연합학습 훈련 프로세스의 신뢰성 확보를 위한 MTV-FL: 멀티에이전트 기반 훈련 호환성 자율 검증 기법

정영환, 최원기, 이상신*

한국전자기술연구원

{cjstntjd, cwk1412}@keti.re.kr, *sslee@keti.re.kr

MTV-FL: A Multi-Agent Based Training Verification Method for Reliable Federated Learning

Jeong Young Hwan, Choi Won Gi, Lee Sang Shin*

Korea Electronics Technology Institute

요 약

연합학습(Federated Learning, FL)은 원천데이터를 공유하지 않고 분산 환경에서 모델을 학습할 수 있는 대안적 학습 패러다임으로 주목 받고 있다. 그러나 실제 FL 플랫폼에서는 모델 개발자가 제출한 훈련 코드가 데이터 보유 주체의 클라이언트 환경에서 직접 실행되며 이 과정에서 다양한 오류 및 악의적 행위가 빈번하게 발생한다. 본 논문에서는 이러한 문제를 해결하기 위해 멀티에이전트 기반 연합학습 훈련 호환성 검증 시스템인 MTV-FL을 제안한다. 제안 시스템은 사용자 제출 코드를 대상으로 플랫폼 관점에서의 실행 흐름 분석, 문법 및 인터페이스 보정, 보안 정책 위배 사항 식별 등을 수행하는 전문화된 에이전트로 구성되어 모델 개발자의 최소한의 개입으로 연합학습 훈련을 수행할 수 있도록 자율적으로 동작한다. 제안된 시스템은 기존 레거시 방식과 비교하여 데이터 은닉 원칙을 침해하지 않으면서도 연합학습 훈련의 안정성 및 확장성을 동시에 확보한다.

Ⅰ. Introduction

연합학습(Federated Learning, FL)은 원천 데이터를 중앙 서버로 집계하지 않고, 각 참여 주체의 로컬 환경에서 모델을 학습한 후 학습 결과만을 공유하는 분산 학습 패러다임이다.[1] 이러한 특성으로 연합학습은 개인정보 보호 및 데이터 주권 보장이 요구되는 다양한 응용 분야에서 효과적인 학습 방식으로 주목받고 있다.[2]

일반적인 연합학습 플랫폼에서는 모델 개발자가 작성한 훈련 코드가 데이터 보유 주체의 클라이언트 환경에서 직접 실행되는 구조를 따른다. 그러나 모델 개발자는 플랫폼 내부의 실행 규약, 데이터 접근 방식, 통신 인터페이스 등을 완전히 인지하기 어렵기 때문에, 훈련 과정에서 실행 오류, 플랫폼 비호환 문제, 또는 의도하지 않은 동작이 빈번하게 발생한다. 이러한 문제는 연합학습 훈련의 안정성을 저해할 뿐만 아니라, 플랫폼 운영 측면에서도 반복적인 수동 검증과 수정 부담을 야기한다.

더 나아가 외부에서 작성된 훈련 코드가 클라이언트 환경에서 실행된다는 점은 연합학습의 신뢰성 측면에서 중요한 위험 요소로 작용한다. 훈련 코드 내부에 비인가 통신, 외부 서버로의 데이터 전송, 또는 플랫폼 정책을 우회하는 로직이 포함될 경우, 이는 연합학습의 핵심 원칙인 데이터 은닉 및 비노출 원칙을 훼손할 수 있다.[3] 기존 연합학습 연구들은 주로 모델 집계 방식이나 개인정보 보호 기법에 초점을 맞추어 왔으며, 훈련 코드 실행 단계에서의 호환성과 행위 검증을 체계적으로 다루는 접근은 제한적이었다.[4] 본 논문에서는 이러한 문제를 해결하기 위해 MTV-FL (Multi-Agent based Training Verification for Federated Learning)을 제안한다. MTV-FL은 연합학습 훈련 실행 이전 단계에서 훈련 코드의 실행 흐름을 분석하고, 플랫폼 비호환 문법 및 인터페이스를 자동으로 보정하며, 샌드박스 기반으로 보안 정책 위배 가능성을 사전에 식별하는 멀티에이전트 기반 자율 검증 기법이다. 각 에이전트는 상호 협력적으로 동작하여 모델 개발자의 개입을 최소화하면서도 플랫폼 정책에 부합하는 안전한 훈련 실행을 보장한다.

또한 다양한 연합학습 훈련 시나리오를 대상으로 한 실험을 통해, MTV-FL이 기존 방식 대비 훈련 오류 발생률을 감소시키고 정책 위반 가능성을 효과적으로 식별함으로써 연합학습 훈련 프로세스의 신뢰성을 유의미하게 향상시킴을 확인하였다.

Ⅱ. MTV-FL(Multi-Agent based Training Verification for FL)

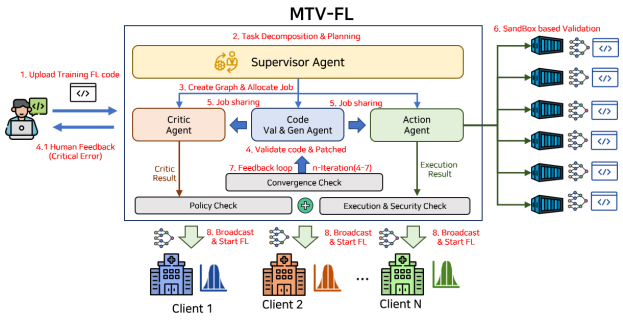

그림1. MTV-FL 구조도

이 장에서는 MTV-FL의 전체 동작 방식과 멀티에이전트 기반 FL 훈련 검증 기법을 설명한다. MTV-FL은 연합학습 훈련 코드의 검증 문제를 단순한 규칙 검사나 정적 분석이 아닌, 플랫폼 정책 제약 하에서의 훈련 실행 정합성 수렴 문제로 정의한다. 이를 위해 MTV-FL은 훈련 코드를 반복적으로 분석·수정·실행하며, 정책 위반 가능성이 허용 범위 내로 감소할 때까지 자율 검증을 수행한다. MTV-FL에서 훈련 코드는 실행 관점에서 다음과 같이 표현된다.

$$P = (G, C, \Pi) \quad (1)$$

여기서 $G = (V, E)$는 훈련코드로부터 추출된 실행 흐름 그래프이고 C는 플랫폼

보안 정책 및 실행 제약의 집합, Π는 연합학습 프로토콜이 허용하는 훈련 행위 공간을 의미한다. MTV-FL의 목적은 훈련 코드가 생성하는 실행 흐름 G가 제약 C를 만족하면서 Π내에서 동작하도록, 최소한의 수정으로 실행 정합성을 확보하는 것이다.

이를 위해 MTV-FL은 멀티에이전트 기반의 계층적 검증 구조를 사용한다. 검증 과정은 감독 에이전트의 계획 하에 코드 생성, 실행 및 분석, 비평에 따른 반복 피드백 단계로 분해되며, 각 단계는 독립적인 에이전트에 의해 수행된다. 이 때 에이전트 간 상호작용은 단순 순차 실행이 아니라, 중간 검증 결과를 공유하며 반복적으로 갱신되는 협업 구조를 따른다.

이를 위해 MTV-FL은 훈련 코드의 정책 정합성을 정량적으로 평가하기 위한 정책 위반 스코어 함수를 정의한다. 각 에이전트와 연계된 평가 함수는 실행 흐름 그래프 G 상의 가능한 실행 경로 집합 $P(G)$)을 대상으로, 각 경로가 플랫폼 제약 C를 위반할 가능성을 평가하여 다음과 같은 스코어를 산출한다.

$$S(G) = \sum_{p \in P(G)} w_p \cdot \Phi(p, C) \qquad (2)$$

여기서 $\Phi(p,C)$는 실행경로 p가 정책 제약 C를 위반할 가능성을 나타내는 평가함수이며, w_p는 해당 경로의 실행 가능성 또는 위험도를 반영한 가중치이다. 이에 $S(G)$는 훈련 코드의 정책 정합성 정도를 나타내는 지표로 사용되어, 초기 검토 단계에서 보완여부를 결정하는 기준이 된다.

검토 결과 $S(G) \geq \alpha$인 경우, 감독 에이전트는 해당 훈련 코드를 실패 상태로 판정하고, 비평 및 액션 에이전트가 식별한 제약 위반 영역과 보완 방향을 코드 생성 에이전트에 전달한다. 코드 생성 에이전트는 이를 입력으로 받아 정책 제약을 만족하도록 훈련 코드를 재구성하며, 수정된 코드는 다시 비평 에이전트와 액션 에이전트에 전달되어 추가 검증을 수행한다. 이와 같은 검증-보완 절차는 다음과 같은 반복적 갱신 과정으로 표현된다.

$$G^{t+1} = \tau(G^t | C, \Delta^t) \qquad (3)$$

여기서 $\tau(\bullet)$는 코드 생성 에이전트에 의해 수행되는 제약 반영 코드 변환 연산자이며, Δ^t는 t번째 반복 단계에서 비평 에이전트와 액션 에이전트로부터 전달된 정책 위반 피드백 및 실행 분석 결과를 의미한다. 각 반복 단계에서 수정된 실행 흐름 G^{t+1}에 대해 스코어 $S(G^{t+1})$가 다시 평가되며, 해당 값이 임계값 이하로 감소할 때까지 검증 루프가 수행된다. 이를 통해 MTV-FL은 다양한 FL 로직에서도 최소한의 개입으로 플랫폼 정책과 데이터 은닉 원칙을 침해하지 않는 통제된 훈련 절차를 보장한다.

Ⅲ. Simulation Result

이 장에서는 MTV-FL의 성능을 평가한다. 성능 평가 요소는 크게 3가지로 코드 호환성 확보 효율, 악의적 작업 제거 성능, 그리고 연합학습 훈련 실행 성공률로 구성된다. 먼저 코드 호환성 확보 효율은 모델 개발자가 제출한 훈련 코드가 FedAvg, FedProx 등 대표적인 연합학습 알고리즘을 정상적으로 수행할 수 있도록 보완되기까지 요구되는 평균 피드백 루프 반복 횟수로 정의한다. 두 번째로 악의적 작업 제거 성능은 훈련 코드 내부에 삽입된 비인가 외부 통신(post 요청 등)과 같은 악의적 작업이 제거되기까지 필요한 평균 반복 횟수로 평가한다. 마지막으로 훈련 실행 성공률은 보정된 훈련 코드가 실제 연합학습 환경에서 오류 없이 훈련을 완료한 비율을 의미한다. 시뮬레이션을 위해 무작위 훈련 참여자 환경을 가정하였으며, 동일한 플랫폼 정책 제약 하에서 MTV-FL과 LLM 기반 코드 보정 방식인 DeepSeek-R1 70B, Llama 3.1 70B를 비교하였다.

그림 2는 세 가지 평가 지표에 대한 비교 결과를 보여준다. 그림에서 알 수 있듯이 MTV-FL은 코드 호환성 확보 측면에서 평균 4.3회의 반복만으로 훈련을 가능하게 한 반면, DeepSeek-R1 70B와 Llama 3.1 70B는 각각 12.6회와 19.2회의 반복이 요구되어 제안하는 방법이 현저히 빠른 수렴 특성을 보였다. 또한 악의적 작업 제거 성능에서도 MTV-FL은 평균 2.7회의 반복으로 정책 위반 요소를 제거한 반면, DeepSeek-R1 70B는 9.1회, Llama 3.1 70B는 13.3회의 반복이 필요하였다. 훈련 실행 성공률 측면에서는 MTV-FL이 95%로 가장 높은 성공률을 기록한 반면, DeepSeek-R1 70B와 Llama 3.1 70B는 각각 55%와 35%에 그쳤으며 아마저도 치명적인 오류의 경우 사용자 피드백을 통해 개선하는 절차를 포함했다. 이는 MTV-FL이 사전에 정의된 정책 평가 함수를 기반으로 코드 실행 흐름을 반복적으로 검증하고, 샌드박스 내 실행 기반 피드백을 통해 비호환 요소와 정책 위반 가능성을 점진적으로 감소시키는 구조를 가지기 때문이다. 결과적으로 MTV-FL은 단순한 코드 생성 또는 수정 방식과 달리, 정책 제약 하에서 훈련 코드를 안정적으로 수렴시키는 검증 기법으로서 연합학습 훈련 프로세스의 신뢰성과 안정성을 효과적으로 향상시킴을 확인할 수 있다.

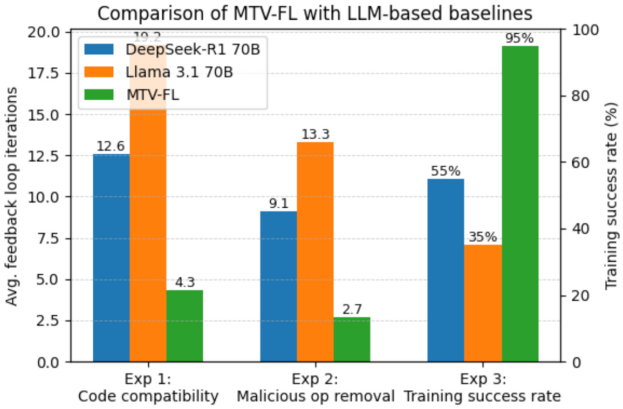

그림2. MTV-FL 호환성 검증 성능 비교

Ⅳ. Conclusion

본 논문에서는 연합학습 환경에서 모델 개발자가 제출한 훈련 코드의 실행 신뢰성과 플랫폼 호환성 문제를 해결하기 위해, 멀티에이전트 기반 훈련 검증 기법인 MTV-FL을 제안하였다. MTV-FL은 훈련 코드의 실행 흐름을 플랫폼 정책 제약 하에서 정합성 수렴 문제로 정의하고, 정책 평가 함수와 샌드박스 기반 실행 검증을 결합한 반복적 피드백 구조를 통해 훈련 코드를 점진적으로 보완한다. 시뮬레이션 결과, MTV-FL은 LLM 기반 코드 보정 방식과 비교하여 코드 호환성 확보 속도, 악의적 작업 제거 효율, 그리고 최종 훈련 실행 성공률 측면에서 일관된 성능 향상을 보였다. 향후 연구에서는 보다 복잡한 연합학습 시나리오와 다양한 정책 제약 환경에서 MTV-FL의 일반화 가능성을 검증하고, 자동화된 정책 학습 및 적응적 검증 전략으로의 확장을 고려할 수 있을 것이다.

ACKNOWLEDGMENT

본 연구는 2026년도 보건복지부 및 과학기술정보통신부의 재원으로 연합학습 기반 신약개발 가속화 프로젝트사업 지원에 의하여 이루어진 것임 (과제 고유번호 : RS-2024-00459866, 기여율 100%)

참 고 문 헌

[1] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A., Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics (pp. 1273-1282). PMLR, 2017.

[2] Li, L., Fan, Y., Tse, M., & Lin, K. Y., A review of applications in federated learning. Computers & Industrial Engineering, 149, 106854, 2020.

[3] Shalom, O., Leshem, A., & Bajwa, W. U. Mitigating data injection attacks on federated learning. InICASSP IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) pp. 9116-9120, 2024

[4] Yang, Y., Hu, X., Gao, Z., Chen, J., Ni, C., Xia, X., & Lo, D. Federated learning for software engineering: A case study of code clone detection and defect prediction.IEEE Transactions on Software Engineering, 50(2), 296-321, 2024