DNN 기반 사이버 물리 시스템의 결함 주입 프레임워크

조은호, 윤주연, 고인영
한국과학기술원 전산학부
{ehcho, juyeon.yoon, iko}@kaist.ac.kr

Fault Injection Framework for DNN-Enabled Cyber-Physical Systems

Eunho Cho, Juyeon Yoon, In-Young Ko
School of Computing, Korea Advanced Institute of Science and Technology(KAIST)

요 약

심층 신경망(Deep Neural Network, DNN) 기술과 사이버 물리 시스템과 같은 최신 소프트웨어 기술의 도입으로, 현대 사회는 물리적 환경과 사이버 환경 양 측면에서 모두 상호 작용하면서, DNN 을 핵심적인 시스템 구성 요소로 포함하는 DNN 기반 사이버 물리 시스템(DECPS)를 요구하게 되었다. 이 시스템은 맞닥뜨리는 환경의 다양성과 복잡성이 높을 뿐만 아니라, 블랙박스로 간주하는 DNN 을 구성 요소로 포함하여, 가능한 모든 환경에서 신뢰성과 안전성 목표를 준수하는지 검증하기 어려운 현실이다. 실제 현장에서 시뮬레이션 기반의 DECPS 검증을 요구하고 있으나, 아직 DECPS 를 위한 결함 주입 도구 지원이 부족한 상황이다. 따라서, 이 논문에서는 DECPS 의 시뮬레이션 기반 테스팅에 활용될 수 있는 결함 주입 프레임워크를 제시한다. 이 프레임워크는 DECPS 를 소프트웨어 구성 요소 결함, 물리적 구성 요소 결함, 상호작용으로 나누어 각 부분에 현실과 유사한 결함을 주입하는 방법을 제시하는 방식으로 이루어진다.

1. 서론

심층 신경망(Deep Neural Network, DNN) 기술의 도입과 발전은 자율 주행 등 다양한 분야에서 손쉽게 인공지능 기술을 사용하도록 유도했다. 또한 사물인터넷과 사이버 물리 시스템(Cyber-Physical System)과 같은 최신 소프트웨어 기술의 도입은 복잡한 환경에서도 DNN 이 활용되는 사례를 계속 생성하고 있다.

이처럼, 현대 사회는 물리적 환경과 사이버 환경 양 측면에서 모두 상호 작용하면서, DNN 을 핵심적인 시스템 구성 요소로 포함하는 DNN 기반 사이버 물리 시스템을 요구하고 있다. 이 시스템은 맞닥뜨리는 환경의 다양성과 복잡성이 높을 뿐만 아니라, 블랙박스로 간주하는 DNN 을 구성 요소로 포함하여, 가능한 모든 환경에서 신뢰성과 안전성 목표를 준수하는지 테스팅하고 검증하기 어려운 현실이다 [1-2].

DNN 기반 사이버 물리 시스템의 테스팅은 실제 현업에서는 필드 테스팅 중심으로 수행된다. 자율주행 차량의 경우, 표준 ISO 21448 [3]에 따라, 안전성과 신뢰성을 보장하기 위하여 일정 수준의 필드 테스팅 혹은 시뮬레이션 기반의 테스팅을 필수적으로 요구하고 있다.

하지만, 시뮬레이션 기반 테스팅의 경우, 관련 도구 지원이 부족한 상황이다. 특히, 다양한 결함을 시스템에 주입할 수 있는 결함 주입 도구는 아직 개발되지 않았다. 또한, DNN 기반 사이버 물리 시스템은 사이버 물리 시스템 혹은 DNN 기반 시스템에 비하여 시스템 오류 주입과 관련된 연구가 많이 진행되지 않았다.

따라서, 이 논문에서는 DNN 기반 사이버 물리 시스템의 시뮬레이션 기반 테스팅에 활용될 수 있는 결함 주입 프레임워크를 제시하고자 한다. 이 프레임워크는 DNN 기반 사이버 물리 시스템을 소프트웨어 결함, 하드웨어 결함, 상호작용으로 나누어 각 부분에 현실과 유사한 결함을 주입하는 방법을 제시하는 방식으로 이루어진다.

이 논문의 구성은 다음과 같다. 2 장에서는 사이버 물리 시스템 혹은 DNN 기반 시스템의 오류 주입 방법의 기존 선행 연구 기법들을 소개한다. 3 장에서는 DNN-based 사이버 물리 시스템의 부분별 오류 주입 프레임워크를 제시한다. 4 장에서는 해당 프레임워크가 향후 연구에 기여할 수 있는 방향과 결론을 제시한다.

2. 선행 연구

사이버 물리 시스템의 결함 주입 방법에 관해 다양한 연구가 진행되었으나, 아직 통일된 결함 유형이나, 오류 주입 방법은 나오지 않았다. T. Fabarisov 외의 연구에서는 사이버 물리 시스템의 결함의 종류를 센서 결함, 컴퓨팅 하드웨어 결함, 네트워크 결함의 세 부분으로 나누고, 각 결함의 유형을

* 본 연구는 국토교통부/국토교통과학기술진흥원의 지원으로 수행되었음 (과제번호 22CTAP-C163794-02).

구분하여 각각의 결함 종류를 생성하는 도구를 개발하였다 [4]. 다양한 결함 종류가 소개되었으나, 여러 물리적 결함이나, 지연, 소실 등 해당 연구의 대상 시스템에만 적용되는 결함만 생성하는 도구를 제시하였다.

J. Fröhlich 외의 연구에서는 사이버 물리 시스템에 결함 주입을 통한 안전성 실험 도구를 개발했다 [5]. 이 연구에서는 중복된 입력 신호와 실패를 유도하는 데이터 프로세싱의 두 가지 결함을 생성하여, 시스템에 결함을 주입하고, 안전성을 검증하는 도구를 제안했다.

Y. Liu 외의 연구에서는 DNN 모델 상에서 결함 주입이 모델에 주는 영향에 대한 실험적 결과를 제시한다 [6]. 예를 들어, 분류 모델에서는 특정 입력 패턴을 모델이 잘못 분류하도록 모델의 파라미터를 일부 조작하는 것을 통해 결함을 주입할 수 있다. 이 연구에서는 DNN 모델 파라미터 상의 결함 주입을 크게 단일 편향 공격과 경사 하강 공격로 나누어, DNN 파라미터를 변경하는 양에 따라 공격이 모델의 결과값에 얼마나 큰 영향을 미치는지를 측정한다.

L. Ma 외의 연구에서는 딥러닝 기반 시스템에 결함을 주입하고, 테스트 데이터의 결함 발견 능력을 효과적으로 측정하는 것을 목적으로 한다 [7]. 이를 위해 학습 데이터와 학습 코드 상에 인공적 결함, 즉 뮤턴트(Mutant)를 주입한다.

선행 연구 조사 결과, DNN 기반 사이버 물리 시스템을 구성하는 두 요소, 물리적 구성 요소와 소프트웨어 구성 요소를 모두 복합적으로 고려한 시스템 오류 주입 프레임워크는 아직 연구되지 않았다. 특히, 사이버 물리 시스템의 결함 주입은 물리적 구성 요소의 결함에 한정된 연구의 비율이 높다.

3. DNN 기반 사이버 물리 시스템 오류 주입 프레임워크

이 연구에서는 DNN 기반 사이버 물리 시스템의 오류 주입을 소프트웨어 구성 요소, 하드웨어 구성 요소, 그리고 각 구성 요소 간의 상호작용으로 나눈다.

소프트웨어 구성 요소는 뮤테이션 테스팅(Mutation Testing) 기법을 기반으로 실제적인 결함을 주입하는 기법을 활용한다. 하드웨어 구성 요소와 상호작용 구성 요소는 기존 사이버 물리 시스템의 연구를 기반으로 센서 및 네트워크 기반 결함을 주입한다.

3.1 소프트웨어 구성 요소 결함 주입

선행 연구를 분석한 결과, 사이버 물리 시스템 관련 연구에서는 소프트웨어 결함을 크게 다루지 않으나, 일반적인 소프트웨어 와 DNN 기반 시스템을 대상으로 한 연구에서는 중점적으로 다루는 차이가 있었다. 따라서, 이 프레임워크에서는 뮤테이션 테스팅에서 활용되는 뮤턴트 생성 기법을 활용해 소프트웨어 결함 주입 자동화를 지원하고자 한다.

소프트웨어의 테스트 스위트 적합성 판별은 보편적으로 커버리지 (Coverage) 수치에 기반한다. 하지만 커버리지는 단순히 테스트가 특정 프로그램 요소를 실행하는지 여부만 확인하기 때문에, 테스트가 잠재적인 결함을 실제로 발견할 수 있는지에 대한 능력을 측정하기 위해서는 더 강한 테스트 적합성 기준이 필요하다. 이로써 제안한 것이, 소프트웨어에 단순한 형태의 인공적 결함, 즉 뮤턴트를 다수 주입하여 테스트가 각 뮤턴트에 의해 실패하는지를 확인하는 뮤테이션 테스팅이다. 뮤테이션 테스팅의 결과를 통해 커버리지보다 더욱 테스트 스위트의 결함 탐지 능력을 근사하게 추정할 수 있다. 그러므로 뮤테이션 테스팅은 현재 이론적으로 가장 강력한 테스트 적합성 기준으로서 활발히 연구되고 있다.

하지만 뮤테이션 테스팅의 효과성에도 불구하고, 실제 개발 환경에서 뮤테이션 테스팅이 도입되기 어려운 이유는 수많은 뮤턴트를 만들고 반복적으로 실행하는 비용 문제 때문이다. 이로 인해 생성된 뮤턴트들 중 효과적인 것들을 선택적으로 사용하거나, 뮤테이션 테스팅의 결과를 머신 러닝을 기반으로 예측하는 등 뮤테이션 테스팅의 비용을 줄이기 위한 다양한 기법들이 제시되었다. 또한, 대부분 단순한 산술 연산의 수정으로 이루어진 간단한 뮤턴트가 실제 개발 환경에서 발생하는 결함을 대표할 수 있는지에 대해서도 의문이 제기되어왔다.

한 가지 유망한 방향은 실제 관찰되는 결함과 유사한 형태로 인공적인 결함을 생성하는 것이다. A. Khanfir 외의 연구에서는 버그리포트 문서에서부터 프로그램에 버그가 생기기 쉬운 위치와 형태를 추출하여 실제 결함과 가까운 형태로 프로그램에 결함을 주입하는 기법을 제안하며, 이를 통해 기존의 뮤테이션 테스팅보다 테스트 스위트(Test Suite)의 결함 발견 능력을 더 정확하게 근사하는 것을 보였다 [8].

이 연구에서는 DNN 기반의 사이버 물리 시스템은 pylot [9]과 같은 자율 주행 모듈을 대상으로 한다. 많은 DNN 기반의 사이버 물리 시스템이 텐서플로우(Tensorflow)등의 딥러닝 프레임워크에 기반하여 python 을 대상으로 하는 것에 비해 기존 A. Khanfir 외의 연구 [8]에서는 자바 언어의 단일 프로젝트를 대상으로 결함 주입 기법을 제안하였다. 따라서, Tensorflow, PyTorch 등 다양한 파이썬 기반의 딥러닝 모델의 학습과 실행 코드들을 대상으로 결함을 주입하기 위해서는 파이썬 언어에 기반한 새로운 결함 주입 모델을 구성해야 한다.

이 프레임워크에서는 그림 1 과 같이 텐서플로우 등 딥러닝 프레임워크 기반으로 구현된 소스코드와 버그 리포트 문서를 수집하여 DNN 기반 사이버 물리 시스템의 소프트웨어 요소에 적합한 결함 주입 모델을 학습한다. 버그 리포트와 연결된 프로그램 수정 기록을 바탕으로 실제 사이버 물리 시스템에 실제 결함과 높은 유사도를 가진 결함을 주입한다.

이와 같이, 실제 결함을 모방한 인공적 결함을 주입함으로써 기존 뮤테이션 테스팅을 사이버 물리 시스템의 소프트웨어 구성요소에 효과적으로 적용할 수 있을 것이며, 사이버 물리

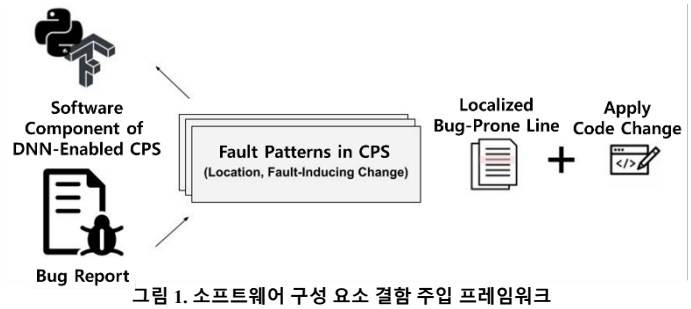

그림 1. 소프트웨어 구성 요소 결함 주입 프레임워크

시스템을 검증하는 테스트들의 결함 탐지 능력을 좀더 신뢰성 있는 수치로 판단할 수 있음을 기대한다.

3.2 물리적 구성 요소 및 상호작용 결함 주입

물리적 구성 요소 및 상호작용 결함은 T. Fabarisov 외의 선행 연구[4]에서 이미 결함 유형과 MATLAB 기반의 결함 삽입 도구가 제시된 바 있다. 이 프레임워크에서는 해당 도구를 좀 더 범용적으로 확장하여, 물리적 구성 요소 및 상호작용 결함을 주입하고자 한다.

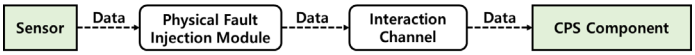

그림 2. 물리적 구성 요소 결함 주입 프레임워크

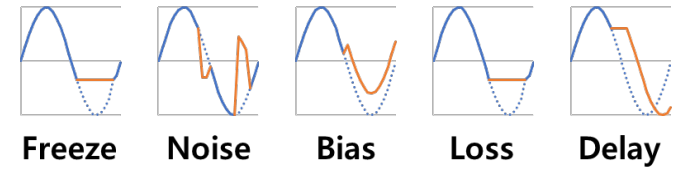

그림 3. 물리적 결함 유형(고정, 잡음, 편향) 및 상호작용 결함 유형(소실, 지연)

그림 2 와 같이 물리적 구성 요소의 결함은 사이버 물리 시스템의 각 센서가 센서 값의 데이터 스트림을 만들어 낼 때, 해당 데이터 스트림을 조작하는 모듈을 설치한다. 해당 모듈은 그림 3 과 같은 세 가지 결함 유형을 삽입한다. 데이터 스트림을 일시 정지시키는 고정 결함, 특정 정규분포에 해당하는 임의의 값을 추가하는 잡음 결함, 특정 임의의 값만큼 편향된 데이터를 송출하는 편향 결함을 사용자의 의도에 따라 삽입하게 된다.

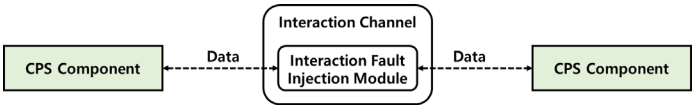

그림 4. 상호작용 결함 주입 프레임워크

상호작용 결함은 그림 4 와 같이 각 구성 요소 간의 상호작용 채널에 데이터를 조작하는 추가적인 모듈을 설치한다. 이 모듈을 통해 그림 3 과 같은 두 가지 결함 유형을 삽입한다. 채널을 잠시 정지시키는 소실 결함과 채널 데이터를 지연하여 전달하는 지연 전달의 두 가지 결함을 사용자의 의도에 따라 삽입하게 된다.

4. 결론 및 향후 연구 방향

이 논문에서는 DNN 기반 사이버 물리 시스템의 오류 주입 프레임워크를 제시한다. 이 프레임워크는 소프트웨어, 하드웨어 구성요소와 각 구성 요소 간의 상호작용으로 시스템을 분리하여, 시스템 오류 주입과 관련된 기존 연구를 바탕으로, 현실적인 시스템 오류를 주입하는 방법에 관해 제시하고 있다.

이 프레임워크는 다양한 환경에서 DNN 기반 사이버 물리 시스템의 견고성과 안전성을 검증하는 데 도움을 줄 수 있을 것이다. 또한, 최근 활발히 연구되고 있는 시뮬레이션 기반 시스템 테스팅의 테스트 스위트 효율성을 확인하고, 검증하는 연구에도 사용될 수 있을 것이다.

다만, 아직 실제적인 오류가 주입되는지와 관련된 실험적인 검증은 부족하다. 따라서, 향후 연구에서는 이 논문에서 제시한 프레임워크로 만든 오류와 실제 시스템에서 나오는 오류를 비교하여 이 프레임워크의 실험적 검증을 진행할 것이다. 또한, 해당 프레임워크에서 다루지 못한 심층 신경망에 오류를 주입하는 기법을 추가하는 연구를 진행할 것이다. 마지막으로, 이 프레임워크를 도구화하여, 시뮬레이션 기반 시스템 테스팅 등 실제 테스팅 연구에 활용할 계획이다.

참고문헌

[1] Ozkaya, Ipek. "What is really different in engineering AI-enabled systems?." *IEEE Software* 37.4 (2020): 3-6.

[2] Cho, Eunho, et al. "Anomaly-aware adaptation approach for self-adaptive cyber-physical system of systems using reinforcement learning." *2022 17th Annual System of Systems Engineering Conference (SOSE)*. IEEE, 2022.

[3] "Road vehicles – Safety of the intended functionality." *International Organization for Standardization. ISO 21448:2022*, 2022.

[4] Fabarisov, Tagir, et al. "Model-based fault injection experiments for the safety analysis of exoskeleton system." *Proceedings of the 30th European Safety and Reliability Conference and the 15th Probabilistic Safety Assessment and Management Conference (2020)*. 2020.

[5] Fröhlich, Joachim, et al. "Testing safety properties of cyber-physical systems with non-intrusive fault injection - an industrial case study." *International Conference on Computer Safety, Reliability, and Security*. Springer, Cham, 2016.

[6] Liu, Yannan, et al. "Fault injection attack on deep neural network." *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2017.

[7] Ma, Lei, et al. "Deepmutation: Mutation testing of deep learning systems." *2018 IEEE 29th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2018.

[8] Khanfir, Ahmed, et al. "Ibir: Bug report driven fault injection." *ACM Journal of the ACM (JACM)* (2020).

[9] Gog, Ionel, et al. "Pylot: A modular platform for exploring latency-accuracy tradeoffs in autonomous vehicles." *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.