

Cost-optimal Preventive Maintenance based on Remaining Useful Life Prediction and Minimum-repair Block Replacement Models

Young-Suk Choo* · Seung-Jun Shin**†

*Graduate School of Technology and Innovation Management, Hanyang University

**School of Interdisciplinary Industrial Studies, Hanyang University

잔여 유효 수명 예측 모형과 최소 수리 블록 교체 모형에 기반한 비용 최적 예방 정비 방법

주영석* · 신승준**†

*한양대학교 기술경영전문대학원

**한양대학교 산업융합학부

Predicting remaining useful life (RUL) becomes significant to implement prognostics and health management of industrial systems. The relevant studies have contributed to creating RUL prediction models and validating their acceptable performance; however, they are confined to drive reasonable preventive maintenance strategies derived from and connected with such predictive models. This paper proposes a data-driven preventive maintenance method that predicts RUL of industrial systems and determines the optimal replacement time intervals to lead to cost minimization in preventive maintenance. The proposed method comprises: (1) generating RUL prediction models through learning historical process data by using machine learning techniques including random forest and extreme gradient boosting, and (2) applying the system failure time derived from the RUL prediction models to the Weibull distribution-based minimum-repair block replacement model for finding the cost-optimal block replacement time. The paper includes a case study to demonstrate the feasibility of the proposed method using an open dataset, wherein sensor data are generated and recorded from turbofan engine systems.

Keywords : Remaining Useful Life, Predictive Maintenance, Preventive Maintenance, Weibull Distribution, Minimum-Repair Block Replacement

1. 서론

최근 설비 시스템의 효과적인 유지 관리를 위하여 시스템 상태 모니터링(monitoring) 데이터를 활용한 고장 예지 및 건전성 관리(Prognostics and Health Management; PHM)가 많이 활용되고 있다[3]. PHM은 시스템의 이상 감지(anomaly

detection)와 고장 진단(diagnostics)뿐만 아니라 잔여 유효 수명(Remaining Useful Life; RUL) 예측을 통해 상태 기반 유지 관리(condition-based maintenance)를 수행한다. 특히, RUL 예측은 사전에 예상치 못한 고장 발생으로 야기되는 손실을 줄일 수 있기 때문에 예측 보전(Predictive Maintenance; PdM)의 핵심적 방법이다[1]. RUL 예측의 정확성을 높이기 위해서 수집된 데이터에 적합한 RUL 예측 분석 방법이 요구되며, 예측된 결과를 유지 관리 방법에 적용함으로써 유지 관리 비용 및 가용도를 최적화하는 것이 중요하다.

Received 14 June 2022; Finally Revised 16 August 2022;

Accepted 24 August 2022

† Corresponding Author : sjshin@hanyang.ac.kr

시스템의 유지 관리 방법으로는 사후 보전(Corrective Maintenance; CM)과 예방 보전(Preventive Maintenance; PM) 및 예측 보전이 있다[2]. 사후 보전은 고장이 발생한 이후에 복구를 통해 시스템을 정상화하는 방법이다. 예방 보전은 고장 발생 이전에 정해진 기준(기대 수명, 교체 주기 등)에 의한 스케줄에 따라 유지 관리를 실시하는 방법이다. 예측 보전은 시스템의 상태를 분석한 후 실시되며, 시스템 상태와 고장 발생 간의 관계가 미리 규명되어야 한다. 예방 보전과 예측 보전은 고장 발생 이전에 실시된다는 공통점이 있다. 최근에는 스마트 공장이 늘어남에 따라 산업용 센서를 활용한 예측 보전의 중요성이 커지고 있다. 따라서 RUL 예측을 통해 시스템의 고장 시점을 예측하고 시스템 특성에 맞는 수리(repair) 또는 교체(replacement)를 위해서는 예측 보전과 예방 보전을 모두 고려한 유지 관리 전략이 필요하다.

기존 RUL 예측 연구들은 RUL을 예측하는 방법(method)과 예측 정확성(accuracy) 검증에 초점을 맞추고 있다[13]. 즉, 연구 결과인 고장 예측 시점을 이용하여 유지 관리 전략의 수립을 위한 연구로의 연계는 많이 다루지 않았다. 이는 RUL 예측을 독립된 연구 결과로서 다루는 이유도 있지만, 동일 연구에서 설비 시스템의 다양한 공정 환경 변수를 고려하여 분석하기에는 어려움이 있기 때문이다. 특히, 개별 시스템이 아닌 여러 설비가 포함된 시스템은 연구 복잡성이 가중된다. 한편, 기존 유지 보수 연구들은 설비의 RUL 예측을 통한 고장 시점이 아닌, 이미 발생했던 고장 시점을 기반으로 이론적 수식을 이용하여 분석하는 경우가 많다. RUL 예측의 궁극적 목적은 고장 예측으로 끝나는 것이 아니라 예측 결과를 활용하여 예기치 않은 고장을 최소화하고 불필요한 유지 보수 비용을 절감하는 데 있다[18]. 따라서 RUL 예측 결과로 얻은 고장 예측 시점을 스케줄 기반 예방 보전 방법에 접목함으로써 효과적인 유지 관리 전략을 수립할 필요가 있다.

스케줄 기반 예방 보전의 대표적인 방법으로는 수명 교체(age replacement) 모델과 블록 교체(block replacement) 모델이 있다[21]. 수명 교체 모델은 부품이나 시스템을 사전에 정해진 교체 수명(replacement age)에 따라 교체하는 방법이다. 만약 교체 수명 전에 고장이 발생하면 고장 발생 즉시 교체된다. 수명 교체 모델은 고장률(failure rate)이 시간의 흐름에 따라 증가하고, 고장 교체 비용이 예방 교체 비용보다 높은 경우에 효과적이다. 블록 교체 모델은 부품 또는 시스템이 수명과 상관없이 사전에 일정하게 정해진 시점(주기)에 따라 예방 교체하는 방법이다. 고장 교체 여부와 상관없이 정해진 시점에서 교체가 되기에 유지 관리가 편리한 장점이 있다. 그러나 새것과 같은 시스템이 교체될 수 있기에 낭비적 요소가

존재할 수 있다. 따라서 블록 교체 모델은 적절한 고장 수리를 통해서 낭비적 요소를 보완하거나 비용 또는 가용도를 고려한 최적 교체 주기 결정이 중요하다.

본 논문에서는 시스템의 학습적 방법을 이용하여 RUL 예측 모델을 생성하고 예측된 RUL을 이용하여 비용 관점에서 최적의 예방 보전 시점을 찾는 방법을 제안한다. 구체적으로 다수의 동일한 설비들로 구성된 시스템을 대상으로 기계학습(machine learning) 기법을 이용하여 RUL 예측 모델을 생성하고 모델의 성능을 평가한다. 그리고 시스템 특성을 감안한 블록 교체 모델을 선택하고 비용 최소화를 달성하는 교체 주기를 도출한다. 이를 통하여, 개별 설비가 아닌 다수의 동일 설비들을 대상으로 RUL을 예측하며, RUL 예측 결과와 예방 보전 전략을 연계한 통합적인 방법을 제안하여, 유지 보수의 비용 최적화를 달성하고자 한다. 제안 방법은 두 개의 파트(part)로 구성된다. 첫 번째 파트에서는 기계학습 기법인 랜덤 포레스트(Random Forest; RF)와 익스트림 그라디언트 부스팅(Extreme Gradient Boosting; XGBoost)을 이용하여 공정 이력(historical) 데이터를 학습하고 RUL 예측 모델들을 생성한다. 이러한 예측 모델들 중에서 정확도가 가장 높은 모델을 이용하여 개별 설비의 고장 예측 시점을 도출한다. 두 번째 파트에서는 개별 설비의 고장 예측 시점을 활용하여 예방 보전 의사결정을 수행한다. 이때, Weibull 분포 기반 최소 수리 블록 교체 모델을 이용하여 비용 최소화 관점에서 수리 및 교체 시점을 결정한다. 사례 연구에서는 CMAPSS(Commercial Modular Aero-Propulsion System Simulation) 터보팬 엔진 데이터셋을 이용하여 시스템의 고장 시점을 예측하고 시스템 특성과 유지 비용을 감안한 최적 교체 주기를 찾는 과정을 보여준다.

본 논문의 구성은 다음과 같다. 2장에서는 RF와 XGBoost를 활용한 RUL 예측 연구 및 블록 교체 모델을 설명한다. 3장에서는 제안된 방법을 설명한다. 4장에서는 사례 연구를 소개하며, 5장에서 결론을 맺는다.

2. 기존 연구

2.1 RF와 XGBoost 활용 잔여 유효 수명 예측

산업용 센서 기술의 발달 및 센서로부터 얻은 모니터링 데이터가 증가함에 따라 공정 빅데이터 분석을 위한 기계학습의 적용 또한 증가하고 있다[1]. 공정 이력 데이터를 기반으로 예측 모델을 구성하는 데이터 기반(data-driven) RUL 예측의 경우, 전통적 통계적 기법과 기계학습 기법이 많이 활용된다[26]. 대표적으로 활용되는 기계학습 기법은 선형·

비선형 회귀분석(regression), 서포트 벡터 머신(Support Vector Machine; SVM), RF, XGBoost, 인공신경망(Artificial Neural Network; ANN) 등이 있다.

RF와 XGBoost는 앙상블(ensemble) 학습 방법이라 불리며 다양한 예측 모델에서 활용되고 있다. 앙상블은 여러 개의 분류기를 생성하고 예측 결과들을 결합하여 정확도를 높이는 기법이다[25]. 앙상블 기법은 기계학습에서 빈번하게 발생하는 과적합(overfitting) 문제를 감소시키고 단일 모형보다 정확성과 적응성을 높이는 장점이 있다. 앙상블 기법으로는 보팅(voting), 배깅(bootstrap aggregation; bagging), 부스팅(boosting) 등이 있다[8]. 보팅은 여러 개의 분류기가 투표를 통해 최종 예측 결과를 결정한다. 배깅은 평균값, 중간값을 이용하여 예측하며 비교적 빠른 수행 속도를 나타낸다. 배깅의 대표적 알고리즘은 여러 개의 의사결정나무(Decision Tree; DT)를 사용하는 RF이다.

RF는 DT가 가지고 있는 학습(train) 데이터 성능과 테스트(test) 데이터의 성능 간의 차이가 크게 발생하는 문제점을 보완하기 위하여 개발되었다[11]. 변수 간의 상호작용 및 비선형성을 다룸에 있어 큰 문제가 없으며 많은 수의 DT 결과를 평균하여 결과값을 도출하기 때문에 비교적 높은 안정성과 예측 정확성을 장점으로 가진다[12]. 한편, 부스팅은 여러 개의 분류기가 순차적 학습을 통해 다음 분류기에 가중치를 부여하는 방식이다. 부스팅의 대표적 알고리즘은 Adaboost와 Gradient boost가 있으며 Gradient boost에서 속도 수행 능력을 높인 XGBoost가 있다[5]. XGBoost는 병렬화와 분산화를 통해서 학습을 수행함으로써 학습 속도가 빠르고 결측값 처리가 가능한 장점이 있다[9, 10].

RF와 XGBoost를 활용한 RUL 예측 연구는 다음과 같다. Chen et al.[6]은 RF 회귀를 사용하고, Aydemir and Acar[1]은 RF와 LSTM(Long Short-Term Memory)을 이용하여 CMAPSS 터보팬(turbofan) 엔진의 RUL을 예측하였다. Patil et al.[18]은 RF와 Gradient boost를 이용하여 베어링(bearing)의 RUL을 예측하였다. 또한 Cailian et al.[4]은 RF에 유전(genetic) 알고리즘을 접목하여 리튬이온(lithium-ion) 배터리 RUL을 예측하였다.

Shi et al.[23]은 베어링의 RUL 예측에 XGBoost와 RF, SVM을 이용하였다. Que and Xu [19]은 XGBoost와 dynamic time warping을 이용하여 스팀 터빈(steam turbine)의 RUL을 예측하였고, Tomg et al.[24]은 리튬이온 배터리의 RUL 예측을 위해 XGBoost, Adaboost, Gradient boost 등을 이용하고 성능 평가를 비교하였다. Li et al.[14]은 XGBoost, CNN(Convolutional Neural Network), LightGBM, RNN(Recurrent Neural Network)를 이용하여 CMAPSS 터보팬 엔진의 RUL을 예측하였다. Liu et al.[16] 또한

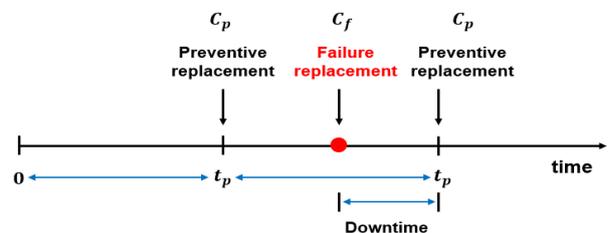
CMAPSS 터보팬 엔진의 수명 예측을 위하여 XGBoost, RNN, DNN(Deep Neural Network)을 이용하였다.

기존 RF와 XGBoost 기반 RUL 예측 연구들의 시사점은 다음과 같다. 첫째, 데이터 기반 RUL 예측 연구가 많았다. 기존 연구들은 센서 데이터 측정값을 이용하여 두 가지 이상의 기계학습 알고리즘을 통해 RUL을 예측하고 이들의 성능 비교를 하는 방식이 많았다. 그러나, 예측한 RUL을 활용하여 예방 보전 전략으로 연계하거나 시스템 교체를 결정하는 방법에 대한 연구는 부족하였다. 둘째, CMAPSS 터보팬 엔진, 리튬이온 배터리, 베어링 등 공개된 참조 데이터 사용이 많았다. 참조 데이터의 경우 데이터 신뢰도를 확보할 수 있으며 RUL 예측값과 실제값 비교가 용이하다. 참고로, 본 연구의 사례 연구에서도 CMAPSS 데이터 셋을 활용하였다.

2.2 블록 교체 모델(Block Replacement Model)

<Figure 1>과 같이, 블록 교체 모델에서 시스템은 일정하게 정해진 시점($t_p, 2t_p, 3t_p, \dots$)에 예방 교체가 이루어진다. 여기서 블록은 여러 개의 구성 요소가 하나의 구역 또는 단위로 합쳐진 것을 의미한다. 정해진 시점(주기)에 따라 예방 보전이 수립되므로, 유지 관리 계획을 세우고 실행하기에 편리하다. 또한, 단일 시스템보다는 동일 기계나 부품이 수가 많을 때 효과적으로 활용된다[22]. 블록 교체 모델은 교체하는 시점 및 방법에 따라 다음과 같은 세 가지 경우가 존재한다.

- i) 고장이 발생한 시스템을 고장 발생 즉시 교체하는 경우
- ii) 고장이 발생한 시스템을 다음 계획된 교체 시간 t_p 까지 대기 후 교체하는 경우
- iii) 고장이 발생한 시스템을 최소 수리를 통하여 활용 후, 다음 계획된 교체 시간 t_p 에 교체하는 경우



<Figure 1> Block Replacement Model

<Figure 1>에서 C_p 는 고장이 발생하기 전에 예방 교체 비용(the total cost of a preventive replacement)이다. C_f 는 고장이 발생한 이후의 교체 비용(the total cost of a failure replacement)이다. C_p 는 예방 교체 비용에 더하여 고

장 교체 시 발생하는 추가적 비용을 포함하므로, 일반적으로 $C_f > C_p$ 관계를 갖는다.

i)의 경우는 정해진 예방 보수 시점 t_p 전에 고장이 발생한 것으로, 고장 발생 즉시 교체된다. 하지만 t_p 가 되면 다시 계획된 예방 보전에 의해서 교체되므로 낭비가 발생할 수 있다. $N(t_p)$ 를 t_p 기간 동안 발생하는 고장 횟수, $M(t_p)$ 는 t_p 기간에 일어나는 평균 횟수, 즉 기대값 $E(N(t_p))$ 이라 하면 단위 시간당 기대 비용(expected cost rate per unit time) $C_1(t_p)$ 는 식 (1)로 나타낼 수 있다[17].

$$C_1(t_p) = \frac{C_p + C_f M(t_p)}{t_p} \quad (1)$$

단위 시간당 최적 비용(optimal cost rate per unit time) $C_1(t_p^*)$ 을 찾기 위해 식 (1)을 t_p 에 관하여 미분하여 0으로 놓으면 식 (2)와 같이 유도된다[17]. 여기서 $m(t_p)$ 는 재생(renewal) 밀도 함수 $m(t_p) = d/dt M(t_p)$ 이다. 이어서, $C_1(t_p^*)$ 는 식 (3)으로 정리된다[17].

$$t_p m(t_p) - \int_0^{t_p} m(t_p) dt = \frac{C_p}{C_f} \quad (2)$$

$$C_1(t_p^*) = C_f m(t_p^*) \quad (3)$$

ii)의 경우는 고장이 발생한 시스템을 다음 계획된 교체 시간 t_p 까지 대기 후 교체하게 된다. i) 경우에 대비하여 비용 측면에서는 유리하나, 다음 예방 보수 시간까지 고장이 발생한 시스템을 가동하지 못하는 단점이 발생한다. C_d 는 고장 발생부터 t_p 까지 경과 시간에 대한 대기비용(downtime cost)일 때, $C_2(t_p)$ 는 식 (4)와 같다[7].

$$C_2(t_p) = \frac{C_p(1-F(t_p)) + C_f F(t_p) + C_d \int_0^{t_p} (t_p - t) f(t) dt}{t_p} \quad (4)$$

$$= \frac{C_p(1-F(t_p)) + C_f F(t_p) + C_d \int_0^{t_p} F(t_p) dt}{t_p}$$

$C_2(t_p^*)$ 를 찾기 위해 식 (4)를 t_p 에 관하여 미분하고 0으로 놓으면 식 (5), (6)으로 정리된다[7]. 이때, 고장 밀도 함수 $f(t)$ 는 $f(t) = d/dt F(t)$ 이다.

$$(C_p - C_f) \int_0^{t_p} f(t_p^*) - f(t) dt \quad (5)$$

$$+ C_d \int_0^{t_p^*} F(t_p^*) - F(t) = C_p$$

$$C_2(t_p^*) = (C_f - C_p) f(t_p^*) + C_d F(t_p^*) \quad (6)$$

iii)의 경우는 t_p 이전에 고장이 발생할 경우로 다음 예방 보전 시점까지 최소 수리를 실시하여 시스템을 고장 직전 상태로 복구하는 것이다. 즉, 최소 수리는 중고 제품 또는 재생 부품 등을 활용하여서 비용 측면에서 최소화를 추구한다. 여러 개의 설비를 동시에 운영하는 경우에 시스템을 중단 없이 활용하며, 교체 비용을 절감하는 장점이 있다. 본 연구의 제안 방법은 시스템 내 다수의 설비가 운영되며, 개별 설비는 고장 발생 시 즉각 복구를 하는 것으로 가정한다. 따라서 시스템 특성을 감안한 유지 비용 최적화를 위하여 iii)의 경우를 적용한다. 이때, 수리에 걸리는 시간은 무시할 정도로 짧고, 수리 이후에도 시스템 고장률 함수 변화는 없는 것으로 가정한다. 고장률 함수를 $h(t)$, 최소 수리 비용을 C_k 라 할 때, $C_3(t_p)$ 는 식 (7)로 나타낼 수 있다[15, 17]. $C_3(t_p^*)$ 를 찾기 위해 식 (7)을 t_p 에 관하여 미분하고 0으로 놓으면 식 (8)로 정리된다[17, 21]. 식 (8)에서 $h(t)$ 가 단조 증가할 경우, 식 (9)를 만족한다[17, 21].

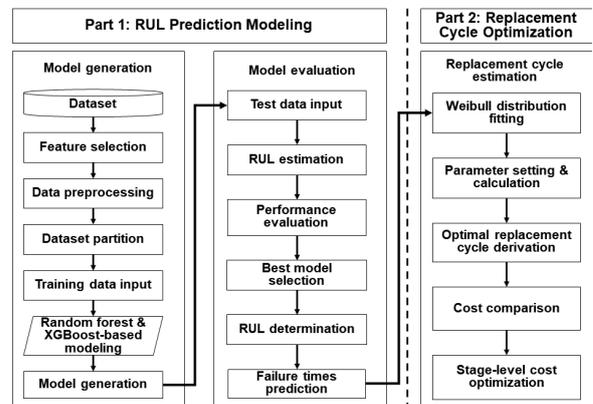
$$C_3(t_p) = \frac{C_p + C_k \int_0^{t_p} h(t) dt}{t_p} \quad (7)$$

$$t_p h(t_p) - \int_0^{t_p} h(t_p) dt = \frac{C_p}{C_k} \quad (8)$$

$$C_3(t_p^*) = C_k h(t_p^*) \quad (9)$$

3. 제안 방법

제안 방법은 <Figure 2>와 같이, 시스템의 RUL를 예측하는 파트 1과 예측된 RUL를 이용하여 최적 예방 교체를 실시하는 파트 2로 구성된다. 파트 1의 첫 번째 단계에서는 RF와 XGBoost를 이용하여 공정 이력 데이터를 학습한 RUL 예측 모델을 생성한다. 두 번째 단계에



<Figure 2> The Proposed Framework

서는 테스트 데이터를 이용하여 RUL 예측 모델의 성능을 평가한다. 이를 통해 가장 예측 정확도가 높은 모델을 찾는다. 파트 2에서는 파트 1의 최상 예측 모델을 이용한 시스템 고장 예측 시간을 사용한다. Weibull 분포 기반 최소 수리 블록 교체 모델을 이용하여 최적 예방 교체 시점을 도출 후, 기존 방법 대비 비용 절감 효과를 분석하고 추가 최적화 방안을 모색한다.

3.1 RUL 예측

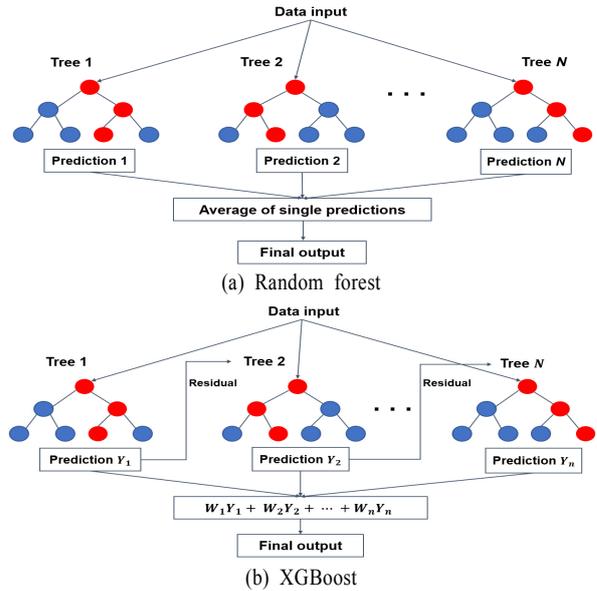
3.1.1 모델 생성

파트 1의 모델 생성은 공정 이력 데이터의 학습을 통하여 RUL 예측 모델을 도출하는 과정이다. 이를 위하여 공정 이력 데이터를 추출하고 전처리 및 정규화 과정을 거친다. 사용되는 공정 이력 데이터는 시간 또는 사이클 단위로 설비 시스템의 상태가 기록된 데이터가 요구된다. 이후, 데이터 분할 과정을 거친다. 데이터 분할은 전체 데이터를 학습 데이터와 테스트 데이터로 분할하는 것으로서, 보편적으로 7:3 또는 8:2 비율을 적용한다.

<Figure 3>은 RF와 XGBoost 구조를 나타낸 것이다. 이 두 기법은 분할된 학습 데이터의 반복적인 학습을 통하여 오차를 줄이는 RUL 예측 모델을 생성한다. 여기서 RF를 선택한 이유는 다양한 센서 변수 특성을 가진 공정 이력 데이터 학습에서 안정적이고 높은 예측력을 보여주기 때문이다. 그리고 XGBoost를 선택한 이유는 비교적 간단한 알고리즘으로 병렬 연산이 가능하므로 데이터가 많은 경우에도 빠르게 학습이 가능한 장점이 있다.

RF는 다수의 DT 예측 결과값들의 평균값을 결과로써 도출하고 다양한 변수를 고려할 수 있기 때문에 예측 안정성이 높다. 한편, XGBoost는 처음 데이터를 학습하고 학습 결과에서 발생한 오차를 그다음 DT에서 학습하는 방식으로 오차를 순차적으로 줄인다. 최종 결과값은 각 DT에 다른 가중치(w_n)를 주어 합하게 된다. 이는 각 DT를 같은 비율로 합할 경우, 정확성이 떨어지기 때문이다.

제안 방법은 두 가지 방법으로 데이터를 학습하였다. 첫 번째는 모든 데이터를 한 번에 넣어서 학습시키는 방법이고, 두 번째는 전체 데이터를 단일 설비 단위로 그룹(group)화한 후, 하나씩 모델에 투입하여 가장 최근의 다섯 사이클에 대해 예측을 하고 평균을 취하는 방법(average of last 5 cycles prediction)이다. 두 가지 방법을 사용한 이유는 같은 방법의 기계학습 알고리즘을 사용하더라도 어떻게 학습시키느냐에 따라서 예측 정확도가 달라진다는 전제에서 기인한 것이다. 즉, 같은 학습 데이터를 사용하더라도 학습 방법에 따라 테스트 데이터에서의 예측 결과값이 달라질 수 있다.



<Figure 3> Structure of Random Forest and XGBoost

3.1.2 모델 성능 평가

모델 생성 단계에서 RF와 XGBoost를 이용하여 예측 모델을 생성하였으므로 2개의 예측 결과값이 도출된다. 이러한 예측 모델이 실제값과 대비하여 얼마나 정확하게 예측하였는지에 대한 성능 평가가 필요하다. 본 연구에서는 보편적으로 사용하는 성능 평가 지표인 평균 제곱근 편차(Root Mean Square Error; RMSE), 평균 제곱 오차(Mean Absolute Error; MAE) 및 결정계수(R-Squared)를 이용한다. RMSE는 MAE에 비해서 오차에 덜 민감한 장점이 있다. MAE는 계산이 비교적 간단하지만 이상치(outlier)가 있을 경우 한계점이 있다. R^2 는 직관적으로 연관성을 나타내며, 0과 1 사이의 값을 가진다. \hat{Y} 가 예측값이고 Y 가 실제값일 때, RMSE, MAE, R^2 는 각각 식 (10), 식 (11), 식 (12)와 같다. 성능 평가 분석은 RMSE와 MAE는 결과값이 낮은 값을 가질수록, R^2 는 높은 값을 가질수록 예측 성능이 높다고 판단한다.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n}} \quad (10)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i| \quad (11)$$

$$R^2 = 1 - \frac{SSE}{SST}, \quad SSE = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2, \quad (12)$$

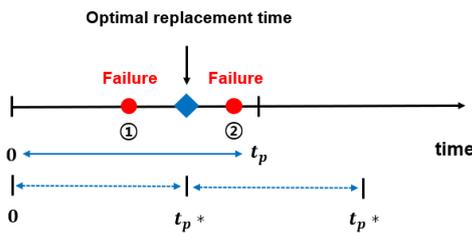
$$SST = \sum_{i=1}^n (Y_i - \bar{Y})^2$$

3.2 최소 수리 블록 교체

파트 2에서는 예측된 고장 시점을 기반으로 개별 설비들이 포함된 전체 시스템의 최적 예방 교체 시점(주기)을 도출한다. 모든 설비는 최적 예방 교체 주기 끝에서 RUL과 상관없이 교체가 실시된다.

본 장에서의 가정 사항은 다음과 같다. 첫째, 예방 보전으로 교체되는 시스템은 다수의 동일한 설비가 포함된 시스템으로서 최소 수리 블록 교체를 실시한다. 2.2절에서 설명한 바와 같이, 최소 수리 블록 교체 모델은 예방 교체 시점 이전에 고장 발생할 경우에 최소 수리를 수행하여 시스템을 운영하다가 정기 교체 시점에 일제히 교체하는 것이다. 이 방법은 수리 비용이 교체 비용보다 경제적이 때 사용하며 다수의 동일한 시스템을 운영할 경우에 효과적이다[15]. 둘째, 개별 설비들 고장 예측 시점은 Weibull 분포를 따른다. Weibull 분포는 부품, 기계 등 고장 데이터 분석에서 가장 널리 쓰이는 분포 중 하나이다.

<Figure 4>는 최소 수리 블록 교체 모델을 나타낸다. t_p 는 현재 운영되는 예방 교체 주기이며, 일반적으로 개별 설비의 고장 시간 평균의 근처값으로 결정된다. 현재 예방 교체 주기 t_p 를 이용할 경우, 고장 ①과 고장 ②는 예방 교체 주기 이내에 고장이 발생한 것으로 최소 수리가 실시된다(①과 ②는 각기 다른 설비의 고장 발생 시점이다). 기존 예방 교체 주기의 경우, 고장 ①의 최소 수리 범위는 고장 ① 시점부터 t_p 까지가 된다. 반면 최적 교체 주기 t_p^* 를 이용할 경우, 고장 ①만 최소 수리가 실시되며, t_p^* 에서 모든 설비가 새로 교체되어 고장 ②는 발생하지 않는다. 이때, 최소 수리 범위는 고장 ① 시점부터 t_p^* 까지가 된다.



<Figure 4> Minimum-repair Block Replacement Model

파트 2에서는 현재 운영되는 예방 교체 주기 t_p 와 최소 수리 블록 교체를 실시할 경우의 예방 교체 주기 t_p^* 에서의 평균 유지 비용을 비교하여 합리적인 의사결정을 도출한다. 파트 2에서는 파트 1에서 도출했던 가장 성능이 좋은 예측 모델을 취하여 시스템 고장 예측 시점을 얻는다. 예측 성능이 좋다는 것은 실제 고장 시점과 비교하여 예측 고장 시점과의 오차가 적은 것을 의미한다.

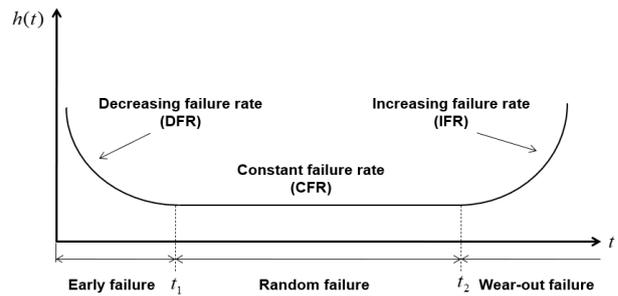
Weibull 분포의 고장 확률 밀도 함수 $f(t)$, 고장 확률 분포 함수 $F(t)$, 신뢰도 함수 $R(t)$ 는 식 (13), (14), (15)로 나타낼 수 있다. α 는 척도 모수(scale parameter), β 는 형상 모수(shape parameter)를 의미한다.

$$f(t) = \frac{\beta}{\alpha} \left(\frac{t}{\alpha}\right)^{\beta-1} \exp\left[-\left(\frac{t}{\alpha}\right)^\beta\right] \quad (13)$$

$$F(t) = 1 - \exp\left[-\left(\frac{t}{\alpha}\right)^\beta\right] \quad (14)$$

$$R(t) = \exp\left[-\left(\frac{t}{\alpha}\right)^\beta\right] \quad (15)$$

<Figure 5>와 같이, Weibull 분포는 고장률 함수 분포에 따라 DFR(Decreasing Failure Rate), CFR(Constant Failure Rate), IFR(Increasing Failure Rate)로 나타낼 수 있다. $\beta < 1$ 이면 DFR, $\beta = 1$ 이면 CFR, $\beta > 1$ 이면 IFR 형태로 나타난다. 이때, $\beta = 1$ 이면 지수 분포, $\beta = 2$ 이면 레일리(Rayleigh) 분포, $\beta > 3$ 이면 정규 분포에 근사한다고 본다.



<Figure 5> Failure Rate Distribution Type

시스템 고장 시간이 Weibull 분포를 따를 때, 최소 수리 블록 교체 모델 식 (7)은 식 (16)으로 나타낼 수 있다.

$$C_3(t_p) = \frac{C_p + C_k \left(\frac{t_p}{\alpha}\right)^\beta}{t_p} \quad (16)$$

여기서 식 (7)의 고장률 함수 $h(t)$ 는 $h(t) = f(t)/R(t) = -\frac{d}{dt} \ln R(t)$ 이고, 고장률 함수를 적분한 것은 식 (17)을 통해 도출한 후 대입하였다.

$$\int h(t) dt = -\ln R(t) = -\ln e^{-\left(\frac{t}{\alpha}\right)^\beta} = \left(\frac{t}{\alpha}\right)^\beta \quad (17)$$

비용 관점에서 가장 효과적인 최적 교체 주기 t_p^* 를 찾기 위해서는 식 (16)을 t_p 에 대해 미분하여 0으로 놓으면 식 (18)로 구할 수 있다.

$$\frac{dC_3(t_p)}{dt_p} = \frac{\beta C_k \left(\frac{t_p}{\alpha}\right)^{\beta-1} t_p - C_p + C_k \left(\frac{t_p}{\alpha}\right)^\beta}{t_p^2} = 0 \quad (18)$$

식 (18)을 정리하면 t_p^* 는 식 (19)로 도출된다[15, 20].

$$t_p^* = \alpha \left(\frac{C_p}{C_k(\beta-1)} \right)^{\frac{1}{\beta}} \quad (19)$$

따라서 시스템 교체 비용과 시스템 수리 비용이 사전에 정해져 있을 경우, 현재 운영되는 예방 교체 주기와 식 (19)에서 도출한 최적 교체 주기를 비교하여 경제성 분석을 실시한다.

4. 사례 연구

4.1 데이터 설명

사례 연구에서는 미국 NASA(National Aeronautics and Space Administration)의 CMAPSS 터보팬 엔진 데이터 셋을 RUL 예측 및 최적 예방 교체 주기 결정을 위하여 사용하였다. CMAPSS 데이터 셋은 공개(open)된 데이터로서, 많은 RUL 예측 연구에서 널리 활용되고 있다[13]. CMAPSS 데이터는 다음과 같은 특성을 가진다.

- 터보팬 엔진의 주요 부품인 팬(fan), 저압 압축기(Low Pressure Compressor; LPC), 고압 압축기(High Pressure Compressor; HPC), 저압 터빈(Low Pressure Turbine; LPT), 고압터빈(High Pressure Turbine; HPT)을 대상으로, 고장 및 열화의 영향을 감안한 센서 데이터를 얻을 수 있다[27].
- 센서 측정이 엔진 초기 가동부터 고장 발생까지(run-to-failure) 완전 자료 형태로서 중도 절단 없이 기록되었다.

- 시간 단위가 아닌 사이클 단위를 기준으로 센서값을 기록하므로, 가동 주기(cycle)마다 RUL 예측이 가능하다.
- 데이터 셋에 실제 RUL이 포함되어 ground truth(일종의 정답값)를 제공한다.
- 개별 데이터 셋은 학습 데이터와 테스트 데이터로 구성되어 있어서 기계학습을 위하여 별도의 데이터 분할이 필요 없다.
- 운용 설정 파라미터와 결합 모드 조합에 따라 FD001부터 FD004까지 4개의 서브셋(subset) 데이터 셋으로 구성된다. FD001과 FD002는 고압 압축기의 결합 모드가 인가된 데이터 셋이며, FD003과 FD004는 팬과 고압 압축기의 결합 모드가 인가된 데이터 셋이다[27].

<Table 1> CMAPSS Sensor Measurements

Symbol	Description	Unit	Use
T2	Total temperature at fan inlet	°R	
T24	Total temperature at LPC outlet	°R	o
T30	Total temperature at HPC outlet	°R	o
T50	Total temperature at LPT outlet	°R	o
P2	Pressure at fan inlet	psia	
P15	Total pressure in bypass-duct	psia	
P30	Total pressure at HPC outlet	psia	o
Nf	Physical fan speed	rpm	o
Nc	Physical core speed	rpm	o
epr	Engine pressure ratio	-	
Ps30	Static pressure at HPC outlet	psia	o
phi	Ratio of fuel flow to Ps30	pps/psi	o
NRf	Corrected fan speed	rpm	o
NRc	Corrected core speed	rpm	
BPR	Bypass ratio	-	o
farB	Burner fuel-air ratio	-	
htBleed	Bleed Enthalpy	-	o
Nf_dmd	Demanded fan speed	rpm	
PCNfR_dmd	Demanded corrected fan speed	rpm	
W31	HPT coolant bleed	lbm/s	o
W32	LPT coolant bleed	lbm/s	o

<Table 2> Overview of CMAPSS Datasets

Attribute	FD001	FD002	FD003	FD004
Engines in training set (unit)	100	260	100	249
Engines in test set (unit)	100	259	100	248
Max/Min for training set (cycle)	362/128	378/128	525/145	543/128
Max/Min for test set (cycle)	303/31	367/21	475/38	486/19

<Table 3> Performance Comparisons of Prediction at One Time

Performance	FD001		FD002		FD003		FD004	
	RF	XGB	RF	XGB	RF	XGB	RF	XGB
RMSE	24.45	24.39	32.20	31.69	42.89	33.23	41.61	38.66
MAE	19.25	20.31	25.78	25.12	31.05	25.74	34.54	31.55
R^2	0.65	0.66	0.64	0.65	-	0.36	0.42	0.50

<Table 1>은 각 데이터 서브셋에 포함된 21개 센서 속성을 나타낸다. 또한, 분석에 포함된 센서는 Use의 ‘o’로 표기하였다. <Table 2>는 데이터 셋 구성을 나타낸다.

4.2 RUL 예측

엔진 설비의 열화 상태 데이터를 학습하여 RUL 예측 모델을 생성한다. 모델 생성 도구는 파이썬(python)을, 기계 학습을 위하여 scikit-learn 라이브러리와 XGBoost 라이브러리를 이용하였다. 먼저 기계학습 알고리즘을 사용하기 위한 데이터 전처리를 실시하였다. CMAPSS 데이터 셋은 텍스트(txt) 파일 형식이므로 분석을 위하여 CSV 파일 형태로 변환하였다. 개별 센서 데이터는 레이블(label)이 안 되어 있는 상태이므로 센서별로 측정된 센서 명칭을 입력하였다. ‘Nf_dmd’, ‘PCNfR_dmd’, ‘P2’, ‘T2’, ‘farB’, ‘epr’ 센서는 일정한 값을 가지므로 분석에서 제외하였다. 그리고 운영 파라미터(operational parameters)인 ‘unit_number’, ‘setting_1’, ‘setting_2’, ‘setting_3’는 분석에 사용되지 않으므로 제거하였다. 또한, 센서 간의 상관분석을 실시하여 중복 경향이 있는 ‘P15’, ‘NRC’ 센서를 제외하였다. 그 후, 최소-최대 정규화(Min-Max normalization) 방법을 사용하여 모든 특성(feature)을 0-1의 범위 값으로 정규화하였다. CMAPSS 데이터 셋은 학습 데이터와 테스트 데이터로 사전에 분할되어 있으므로 이를 그대로 사용하였다.

3.1.2절에 설명한 대로, RF와 XGBoost 알고리즘에 두 가지 방법(① 모든 데이터를 한 번에 학습시키는 방법, ② 단일 엔진 단위로 그룹화한 후 하나씩 모델에 투입하여 가장 최근의 다섯 사이클을 예측하여 평균하는 방법)으로 학습하였다. <Table 3>은 ① 방법의 분석 결과, <Table 4>는 ② 방법의 분석 결과이다.

<Table 3>의 경우, 4개의 개별 셋을 비교할 때, FD001이 RF와 XGBoost 알고리즘 모두 RMSE, MAE 값이 낮고 R^2 이 높게 나타나서 가장 예측이 잘 되었다. 반면, FD003과 FD004는 예측 정확성이 상대적으로 낮은 것을 알 수 있다.

<Table 4>의 경우, FD001이 RF와 XGBoost 알고리즘 모두 RMSE, MAE 값은 낮고 R^2 이 높게 나타났다. 특히, ① 방법과 ② 방법을 비교한 경우에는 ② 방법이 예측

정확도가 높은 것으로 나타났다. 이는 동일한 데이터 셋과 동일한 알고리즘을 사용하더라도 학습 방법을 구체화시켰을 때 정확도가 높아짐을 확인할 수 있다. 그리고, FD001과 FD002에서는 일정 수준 이상의 성능이 나타났으나, FD003과 FD004에서는 낮은 성능이 나타났다. 이는 FD001과 FD002는 단수 개의 부품 고장 모드(고압 압축기 결함)만을 포함하고 있어 RUL 예측 정확도가 높았던 반면, FD003과 FD004는 복수 개의 부품 고장 모드(팬과 고압 압축기 결함)가 혼재되어 있어 열화 성능의 예측이 더 어렵기 때문이라고 추정된다.

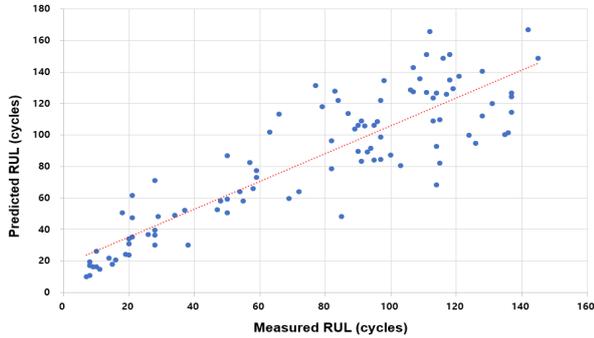
추가적으로, 사용된 RF와 XGBoost 알고리즘과 다른 기계학습 비선형 알고리즘간 성능을 비교하였다. 비선형 알고리즘으로는 보편적으로 사용하는 SVM과 ANN을 이용하였다. SVM은 RMSE=22.89, MAE=16.6, $R^2=0.69$ 의 성능을, ANN은 RMSE=22.42, MAE=15.4, $R^2=0.70$ 성능을 나타냈다. 이 경우에도 XGBoost가 성능이 다소 우월하게 나타났다. 그 이유는 XGBoost 장점인 DT의 오차를 순차적으로 줄이면서 예측 성능이 높아진 것이라고 추정된다.

<Figure 6>은 ② 방법을 활용한 FD001의 RUL 실제값과 예측값을 나타낸다. X축은 데이터 셋에서 주어진 측정값, Y축은 예측값을 나타낸다. 그리고 붉은색 점선은 선형 추세선(trend line)을 나타낸 것이다.

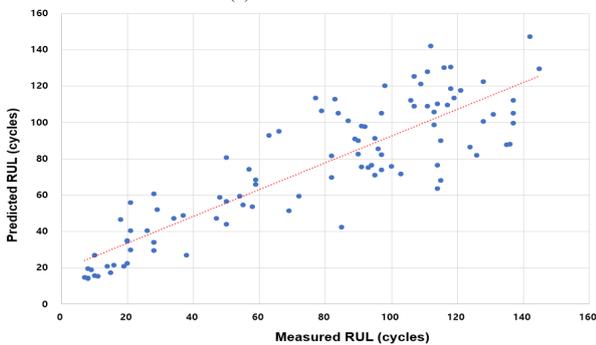
<Figure 6>의 예측값들은 학습된 모델에 테스트 데이터를 넣었을 때 결과값으로 얻은 RUL 예측 사이클이다. 각 엔진의 테스트 데이터 사이클에 RUL 예측 사이클을 합하면 고장 예측 시점을 얻을 수 있다. 예를 들어, ② 방법으로 학습하고 XGBoost를 이용한 1번 엔진의 테스트 데이터는 31 사이클이고, RUL 예측은 142 사이클이다. 따라서 이 엔진의 고장 예측 사이클은 31과 142의 합인, 173 사이클이 되는 것이다. <Table 4>에서 최소 RMSE와 MAE 그리고 최대 R^2 을 갖는 즉, 가장 예측이 잘 된 FD001의 XGBoost를 이용하여 최소 수리 블록 교체 시점을 결정하였다. <Table 5>는 FD001에 속한 100개 엔진의 각 고장 예측 시점을 도출한 결과이다. 이 고장 예측 시점을 활용하여 최적 예방 교체 전략을 수립하게 된다.

<Table 4> Performance Comparisons of Average of Last 5 Predictions

Performance	FD001		FD002		FD003		FD004	
	RF	XGB	RF	XGB	RF	XGB	RF	XGB
RMSE	22.17	20.49	28.15	32.48	36.12	27.77	32.75	31.36
MAE	17.78	16.01	21.73	24.18	26.21	20.44	26.43	24.78
R^2	0.72	0.76	0.73	0.64	0.24	0.55	0.64	0.67



(a) Random Forest



(b) XGBoost

<Figure 6> Comparison of Measured and Predicted RUL at FD0001

<Table 5> Failure Prediction of FD001(cycles)

Engine No.	Test	RUL	Failure prediction
1	31	142	173
2	49	120	169
3	126	51	177
4	106	70	176
5	98	75	173
...
98	121	66	187
99	97	109	206
100	198	34	232
Average	131	74	205

4.3 최소 수리 블록 교체

4.2절의 고장 예측 사이클을 기반으로 전체 엔진의 최적 예방 교체 주기를 도출한다. 이를 위하여 단위 사이클 당 평균 비용을 최소화 하는 시점을 구한다. CMAPSS 데이터는 사이클 단위를 활용하여 센서값을 기록하기 때문에 예방 보전을 실시하는 각 블록의 단위도 사이클 단위를 이용하여 분석하였다. 100개 엔진의 고장 예측 사이클 특징을 보면 가장 빠르게 고장이 예측된 엔진은 36

번 엔진의 147 사이클, 가장 늦게 고장이 예측된 엔진은 49번 엔진의 333 사이클이었다. 예측된 엔진의 평균 고장 사이클은 205 사이클이었다. 3.2절에 언급한 대로, <Table 5>의 예측된 고장 사이클이 Weibull 분포를 따른다고 가정하였기에 최대 우도 추정(Maximum Likelihood Estimation; MLE)으로 형상 모수와 척도 모수를 추정한다. <Table 6>은 추정된 결과이다.

<Table 6> Weibull Distribution Parameters

Dataset	shape parameter(β)	scale parameter(α)
FD001	5.58	220.65

최소 수리 블록 교체 모델을 활용하기 위해서는 개별 엔진을 새것으로 교체할 때의 비용과 최소 수리할 때의 비용이 사전에 정해져 있어야 한다. 최소 수리 비용은 두 가지 형태로 고려된다. 첫 번째는 고장 발생 시, 바로 대처하기 위해 재고로 준비된 중고 엔진을 사용하는 방법이다. 두 번째는 고장 발생 시, 고장 난 부분을 찾아서 부품을 수리하는 방법이다. 첫 번째 방법은 두 번째 방법보다 최소 수리 비용이 높으며, 고장 난 개별 부품을 분리하여 수리하기 어려운 경우에 적용한다. 두 번째 방법은 첫 번째 방법에 비해 비용은 적게 들지만 같은 주기 내에서 다른 부품이 고장 나면 또 수리를 위해 멈춰야 하는 단점이 있다. 본 사례 연구는 시스템 복잡성과 동일 주기 내 추가 고장 방지를 고려하여 첫 번째 방법을 적용한다. 일반적으로 수리 비용이 교체 비용보다 낮을 경우에 최소 수리 블록 교체가 실시되며, 두 방법은 최소 수리 비용의 차이만 있을 뿐, 방정식 적용에 있어서 차이는 없다. 현재 전체 엔진 교체 주기(t_p)는 200 사이클로 결정한다. 이는 <Table 5>의 고장 예측 사이클 평균값인 205 사이클의 내림 값이다. 또한, 개별 엔진의 새 엔진으로 교체 비용(C_p)은 2억원, 개별 엔진의 최소 수리 비용(C_k)은 1.6억 원으로 가정한다. 비용의 가정은 관련 문헌[22] 및 실제 엔진 가격 단위를 참고하였다. 이는 CMAPSS 데이터에 실제 유지 보수 비용이 포함되지 않은 것에서 기인한다.

식 (16)을 이용하여 엔진 교체 주기 200 사이클에서의 단위 사이클 당 평균 비용을 구하였다. 이 경우, 평균 비용은 146만 원이 소요된다.

$$C_3(200) = \frac{C_p + C_k \left(\frac{t_p}{\alpha}\right)^\beta}{t_p} = \frac{2 + 1.6 \left(\frac{200}{220.65}\right)^{5.58}}{200} = 0.0146 \text{ (억원)}$$

식 (19)를 이용하여 최소 수리 블록 교체에서의 최적

교체 시점 t_p^* 를 구하였다. 최적 교체 주기는 174.84 사이클이고, 이때의 평균 비용은 139만 원이다.

$$t_p^* = \alpha \left(\frac{C_p}{C_k(\beta-1)} \right)^{\frac{1}{\beta}}$$

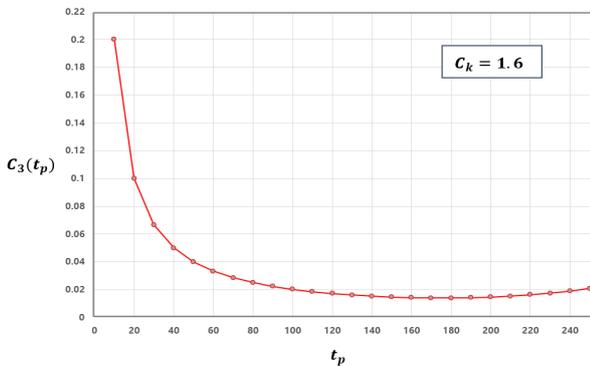
$$= 220.65 \left(\frac{2}{1.6(5.58-1)} \right)^{\frac{1}{5.58}} = 174.84$$

$$C_3(174.84) = \frac{C_p + C_k \left(\frac{t_p}{\alpha} \right)^\beta}{t_p}$$

$$= \frac{2 + 1.6 \left(\frac{174.84}{220.65} \right)^{5.58}}{174.84} = 0.0139(\text{억원})$$

현재 운영 중인 교체 주기와 제안 방법으로부터 도출된 최적 교체 주기는 약 16 사이클의 차이가 존재한다. 최적 교체 주기를 활용할 경우, 사이클 당 평균 비용은 기존 146만 원에서 139만 원으로 약 7만 원의 비용 감소 효과가 기대되었다.

<Figure 7>은 최소 수리 비용이 1.6억원으로 고정될 때, 최적 교체 주기와 평균 비용 관계를 그래프로 나타낸 것이다. 최적 교체 주기 174 사이클 전까지는 하강한 후, 그 이후로는 급격한 변동 없이 완만한 상승을 보인다.



<Figure 7> Relationship of $C_3(t_p)$ with respect to t_p when C_p is 1.6

<Table 7>은 최소 수리 비용의 변화와 새 엔진으로 교체 비용 변화에 따른 민감도 분석을 실시한 결과이다. 최소 수리 비용이 1억 원에서 1.8억 원으로 증가함에 따라서 최적 교체 주기는 빨라지고 평균 비용은 증가함을 알 수 있다. <Figure 7>과 <Table 7>을 통해서 최소 수리 비용이 고정될 때와 변화할 때의 최적 교체 주기 및 평균 비용을 비교할 수 있으며 이를 유지 보수 전략에 활용한다. <Table 8>은 최적 교체 주기에 변화에 따른 $f(t_p^*)$, $R(t_p^*)$, $h(t_p^*)$ 를 나타낸다. 최적 교체 주기가 짧아질수록 신뢰도는 증가하며 고장

률은 낮아짐을 알 수 있다. 본 사례 연구에서는 β 가 1보다 크게 나타나는 IFR 형태로서 식 (9)의 조건을 만족한다. 따라서, 최적 교체 주기에서의 고장률을 이용해서도 아래와 같이 평균 비용을 도출할 수 있다.

$$C_3(174.84) = C_k \times h(174.84)$$

$$= 1.6 \times 0.0087 = 0.0139(\text{억원})$$

<Table 7> Sensitivity Analysis Result for C_k (t_p^* : cycles, C_p , C_k and $C_3(t_p^*)$: 100 million won)

C_k	$C_p = 1.8$		$C_p = 2.0$		$C_p = 2.2$	
	t_p^*	$C_3(t_p^*)$	t_p^*	$C_3(t_p^*)$	t_p^*	$C_3(t_p^*)$
1.0	186.65	0.0117	190.20	0.0128	193.48	0.0139
1.2	180.65	0.0121	184.09	0.0132	187.26	0.0143
1.4	175.72	0.0125	179.07	0.0136	182.16	0.0147
1.6	171.57	0.0128	174.84	0.0139	177.85	0.0151
1.8	167.98	0.0131	171.19	0.0142	174.14	0.0154

<Table 8> $f(t_p^*)$, $R(t_p^*)$ and $h(t_p^*)$ According to t_p^*

t_p^*	$f(t_p^*)$	$R(t_p^*)$	$h(t_p^*)$
190.20	0.0083	0.6462	0.0128
184.09	0.0077	0.6949	0.0111
179.07	0.0071	0.7321	0.0097
174.84	0.0066	0.7611	0.0087
171.19	0.0062	0.7845	0.0079

4.4 평균 비용 최적화

기존의 200 사이클 주기로 교체를 실시할 경우, 100개의 엔진 중 47개 엔진이 한 블록 주기 내에서 고장이 발생하여 최소 수리가 실시된다. 최적 블록 주기인 174 사이클 주기로 교체를 실시할 경우, 최소 수리 교체 엔진이 19개로 줄게 된다. 단일 블록 주기 안에서 고장으로 인한 최소 수리 엔진의 수를 줄이기 위해서 블록 주기를 짧게 가져갈 수 있다. 그러나 성능이 강한 엔진들은 RUL이 많이 남았음에도 불구하고 함께 교체되므로 비용적 낭비가 발생할 수 있다. 반면 블록 주기를 길게 하면, 블록 주기 안에서 최소 수리 엔진이 증가하므로 유지 비용 부담이 증가하는 문제가 발생한다. 이러한 상충 관계에서의 최적 의사 결정을 위하여 비용 관점의 최적화를 실시하였다. 100개 엔진을 고장 예측 사이클에 따라 나누어 특정 구간에 할당하고, 구간별 최소 유지 비용을 도출한다. 기존 최적화 방법은 100개의 엔진이 단일 블록 주기 끝에서 일제히 교체되었다. 비용 관점의 최적화 방법은 성능이 약한 엔진은 블록 주기를 짧게 하여서 최소 수리 엔진을 줄이는 방법으로 비용을 감소시킨다. 또한,

성능이 강한 엔진은 블록 주기를 길게 하여서 RUL이 남아있어도 이른 시기에 교체되는 단점을 보완하여 평균 유지 비용을 줄인다. 여기서 엔진 성능의 판단은 기존 최적화 블록 주기에 따라 결정된다. 예를 들어, 개별 엔진의 고장 예측 사이클이 기존 최적 교체 주기인 174 사이클보다 빠르다면 성능이 약한 엔진으로 간주한다.

<Table 9>와 같이 100개 엔진을 성능 순으로 3개 구간(stage)으로 나누어 분석하였다. 1 구간은 최적 교체 주기인 174 사이클보다 낮은 성능(하위 19%)을 보이는 구간이며, 엔진 수는 19개이다. 2 구간은 중간 성능 구간으로 상위 구간과 하위 구간을 제외한 엔진이며 56개, 3 구간은 상위 25% 이상 구간이며 25개 엔진으로 구성된다.

<Table 9> Overview of Stage Analysis

Attribute	Stage 1	Stage 2	Stage 3
Number of engines	19	56	25
Max/Min cycles	173/147	222/175	333/223
Mean cycles	164	199	252

<Table 10>은 앞서 가정한 엔진 비용과 새로 추정된 모수를 이용하여 최적 교체 주기 및 사이클 당 평균 비용을 도출한 결과이다. 1 구간과 2 구간의 엔진 수를 합하고, 2 구간과 3 구간을 합하여서 형상 모수와 척도 모수를 MLE로 추정하였다. 이렇게 하위 구간과 상위 구간을 제외하고 추정한 이유는 각 구간 단위로 분석할 경우, 엔진 수가 적어서 모수 추정이 부정확할 수 있으며 통계적으로 모집단 대표성을 잃을 수 있기 때문이다. 1 구간에서 164.35 사이클 블록 주기를 활용할 경우, 최소 수리 블록 교체는 기존의 19개 엔진에서 9개 엔진으로 줄게 된다. 평균 비용은 사이클 당 139만 원에서 134만 원으로 감소한다. 또 2 구간과 3 구간에서 182.26 사이클 블록 주기를 활용할 경우, 사이클 당 131만 원으로 분석되어 현재보다 약 8만 원 정도 비용 감소 효과가 기대되었다.

<Table 10> Result of Stage Analysis (t_p^* : cycles, $C_3(t_p^*)$: 100 million won)

Stages	β	α	t_p^*	$C_3(t_p^*)$
Stage 1 + Stage 2	10.77	198.92	164.35	0.0134
Stage 2 + Stage 3	6.15	229.44	182.26	0.0131

5. 결론

본 논문에서는 설비 시스템의 RUL을 예측하고 최소 수리 블록 교체 모델을 활용하여 최적 예방 정비 시점

의 도출 방법을 제안하였다. 이를 위하여 기계학습 알고리즘인 RF와 XGBoost를 이용하여 다수의 동일 설비로 이루어진 시스템의 RUL을 예측한 후, 경제성 관점에서 최적 교체 주기를 도출하는 세부 방법을 개발하였다. RUL 예측 방법이나 성능 평가에 초점을 맞춘 기존 연구에 대비하여, 본 논문은 RUL 예측 결과를 스케줄 기반 예방 보전에 적용함으로써 어느 시점에서 수리 및 교체가 이루어져야 비용면에서 효과적인지 찾는 방법을 제안한 것이다. 이 제안 방법을 통해서 고장률이 증가하는 시스템의 예상치 못한 고장을 사전에 예방하고 유지 비용의 절감 효과를 기대할 수 있다. 특히 다수의 동일 설비 시스템에서 구간을 나누어 최적화를 실시함으로써, 설비 성능이 약한 구간에서 교체 주기를 짧게 실시할 수 있게 되어 수리 비용 감소 효과를 증진할 수 있다.

본 연구의 한계 및 추후 연구는 다음과 같다. 첫째, RUL 예측 방법으로 기계학습의 앙상블 기법을 활용하였지만 실제 현장의 다양한 공정 데이터를 적합하게 분석하기에는 한계가 있다. 이는 주어진 공정 및 시스템에서 특정 알고리즘이 항상 높은 예측 정확성을 보장하지 않기 때문이다. 특히, 사례 연구에서는 RUL 예측이 우수한 데이터 셋을 이용하여 제안 방법의 타당성을 확인하였다. 그러나, RUL 예측이 항상 잘 되는 것은 아니다. 일부 데이터 셋에서는 RUL 예측 성능이 높게 나타나지 않았으므로, 제안 방법을 그대로 적용하기에는 어려움이 존재한다. 추후 연구에서는 고장 모드에 따른 RUL 예측이 가능하도록 FDC(fault detection and classification)나 FMEA(failure mode and effects analysis)와 같은 물리적인 고장 모드 분석과 병행하여, 데이터 전처리 방법 및 기계학습 또는 딥러닝 알고리즘을 활용할 계획이다. 둘째, 스케줄 기반 예방 보전의 영역을 다루었으나 상태 기반 보전의 영역을 다루지는 못하였다. 예상하지 못한 고장을 줄이기 위해서는 시스템 열화 상태를 감안한 상태 기반 접근법이 필요하다. 추후에는 시스템의 이상 감지 및 고장 진단의 효과적 분석을 위하여 다변량 관리도와 같은 모니터링 기법을 RUL 예측에 접목할 예정이다.

Acknowledgement

This research was supported by the Ministry of SMEs and Startups, Republic of Korea, under ‘Continuous Process Manufacturing Standardization of Shared Data between Facilities/Factories/Businesses in Characteristic Industries’ in ‘Smart Manufacturing Innovation R&D Program’ (RS-2022-00140694).

References

- [1] Aydemir, G. and Acar, B., Anomaly monitoring improves remaining useful life estimation of industrial machinery, *Journal of Manufacturing Systems*, 2020, Vol. 56, pp. 463-469.
- [2] Basri, E.I., Abdul Razak, I.H., Ab-Samat, H., and Kamaruddin, S., Preventive maintenance (PM) planning: a review, *Journal of Quality in Maintenance Engineering*, 2017, Vol. 23, No. 2, pp. 114-143.
- [3] Biggio, L. and Kastanis, I., Prognostics and health management of industrial assets: Current progress and road ahead, *Frontiers in Artificial Intelligence*, 2020, Vol. 3, 578613.
- [4] Cailian, L. and Chun, Z., Life prediction of battery based on random forest optimized by genetic algorithm, *IEEE International Conference on Prognostics and Health Management (ICPHM)*, 2020, 19950220.
- [5] Chen, T. and Guestrin, C., XGBoost: A scalable tree boosting system, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785-794.
- [6] Chen, X., Jin, G., Qiu, S., Lu, M., and Yu, D., Direct remaining useful life estimation based on random forest regression, *Global Reliability and Prognostics and Health Management (PHM-Shanghai)*, 2020, 20297708.
- [7] Dong, W., Liu, s., Cao, Y., and Bae S.J., Time-based replacement policies for a fault tolerant system subject to degradation and two types of shocks, *Quality and Reliability Engineering International*, 2020, Vol. 36, No. 7, pp. 2338-2350.
- [8] Dong, X., Yu, Z., Cao, W., Shi, Y., and Ma, Q., A survey on ensemble learning, *Frontiers of Computer Science*, 2020, Vol. 14, No. 2, pp. 241-258.
- [9] Han, Y.C. and Kim, J.Y., Developing a XGBoost trading system based on N-period volatility labeling in the stock market, *Journal of the Korean Data & Information Science Society*, 2021, Vol. 32, No. 5, pp. 1049-1070.
- [10] Han, Y.J. and Joe, I.W., Prediction of Ad Clicks Using Early Stop Based on XGBoost, *The Journal of Korean Institute of Communications and Information Sciences*, 2021, Vol. 46, No. 6, pp. 993-1000.
- [11] Kim, R., Kim, K.M., and Ahn, J.H., Comparison between random forest and recurrent neural network for photovoltaic power forecasting, *Journal of Korean Society of Environmental Engineers*, 2021, Vol. 45, No. 3, pp. 347-355.
- [12] Lee, Y.D., Lim, M.K., and Choi, D.G., Short-term load forecasting for an individual building using shape-based clustering and random forest, *Journal of the Korean Operations Research and Management Science Society*, 2020, Vol. 37, No.4, pp. 21-31.
- [13] Lei, Y., Li, N., Guo, L., Li N., Yan, T., and Lin, J., Machinery health prognostics: A systematic review from data acquisition to RUL prediction, *Mechanical Systems and Signal Processing*, 2018, Vol. 104, pp. 799-834.
- [14] Li, F., Zhang, L., Chen, B., Gao, D., Cheng, Y., Zhang, X., Yang, Y., Gao, K., Huang, Z., and Peng, J., A Light Gradient Boosting Machine for Remaining Useful Life Estimation of Aircraft Engines, *21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, 18308685.
- [15] Lim, J.H., Won, D.Y., Sim, H.S., Park, C.H., Koh, K.J., Kang, J.G., and Kim, Y.S., A study on condition-based maintenance policy using minimum-repair block replacement, *Journal of Applied Reliability*, 2018, Vol. 18, No. 2, pp. 114-121.
- [16] Liu, L., Wang, L., and Yu, Z., Remaining useful life estimation of aircraft engines based on deep convolution neural network and lightGBM combination model, *International Journal of Computational Intelligence Systems*, 2021, Vol. 14, 165.
- [17] Nakagawa, T., A summary of block replacement policies, *Operations Research*, 1979, Vol. 13, No. 4, pp. 351-361.
- [18] Patil, S., Patil, A., Handikherkar, V., Desai, S., Phalle, V.M., and Kazi, F.S., Remaining useful life (rul) prediction of rolling element bearing using random forest and gradient boosting technique, *International Mechanical Engineering Congress and Exposition, IMECE 2018-87623*, 2018.
- [19] Que, Z. and Xu, Z., A data-driven health prognostics approach for steam turbines based on XGBoost and DTW, *IEEE Access*, 2019, Vol. 7, pp. 93131-93138.
- [20] Rebaiaia, M.L. and Ait-kadi, D., Maintenance policies with minimal repair and replacement on failures: analysis and comparison, *International Journal of Production Research*, 2020, Vol. 59, No. 23, pp. 6995-7017.
- [21] Rebaiaia, M.L., Ait-Kadi, D., and Jamshidi, A., Periodic replacement strategies: optimality conditions and numerical performance comparisons, *International Journal of Production Research*, 2017, Vol. 55, No. 23, pp. 7135-7152.
- [22] Seo, S.G., Kim, H.G., Bae, S.J., and Yun, W.Y., Reliability

engineering, (3rd ed), Gyomoonsa, 2020.

- [23] Shi, J., Yu, T., Goebel, K., and Wu, D., Remaining useful life prediction of bearings using ensemble learning: The impact of diversity in base learners and features, *Journal of Computing and Information Science in Engineering*, 2021, Vol. 21, No. 2, 021004.
- [24] Tong, Z., Miao, J., Tong, S., and Lu, Y., Early prediction of remaining useful life for Lithium-ion batteries based on a hybrid machine learning method, *Journal of Cleaner Production*, 2021, Vol. 317, 128265.
- [25] Wang, X., Zeng, D., Dai, H., and Zhu, Y., Making the right business decision: Forecasting the binary NPD strategy in Chinese automotive industry with machine learning methods, *Technological Forecasting & Social Change*, 2020, Vol. 155, 120032.
- [26] Wu, J.Y., Wu, M., Chen, Z., Li, X., and Yan, R., A joint classification-regression method for multi-stage remaining useful life prediction, *Journal of Manufacturing Systems*, 2021, Vol. 58, pp. 109-119.
- [27] Yun, Y., Kim, S., Cho, S.H., and Choi, J.H., Neural network based aircraft engine health management using C-MAPSS data, *Journal of Aerospace System Engineering*, 2019, Vol. 13, No. 6, pp. 17-25.

ORCID

Youngsuk Choo | <http://orcid.org/0000-0001-7071-9721>

Seung-Jun Shin | <http://orcid.org/0000-0002-3587-5507>