사물인터넷

연합 학습 기반 IoT 이상 탐지 논문 발표

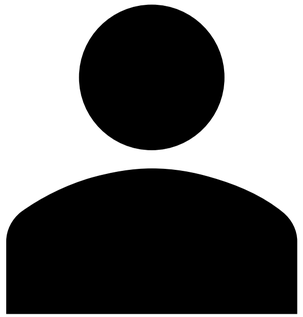
팀원
권우현 박준성 김준서

발표자
권우현

발표일시
2025년 12월 5일

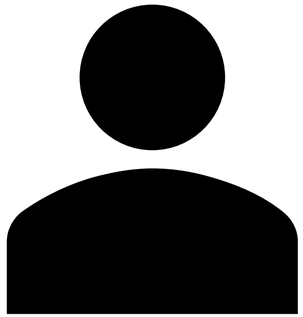
목차

팀원 소개



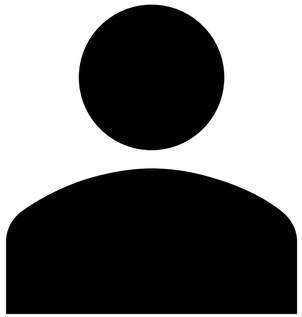
권우현

- 팀장 및 발표자
- 실험
- FedDetect 알고리즘 구현
- 결과 시각화



박준성

- 실험
- 데이터 전처리
- PPT 제작



김준서

- 실험
- 데이터셋 분석
- PPT 제작

연구 배경 및 필요성

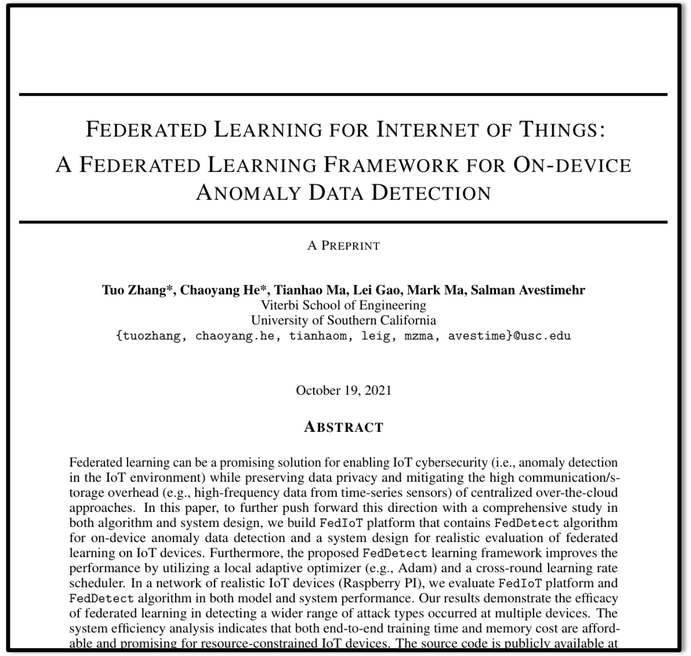
논문 선정 이유

논문 선정 제목:

Federated Learning fore Internet of Things:

A Federated Learning Framework for On-Device Anomaly Data Detetion (ACM SenSys 2021)

➢ 번역: IoT를 위한 연합 학습: 온-디바이스 이상 데이터 탐지를 위한 연합 학습 프레임워크



FEDERATED LEARNING FOR INTERNET OF THINGS:
A FEDERATED LEARNING FRAMEWORK FOR ON-DEVICE ANOMALY DATA DETECTION

A PREPRINT

Tuo Zhang*, Chaoyang He*, Tianhao Ma, Lei Gao, Mark Ma, Salman Avestimehr
Viterbi School of Engineering
University of Southern California
{tuozhang, chaoyang.he, tianhaom, leig, mzma, avestime}@usc.edu

October 19, 2021

ABSTRACT

Federated learning can be a promising solution for enabling IoT cybersecurity (i.e., anomaly detection in the IoT environment) while preserving data privacy and mitigating the high communication/storage overhead (e.g., high-frequency data from time-series sensors) of centralized over-the-cloud approaches. In this paper, to further push forward this direction with a comprehensive study in both algorithm and system design, we build `FedIoT` platform that contains `FedDetect` algorithm for on-device anomaly data detection and a system design for realistic evaluation of federated learning on IoT devices. Furthermore, the proposed `FedDetect` learning framework improves the performance by utilizing a local adaptive optimizer (e.g., Adam) and a cross-round learning rate scheduler. In a network of realistic IoT devices (Raspberry PI), we evaluate `FedIoT` platform and `FedDetect` algorithm in both model and system performance. Our results demonstrate the efficacy of federated learning in detecting a wider range of attack types occurred at multiple devices. The system efficiency analysis indicates that both end-to-end training time and memory cost are affordable and promising for resource-constrained IoT devices. The source code is publicly available at

선정 이유

- 단순 FL 알고리즘이 아닌, IoT 기기 환경에서 발생하는 '이상 탐지'라는 핵심 문제가 주제에 적합하다고 판단
- Federated learning에 대한 기초적인 지식에 대한 기반을 함께 학습할 수 있음
- FedDetect라는 개념을 도입하여 기존의 이상 탐지에 한층 더 고수준의 알고리즘을 구현하고, 플랫폼 구축을 목적으로 하여 논문의 실용성 및 목적성이 뛰어남

수행 계획

1) 논문 심층 분석 및 환경 구축

- 논문의 전반적인 내용을 깊이 있게 이해하고, 충분히 설명할 수 있을 정도의 지식 습득
- 논문에서 제시한 FedIoT 아키텍처 및 Deep Autoencoder 모델 구조 파악
- FL 시뮬레이션 환경(FedML/Python) 구축 및 Github 코드 분석

2) 데이터 분할 및 로컬 구현

- IoT 이상 탐지 데이터셋(추가 데이터셋)을 클라이언트별로 분할
- Autoencoder 모델을 구성하고, 각 클라이언트에서 로컬 학습 기능 구현

3) FedIoT 알고리즘 재구현 및 검증

- FedAvg를 기반으로 이상 탐지 모델 업데이트 및 중앙 서버 집계 로직 구현
- 논문에서 제시된 지표를 사용해 성능 평가(Accuracy, Recall)

4) 최종 정리 및 발표 준비

- 구현 결과 정리 및 시각화
- 논문 분석 내용, 구현 과정의 난점 및 해결책을 중심으로 발표 자료 준비

연구 배경: IoT 보안의 중요성

- 전 세계 수십억 **IoT** 기기의 확산으로 인해 공격 표면이 기하급수적으로 확대
- IoT 장치가 수집하는 개인정보 및 운영 데이터의 민감도 상승으로 인한 보안 위험 증가
- 에지에서 발생하는 고빈도/대용량 네트워크 트래픽으로 인한 효율적인 이상 탐지의 어려움
- 중앙 서버 의존 시 프라이버시, 비용, 지연 문제 심화
- **Mirai, BASHLITE** 등 IoT 타깃 봇넷 공격 증가로 분산 방어 체계 필요성 대두

문제 정의 및 기존의 한계

문제 정의 및 기존 연구 한계

클라우드 중심 접근법의 한계

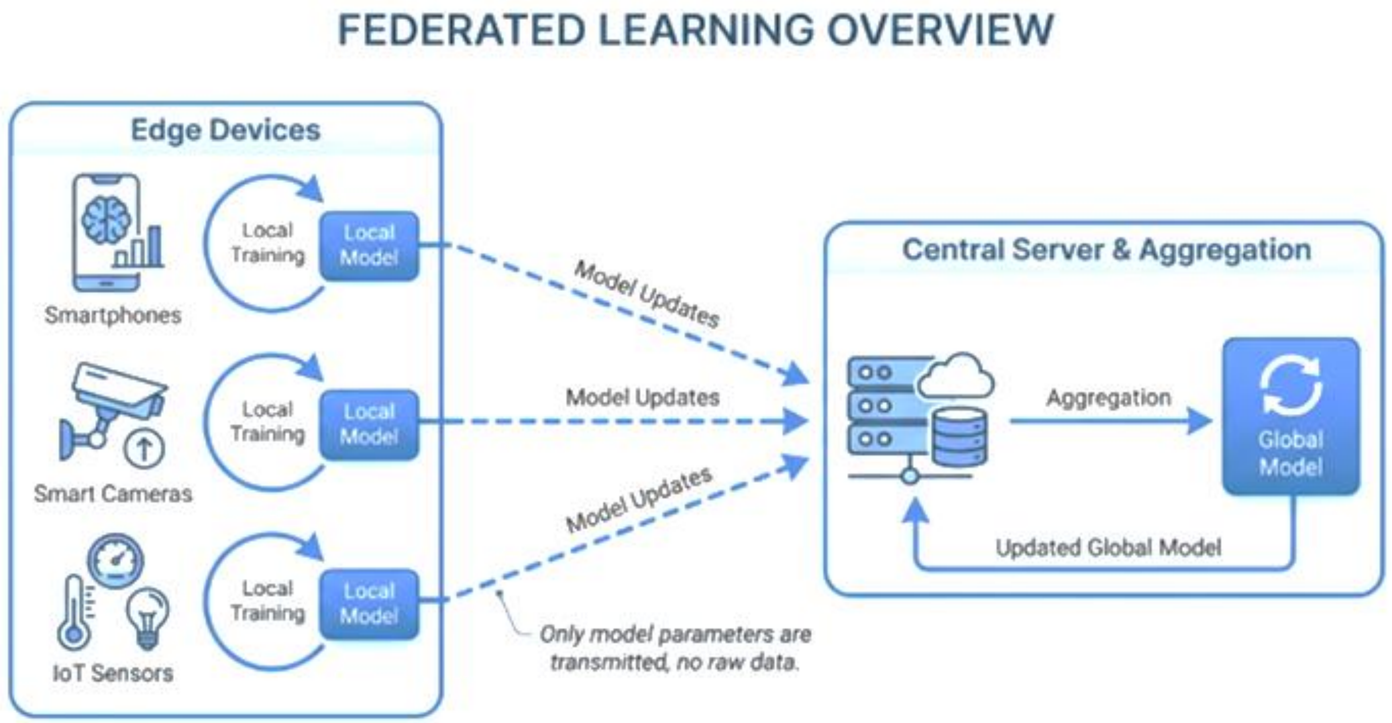
- ❑ 데이터 수집·전송 오버헤드
  - ➢ 고빈도 IoT 데이터의 중앙 서버 전송 시 대역폭/지연 문제
- ❑ **개인정보 민감성**
  - ➢ 민감한 IoT 데이터의 중앙화로 인한 개인정보 및 규제 리스크 증가
- ❑ Non-IID/**불균형 데이터**
  - ➢ 장치별 다양한 데이터 특성에 중앙 모델 대응 어려움
- ❑ **단일 장애점**(Single Point Of Failure)
  - ➢ 중앙 서버 장애 시 전체 시스템 취약

필요한 접근 방식 요구사항

- ❑ 온-디바이스 학습과 프라이버시
  - ➢ 민감 데이터를 로컬에 유지하며 학습
- ❑ 통신량 절감 및 장치 확장
  - ➢ 모델 파라미터만 교환, 다수 장치 확장 가능
- ❑ 다양한 장치 지원
  - ➢ 다양한 리소스와 데이터 특성을 가진 장치들의 협업 가능
- ❑ 개인 및 글로벌 조화
  - ➢ 장치별 특성 유지하며 전체 모델 개선

제안 방법

연합 학습(Federated Learning) 개념



1. 장치 로컬에서 모델 학습
   데이터는 장치에 유지
2. 모델 파라미터만 서버로 전송
   프라이버시 보호
3. 서버는 파라미터 집계
   글로벌 모델 생성
4. 통신 라운드 반복, 협업 학습 실현

관련 연구

01. FL 기초 알고리즘

- FedAvg: 기본 FL 알고리즘, 클라이언트 모델 가중치 평균화
- FedProx: 이질성 해결, 근접 항 추가해 로컬 최적화 제약
- SCAFFOLD: 클라이언트 드리프트 수정 통해 수렴 가속
- FedOpt/FedAdam: 적응형 옵티마이저 도입한 서버 측 집계

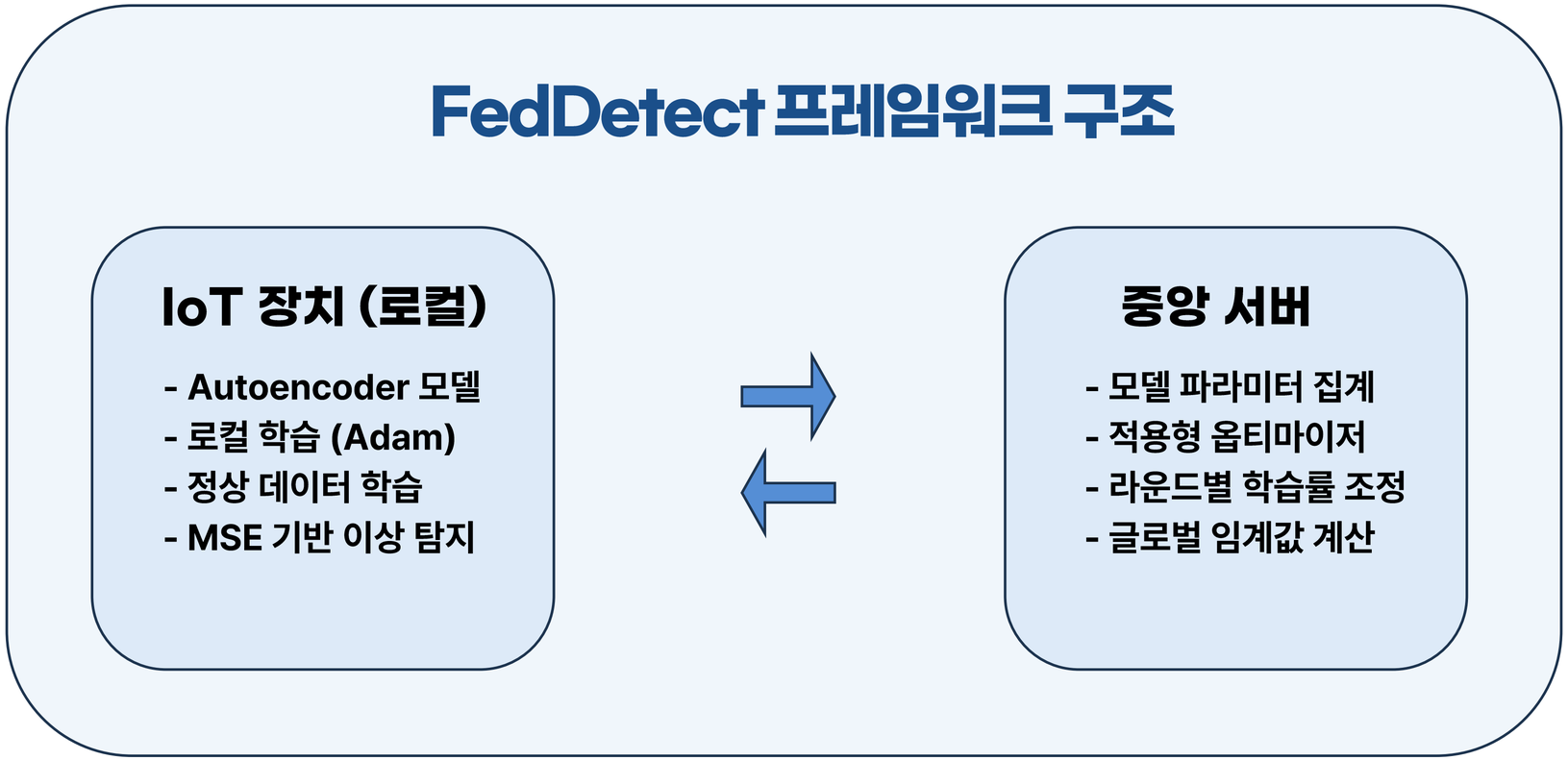
02. IoT 침입 탐지 연구

- Autoencoder 기반 이상 탐지: 재구성 오류 기반 비지도 탐지
- N-BaloT 활용 연구: 실제 IoT 기기 공격 데이터로 검증

03. 최신 동향

- 블록체인 결합: 검증 및 투명성 향상 위한 통합 연구
- 개인화 FL: 특정 클라이언트 맞춤형 모델 최적화 연구
- 통신 효율화: 압축, 양자화, 희소 업데이트 연구
- 비동기 FL: 클라이언트 동기화 없이 학습 최적화

FedDetect 프레임워크 개요



1. 목적: 온-디바이스 이상 데이터 탐지 성능 극대화
2. 핵심 구성: Autoencoder 기반 로컬 학습 + 서버 적용형 집계
3. 차별점: Adam 옵티마이저 + 라운드 간 학습률 스케줄러
4. Global Threshold 모듈: 장치별 통합 임계값 산출

FedDetect **핵심 알고리즘**

01. 초기화: 글로벌 AE 파라미터

- 서버에서 초기 **Autoencoder 모델 파라미터**를 생성하여 모든 참여 IoT 장치에 배포
- 입력 115차원, 4개 은닉층 구성(75%, 50%, 33%, 25%)

02. 로컬 학습: Adam 옵티마이저

- 각 IoT 장치에서 **Adam 옵티마이저**를 사용하여 로컬 데이터로 학습
- 배치 크기 64, T 라운드 반복 수행하며 MSE 손실함수 최소화

03. 서버 집계: 적응형 옵티마이저

- 서버는 **FedAdam** 계열의 적응형 옵티마이저로 장치별 모델을 집계
- 기존 FedAvg와 달리 모멘텀과 적응적 학습률 활용으로 수렴 속도 개선

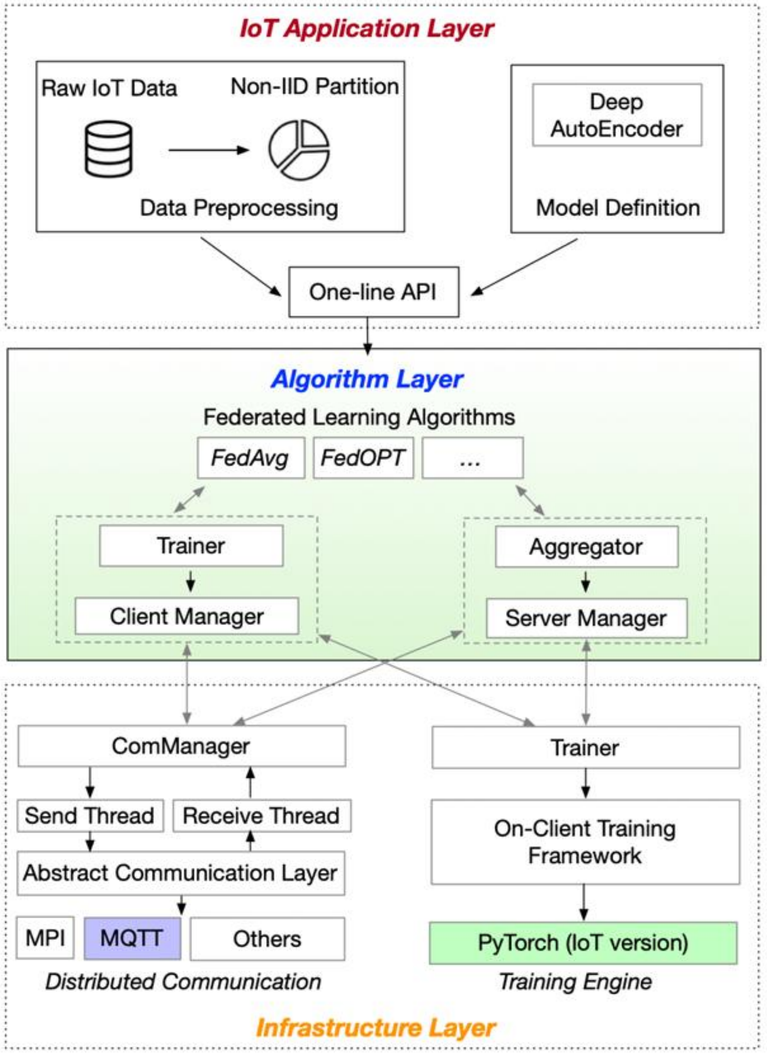
04. 학습률 스케줄링

- **라운드 간 학습률 동적 조정**으로 초기에는 빠른 학습, 후기에는 미세 조정이 가능하도록 설계
- 비선형 감쇠 또는 코사인 스케줄링 적용

05. 글로벌 임계값 산출

- 각 장치에서 정상 데이터로 **MSE 시퀀스**를 생성하고 서버에서 통합하여 장치별 최적화된 이상 탐지 임계값을 결정
- 이상 탐지 정확도 향상의 핵심 차별점

FedIoT 플랫폼 : 3계층 아키텍처



Application Layer

사용자 인터페이스와 API 제공

- 한 줄 API로 Autoencoder와 데이터 로더 등록
- 간편한 FL 설정 및 모델 훈련 시작 기능
- 데이터 전처리 및 결과 시각화 도구 제공

Algorithm Layer

FL 알고리즘과 학습 프로세스 관리

- FedDetect, FedAvg, FedOpt 등 다양한 알고리즘 지원
- 사용자 정의 트레이너와 집계 함수 확장 가능
- 클라이언트/서버 매니저로 인스턴스 관리

Infrastructure Layer

통신 및 하드웨어 인프라 관리

- MQTT/MPI 통신 백엔드로 효율적 메시징
- Raspberry Pi, Jetson Nano 등 엣지 장치 지원
- 맞춤형 PyTorch 라이브러리 기반 실장치 학습

데이터 및 시스템 구성

실험 설계: 데이터셋·모델·환경

01. N-BaIoT 데이터셋

- 데이터 구성: 9개 상용 IoT 장치의 네트워크 트래픽
- 공격 유형: Mirai 및 BASHLITE 계열 악성코드 공격
- 데이터 분할: 정상 5000(학습)/3000(평가), 균형 테스트셋 생성
- 테스트셋 합성: 각 공격유형 500개 + 정상 500개로 균형화

02. 모델 구조

- 입력 차원: 115D 네트워크 트래픽 특성 벡터
- Autoencoder: 인코더 4층 (75%-50%-33%-25%), tanh 활성화
- 하이퍼파라미터: batch size 64, epoch 120, Rounds 30
- 로컬 옵티마이저: Adam, 서버 집계: FedAdam

03. 실험 환경

- 하드웨어: 9×Raspberry Pi 4B (클라이언트) + GPU 서버
- 통신: MQTT 프로토콜, 무선 환경
- 비교 기준: CL-Single vs. CL-Combined vs. FL-FedDetect
- 평가 지표:
ACC (Accuracy), TPR (True Positive Rate), FPR (False Positive Rate), TNR(True Negative Rate)

실험 결과 및 실제 구현 결과

실험 결과(1): 탐지 성능 비교

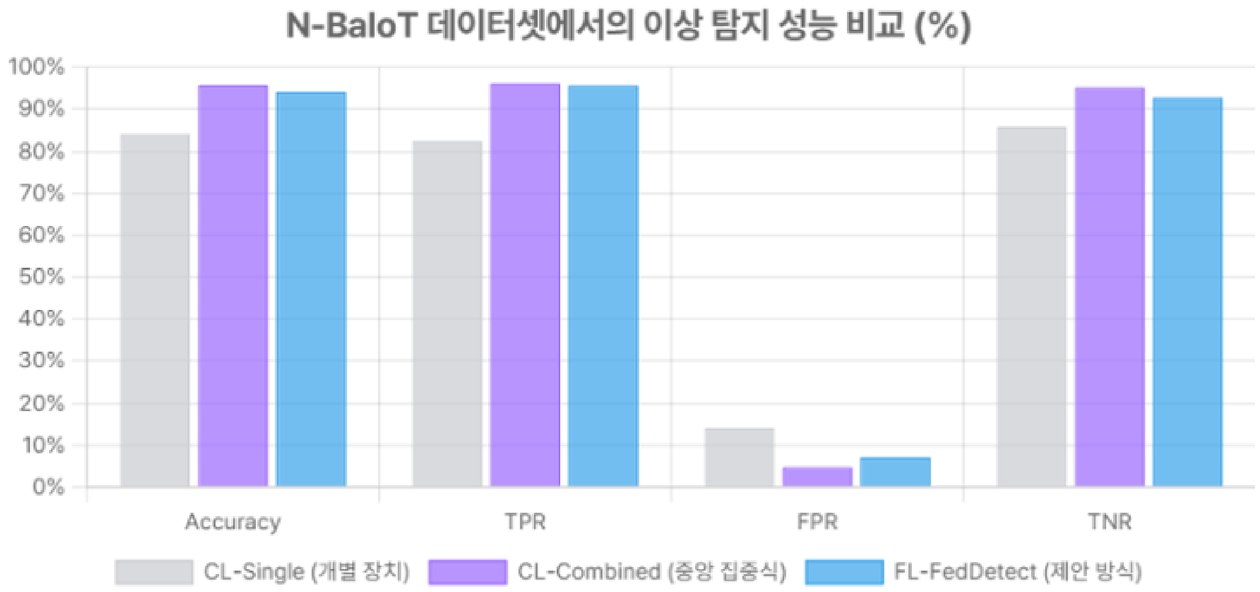


Table 1: Performance of anomaly detection under both centralized training and federated training

	Acc	FPR	TPR	TNR
CL-Single	73.82%	37.50%	86.56%	62.50%
CL-Combined	98.64%	2.71%	99.99%	97.29%
FL-FedDect	98.27%	3.45%	99.99%	96.55%

Table 2: CPU/GPU Training v.s. IoT Edge Training

	Acc	FPR	TPR	TNR
Simulation	98.27%	3.45%	99.99%	96.55%
Raspberry Pi	97.47%	4.78%	99.99%	95.22%

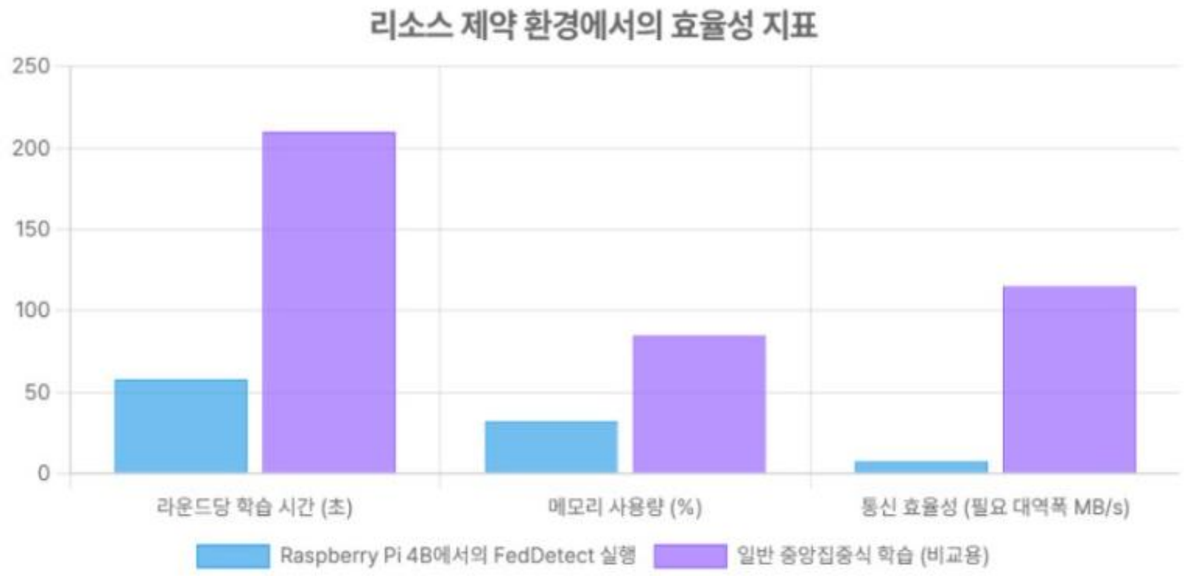
비교 결과 요약

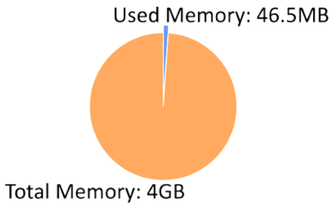
- FedDetect는 중앙집중식 학습(CL-Combined) 상황에 근접한 우수한 성능을 보이며, 개별 장치 학습(CL-Single) 대비 우수한 이상 탐지 정확도 달성
- 특히 IoT 디바이스 간 데이터 협업을 통해 TPR(True Positive Rate)이 크게 향상되었으며, FPR(False Positive Rate)은 적절히 유지

성능 지표 설명

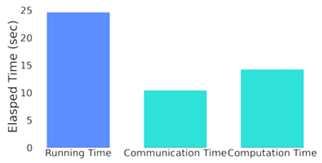
- Accuracy: 전체 탐지 정확도
- TPR: 공격 탐지 성공률(재현율)
- FPR: 오탐률(위양성율)
- TNR: 정상 트래픽 정확도(특이도)

실험 결과(2): 효율성 분석





(a) Memory Percentage



(b) Running Time Per Round

시스템 효율성 요약

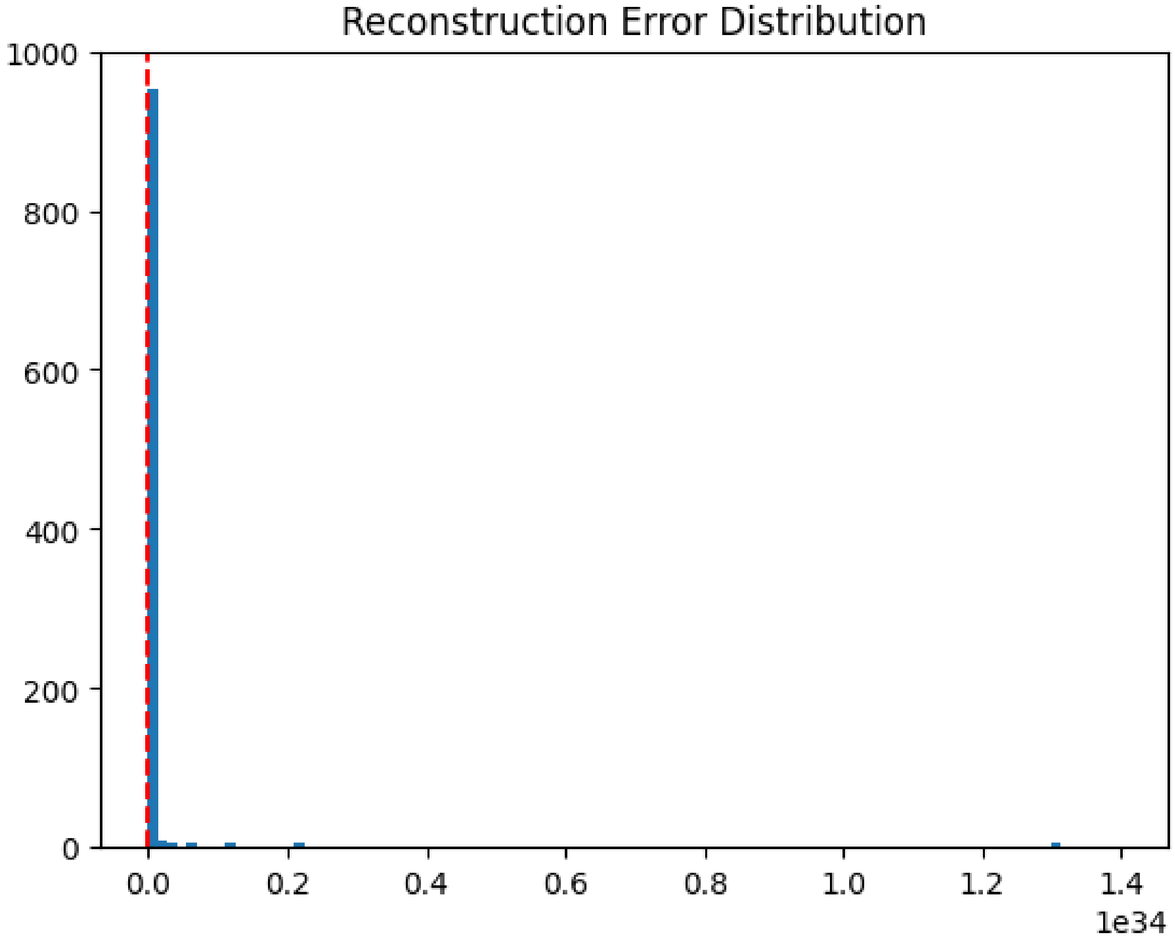
- FedDetect는 Raspberry Pi 4B와 같은 제한된 리소스 환경에서도 효율적으로 작동 가능
- 라운드당 학습 시간은 1분 미만으로, 실용적인 온-디바이스 학습이 가능

- 메모리 사용량은 제한된 장치에서도 부담이 없으며, 7.65MB/s의 일반적인 무선 환경에서도 원활한 연합 학습을 수행할 수 있음

통신 효율성

- FedIoT 플랫폼은 MQTT 기반 경량 메시징을 사용하여 통신 오버헤드를 최소화합니다. 모델 파라미터만 전송하므로 중앙 집중식 방식 대비 데이터 전송량이 크게 감소
- FedDetect의 효율성은 다양한 IoT 환경에서 실시간 이상 탐지를 위한 실제 배포 가능성을 보여줌

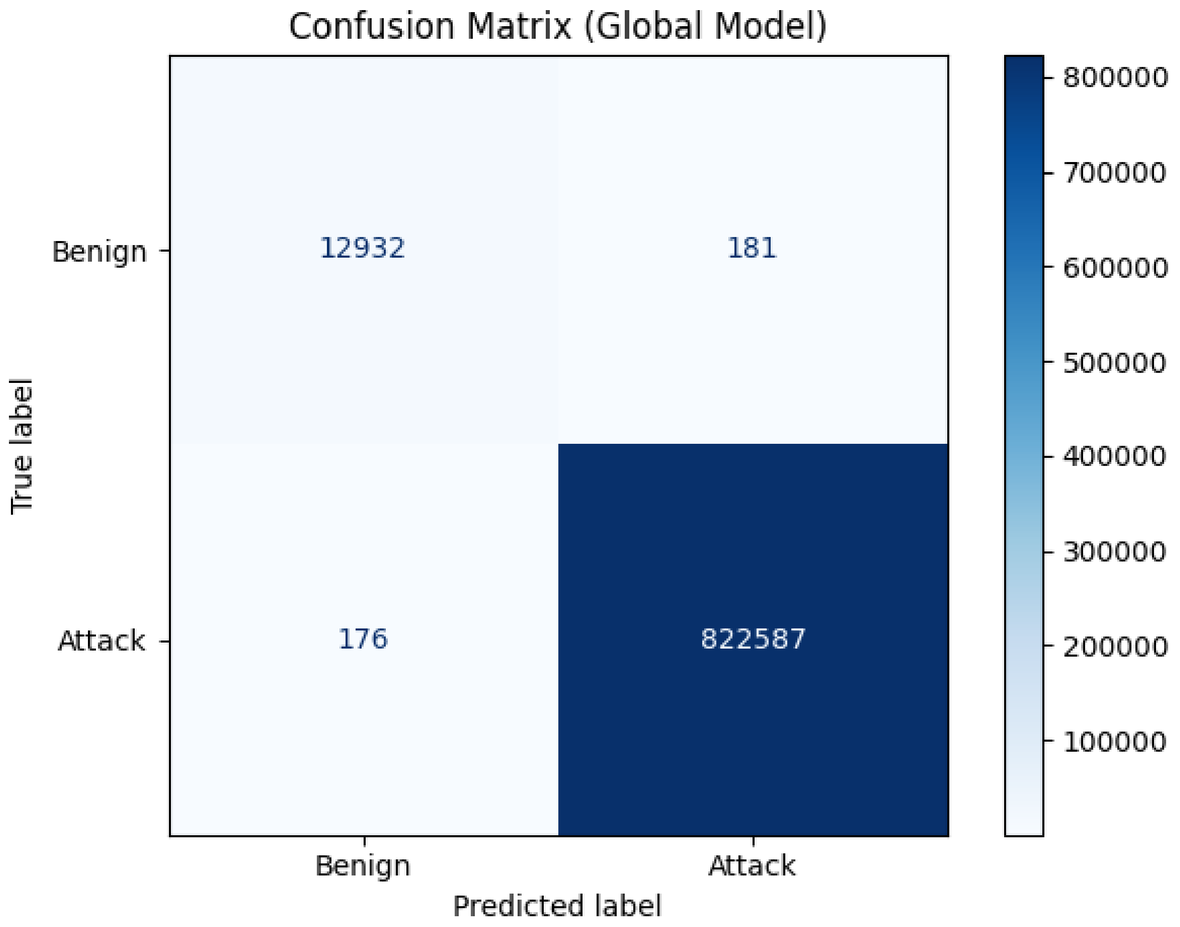
실제 재현 결과(1): Threshold



Global Threshold

- FedIoT는 각 클라이언트들의 개별 학습 후, Threshold를 중앙으로 보내 FedAvg를 통하여 Global Threshold를 정의
- Threshold 값은 약 0.0324로 수렴하였고, 이는 정상데이터만을 학습했기 때문에 생긴 값
- 차후 Threshold는 공격 탐지에 대한 이진 분류에서 사용됨

실제 재현 결과(2): Confusion Matrix



Confusion Matrix

- 행렬의 각 면은 정상 데이터를 정상 판단 및 오탐, 공격 데이터의 오탐 및 공격 판단으로 나뉘어짐
- 일반적으로, 네트워크 침입은 공격 트래픽이 많기 때문에, 정상데이터 보다 공격 데이터의 수가 많음

평가 지표

- 오차 행렬의 평가 지표로 Accuracy, Recall, Precision, F1-score를 선택
- 지표는 각각 0.99, 0.98, 0.98, 0.98로 실험 값과 유사한 수치 달성

결론 및 향후 연구

논문의 기여점 결론

01. 알고리즘 기여 (FedDetect)

- 적용형 옵티마이저: FedAvg 대신 Adam 기반 집계로 성능 향상
- 라운드 간 LR 스케줄러: 라운드별 학습률 동적 조정으로 안정적 수렴
- Global Threshold: 디바이스별 MSE 시퀀스로 통합 임계값 산출

02. 시스템 기여 (FedIoT)

- 실제 하드웨어 테스트베드: Raspberry Pi 기반 온디바이스 검증
- 3계층 아키텍처: 애플리케이션/알고리즘/인프라 계층 분리로 확장성 제공
- Python 통합 개발: 단일 언어로 시스템 프로토타입 용이성

03. 방법론 & 실증 결과

- 테스트셋 합성 규칙: 재현 가능한 공정한 평가 시나리오 제안
- 다양한 공격 탐지: 여러 장치 협력으로 광범위한 공격 유형 탐지 가능
- 성능 근접도: 중앙학습(상한) 수준에 근접한 FL 성능 입증
- 자원 제약 실용성: 메모리/시간 효율적, 실제 IoT 환경 가능성 확인

한계점 및 향후 연구

현재 한계점

- **Non-IID 강도 증가**: 데이터 편차가 클수록 FPR 상승, 성능 저하 가능성
- **데이터 편향**: 단일 장치가 모든 공격 유형을 관찰하지 못함
- **모델 다양성**: Autoencoder 외 고급 모델 적용 시 추가 엔지니어링 비용
- **임계값 설정**: 장치별 특성 차이로 통합 임계값의 효과 제한적일 수 있음

향후 연구 방향

- **개인화 및 도메인 적응**: 장치별 특성 보존하며 지식 공유 (FedBN, FedPer)
- **비동기/셀렉티브 FL**: 모든 장치 동기화 없이 효율적 학습 방식 연구
- **임계값 학습 자동화**: 장치별 최적 임계값 동적 조정 메커니즘 개발
- **대규모 실제 배치 평가**: 다양한 실제 IoT 네트워크에서 검증 필요

감사합니다