논문 2024-19-19

연속학습을 활용한 경량 온-디바이스 AI 기반 실시간 기계 결함 진단 시스템 설계 및 구현

(Design and Implementation of a Lightweight On-Device AI-Based Real-time Fault Diagnosis System using Continual Learning)

김 영 준, 김 태 완, 김 수 현, 이 성 재, 김 태 현*
(Youngjun Kim, Taewan Kim, Suhyun Kim, Seongjae Lee, Taehyoun Kim)

Abstract : Although on-device artificial intelligence (AI) has gained attention to diagnosing machine faults in real time, most previous studies did not consider the model retraining and redeployment processes that must be performed in real-world industrial environments. Our study addresses this challenge by proposing an on-device AI-based real-time machine fault diagnosis system that utilizes continual learning. Our proposed system includes a lightweight convolutional neural network (CNN) model, a continual learning algorithm, and a real-time monitoring service. First, we developed a lightweight 1D CNN model to reduce the cost of model deployment and enable real-time inference on the target edge device with limited computing resources. We then compared the performance of five continual learning algorithms with three public bearing fault datasets and selected the most effective algorithm for our system. Finally, we implemented a real-time monitoring service using an open-source data visualization framework. In the performance comparison results between continual learning algorithms, we found that the replay-based algorithms outperformed the regularization-based algorithms, and the experience replay (ER) algorithm had the best diagnostic accuracy. We further tuned the number and length of data samples used for a memory buffer of the ER algorithm to maximize its performance. We confirmed that the performance of the ER algorithm becomes higher when a longer data length is used. Consequently, the proposed system showed an accuracy of 98.7%, while only 16.5% of the previous data was stored in memory buffer. Our lightweight CNN model was also able to diagnose a fault type of one data sample within 3.76 ms on the Raspberry Pi 4B device.

Keywords : Continual Learning, Deep Learning, Machine Fault Diagnosis, On-Device AI

Ⅰ. 서 론

회전 기계는 가스 터빈, 제트 엔진과 같이 산업 현장에서 주요한 역할을 하는 기계 장치의 핵심 구성 요소이다. 회전 기계에서 결함이 발생하면 기계 유지 보수 비용과 사고 위험성이 높아지므로 회전 기계 결함을 신속하고 정확하게 진단하는 것은 중요한 과제이다. 특히 베어링은 회전 기계에서 결함이 빈번히 발생하는 구성 요소로, 회전 기계의 한 종류인 유도 전동기에서는 베어링 결함이 전체 결함 원인의 44%를 차지한다 [1]. 초기 연구들은 푸리에 변환, 웨이블릿 변환 등의 신호 처리 기반 특징 추출 방법을 사용하여 회전 기계 결함 여부를 판단하였다. 그러나 이러한 방법은 전문적인 신호 처리 지식을 필요로 한다는 한계가 있다 [2, 3].

최근에는 이 문제를 해결하기 위해 딥 러닝 (deep learning)을 활용하여 기계 결함 여부를 진단하는 연구가 많이 수행되고 있다 [2-6]. 딥 러닝을 사용하면 모델이 데이터 특징을 자동으로 추출할 수 있다는 장점이 있지만, 이를 위해서는 다량의 학습 데이터가 필요하다. 또한 딥 러닝 모델의 연산량은 기존의 신호 처리 또는 머신러닝 기반 기계 결함 진단 방법의 연산량보다 크기 때문에 딥 러닝 모델의 학습과 추론은 일반적으로 클라우드 서버에서 수행된다.

클라우드 서버에서 모델 추론을 수행하기 위해서는 데이터를 수집하는 엣지 디바이스 (edge device)와 클라우드 서버 간 데이터 통신이 선행되어야 하는데, 통신 과정에서 네트워크 환경 변화에 따른 시간 지연이 발생할 수 있다 [7]. 이러한 시간 지연은 안전과 직결된 기계 결함 진단을 실시간으로 수행하는 데에 큰 문제로 작용한다.

클라우드 서버 기반 추론 방식의 대안으로, 클라우드에서 학습된 모델을 엣지 디바이스로 배포하여 엣지 디바이스가 데이터 수집과 모델 추론을 모두 수행하도록 하는 온-디바이스 (on-device) AI (Artificial Intelligence) 기반 시스템이 최근에 주목받고 있다 [8, 9]. 엣지 디바이스는 클라우드 서버보다 연산 능력과 메모리 공간이 제한되어 있으므로 기존 온-디바이

*Corresponding Author (thkim@uos.ac.kr)
Received: Jan. 26, 2024, Revised: Mar. 7, 2024, Accepted: Apr. 13, 2024.
Y. J. Kim: University of Seoul (M.S. Student)
T. W. Kim: University of Seoul (M.S. Student)
S. H. Kim: University of Seoul (M.S. Student)
S. J. Lee: University of Seoul (Ph.D. Candidate)
T. H. Kim: University of Seoul (Prof.)
※ 이 논문은 정부 (과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2022R1F1A1060231).

스 AI 기반 실시간 기계 결함 진단 연구는 모델을 경량으로 설계하거나 모델 연산을 최적화하는 데에 집중한다 [10, 11]. 그러나 이 연구들은 모델이 현장의 운용 환경 변화에 따라 새로 추가될 수 있는 데이터를 재학습하고 엣지 디바이스에 재배포되는 과정을 고려하지 않는다는 한계점이 있다.

모델을 재학습하는 일반적인 방법으로는 조인트 훈련 (joint training), 파인튜닝 (fine-tuning), 전이학습 (transfer learning)이 있다. 조인트 훈련 방식은 데이터 갱신 시 과거 학습에 사용한 전체 데이터와 새로운 데이터를 모두 포함하여 모델을 재학습한다. 이 방식은 추론 성능이 좋지만, 데이터 누적으로 인해 학습과 데이터 저장에 사용되는 비용이 지속적으로 증가한다는 문제점이 있다 [12]. 파인튜닝과 전이학습은 과거 데이터를 사용하지 않고 새로운 데이터만을 모델 재학습에 사용한다 [13, 14]. 그러나 파인튜닝과 전이학습을 사용하면 새로운 데이터를 학습하는 과정에서 이전에 학습한 데이터에 대한 판별 성능이 저하되는 상황인 파괴적 망각 (catastrophic forgetting)이 발생할 수 있다 [15]. 따라서 일반적인 모델 재학습 방식을 온-디바이스 AI 기반 기계 결함 진단 시스템에 직접 적용하기에는 어려움이 따른다.

최근에는 기존 모델 재학습 방식의 한계를 극복하기 위해 연속학습 (continual learning) 기법이 활발히 연구되고 있다 [16]. 연속학습 기법은 딥 러닝 모델이 새로운 데이터를 학습하는 동시에 이전 데이터로부터 학습한 지식 (knowledge)을 유지하는 방법으로, 파인튜닝과 전이학습의 문제점인 파괴적 망각을 극복할 수 있다. 또한 연속학습은 모델 재학습 시 조인트 훈련과 달리, 이전 데이터를 사용하지 않거나 일부분만을 활용하기 때문에 학습과 데이터 저장에 사용되는 비용이 크게 줄어든다.

본 연구는 연속학습 기법과 온-디바이스 AI를 활용한 실시간 기계 결함 진단 시스템을 제안한다. 제안하는 시스템은 연속학습, 모델 배포, 모니터링 서비스를 수행하는 클라우드 서버와 실시간 결함 진단을 수행하는 엣지 디바이스로 구성된다. 우선 모델 재배포 비용을 감소시키고 엣지 디바이스 상에서의 실시간 추론을 용이하게 하기 위해, 경량 1차원 합성곱 신경망 (1D convolutional neural network) 모델을 개발하였다. 또한, 다섯 개의 대표적인 연속학습 알고리즘을 비교하여 결함 진단 응용 특성에 가장 적합한 알고리즘을 선정한 후 알고리즘의 하이퍼파라미터를 조정하여 성능을 최대화하였다. 엣지 디바이스에서 수집된 데이터 및 결함 추론 결과는 클라우드로 전송된 후 웹 기반 모니터링 서비스를 통해 사용자가 원격으로 기계 상태를 모니터링할 수 있도록 하였다. 최종적으로, 클라우드 서버와 Raspberry Pi 4B 엣지 디바이스 상에서 세 종류의 공개 베어링 결함 데이터셋이 연속적으로 추가되는 상황을 가정하여 시스템의 성능을 평가하였다. 제안된 시스템은 데이터 분포가 변화하는 환경에 적응하면서 실시간 결함 진단을 수행할 수 있다는 점에서 의미가 있다.

Ⅱ. 관련 연구

1. 온-디바이스 AI 기반 기계 결함 진단 및 모니터링 시스템

최근 클라우드 서버에서 훈련된 모델을 엣지 디바이스로 배포하여 실시간 기계 결함 진단을 수행하기 위한 경량 딥 러닝 모델을 개발하는 연구가 많이 이루어지고 있다 [17-19]. 대표적으로 데이터 전처리와 연산 비용을 줄이기 위해 1차원 합성곱 신경망 기반 경량 결함 진단 모델을 개발하고 이를 원본 진동 신호에 대해 학습시켜 훈련과 추론에 걸리는 시간을 크게 감소시킨 사례가 있다 [17, 18]. Hou 등의 연구 [19]는 1차원 합성곱 신경망 기반 경량 모델을 컴퓨팅 자원이 제한된 무선 센서 네트워크 (wireless sensor network)의 센서 노드에 내장시킨 뒤 진단 결과만을 서버로 전송하여, 전송되는 데이터의 양과 전력 소비를 감소시키는 결함 진단 방식을 제안하였다.

또한 센서와 컴퓨팅 시스템 등의 급격한 발전으로 데이터 기반 기계 상태 모니터링에 관한 관심이 증가함에 따라 엣지 디바이스 상에서 실시간 기계 결함 진단을 수행하는 동시에 데이터의 변화 추이와 통계량을 실시간으로 확인할 수 있는 모니터링 시스템에 관한 연구도 많이 이루어지고 있다 [20-22]. Gültekin 등의 연구 [20]는 LeNet-5 합성곱 신경망 기반 모델과 모니터링 프레임워크를 사용하여 실시간 기계 결함 진단 및 상태 모니터링 시스템을 구현하고 자율 이동 차량 (autonomous transfer vehicle)을 대상으로 성능 검증을 수행하여 높은 결함 판별 정확도를 얻었다. Qian 등의 연구 [21]는 심각도가 높은 결함이 감지될 때 LCD로 결함 진단 결과를 모니터링하고 엣지 컴퓨팅 노드에 배포된 모델을 사용하여 회전 기계의 결함 진단과 동적 제어를 동시에 수행하는 시스템을 제시하였다. Ding 등의 연구 [22]는 AlexNet을 단순화한 모델을 STM32H743 보드에 배포한 뒤 해당 모델이 실시간으로 추론한 베어링 결함 상태 및 진동 데이터의 변동 추이를 LCD 화면을 통해 확인할 수 있도록 하였다. 그러나 위 연구들은 특정 시점에서 수집된 데이터셋을 사용하여 모델을 학습하고 검증하기 때문에, 시간이 지남에 따라 데이터의 분포와 특징이 변화하는 실제 산업 현장 환경에서의 성능을 보장하기 어렵다.

2. 연속학습

데이터가 변화하는 환경에 적응할 수 있는 딥 러닝 모델의 필요성이 증가함에 따라, 이를 충족시키기 위해 다양한 연속학습 알고리즘이 연구되고 있다 [15, 23-29]. 연속학습 알고리즘은 크게 재연 (replay) 방식 알고리즘과 정규화 (regularization) 방식 알고리즘으로 나눌 수 있다 [23].

재연 방식 알고리즘은 이전 학습 단계의 데이터 샘플 일부를 메모리 버퍼에 저장해 둔 후, 새로운 학습 단계에서 현재 데이터와 함께 학습에 사용하여 모델을 업데이트하는 방식이다. 대표적인 재연 방식 알고리즘으로는 ER (Experience Replay) [24], GEM (Gradient Episodic Memory) [25], A-GEM (Averaged GEM) [26] 등이 있다. ER은 과거 데이터의 일부를 버퍼에 저장하고 새로운 데이터와 함께 재학습하는 방법이며, GEM과 A-GEM은 손실 함수의 기울기 (gradient) 정보를 활용하여, 새로운 데이터를 학습하면서도 이전 데이터에 대한 성능이 크게 감소하지 않도록 제어하는 방법이다. Vödisch 등의 연구 [27]는 자동차 도로 이미지에

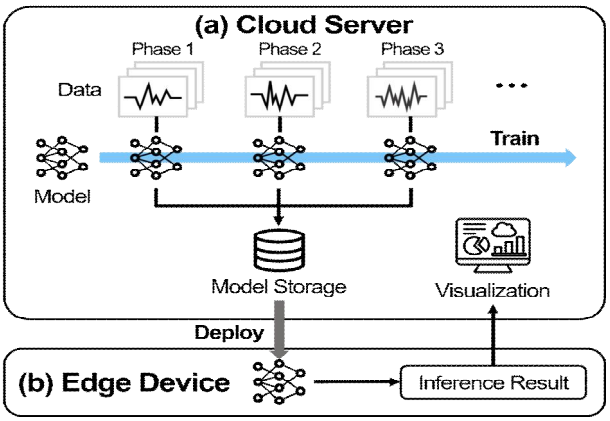


그림 1. 제안하는 시스템 구조
Fig. 1. Schematic architecture of the proposed system

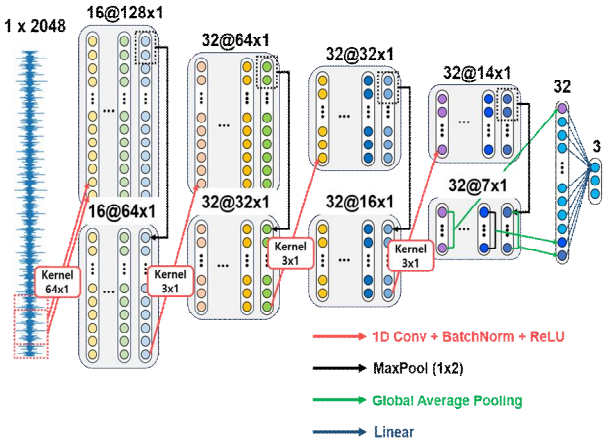


그림 2. 경량 딥 러닝 모델의 구조
Fig. 2. Lightweight deep learning model architecture

대한 깊이 추정 (depth estimation)과 이미지 세그멘테이션 (image segmentation) 작업을 수행하는 CoDEPS 시스템을 개발하였다. CoDEPS는 ER 알고리즘을 사용하여 실제 주행 환경에서 모델의 성능을 유지하도록 하였다.

반면 정규화 방식 알고리즘은 이전 단계에서 학습된 패턴을 유지하기 위해 가중치에 정규화 항을 적용하여 새로운 데이터를 학습할 때 발생하는 가중치의 급격한 변화를 제한하는 방식이다. 대표적인 정규화 방식 알고리즘으로는 LwF (Learning without Forgetting) [28]와 EWC (Elastic Weight Consolidation) [15] 등이 있다. LwF는 새로운 데이터를 학습할 때 이전 모델의 출력값을 보존하는 정규화 항을 사용하여 이전 지식을 보존하는 방법이고, EWC는 모델 파라미터의 중요도를 평가하고 이전 단계 데이터에서 중요하게 사용된 파라미터의 변화를 제한함으로써 이전 지식을 유지하는 방법이다. Maschler 등의 연구 [29]는 터보팬 엔진의 잔여 수명 예측을 위해 LSTM (Long Short-Term Memory) 모델에 EWC 알고리즘을 적용하였다. 이 연구에서 사용된 TEDSDS (Turbofan Engine Degradation Simulation Data Set) [30]는 각각 다른 조건과 시나리오에서 운행되는 여러 대의 터보팬 엔진의 시계열 데이터를 포함하고 있다. 이 데이터를 통해 다양한 데이터 분포 환경에서도 터보팬 엔진의 잔여 수명을 예측할 수 있는 모델을 제시하였다.

앞선 연구들은 연속학습을 도입한 딥 러닝 모델을 제시하였으나, 차량 혹은 터보팬 엔진과 같은 디바이스에서 수집된 데이터를 바탕으로 서버에서 결함을 판별하는 클라우드 서버 기반 추론 방식을 사용한다. 이 경우, 데이터를 서버로 전송하는 데 추가적인 시간이 발생하고 추론 결과를 얻기까지 시간이 지연될 수 있다는 한계점이 있다.

Ⅲ. 본 론

1. 시스템 설계

본 연구에서 제안하는 시스템의 전체적인 구성은 그림 1과 같다. 그림 1 (a)와 같이, 클라우드 서버에서는 모델이 시간에 따라 분포와 특성이 변화하는 데이터셋을 단계 (phase) 별로 학습한다. 단계란 모델 학습이 수행되는 시점을 의미하며, 각 단계에서는 연속학습 알고리즘을 사용하여 이전 단계의 데이터 특징을 재학습한다. 단계를 거치며 학습된 모델은 버전별로 클라우드 서버의 모델 스토리지에 저장되며 모델이 업데이트되면 엣지 디바이스로 모델의 최신 버전이 배포된다. 엣지 디바이스는 그림 1 (b)와 같이 클라우드로부터 학습된 모델을 전송받아 실시간으로 기계 결함을 진단한다. 수집된 데이터의 통계량과 결함 진단 결과는 클라우드 서버로 전송되며, 오픈소스 모니터링 프레임워크인 Grafana [31]를 통해 시각화된다. 시스템 사용자는 웹 환경에서 모니터링 결과를 확인하고 기계 상태를 판단할 수 있다.

또한 본 시스템은 분포가 다른 데이터가 지속적으로 수집되는 환경을 가정하므로, 모델의 훈련 및 배포가 빈번히 이루어져야 한다. 이 과정에서 소요되는 시간과 네트워크 리소스 사용 경감을 위하여 1차원 합성곱 신경망 기반 경량 딥 러닝 모델을 그림 2와 같이 설계하였다. 먼저 첫 번째 합성곱 계층의 커널 크기를 크게 할 경우 고주파수 잡음의 영향을 최소화하면서 기계의 저주파, 중주파수 진동 특징을 집중적으로 추출할 수 있다는 것을 보인 기존 연구 [5]를 참고하여 첫 번째 합성곱 계층의 커널 크기를 64로 설정하였다. 이후 나머지 합성곱 계층의 커널 크기를 3으로 설정하여 기계에서 주로 발생하는 진동의 복잡한 특징을 추출할 수 있도록 하였다. 모델은 진동 데이터의 특징을 추출하는 네 개의 합성곱 계층과 데이터의 클래스를 판별하는 완전 연결 (fully connected) 계층으로 구분되며, 각 합성곱 계층에서는 설정된 커널 크기에 따른 1차원 합성곱, 배치 정규화 (batch normalization), ReLU 활성화 함수, 길이가 2인 최대 풀링 (max pooling) 연산이 순차적으로 수행된다. 네 번째 합성곱 계층의 최대 풀링 연산 후에는 완전 연결 계층에서의 연산을 최소화하기 위해 전역 평균 풀링 (global average pooling)을 사용하여 각 채널의 노드들을 하나의 평균값으로 계산한 뒤, 완전 연결 계층의 입력으로 넣어 클래스에 대한 판별을 진행하도록 하였다. 입력 데이터 길이가 2,048일 때 모델 구조 세부사항은 표 1과 같으며 C는 채널 수, K는 커널 크기, Pool은 풀링 크기, S는 스트라이드 크기, Pad는 패딩 크기를 의미한다.

표 1. 모델 구조의 세부사항
Table 1. Details of the model architecture

Layer	C	K	Pool	S	Pad	Output Size
Input	-	-	-	-	-	2,048
Conv[1)]	16	64	2	16	24	16×64
Conv	32	3	2	1	same	32×32
Conv	32	3	2	1	same	32×16
Conv	32	3	2	1	0	32×7
GAP[2)]	-	-	-	-	-	32
FC[3)]	-	-	-	-	-	3

1) Conv: Convolutional Layer,
2) GAP: Global Average Pooling Layer,
3) FC: Fully Connected Layer

2. 데이터셋

본 연구는 단계 변화에 따라 데이터의 특성이 달라지는 상황을 연출하기 위해 특성이 다른 공개 회전 기계 베어링 결함 데이터셋인 CWRU (Case Western Reserve University) [32], MFPT (Machinery Failure Prevention Technology) [33], Ottawa (Ottawa University) [34]를 각각 그림 1 (a)에서 제시한 단계 1, 2, 3의 학습 데이터로 사용하였다. 각 데이터셋은 구름 베어링 (rolling element bearing)의 다양한 결함 상황에서 수집된 진동 데이터를 제공한다. 회전 기계에서 많이 쓰이는 구름 베어링은 내륜 (inner race), 외륜 (outer race), 볼 (ball)의 세 가지 요소로 구성된다. 구름 베어링의 경우 반복적인 사용으로 각 요소에 결함이 발생할 수 있으며, 결함이 발생한 요소가 다른 구성 요소와 부딪힐 때마다 발생하는 충격으로 인해 진동 데이터의 파형이 변화한다 [35]. 그림 3은 구름 베어링에서 발생할 수 있는 대표적인 결함인 내륜 결함, 외륜 결함, 볼 결함을 보여 준다.

표 2는 각 데이터셋에서 제공하는 데이터의 세부 특성을 보여 준다. 본 연구에서는 결함을 정상 (N), 내륜 결함 (IR), 외륜 결함 (OR), 볼 결함 (B)과 두 종류 이상의 복합적인 결함 (compound fault, C)으로 구분하였다. MFPT 데이터셋은 B, C에 해당하는 결함을 제공하지 않으므로 본 연구에서는 N, IR, OR 상태만을 대상으로 단계별 데이터를 구성했다. 또한, 각 데이터셋의 규모가 다르다는 점을 고려하여 표 3과 같이 각 단계마다 동일한 수의 샘플을 무작위로 추출하여 데이터를 구성하고, train, validation, test 데이터를 6:2:2 비율로 분할하였다.

3. 연속학습 알고리즘 선정

본 연구에서는 제안하는 시스템에 가장 적합한 연속학습 알고리즘을 선정하기 위해 대표적인 재연 방식 알고리즘인 ER, GEM, A-GEM과 정규화 방식 알고리즘인 LwF, EWC를 후보로 선정하고 성능 비교 실험을 진행하였다. 실험은 표 4와 같은 사양의 클라우드 컴퓨팅 환경에서 수행되었으며, 연속학습을 적용하지 않는 Naive 알고리즘과의 성능 비교도 함께 진행하였다.

성능 평가를 위한 주요 지표로는 각 단계의 학습을 마친 후 측정된 테스트 데이터셋에 대한 정확도 (accuracy)와 망각도 (forgetting)를 사용하였다. 정확도는 각 단계의 학습이 완료될 때마다, 현재 단계를 포함한 모든 이전 단계의 테스트 데이터를 사용하여 계산된다. 예를 들어, 단계 2에서의 정확도는 단계 1과 단계 2의 테스트 데이터를 통해 계산된다. 모델이 새로운 단계의 데이터를 학습하면서 해당 데이터에 과적합되는 경향으로 인해, 누적된 테스트 데이터 전체에 대한 정확도는 다소 감소할 수 있다.

다른 평가 지표인 망각도는 모델이 새로운 데이터셋을 학습한 후 발생하는 이전 단계 데이터셋에 대한 추론 성능 저

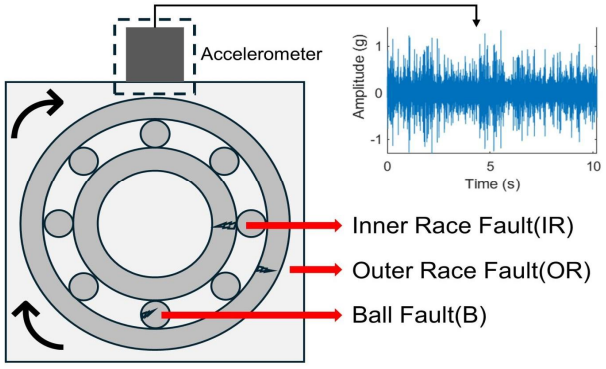


그림 3. 구름 베어링의 결함 종류 및 가속도계에서 측정되는 진동 데이터 예시
Fig. 3. Fault types of the rolling element bearing and an example of vibration data measured from an accelerometer

표 2. 회전 기계 베어링 결함 데이터셋의 특성
Table 2. Characteristics of rotating machinery bearing fault datasets

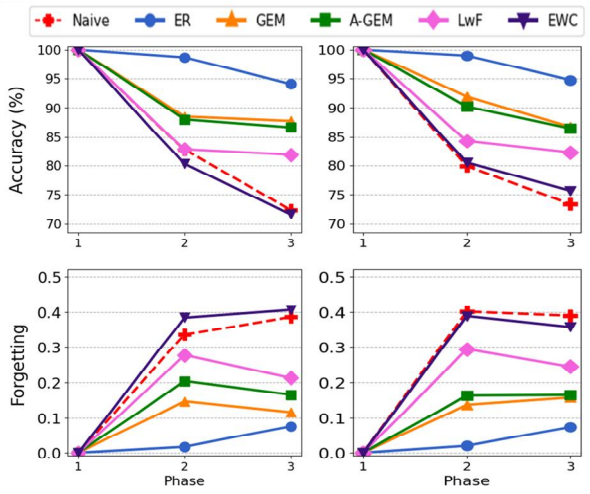
Dataset (Phase)	Shaft Rotating Speed	Class	Sampling Rate
CWRU (Phase 1)	about 30 Hz	N, IR, OR, B	12,000 Hz
MFPT (Phase 2)	25 Hz	N, IR, OR	97,656 Hz or 48,828 Hz
Ottawa (Phase 3)	dynamically changing	N, IR, OR, B, C	200,000 Hz

표 3. 각 단계에서 사용하는 데이터셋의 클래스별 샘플 수
Table 3. Number of samples of the dataset per class used in a single phase

Class	Train	Validation	Test	Total
N	252	84	84	420
IR	252	84	84	420
OR	252	84	84	420

표 4. 클라우드 서버 사양
Table 4. Specification of the cloud server

CPU	Intel® Core™ i9-10900K CPU @ 3.70GHz × 20
Memory	128 GB
GPU	NVIDIA GeForce RTX 2080 SUPER
OS	Ubuntu 20.04 LTS
Framework	PyTorch 2.0.0



(a) L_{sample}= 2,048, $N_{samples}$= 100 (b) L_{sample}= 4,096, $N_{samples}$= 50

그림 4. 연속학습 알고리즘 간 성능 비교 결과
Fig. 4. Performance comparison results of continual learning algorithms

하 정도를 나타내는 지표로, 식 (1)과 같이 정의된다 [25].

$$\text{Forgetting} = -\frac{1}{T-1}\sum_{i=1}^{T-1}(R_{T,i}-R_{i,i}). \quad (1)$$

식 (1)에서 T는 지금까지 학습한 전체 단계의 수를 나타내고 $R_{i,j}$는 i번째 단계의 데이터까지 학습한 후 j번째 단계의 데이터에 대해 측정한 판별 정확도를 의미한다. 즉, 망각도는 이전 단계의 데이터에 대한 정확도 감소량의 평균으로 계산되며, 값이 클수록 이전 데이터에 대한 성능 저하가 심각함을 나타낸다.

$$\text{Memory Buffer Size (MB)} = L_{sample} \times N_{samples} \times \text{sizeof(float)}. \quad (2)$$

재연 방식 알고리즘의 경우 이전 학습 단계의 데이터 샘플 일부를 메모리 버퍼에 저장한다. 메모리 버퍼의 크기(memory buffer size)는 식 (2)와 같이 데이터 샘플 하나의 길이인 L_{sample}과 버퍼에 저장되는 데이터 샘플의 개수인 $N_{samples}$의 곱으로 결정된다. L_{sample}이 커질수록 데이터 샘플 하나에 진동 데이터의 특성이 더 많이 반영되지만, 데이터 저장 비용이 증가한다.

Naive 알고리즘과 다양한 연속학습 알고리즘의 성능을 비교한 결과는 그림 4와 같다. 같은 데이터 저장 비용 조건에서 성능을 비교하기 위해 재연 방식 알고리즘의 L_{sample}이 2,048일 때는 $N_{samples}$를 100으로, L_{sample}이 4,096일 때는 $N_{samples}$를 50으로 설정하여 실험을 진행하였다. 실험 결과, 연속학습 알고리즘을 사용하면 대체로 Naive 알고리즘을 사용했을 때보다 결함 진단 성능이 크게 향상되었음을 알 수 있다. 연속학습 알고리즘 간 성능을 비교했을 때는 ER, GEM, A-GEM, LwF, EWC 알고리즘 순으로 정확도가 높았으며, 재연 방식 알고리즘이 정규화 방식 알고리즘에 비해 성능이 더 좋은 것을 확인하였다. 재연 방식 알고리즘은 메모리 버퍼에 저장된 이전 학습 데이터를 활용하여 지속적으로 변화하는 데이터의 다양한 패턴을 더 정확하게 반영한 것으로 보이며, 정규화 방식 알고리즘은 이전에 학습한 데이터의 분포를 유지하는 것에 중점을 두어 새로운 데이터에 대한 적응 능력이 더 낮게 나온 것으로 보인다. 실험 결과를 바탕으로, 본 연구는 재연 방식 알고리즘 중에서도 가장 우수한 성능을 보이는 ER 알고리즘을 제안하는 시스템의 연속학습 알고리즘으로 선정하였다.

표 5. 데이터 샘플의 길이 및 개수에 따른 메모리 버퍼 크기
Table 5. Memory buffer size depending on L_{sample} and $N_{samples}$

L_{sample}	$N_{samples}$				
2,048	100	150	200	250	300
4,096	50	75	100	125	150
Memory Buffer Size (MB)	0.8	1.2	1.6	2.0	2.4

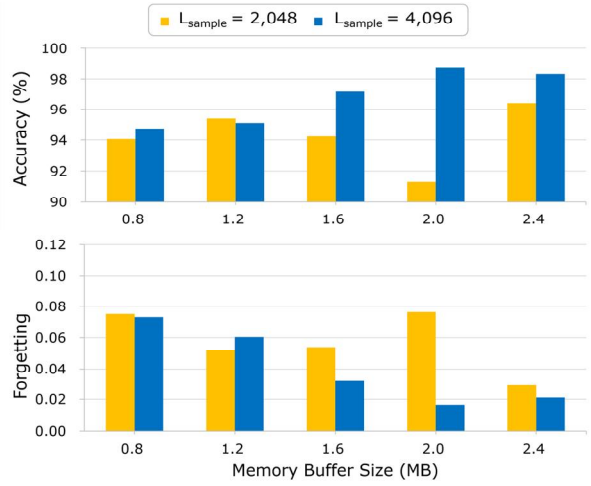


그림 5. 메모리 버퍼 크기에 따른 ER 알고리즘 성능 평가
Fig. 5. Performance evaluation results of ER algorithm by memory buffer size

4. 시스템 성능 평가

시스템 성능 평가는 다음과 같이 진행된다. 먼저 ER 알고리즘은 메모리 버퍼의 구성에 따라 모델의 학습 성능과 메모리 사용량이 달라지기 때문에 메모리 버퍼를 구성하는 데이터 샘플의 길이와 개수를 조정하여 학습 성능을 최대화하였다. 이후 학습시킨 모델을 엣지 디바이스에 배포한 후 추론 속도를 측정하고 모니터링 서비스의 동작을 확인하여 시스템의 전반적인 성능을 평가하였다.

ER 알고리즘의 성능 최적화를 위해, L_{sample}이 2,048과 4,096일 때를 대상으로 ER 알고리즘의 주요 변수인 메모리 버퍼 크기 증가에 따른 성능 변화를 실험하였다. 표 5는 실험에서 사용된 L_{sample}과 메모리 버퍼 크기 값들을 보여 준다. L_{sample}이 4,096일 경우의 $N_{samples}$는 2,048일 때의 절반으로 설정하여, 두 L_{sample}에서 메모리 버퍼 크기를 동일하게 유지하였다.

그림 5는 각각 L_{sample}이 2,048일 때와 4,096일 때, 모든 단계의 학습을 마친 상황에서 메모리 버퍼 크기에 따른 모델의

표 6. 타겟 엣지 디바이스 사양
Table 6. Specification of the target edge device

Hardware	Broadcom BCM2711, Quad core Cortex A72 (ARM v8) 64-bit SoC @ 1.8GHz, 4GB RAM
OS	Raspberry Pi OS (64-bit)
Framework	PyTorch 2.0.1

표 7. 라즈베리파이 4B에서의 추론 시간 측정 결과 (단위: ms)
Table 7. Inference time measurement results in Raspberry Pi 4B (unit: ms)

mean	std.	min.	max.
2.88	0.15	2.73	3.76

정확도와 망각도 변화를 보여 준다. 동일한 메모리 버퍼 크기를 가진 경우에 대부분 L_{sample}이 2,048일 때보다 4,096일 때 더 높은 정확도와 낮은 망각도를 보였다. 이러한 결과는 본 실험 조건에서 샘플 하나의 길이를 증가시키면 적은 수의 샘플에서도 모델이 데이터 특징을 효과적으로 학습함을 나타낸다.

동일한 메모리 버퍼 크기 조건에서 더 좋은 성능을 보인 L_{sample}이 4,096인 경우에 대해 버퍼 크기 변화에 따른 모델 성능을 분석해 보면, 메모리 버퍼 크기가 2MB가 될 때까지는 버퍼 크기가 커짐에 따라 모델의 성능이 지속적으로 좋아짐을 알 수 있다. 그러나 버퍼 크기가 2MB보다 더 커지더라도 모델은 약 98 %의 정확도와 0.02의 망각도를 유지하며 추가적인 성능 향상이 관찰되지 않았다. 따라서 모델의 연속학습 성능을 최대화하는 동시에 연속학습 시의 메모리 사용량을 줄이기 위해 최종적으로 $N_{samples}$값은 메모리 버퍼 크기 2MB일 때인 125개로 결정하였다.

결과적으로 L_{sample}과 $N_{samples}$가 각각 4,096, 125일 때, ER 알고리즘으로 학습시킨 모델은 2MB 크기의 메모리 버퍼를 사용하며 세 단계의 학습을 모두 마쳤을 때 Naive 알고리즘 대비 26.3 %p 높은 98.7 %의 정확도를 달성했다. 이 결과는 시스템에서 연속학습을 수행하면 단계별 학습 데이터 총량의 16.5 % 가량의 추가 버퍼만으로도 이전 데이터의 판별 성능을 성공적으로 유지할 수 있음을 보여 준다.

또한 해당 모델을 표 6과 같은 환경의 Raspberry Pi 4B 디바이스로 배포하여 추론 시간을 측정하였다. 모델 추론은 PyTorch 프레임워크 상에서 진행되었으며, 1,000회의 warm-up 연산 후 100회 반복 추론 시간을 측정하고 그 결과를 표 7에 제시하였다. 측정된 경량 딥 러닝 모델의 추론 시간은 평균 2.88 ms, 최대 3.76 ms로, 본 연구에서 제안한 모델이 컴퓨팅 자원이 제한된 엣지 디바이스에서도 실시간 추론을 수행할 수 있음을 확인하였다.

마지막으로 엣지 디바이스에서 수집된 데이터를 시각화하는 실시간 모니터링 서비스를 구현하고 동작을 확인하였다. 그림 6과 같이 모니터링 서비스 화면에서는 모니터링 대상 기계의 기본 정보를 확인할 수 있고, 진동 데이터에서 측정된 진동 신호의 최대값 (peak value), RMS (Root Mean Square), 각 클래스에 대한 신뢰도 점수 (confidence score)

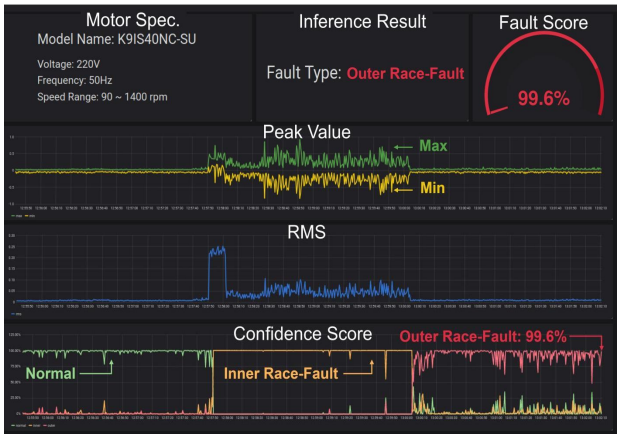


그림 6. 실시간 기계 결함 모니터링 서비스 화면
Fig. 6. Snapshot of the real-time machine fault monitoring service

가 실시간으로 업데이트된다. 따라서, 본 연구에서 구현한 서비스를 활용하여 사용자는 기계에서 수집된 진동 데이터의 변화 추이를 확인할 수 있으며 딥 러닝 모델이 추론한 베어링 상태의 가장 높은 신뢰도 점수 값을 보고 베어링 상태를 점검할 수 있다.

Ⅳ. 결 론

본 논문은 연속학습을 활용하여 변화하는 데이터 분포에 적응할 수 있는 경량 온-디바이스 AI 기반 기계 결함 진단 시스템을 제안하고 구현하였다. 이를 위해 엣지 디바이스에서 실시간 추론을 수행할 수 있는 1차원 합성곱 신경망 기반 경량 딥 러닝 모델을 설계하고 연속학습 알고리즘 비교 실험을 통해 기계 결함 진단 응용에서 가장 높은 성능을 보이는 알고리즘을 선정한 후, 연속학습을 위한 데이터 버퍼에 저장될 데이터의 길이와 개수를 조정하였다. 마지막으로 엣지 디바이스에서 취득한 데이터의 통계량과 결함 진단 결과를 오픈소스 모니터링 프레임워크 Grafana로 시각화하였다.

제안한 시스템은 이전에 수집된 데이터의 16.5 %만을 활용하여 CWRU, MFPT, Ottawa 데이터셋의 데이터가 단계별로 들어오는 연속학습 시나리오에서 98.7 %의 정확도로 베어링 결함을 판별할 수 있으며, 학습된 모델을 Raspberry Pi 4B 디바이스에 배포했을 때 최대 3.76 ms의 추론 시간으로 기계 결함을 진단할 수 있다. 이 결과는 본 연구의 시스템을 활용하면 데이터 변화에 적응하여 기계 결함을 정확하게 판별하는 동시에 클라우드와 엣지 디바이스 간의 진동 데이터 교환 없이 엣지 디바이스에서 실시간으로 딥 러닝 추론을 수행할 수 있음을 보여 준다. 또한 시스템 사용자는 모니터링 서비스를 통해 딥 러닝 모델의 결함 판별 결과와 진동 신호의 통계값을 실시간으로 확인할 수 있다. 향후 연구로 시스템 작동 중에 발생하는 추론 데이터 전송, 모델 가중치 배포와 같은 데이터 통신 작업의 부하를 평가하고 최적화할 계획이다.

References

[1] P. Zhang, Y. Du, T. G. Habetler, B. Lu, "A Survey of Condition Monitoring and Protection Methods for Medium-voltage Induction Motors," IEEE Transactions on Industry Applications, Vol. 47, No. 1, pp. 34-46, 2010.

[2] J. P. Yun, M. S. Kim, G. Koo, C. Sin, "Fault Diagnosis and Analysis Based on Transfer Learning and Vibration Signals," IEMEK J. Embed. Sys. Appl., Vol. 14, No. 6, pp. 287-294, 2019 (in Korean).

[3] C. Y. Lee, G. L. Zhuo, T. A. Le, "A Robust Deep Neural Network for Rolling Element Fault Diagnosis Under Various Operating and Noisy Conditions," Sensors, Vol. 22, No. 13, pp. 4705, 2022.

[4] D. T. Hoang, H. J. Kang, "A Survey on Deep Learning Based Bearing Fault Diagnosis," Neurocomputing, Vol. 335, pp. 327-335, 2019.

[5] W. Zhang, G. Peng, C. Li, Y. Chen, Z. Zhang, "A New Deep Learning Model for Fault Diagnosis with Good Anti-noise and Domain Adaptation Ability on Raw Vibration Signals," Sensors, Vol. 17, No. 2, pp. 425, 2017.

[6] W. Zhang, C. Li, G. Peng, Y. Chen, Z. Zhang, "A Deep Convolutional Neural Network with New Training Methods for Bearing Fault Diagnosis Under Noisy Environment and Different Working Load," Mechanical Systems and Signal Processing, Vol. 100, pp. 439-453, 2018.

[7] S. Dhar, J. Guo, J. Liu, S. Tripathi, U. Kurup, M. Shah, "A Survey of On-device Machine Learning: An Algorithms and Learning Theory Perspective," ACM Transactions on Internet of Things, Vol. 2, No. 3, pp. 1-49, 2021.

[8] H. Hua, Y. Li, T. Wang, N. Dong, W. Li, J. Cao, "Edge Computing with Artificial Intelligence: A Machine Learning Perspective," ACM Computing Surveys, Vol. 55, No. 9, pp. 1-35, 2023.

[9] S. Lu, J. Lu, K. An, X. Wang, Q. He, "Edge Computing on IoT for Machine Signal Processing and Fault Diagnosis: A Review," IEEE Internet of Things Journal, Vol. 10, No. 13, pp. 11093-11116, 2023.

[10] L. Fu, K. Yan, Y. Zhang, R. Chen, Z. Ma, F. Xu, T. Zhu, "EdgeCog: A Real-time Bearing Fault Diagnosis System Based on Lightweight Edge Computing," IEEE Transactions on Instrumentation and Measurement, Vol. 72, pp.1-11, 2023.

[11] S. Afrasiabi, M. Afrasiabi, B. Parang, M. Mohammadi, "Real-Time Bearing Fault Diagnosis of Induction Motors with Accelerated Deep Learning Approach," 2019 10th International Power Electronics, Drive Systems and Technologies Conference (PEDSTC), pp. 155-159, 2019.

[12] B. Chen, C. Shen, J. Shi, L. Kong, L. Tan, D. Wang, Z. Zhu, "Continual Learning Fault Diagnosis: A Dual-branch Aaptive Aggregation Residual Network for Fault Diagnosis with Machine Increments," Chinese Journal of Aeronautics, Vol. 36, No. 6, pp. 361-377, 2023.

[13] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, C. Liu, "A Survey on Deep Transfer Learning," ICANN 2018: 27th International Conference on Artificial Neural Networks, pp. 270-279, 2018.

[14] Y. Bengio, "Deep Learning of Representations for Unsupervised and Transfer Learning," Proceedings of ICML Workshop on Unsupervised and Transfer Learning, PMLR, Vol. 27, pp. 17-36, 2012.

[15] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A.A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, R. Hadsell, "Overcoming Catastrophic Forgetting in Neural Networks," Proceedings of the National Academy of Sciences, Vol. 114, No. 13, pp. 3521-3526, 2017.

[16] R. Kemker, M. McClure, A. Abitino, T. Hayes, C. Kanan, "Measuring Catastrophic Forgetting in Neural Networks," Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, No. 1, pp. 3390-3398 2018.

[17] J. Chuya-Sumba, L. M. Alonso-Valerdi, D. I. Ibarra-Zarate, "Deep-learning Method Based on 1D Convolutional Neural Network for Intelligent Fault Diagnosis of Rotating Machines," Applied Sciences, Vol. 12, No. 4, pp. 2158, 2022.

[18] X. Liu, Q. Zhou, H. Shen. "Real-Time Fault Diagnosis of Rotating Machinery Using 1-D Convolutional Neural Network," 2018 5th International Conference on Soft Computing & Machine Intelligence (ISCMI), pp.104-108, 2018.

[19] L. Hou, L. Liu, G. Mao, "Machine Fault Diagnosis Method Using Lightweight 1-D Separable Convolution and WSNs with Sensor Computing," IEEE Transactions on Instrumentation and Measurement, Vol. 71, pp. 1-8, 2022.

[20] Ö. Gültekin, E. Cinar, K. Özkan, A. Yazıcı, "Real-Time Fault Detection and Condition Monitoring for Industrial Autonomous Transfer Vehicles Utilizing Edge Artificial Intelligence," Sensors, Vol. 22, No. 9, p. 3208, 2022.

[21] G. Qian, S. Lu, D. Pan, H. Tang, Y. Liu, Q. Wang, "Edge Computing: A Promising Framework for Real-time Fault Diagnosis and Dynamic Control of Rotating Machines Using Multi-sensor Data," IEEE Sensors Journal, Vol. 19, No. 11, pp. 4211-4220, 2019.

[22] X. Ding, H. Wang, Z. Cao, X. Liu, Y. Liu, Z. Huang, "An Edge Intelligent Method for Bearing Fault Diagnosis Based on a Parameter Transplantation Convolutional Neural Network," Electronics, Vol. 12, No. 8, pp. 1816, 2023.

[23] G. M. van de Ven, T. Tuytelaars, A. S. Tolias, "Three Types of Incremental Learning," Nature Machine Intelligence, Vol. 4, No. 12, pp. 1185-1197, 2022.

[24] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, G. Wayne, "Experience Replay for Continual Learning," Advances in Neural Information Processing Systems, Vol. 32, pp. 1-11, 2019.

[25] D. Lopez-Paz, M.A. Ranzato, "Gradient Episodic Memory for Continual Learning," Advances in Neural Information Processing Systems, Vol. 30, pp.1-10, 2017.

[26] A. Chaudhry, M.A. Ranzato, M. Rohrbach, M. Elhoseiny, "Efficient Lifelong Learning with A-GEM," arXiv:1812.00420, 2019.

[27] N. Vödisch, K. Petek, W. Burgard, A. Valada, "CoDEPS: Online Continual Learning for Depth Estimation and Panoptic Segmentation," arXiv:2303.10147, 2023.

[28] Z. Li, D. Hoiem, "Learning Without Forgetting," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 40, No. 12, pp. 2935-2947, 2017.

[29] B. Maschler, H. Vietz, N. Jazdi, M. Weyrich, "Continual Learning of Fault Prediction for Turbofan Engines Using Deep Learning with Elastic Weight Consolidation," 2020 25th IEEE International Conference on Emerging Technologies

and Factory Automation (ETFA), Vol. 1, pp. 959-966, 2020.
[30] https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/
[31] https://grafana.com/
[32] https://engineering.case.edu/bearingdatacenter/
[33] https://www.mfpt.org/fault-data-sets/
[34] H. Huang, N. Baddour, "Bearing Vibration Data Collected Under Time-varying Rotational Speed Conditions," Data in Brief, Vol. 21, pp. 1745-1749, 2018.
[35] R. B. Randall, J. Antoni, "Rolling Element Bearing Diagnostics—A Tutorial," Mechanical Systems and Signal Processing, Vol. 25, No. 2, pp. 485-520, 2011.

Youngjun Kim (김 영 준)

2024 Mechanical and Information Engineering from University of Seoul (B.S.)
2024~Department of Mechanical and Information Engineering/Smart Cities, University of Seoul (M.S. Student)

Field of Interests: Continual Learning
Email: youngjr0527@naver.com

Taewan Kim (김 태 완)

2024 Mechanical and Information Engineering from University of Seoul (B.S.)
2024~Department of Mechanical and Information Engineering/Smart Cities, University of Seoul (M.S. Student)

Field of Interests: On-device AI, Fault Diagnosis
Email: ktwktw109@uos.ac.kr

Suhyun Kim (김 수 현)

2024 Mechanical and Information Engineering from University of Seoul (B.S.)
2024~Department of Mechanical and Information Engineering/Smart Cities, University of Seoul (M.S. Student)

Field of Interests: Edge Deep Learning
Email: happy113200@naver.com

Seongjae Lee (이 성 재)

2019 Mechanical and Information Engineering from University of Seoul (B.S.)
2019~Department of Mechanical and Information Engineering/Smart Cities, University of Seoul (Ph.D. Candidate)

Field of Interests: On-device AI, Parallel Computing, Machine Fault Diagnosis
Email: junior209@uos.ac.kr

Taehyoun Kim (김 태 현)

1994 Computer Engineering from Seoul National University (B.S.)
1996 Computer Engineering from Seoul National University (M.S.)
2001 Electrical and Computer Engineering from Seoul National University (Ph.D.)

Career:
2001~2005 R&D Manager, GCT Research, Inc.
2005~Professor, Dept. of Mechanical & Information Eng./Smart Cities, University of Seoul

Field of Interests: Embedded Real-Time Systems, Edge AI Solution, Industrial Automation
Email: thkim@uos.ac.kr