shows the average Fault Recall values for all window sizes combined, classified by the MLP model with Standard Scaling data. Therefore, aircraft maneuvers influence FCR fault occurrence, but approximately 10 s of maneuvers appear to be sufficient to affect FCR equipment. It seems difficult to conclude that particularly long-duration maneuvers have more influence on fault occurrence than short-duration ones.

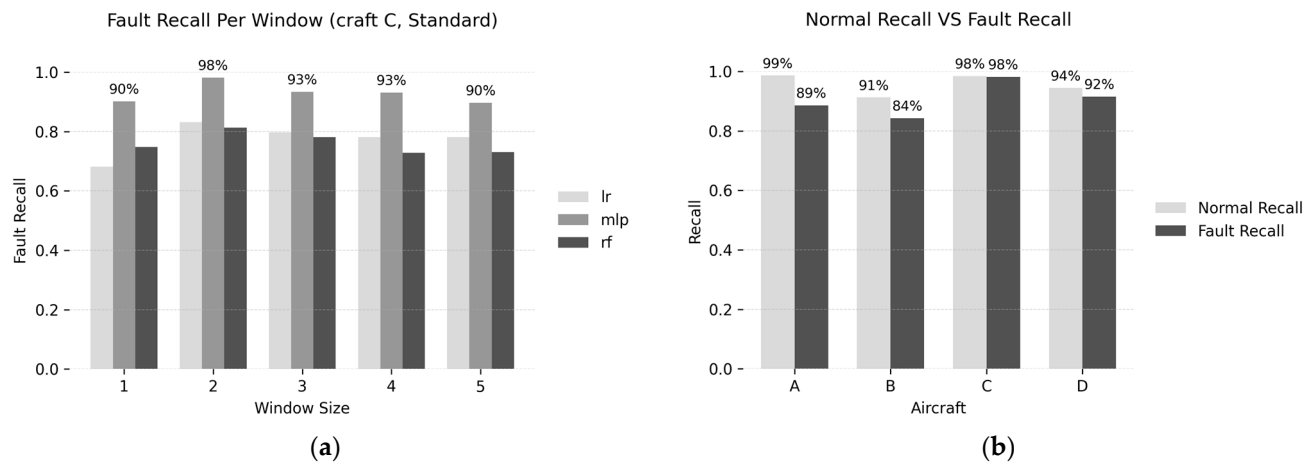


Figure 7. Validation of prediction performance: (a) Fault Recall across different window sizes. It shows optimal performance at window size 2, and (b) comparison of Normal Recall versus Fault Recall across all aircraft. It demonstrates balanced performance with near-perfect Normal Recall and high Fault Recall, confirming no overfitting to fault cases.

5.3.3. Overfitting Validation Result

Additionally, comparing the distributions of Normal Recall and Fault Recall shows that learning was not over-fitted to fault occurrence. Through this, it was demonstrated that there are likely specific maneuver features for each aircraft that cause FCR faults. In Figure 7b, showing Normal Recall and Fault Recall for each craft when using the MLP model, window 2 showed the best performance, and Normal Recall is confirmed to be close to nearly 100%. As shown in Table 4, not only Fault Recall but also other evaluation metrics like Accuracy and Precision recorded high scores, meaning this methodology learned well to predict only actual faults without overfitting to fault positives. This proves it is reasonable to view that each aircraft has specific maneuver features that cause FCR faults.

Table 4. Performance comparison by scaling method and predictor for each aircraft. In addition to Fault Recall, metrics such as Accuracy, Precision, F1 Score, Recall, and Normal Recall are presented. The combination of Standard Scaling and the MLP predictor demonstrates the best overall performance.

Scaling	Craft	Predictor	Fault Recall	Accuracy	Precision	F1 Score	Recall	Normal Recall
Standard	A	lr	0.885	0.964	0.897	0.912	0.929	0.974
		mlp	0.886	0.975	0.936	0.935	0.936	0.986
		rf	0.884	0.984	0.976	0.957	0.94	0.996
	B	lr	0.468	0.875	0.748	0.72	0.707	0.946
		mlp	0.842	0.9	0.801	0.827	0.877	0.912
		rf	0.695	0.927	0.874	0.843	0.832	0.97
	C	lr	0.831	0.979	0.973	0.94	0.913	0.996
		mlp	0.981	0.984	0.945	0.961	0.983	0.984
		rf	0.813	0.979	0.979	0.938	0.906	0.998
	D	lr	0.825	0.923	0.877	0.881	0.886	0.948
		mlp	0.915	0.938	0.892	0.907	0.93	0.944
		rf	0.899	0.96	0.939	0.937	0.937	0.976

Table 4. Cont.

Scaling	Craft	Predictor	Fault Recall	Accuracy	Precision	F1 Score	Recall	Normal Recall
MinMax	A	lr	0.000	0.891	0.446	0.471	0.499	0.998
		mlp	0.367	0.900	0.623	0.631	0.665	0.962
		rf	0.902	0.989	0.994	0.971	0.951	1.000
	B	lr	0.242	0.859	0.795	0.615	0.606	0.970
		mlp	0.458	0.886	0.784	0.734	0.710	0.962
		rf	0.780	0.946	0.912	0.889	0.878	0.976
	C	lr	0.000	0.894	0.447	0.472	0.500	1.000
		mlp	0.781	0.966	0.929	0.905	0.884	0.988
		rf	0.901	0.987	0.985	0.966	0.949	0.998
	D	lr	0.142	0.827	0.898	0.569	0.570	0.998
		mlp	0.545	0.864	0.705	0.713	0.743	0.941
		rf	0.889	0.963	0.950	0.942	0.935	0.982

5.3.4. Undersampling Validation Result

To validate potential information loss from undersampling normal data, the following verification experiments were conducted. The experimental results validating the appropriateness of normal data undersampling are as follows. As shown in Figure 8a, the comparison of model performance using 500 versus 1000 Normal Samples demonstrates that Fault Recall remains consistently high without significant degradation. Across all models (LR, MLP, RF), the difference in Fault Recall between the two conditions was less than 5%, confirming that 500 Normal Samples contain sufficient information for fault detection without performance loss. The KDE distribution analysis in Figure 8b reveals that the data distributions before and after undersampling are nearly indistinguishable, with curves overlapping closely. This indicates that the distributional characteristics of normal data were well preserved despite random undersampling. These results demonstrate that the undersampling strategy adopted in this study effectively addresses the data imbalance problem while minimizing information loss.

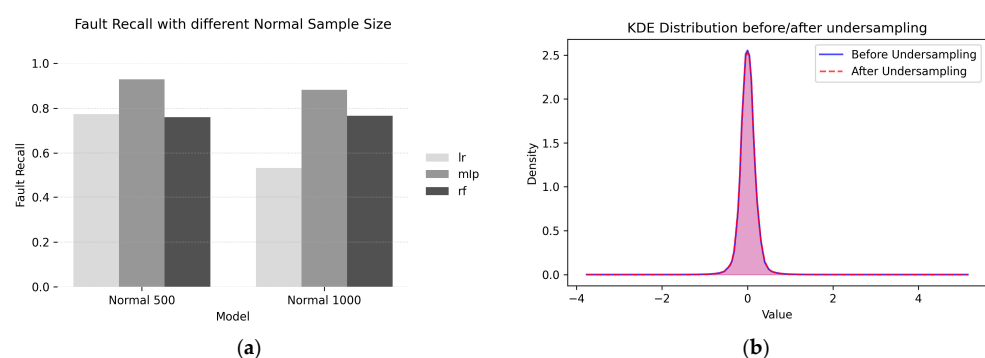


Figure 8. Validation results for undersampling. (a) Fault Recall versus the number of undersampled instances, indicating that performance remains largely unaffected by the degree of undersampling. (b) KDE distributions before and after undersampling, showing that the two data distributions are nearly indistinguishable.

5.3.5. Summary and Discussion

These results show that there is a meaningful correlation between maneuver data and faults. All aircraft show quite high performance, and slight differences appear to reflect some external factors, such as differences in data acquisition rates by aircraft and

aircraft-specific characteristics. Furthermore, the experimental results show that Standard Scaling is effective for preprocessing data to use maneuver data regardless of data outliers, and that there are nonlinear relationships between maneuvers and faults. A limitation of this study is that specific characteristics of maneuvers with high fault-causing probability cannot be disclosed due to military data security reasons. However, this study is significant in proving that a meaningful correlation exists between maneuver data and faults.

6. Conclusions

This study demonstrated that there is a meaningful correlation between aircraft maneuvers and FCR equipment faults, and verified the applicability of CBM+ systems based on this finding. By using actual operational data rather than simulation data, the possibility of application to real environments was enhanced. Additionally, differentiated from existing CBM+ systems that collect data by attaching new sensors to target equipment, a more realistic approach was proposed that utilizes existing data without additional sensor installation.

The proposed methodology consists of a feature extraction stage using CNN-LSTM encoders and a machine learning-based fault prediction stage. Through an architecture combining CNN, which is effective for short-term feature extraction of complex maneuver data, with LSTM, which can reflect time-series characteristics, approximately 10 s of complex maneuver data was effectively compressed and represented as meaningful features. Through experiments with various data preprocessing methods and prediction models, an MLP model trained on Standard Scaled data achieved an average Fault Recall of 91% with a maximum of 98%, fully demonstrating the correlation between aircraft maneuvers and FCR faults. This study has limitations in that it used data from a limited number of aircraft of a specific type and focused only on FCR systems. Additionally, more diverse operational environments and long-term data validation are required for actual field application.

Future research will first analyze the extracted features to identify which specific maneuvers they represent and determine the maneuvers that impose stress on the equipment, and based on this understanding, develop algorithms capable of predicting FCR faults using FDR data alone, without relying on MUX data, thereby establishing a sustainable CBM+ system. Building upon this foundation, the methodology can extend beyond binary fault classification to quantify maneuver-induced stress, estimate equipment degradation trajectories, and predict RUL to optimize maintenance scheduling. Furthermore, the maneuver-based fault prediction approach proposed in this study can be extended to other avionics systems, enabling comprehensive CBM+ implementation across avionics equipment. In the long term, through processes such as model refinement and stability verification, this approach can evolve into a real-time monitoring system that analyzes in-flight maneuver patterns to continuously monitor avionics equipment stress.

Ultimately, this research represents an important first step contributing to the construction of data-based FCR CBM+ systems, and the development of such technology, which can improve the safety, economic efficiency, and effectiveness of aircraft operations, is essential for sustainable growth in the future aviation industry.

Author Contributions: Conceptualization, M.K. and I.L.; methodology, M.K.; software, M.K.; validation, M.K., I.L., S.-H.J. and D.A.; formal analysis, M.K.; investigation, M.K. and I.L.; resources, B.S.; data curation, I.L.; writing—original draft preparation, M.K.; writing—review and editing, I.L. and S.-H.J.; visualization, M.K.; supervision, S.-H.J.; project administration, S.-H.J.; funding acquisition, B.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Korea Research Institute for Defense Technology Planning and Advancement (KRIT)— a grant funded by the Defense Acquisition Program Administration (DAPA) (KRIT-CT-22-081, Weapon System CBM+ Research Center).

Data Availability Statement: Access to the data is granted upon reasonable request to the corresponding author.

Conflicts of Interest: Authors Minyoung Kim, Ikgyu Lee, Seon-Ho Jeong and Byoungserb Shim were employed by the company Korea Aerospace Industries, Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AA	Aircraft Availability
ALIS	Autonomic Logistics Information System
BC	Bus Controller
CBM+	Condition Maintenance Management Plus
CI	Condition Indicators
CNN	Convolutional Neural Network
DoD	Department of Defense
FCR	Fire Control Radar
FDR	Flight Data Recorder
FN	False Negative
FP	False Positive
HUMS	Health and Usage Monitoring Systems
KDE	Kernel Density Estimation
LR	Logistic Regression
LRU	Line Replaceable Unit
LSTM	Long-Short Term Memory
MFL	Maintenance Fault List
MLP	Multi-Layer Perceptron
MSE	Mean-Squared Error
NMCM	Non-Mission Capable Maintenance
NMCMU	Non-Mission Capable Maintenance Unscheduled
ODIN	Operational Data Integrated Network
PHM	Prognostic Health Management
RF	Random Forest
RT	Remote Terminal
RUL	Remaining Useful Life
SI	System Integration
TR	True Positive

Appendix A

To examine whether the smoothing process caused over-smoothing, an experiment was conducted to compare the data distributions before and after applying the smoothing filter used in this study. Figure A1 compares the time series data of some randomly selected parameters from one flight data used in this study, before and after applying the Savitzky–Golay filter. Smoothing was performed after scaling, and the original data is also the scaled original data. Each subplot overlays the original data and smoothed data for the entire flight section. Analyzing the smoothing results, the main trends and variation patterns of the data match the original across all parameters, confirming that essential information has been preserved. On the other hand, abrupt spike-type outliers caused by sensor measurement

errors or communication interference have been effectively removed. This is because the Savitzky–Golay filter uses local polynomial approximation to preserve the morphological characteristics of the signal better than simple averaging filters. Consequently, it was confirmed that the applied filter parameters (window length = 11, polynomial order = 3) are appropriate settings that effectively remove noise without excessive smoothing.

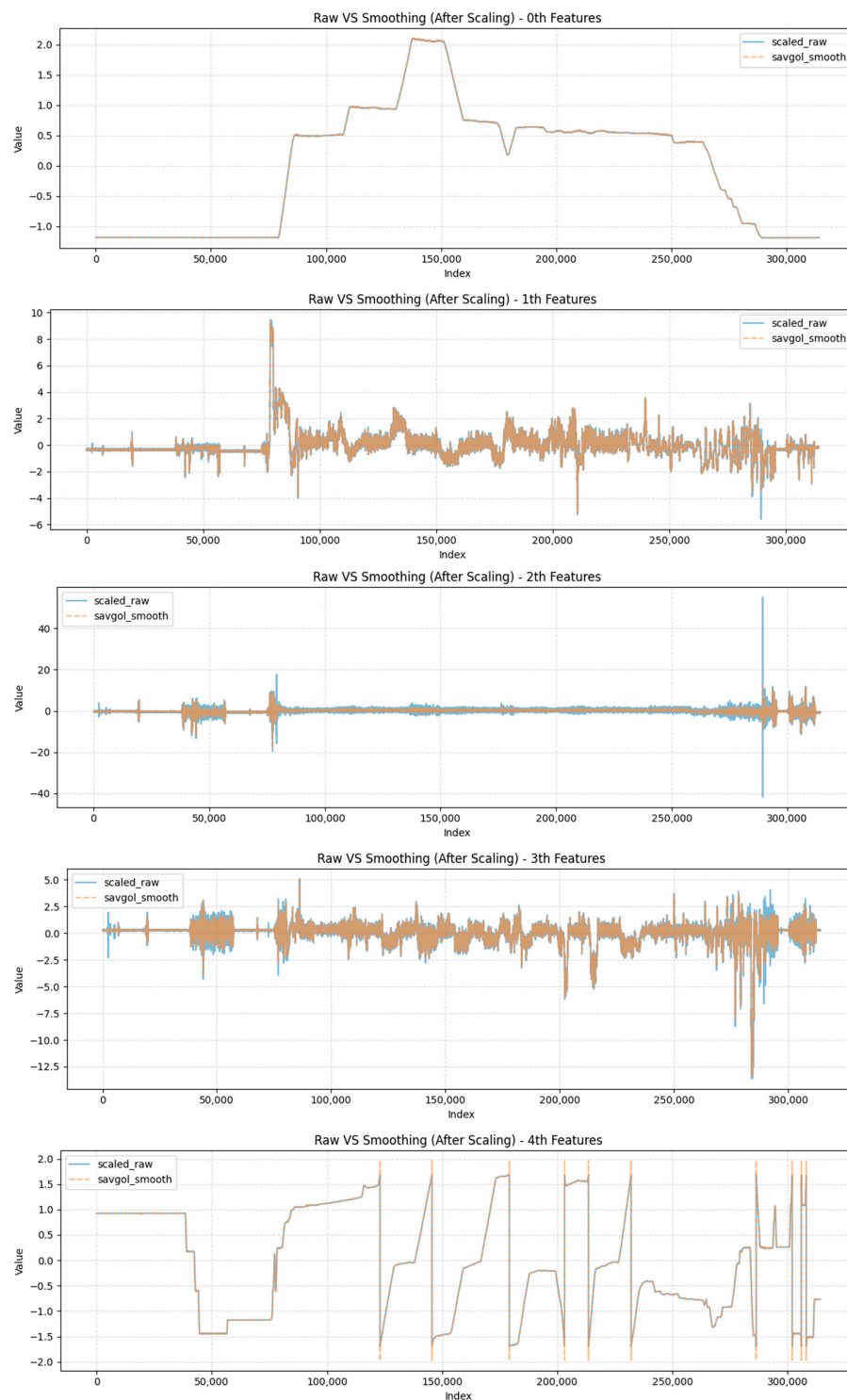


Figure A1. Comparison of scaled raw data and smoothing data for each parameter. Comparison of time-series data before and after applying the Savitzky–Golay filter for several randomly selected parameters from one flight. Smoothed signals are overlaid with the scaled original data. Subplot titles are shown as arbitrary column numbers (Col 1, Col 2, etc.) due to security restrictions. The results confirm that the main trends were preserved while spike-type outliers were effectively removed.

References

1. Ritschel, J.; Ritschel, T.; York, N. Providing a Piece of the Puzzle: Insights into the Aircraft Availability Conundrum. *J. Def. Anal. Logist.* **2019**, *3*, 29–40. [\[CrossRef\]](#)
2. U.S. Army. *Aeronautical Design Standard: Handbook for Condition Based Maintenance Systems for US Army Aircraft Systems (ADS-79D-HDBK)*; U.S. Army: Washington, DC, USA, 2013.
3. U.S. Department of Defense. *DoD Directive 4151.18: Maintenance of Military Materiel*; U.S. Department of Defense: Washington, DC, USA, 2004.
4. U.S. Department of Defense. *DoD Instruction 4151.22: Condition-Based Maintenance Plus for Materiel Maintenance*; U.S. Department of Defense: Washington, DC, USA, 2018.
5. U.S. Department of Defense. *DoD Condition-Based Maintenance Plus (CBM+) Guidebook*; Defense Acquisition University: Washington, DC, USA, 2024.
6. U.S. Air Force. CBM Redefines Aircraft Maintenance. Available online: <https://www.af.mil/News/Article-Display/Article/2438612/cbm-redefines-aircraft-maintenance> (accessed on 15 December 2024).
7. Army Aviation Magazine. Why Do We Need CBM? Available online: <https://armyaviationmagazine.com/why-do-we-need-cbm/> (accessed on 15 December 2024).
8. F-35 Official Site. Sustainment. Available online: <https://www.f35.com/f35/about/sustainment.html> (accessed on 10 April 2025).
9. Li, Y.; Peng, S.; Li, Y.; Jiang, W. A Review of Condition-Based Maintenance: Its Prognostic and Operational Aspects. *Front. Eng. Manag.* **2020**, *7*, 323–334. [\[CrossRef\]](#)
10. Wang, H.; Zhong, D.; Zhao, T. Avionics System Failure Analysis and Verification Based on Model Checking. *Eng. Fail. Anal.* **2019**, *105*, 373–385. [\[CrossRef\]](#)
11. Fioriti, M.; Vercella, V.; Viola, N. Cost-Estimating Model for Aircraft Maintenance. *J. Aircr.* **2018**, *55*, 1564–1575. [\[CrossRef\]](#)
12. Susinni, G.; Rizzo, S.A.; Iannuzzo, F. Two Decades of Condition Monitoring Methods for Power Devices. *Electronics* **2021**, *10*, 683. [\[CrossRef\]](#)
13. Yang, S.; Xiang, D.; Bryant, A.; Mawby, P.; Ran, L.; Tavner, P. Condition Monitoring for Device Reliability in Power Electronic Converters: A Review. *IEEE Trans. Power Electron.* **2010**, *25*, 2734–2752. [\[CrossRef\]](#)
14. Barakat, E.; Sinno, N.; Keyrouz, C. A Remote Monitoring System for Voltage, Current, Power and Temperature Measurements. *Phys. Procedia* **2014**, *55*, 421–428. [\[CrossRef\]](#)
15. El Mir, H.; King, S.; Skote, M.; Perinanayagam, S. Machine Learning Requirements for the Airworthiness of Structural Health Monitoring Systems in Aircraft. In Proceedings of the 38th Conference and 31st Symposium of the International Committee on Aeronautical Fatigue and Structural Integrity (ICAF 2023), Delft, The Netherlands, 26–29 June 2023.
16. Lee, I.; Jeong, S.; Shim, B. Analysis of Maintenance Records by System and Equipment for CBM+ Implementation. In Proceedings of the KSAS 2025 Spring Conference, Jeju, Republic of Korea, 2–4 April 2025.
17. Lei, Y.; Li, N.; Guo, L.; Li, N.; Yan, T.; Lin, J. Machinery Health Prognostics: A Systematic Review from Data Acquisition to RUL Prediction. *Mech. Syst. Signal Process.* **2018**, *104*, 799–834. [\[CrossRef\]](#)
18. Zhang, W.; Yang, D.; Wang, H. Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey. *IEEE Syst. J.* **2019**, *13*, 2213–2227. [\[CrossRef\]](#)
19. Parashare, G.R. Data-Driven Process Optimization Using AI and Statistical Methods in High-Tech Manufacturing. *Int. J. Sci. Res. Manag.* **2024**, *12*, 1854–1874. [\[CrossRef\]](#)
20. Stanton, I.; Munir, K.; Ikram, A.; El-Bakry, M. Predictive Maintenance Analytics and Implementation for Aircraft: Challenges and Opportunities. *Syst. Eng.* **2023**, *26*, 216–237. [\[CrossRef\]](#)
21. Chen, C.; Shi, J.; Lu, N.; Zhu, G.; Jiang, B. Data-Driven Predictive Maintenance Strategy Considering the Uncertainty in Remaining Useful Life Prediction. *Neurocomputing* **2022**, *494*, 618–631. [\[CrossRef\]](#)
22. Fitrayudha, A.; Sastra, K. Predictive Maintenance for Aircraft Engine Using Machine Learning: Trends and Challenges. *AVIA Int. J. Aviat. Sci. Eng.* **2021**, *3*, 37–44. [\[CrossRef\]](#)
23. Kosova, F.; Altay, O.; Ünver, H.O. Structural Health Monitoring in Aviation: A Comprehensive Review and Future Directions for Machine Learning. *Nondestruct. Test. Eval.* **2024**, *40*, 1–60. [\[CrossRef\]](#)
24. Fu, S.; Avdelidis, N.P. Prognostic and Health Management of Critical Aircraft Systems and Components: An Overview. *Sensors* **2023**, *23*, 8124. [\[CrossRef\]](#) [\[PubMed\]](#)
25. Skaf, Z.A. A Rare Failure Detection Model for Aircraft Predictive Maintenance Using a Deep Hybrid Learning Approach. *Neural Comput. Appl.* **2022**, *34*, 12415–12428. [\[CrossRef\]](#)
26. Bharadwaj, R.; Kim, K.; Kulkarni, C.; Biswas, G. Model-Based Avionics System Fault Simulation and Detection. In Proceedings of the AIAA Infotech@Aerospace Conference, Atlanta, GA, USA, 20–22 April 2010.
27. Chen, S.; Ge, H.; Li, J.; Pecht, M. Progressive Improved Convolutional Neural Network for Avionics Fault Diagnosis. *IEEE Access* **2019**, *7*, 177362–177375. [\[CrossRef\]](#)

28. Tirpak, J.A. F-35 Program Dumps ALIS for ODIN. Air & Space Forces. 2020. Available online: <https://www.airandspaceforces.com/f-35-program-dumps-alis-for-odin/> (accessed on 15 September 2025).
29. U.S. Government Accountability Office (GAO). *F-35 Joint Strike Fighter: DOD Needs to Address ALIS Design and Performance Challenges*; GAO-20-316; U.S. Government Accountability Office (GAO): Washington, DC, USA, 2020.
30. Tucker, J.; Rajamani, R. *Health Monitoring Survey of Bell 412EP Transmissions*; NASA Technical Memorandum NASA/TM-2016-219202; NASA: Washington, DC, USA, 2016.
31. Savitzky, A.; Golay, M.J.E. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Anal. Chem.* **1964**, *36*, 1627–1639. [[CrossRef](#)]
32. Lakshmaiah, A.; Ray, K.P.; Prasad, N.N.S.S.R.K. Analysis of Nosecone Radome Effects of Active Electronically Scanned Array (AESA) Radar Performance of Fighter Aircrafts. In Proceedings of the 2019 IEEE Indian Conference on Antennas and Propagation (InCAP), Visakhapatnam, India, 19–22 December 2019; pp. 1–3. [[CrossRef](#)]
33. Voß, A.; Klimmek, T. Parametric Aeroelastic Modeling, Maneuver Loads Analysis Using CFD Methods and Structural Design of a Fighter Aircraft. *Aerosp. Sci. Technol.* **2023**, *136*, 108231. [[CrossRef](#)]
34. Fidan, Ş.S.; Ünal, R. A Survey on Ceramic Radome Failure Types and the Importance of Defect Determination. *Eng. Fail. Anal.* **2023**, *149*, 107234. [[CrossRef](#)]
35. Garg, S.; Krishnamurthi, R. A CNN Encoder Decoder LSTM Model for Sustainable Wind Power Predictive Analytics. *Sustain. Comput. Inform. Syst.* **2023**, *38*, 100869. [[CrossRef](#)]
36. Liu, Y.; Young, R.; Jafarpour, B. Long–Short-Term Memory Encoder–Decoder with Regularized Hidden Dynamics for Fault Detection in Industrial Processes. *J. Process Control* **2023**, *124*, 166–178. [[CrossRef](#)]
37. Jadon, A.; Patil, A.; Jadon, S. A Comprehensive Survey of Regression-Based Loss Functions for Time Series Forecasting. In *Data Management, Analytics and Innovation*; Sharma, N., Goje, A.C., Chakrabarti, A., Bruckstein, A.M., Eds.; Springer: Singapore, 2024; Volume 998, pp. 1–15.
38. Ge, R.; Kakade, S.M.; Kidambi, R.; Netrapalli, P. The Step Decay Schedule: A Near Optimal, Geometrically Decaying Learning Rate Procedure for Least Squares. *arXiv* **2019**, arXiv:1909.09821. [[CrossRef](#)]
39. Luo, Y.; Zhao, X.; Liu, B.; He, S. Condition-Based Maintenance Policy for Systems under Dynamic Environment. *Reliab. Eng. Syst. Saf.* **2024**, *246*, 110072. [[CrossRef](#)]
40. Crespo del Castillo, A.; Parlikad, A.K. Dynamic Fleet Management: Integrating Predictive and Preventive Maintenance with Operation Workload Balance to Minimise Cost. *Reliab. Eng. Syst. Saf.* **2024**, *249*, 110243. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.