

AI-Based Maintenance Code Correction for Reliable MTBF Analysis

Minyoung Kim[1,*], Seon-Ho Jeong[1], Ikgyu Lee[1] & Byoungserb Shim[1]

1Korea Aerospace Industries, LTD. AI Development Team, Seoul 06151, Republic of Korea

Abstract

In aircraft reliability analysis, Mean Time Between Failures (MTBF) is a critical metric that directly determines the stability and operational efficiency of an aircraft. MTBF is derived from actual maintenance data codes, making accurate code entry essential. However, in real-world operational environments, incorrect code entries are common, leading to manual code corrections by experts. While these corrections are accurate, they are time-consuming and costly, limiting their scalability for large-scale data processing. Therefore, this study proposes an AI-based code correction system that predicts maintenance codes using natural language descriptions of defects and actions taken. Experiments and evaluations using expert-corrected data resulted in a prediction code accuracy of up to 98%. The proposed model utilizes a structure that reflects the characteristics of the codes, improving prediction accuracy by 27%p. It also adheres to the logic of Technical Orders, achieving an outstanding performance with over 98% accuracy in classifying failure types. This study aims to automate large-scale maintenance data code correction, reducing time and labor costs while enhancing the quality of maintenance data. It will be able to contribute to more efficient and reliable operational analysis methods.

Keywords: Aircraft Maintenance, Natural Language Processing, Text Classification, Reliability Analysis

1. Introduction

Aircraft operation and reliability analysis are essential for evaluating system performance and developing maintenance strategies. Mean Time Between Failures (MTBF) is a critical metric for quantifying aircraft reliability and is widely used across various operational aspects, including determining component replacement cycles, inventory planning. The reliability of MTBF heavily depends on the accuracy of maintenance data, particularly the precise entry of How Malfunctioned Code (HMC), Action Taken Code (ATC), and Work Unit Code (WUC). However, in real-world operational environments, maintainer often struggle to fully memorize the extensive code system, leading to frequent habitual misuse or arbitrary selection during data entry.

Analysis of actual maintenance data by domain experts revealed that error rates for each code exceed 50%, with 99% of all maintenance records containing at least one error among the three codes. Furthermore, it was found that in 70% of cases, incorrect analysis of failure causes led to non-inherent failures being recorded as inherent failures, directly distorting MTBF calculations. To address these issues, current practices involve domain experts manually reviewing and correcting each code based on natural language descriptions of defects and actions provided by aircraft maintainer. This process is time-consuming and labor-intensive. To overcome these challenges, this study proposes a maintenance code correction model that leverages multilingual Pre-trained Language Models(PLM). By incorporating domain-specific text preprocessing and a model structure designed to reflect the characteristics of the codes, the proposed model aims to achieve high accuracy in code correction. This will enhance the reliability of MTBF calculations and overall aircraft operational analysis by improving the quality of large-scale maintenance data.

*Corresponding author : minyoung.kim@koreaaero.com
https://doi.org/10.71945/icas2026_0770

2. Backgrounds

2.1 Code System and Failure Type Classification

According to the internationally recognized standard for aircraft maintenance and reliability, TO-00-20-2[3], failures are classified into three primary types.

- Type 1(inherent failure) - Inherent, an actual failure of the item.
- Type 2(induce failure) - Induced, the failure of the item was caused by an outside influence.
- Type 6(no failure) - No defect, no actual failure.

The classification of failures is based on a logical framework using codes such as HMC, ATC, WUC. The specific counts and descriptions of these codes are provided in Table 1.

Table 1 Code Description and the number of class in TO-00-20-2

Code	Description	# of Class
HMC	Indicates how or why a piece of equipment malfunctioned	About 300
ATC	Identify a complete unit of work, a maintenance task, or action	About 30
WUC	Unique identification code that hierarchically distinguishes aircraft or equipment by system-subsystem-component level.	About 2,200

Mean Time Between Failures (MTBF) is a critical metric that calculates the average time between failures of aircraft equipment or components. It is widely used for determining replacement cycles and calculating maintenance operational costs. According to TO-00-20-2, MTBF is calculated by dividing the total operational time by the number of inherent failure occurrences. To ensure accurate calculation, it is essential to correctly classify failure types and identify inherent failures without errors. Therefore, this study focuses on the three codes (HMC, ATC, WUC) that play a crucial role in failure type classification and MTBF calculation. Especially, HMC is classified each defect into one of the three defect types which has the most significant impact on failure type classification. And these 3 codes are not independent; they should follow logical rules defined in TO-00-20-2. Therefore, adhering to the defined logic is crucial for ensuring accurate MTBF calculations.

2.2 Related Works

The most directly relevant work on code prediction is that of Anand and Keefer [2], who reported that structured code fields in U.S. Air Force C-17 maintenance data had an accuracy of only 60%, and proposed a pipeline combining TF-IDF vectorization with Logistic Regression to achieve an average accuracy of 93% across five maintenance codes. However, their approach does not leverage pre-trained language models, resulting in limited semantic representation, and achieved only 72% accuracy for WUC prediction.

3. Methodology

3.1 Data

This study utilized about 40,000 aircraft maintenance records created by the Korean Air Force from August 2013 to July 2020. Each record includes a free-form defect and action description written by aircraft maintainer, as well as HMC, ATC, and WUC corrected by experts based on these descriptions. With considering the imbalance in code distribution, we split the data using HMC to ensure that all failure types were evenly distributed across the training (70%), validation (20%), and test (10%) sets. This approach prevented the model from being biased towards specific failure types.

3.2 Text Preprocessing

To improve the quality of natural language data, a preprocessing step was performed. Aircraft maintainers often describe the same concept in various ways, which can cause the model to interpret semantically identical texts as different during training, thereby reducing learning efficiency. First,

meaningful special characters were preserved based on an analysis of maintenance history data, and meaningless ones were removed. In addition, frequent variant expressions of equipment names and work-related terms were manually extracted, and a replacement dictionary was constructed to standardize these variations.

3.3 Model Architecture

This study used an encoder-based multilingual PLM as the backbone to reflect the characteristics of data containing both Korean and English. Classification models were added for each code type, where the maintenance text is encoded by the PLM and the generated embeddings are used for code classification. Each classifier learns independent weights to capture the unique characteristics of each code system. Batch Normalization was applied to each classifier to improve training stability.

3.4 WUC 2-stage Classification Strategy

The key feature of our model is its two-stage prediction strategy for WUC. WUC consists of a five-digit code with approximately 2,000 combinations. Treating this as a single classification problem would result in a large number of classes and fail to account for the structure of WUC. WUC has a hierarchical structure where the first two digits represent the subsystem and the remains represent the component within that subsystem. Therefore, it is more reasonable to model these explicitly. Our approach involves separately predicting the subsystem and component. This strategy reduces the number of classes each model needs to predict, leading to more stable training and better learning of the hierarchical structure of WUC, mimicking the reasoning process of experts.

3.5 Training

Model training involved fine-tuning both the pre-trained language model and the classifier simultaneously. We conducted experiments using three different multilingual pre-trained language models as backbones. The loss function was the sum of cross-entropy losses for each code, and accuracy was used as the evaluation metric for each code.

4. Result

4.1 Overall performance

To evaluate the performance of the proposed AI-based code correction system, experiments were conducted using three different multilingual PLMs as backbones. Additionally, a baseline model combining TF-IDF vectorization with Logistic Regression was implemented. Table 2 shows the accuracy for each model across all code types. The results indicate that the BGE-M3 model achieved the best performance across all classification tasks. The ATC task recorded the highest accuracy at 97.6%, due to the relatively small number of classes and the consistent use of action terms.

Table 2 Classification Accuracy by Model and Task (%)

Model	HMC	HMC Type Defect	ATC	WUC	WUC Subsystem
bge-m3[4]	***95.2%***	***98.5%***	***97.6%***	***91.9%***	***98.0%***
LaBSE[5]	94.0%	98.4%	97.0%	90.8%	97.7%
Multilingual-e5-base[6]	93.6%	98.2%	96.8%	90.8%	97.8%
TF-IDF+LR[2]	90.1%	96.6%	94.0%	21.0%	92.2%

4.2 HMC Type Classification

A key evaluation criterion for HMC classification is the accuracy of type defect classification. Among the total test data, only 37 cases were incorrectly predicted the type defect. This indicates that even when the model misclassifies an HMC, it does so within the same type defect. Notably, the recall rate for Type 1 defect was approximately 98%, which is crucial for reliability analysis.

4.3 WUC Classification

To validate the proposed 2-stage classification strategy for WUC, we compared it with a baseline model which directly classifies the entire WUC. The baseline model achieved 65% accuracy, whereas

the proposed model reached an overall WUC accuracy of 92%. This demonstrates that explicitly modeling the hierarchical structure of WUC leads to a significant performance improvement.

4.4 Code Logic Compliance

We verified compliance with the code assignment logic defined in TO-00-20-2. No logical violations were observed in the model's predictions. This indicates that the model not only predicts individual codes independently but also learns the relationships between codes to generate consistent combinations.

4.5 Error Case Analysis

An analysis of misclassified cases revealed that rare codes tended to be misclassified, which can be solved by data augmentation. Additionally, we observed that misclassified codes had lower confidence scores, which can be used to trigger expert review when the confidence score falls below a threshold, thereby improving practical applicability.

5. Conclusion

This study proposes an automatic code correction system based on multilingual pre-trained language models to address input code errors in aircraft maintenance data. The proposed system automatically predicts correct codes from defect and action descriptions provided by aircraft maintainer. The system achieved accuracies of 95% for HMC, 98% for ATC, 98% for WUC systems, and 92% for WUC components, demonstrating its capability to automatically correct most maintenance code errors. The main achievements of this study can be summarized as follows. First, the system achieved over 98.5% accuracy in HMC type classification, a critical component for MTBF calculation. Even when individual HMC codes were misclassified, they achieved the key requirement of distinguishing inherent from non-inherent defects for reliability analysis. Second, the proposed two-stage strategy, which explicitly reflects the hierarchical structure of WUC, led to a significant performance improvement over single-step classification, validating the effectiveness of integrating domain knowledge into the model architecture. Additionally, the model's predictions did not violate any code logic rules, indicating that it generates high-quality data suitable for subsequent reliability analysis. This study has overcome the limitations of time-consuming and costly manual corrections in large-scale maintenance data, enhancing the efficiency and performance of reliability analysis. Future research will focus on improving practical applicability by performing expert reviews based on model prediction confidence levels and applying data augmentation techniques to low-frequency codes to enhance overall generalization performance across all code categories.

Acknowledgement

This work was supported by Korea Research Institute for defense Technology planning and advancement(KRIT) - Grant funded by Defense Acquisition Program Administration (DAPA) (KRIT-CT-22-081, Weapon System CBM+ Research Center)

Copyright statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

References

[1] M. Rausand and A. Høyland, *System Reliability Theory: Models, Statistical Methods, and Applications*, 2nd ed., Wiley-Interscience, 2004.

[2] S. Anand and R. Keefer, "From description to code: A method to predict maintenance codes from maintainer descriptions," *Maintenance, Reliability and Condition Monitoring*, vol. 2, no. 2, pp. 35–44, Oct. 2022. DOI:

10.21595/marc.2022.22798

[3] U.S. Air Force, "Technical Manual: Maintenance Data Documentation," TO 00-20-2, 2019.

[4] Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D. and Liu, Z. M3-Embedding: multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. arXiv preprint arXiv:2402.03216, 2024.

[5] Feng, F., Yang, Y., Cer, D., Arivazhagan, N. and Wang, W. Language-agnostic BERT sentence embedding. arXiv preprint arXiv:2007.01852, 2022.

[6] Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R. and Wei, F. Multilingual E5 text embeddings: a technical report. arXiv preprint arXiv:2402.05672, 2024.