한국빅데이터학회지
제9권 제1호, 2024, pp. 101-113 https://doi.org/10.36498/kbigdt.2024.9.1.101

연합 학습 환경에서의 랜덤 포레스트 알고리즘 최적화 전략 연구

Research on Optimization Strategies for Random Forest Algorithms in Federated Learning Environments

송인서 · 이강윤†

가천대학교 IT융합공학과 컴퓨터공학부

요 약

연합 학습은 분산 환경에서 데이터 프라이버시와 보안을 유지하면서 효율적으로 머신러닝 모델을 학습하는 방법으로 주목받고 있다. 본 연구에서는 이러한 연합 학습 환경에서 랜덤 포레스트 모델의 성능을 최적화하기 위해 새로운 FedRFBagging 알고리즘을 제안한다. 클라이언트별 데이터 특성에 기반하여 로컬 랜덤 포레스트 모델의 트리를 동적으로 조정함으로써 통신 비용을 줄이고, 다수의 클라이언트 환경에서도 높은 예측 정확도를 달성할 수 있다. 제안하는 방법은 다양한 데이터 조건에 적응하여 모델의 안정성과 학습 속도를 크게 향상시킨다. 랜덤 포레스트 모델은 여러 개의 결정 트리로 구성되나, 연합 학습 환경에서 모든 트리를 서버로 전송하면 통신 오버헤드가 기하급수적으로 증가하여 사용이 어려워진다. 또한 클라이언트 간 데이터 분포의 차이로 인해 트리의 품질 불균형이 발생할 수 있다. 이를 해결하기 위해 FedRFBagging 알고리즘을 제안하며 이는 각 클라이언트에서 성능이 높은 트리만을 선택해 서버로 전송하고, 서버는 불순도 값을 기준으로 트리들을 선택하여 최적의 글로벌 모델을 구성한다. 이를 통해 통신 오버헤드를 줄이고 다양한 데이터 분포에서도 높은 예측 성능을 유지할 수 있다. 글로벌 모델은 다양한 클라이언트 데이터를 반영하지만, 각 클라이언트의 데이터 특성은 다를 수 있다. 이를 보완하기 위해 클라이언트는 글로벌 모델에 추가 트리를 학습하여 로컬 데이터에 맞춘 최적화를 수행한다. 이를 통해 전체 모델의 예측 정확도를 높이고 변화하는 데이터 분포에 적응할 수 있다. 본 연구는 연합 학습 환경에서 랜덤 포레스트 모델이 가지는 통신 비용과 성능 문제를 효과적으로 해결하여 적용 가능한 연합 학습 환경에서 랜덤 포레스트 모델을 위한 알고리즘임을 시사한다.

■ 중심어 : 연합 학습, 랜덤 포레스트, 분산 환경

Abstract

Federated learning has garnered attention as an efficient method for training machine learning models in a distributed environment while maintaining data privacy and security. This study proposes a novel FedRFBagging algorithm to optimize the performance of random forest models in such federated learning environments. By dynamically adjusting the trees of local random forest models based on client-specific data characteristics, the proposed approach reduces communication costs and achieves high prediction accuracy even in environments with numerous clients. This method adapts to various data conditions, significantly enhancing model stability and training speed. While random forest models consist of multiple decision trees, transmitting all trees to the server in a federated learning environment results in exponentially increasing communication overhead, making their use impractical. Additionally, differences in data distribution among clients can

2024년 05월 27일 접수; 2024년 06월 17일 수정본 접수; 2024년 06월 24일 게재 확정.

* 본 연구는 보건복지부의 재원으로 한국보건산업진흥원의 보건의료기술연구개발사업 지원(과제: HI22C1651)과 한국연구재단의 기초연구사업 (grant number: NRF-2022R1F1A1069069) 지원에 의하여 이루어진 것입니다.

†교신저자 (keylee@gachon.ac.kr)

lead to quality imbalances in the trees. To address this, the FedRFBagging algorithm selects only the highest-performing trees from each client for transmission to the server, which then reselects trees based on impurity values to construct the optimal global model. This reduces communication overhead and maintains high prediction performance across diverse data distributions. Although the global model reflects data from various clients, the data characteristics of each client may differ. To compensate for this, clients further train additional trees on the global model to perform local optimizations tailored to their data. This improves the overall model's prediction accuracy and adapts to changing data distributions. Our study demonstrates that the FedRFBagging algorithm effectively addresses the communication cost and performance issues associated with random forest models in federated learning environments, suggesting its applicability in such settings.

■ Keyword : Federated Learning, Random Forest, Distributed Environment

Ⅰ. 서론

연합 학습은 분산 환경에서 데이터 프라이버시와 보안을 유지하면서 효율적으로 머신러닝 모델을 학습하는 방법으로 최근 많은 주목을 받고 있다.[1-2] 데이터가 로컬에서 학습되고 모델 업데이트만 중앙 서버로 전송하는 구조를 통해 데이터의 직접적인 이동 없이도 학습을 가능하게 하는데, 이러한 특징은 특히 민감한 개인정보를 다루는 응용 분야에서 중요한 이점을 제공한다.[3]

기존의 중앙 집중화 기반 환경에서 랜덤 포레스트, XGBoost와 같은 트리 기반의 머신러닝 모델들이 효과적으로 사용되어 왔다.[4-6] 특히, 신약 개발 분야에서 랜덤 포레스트 모델은 현재도 많이 사용되는 모습을 보이는데, 신약 후보 물질의 활성화 예측, 독성 예측, 약물-표적 상호작용 예측 등에서 우수한 성능을 나타내기 때문이다.[7-11]

연합 학습의 대부분의 연구는 딥러닝 기반 모델들 위주로 진행되어 왔으며, 이미 우수한 성능을 보이고 있다.[12-13] 현재 중요한 과제는 기존의 머신러닝 모델들을 어떻게 효과적이고 효율적으로 연합 학습 환경으로 전환할 것인가이다.[14-15] XGBoost는 연합 학습 환경에서 유망한 결과를 보였으나[16], 랜덤 포레스트를 적용하기 위한 연구들은 핵심 개념인 배깅을 사용하지 않고 부스팅을 활용했으며 확립된 방법은 없는 상황이다.

따라서, 중앙 집중화 기반 머신러닝 환경에서 사용하는 기존의 모델들을 새로운 연합 학습 환경에서 그대로 사용할 수 있는 방법이 필요하다. 본 연구에서는 랜덤 포레스트 모델에 초점을 맞추어, 기존의 모델들을 연합 학습 환경에서 효과적이고 손실 없이 사용할 수 있도록 지원하는 FedRFBagging 알고리즘을 제안한다.

랜덤 포레스트 모델을 분산 환경, 특히 연합 학습 환경에서 사용하는 데는 여러 가지 문제가 있다. 그 중 가장 큰 문제는 통신 오버헤드 문제이다. 랜덤 포레스트는 여러 개의 트리를 배깅하여 사용하지만, 각 클라이언트에서 배깅된 트리를 서버로 전송하고 집계하는 과정에서 라운드 수와 클라이언트 수가 증가함에 따라 통신 오버헤드가 기하급수적으로 증가하는데, 이러한 이유로 실제 연합 학습 환경에서 랜덤 포레스트 모델을 사용하는 것이 어려운 실정이다.

이 문제를 해결하기 위해 우리는 FedRFBagging 알고리즘을 연구했으며, 다른 연구들은 주로 랜덤 포레스트에 부스팅을 적용하여 통신 오버헤드 문제를 해결하려 했으나[17], 우리는 랜덤 포레스트의 핵심 개념인 배깅(Bagging)을 그대로 유지하면서 이 문제를 해결하고자 한다. 이는

FedRFBagging 알고리즘의 중요한 장점이며, 기존 연구들과의 차별점이다.

또한, 본 연구의 접근 방식은 클라이언트별 데이터 특성에 기반하여 로컬 랜덤 포레스트 모델의 트리를 동적으로 조정하여 각 클라이언트에서 성능이 높은 트리만 선택하여 서버로 전송함으로써 통신 비용을 줄이고, 연합 학습 환경에서 중앙 집중화 기반 환경에서의 학습된 머신러닝 모델과 비교하여 성능 차이가 거의 없으며, 데이터 특성에 따라 더 높은 예측 정확도를 달성할 수 있다.

본 연구는 기존에 진행되었던 관련 연구 검토를 통해 기존의 머신러닝 모델을 연합 학습 환경으로 전환하기 위한 접근법과 한계점을 분석하여 이를 극복하고자 한다. 제안하는 FedRFBagging 알고리즘의 개요를 자세히 설명하며 중앙 집중화 기반 환경과 연합 학습 환경에서의 랜덤 포레스트 모델의 성능 평가를 위해 사용한 데이터와 전처리 과정을 설명한다. 글로벌 모델과 클라이언트에서의 글로벌 모델의 성능 평가를 통해 도출된 결과를 바탕으로 제안하는 FedRFBagging 알고리즘이 기여하는 바를 논의한다.

Ⅱ. 관련 연구

2.1 연합 학습 환경에서의 XGBoost모델을 효과적으로 적용하기 위한 프레임워크

기존의 연합 학습 환경에서 XGBoost를 적용할 때 발생하는 통신 빈도 문제, 프라이버시 문제를 해결하는 프레임워크를 제시한다.[16] 해당 연구는 연합 학습에서 XGBoost의 학습률을 학습 가능하게 만듦으로써 앞에서의 두 문제를 해결했으며 이를 위해, 각 클라이언트에서 생성된 트리 앙상블의 학습률을 학습하는 1D CNN을 도입했다. 그래디언트와 헤시안을 공유할 필요가 없어 프라이버시 보호가 강화되었으며 실험 결과, 기존의 방법과 비교하여 성능이 유사하거나 특정 데이터셋에 대하여 더 우수했으며, 통신 효율성 측면에서 25배에서 700배까지 향상되었다. 하지만 클라이언트 수가 증가할 수록 구축된 모델의 예측 정확도가 저하되는 문제를 가지고 있다.

2.2 부스팅 기반의 연합 랜덤 포레스트 알고리즘

기존의 방법이 품질이 낮거나 불균형한 데이터를 가진 클라이언트에서는 효과가 미미하다는 한계를 해결하기 위해 수평적으로 분할된 데이터에 대해 부스팅 기반 연합 랜덤 포레스트 알고리즘 BOFRF (Boosting-based Federated Random Forest)를 제시했다.[17] 이 알고리즘은 부스팅과 배깅을 결합하여 각 클라이언트의 로컬 랜덤 포레스트 모델을 구축하고, 다른 클라이언트와 공유한다. 공유된 모델은 다른 클라이언트의 데이터셋에서 실행되어 매튜스 상관계수를 사용한 성능 측정을 진행한다. 이를 기반으로 가중치를 할당하여 최종 연합 모델을 생성한다. 네 가지 의료 데이터셋을 사용하여 성능을 평가했으며 기존의 로컬 랜덤 포레스트 모델에 비해 향상된 성능의 예측 결과를 확인했다. 하지만 이 연구는 클라이언트 간의 결정 트리 공유 시 데이터 통신 오버헤드를 줄이기 위한 최적화 방안이 다루어지지 않았으며 통신 비용과 시간 비용을 증가시킨다는 문제를 가지고 있다.

2.3 수직적 연합 학습에서의 랜덤 포레스트 시스템

대규모 현실 데이터셋에서 데이터 프라이버시를 보호하면서 효율적이고 견고한 수직적 연합 학습 랜덤 포레스트 시스템을 제시했다.[18] 해당 시스템은 SOTA (SecureBoost) 모델보다 최대 83배 빠른 학습 및 서빙 속도를 달성하였으며 병렬화 기법과 효율적인 분산 시스템 설계

를 통해 가능하다. 랜덤 포레스트의 특성상 개별 트리가 독립적으로 학습될 수 있기 때문에, 병렬 처리를 극대화하여 학습 시간을 단축했는데 전통적인 방식에서는 트리 깊이가 길어질수록 노드 간의 통신 지연이 커지는 문제를 해결하기 위해 예측 경로를 미리 계산하여 필요한 통신을 최소화하였다. 하지만 수직적 연합 학습은 각 조직이 서로 다른 특성의 데이터를 보유한 경우에 적용되는데, 실제 응용에서 데이터 특성에 따라 일반화 성능이 저하될 수 있으며, 트리의 깊이가 깊어질수록, 참여 조직 및 클라이언트의 수가 많아질 수록 추론 성능이 저하된다는 문제가 있다.

Ⅲ. 연구내용

랜덤 포레스트 모델은 여러 개의 결정 트리로 구성되며, 연합 학습 환경에서 각 클라이언트가 학습을 통해 배깅된 트리들을 서버로 전송하는 과정에서 클라이언트 수, 트리 수, 라운드 수에 따라 통신 비용이 기하급수적으로 증가하는 문제가 있다. 기존의 다른 연구들은 통신 오버헤드 문제를 해결하기 위해 랜덤 포레스트 모델에 부스팅을 사용했으나[17], 본 연구는 랜덤 포레스트 모델의 특성이며 안정적인 예측 성능을 제공하는 배깅을[19] 유지하여 문제를 해결하고자 한다.

연합 학습 환경에서 랜덤 포레스트 모델의 문제점인 통신 오버헤드를 줄이고, 중앙 집중식 환경에서의 성능을 유지하여 실용적으로 적용할 수 있는 새로운 FedRFBagging 알고리즘을 제안한다. FedRFBagging은 클라이언트별 데이터 특성에 기반하여 로컬 랜덤 포레스트 모델의 트리를 동적으로 조정함으로써 통신 오버헤드를 줄이고, 다양한 데이터 조건에 적응할 수 있다. 연합 학습 환경에서 중앙 집중화 기반 환경에 준하며 데이터 특성에 따라 더 높은 정확도를 달성할 수 있다. 제안하는 알고리즘의 주요 구성 요소와 절차는 <표 1>과 같다.

〈표 1〉 FedRFBagging 알고리즘

Algorithm 1: FedRFBagging

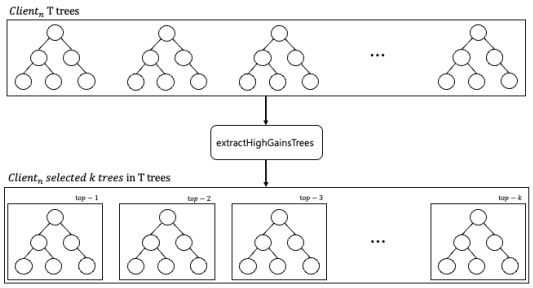
Input: Clients $\{C_1, C_2, ..., C_n\}$ with local datasets $\{D_1, D_2, ..., D_n\}$, Number of trees per client T, Number of high-gain trees to select num_tree_select, Server S

Output: Global Random Forest Model G, Evaluation Metrics $globalmodel\_metrics$

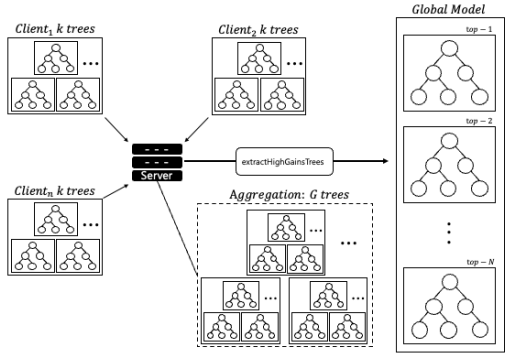
Initialization:
  Initialize global model G with an empty set of trees;
Client-side training:
  for *each client* C_i *in Clients* **do**
    Initialize local random forest model RF_i with T trees;
    Train RF_i on local dataset D_i;
    Calculate accuracy for each tree in RF_i;
    Select top-k trees based on accuracy to form $selected\_trees_i$;
    Convert $selected\_trees_i$ to byte format and send to Server S;
Server-side aggregation:
  Initialize empty set G_trees;
  for *each* $selected\_trees_i$ *received from client* C_i **do**
    Convert $selected\_trees_i$ from byte format to tree objects;
    Add $selected\_trees_i$ to G_trees;
  Calculate impurity for each tree in G_trees;
  Sort all trees in G_trees based on impurity;
  Select top-N trees from G_trees to form global model G;
  Convert global model G to byte format and store in $global\_model$;
Client-side fine-tuning (after the first round):
  for *each client* C_i *in Clients* **do**
    Download global model G from Server S;
    Convert global model G from byte format to tree objects;
    Add additional T trees to G and train on local dataset D_i;
    Calculate accuracy for all trees in G;
    Select top-k trees based on accuracy to form $new\_selected\_trees_i$;
    Convert $new\_selected\_trees_i$ to byte format and send to Server$S$;
    Store local model G_i (including additional trees) for local use;
Evaluation:
  for *each client* C_i *in Clients* **do**
    Evaluate local model G_i on local test dataset;
    Collect and aggregate evaluation metrics;
Global model evaluation:
  Load evaluation dataset on server;
  Convert global model G from byte format to tree objects;
  Use global model G to make predictions on evaluation dataset;
  Calculate evaluation metrics;
  Store metrics in $globalmodel\_metrics$;
Output:
  Output global model G and evaluation results $globalmodel\_metrics$;

알고리즘은 초기화 단계, 로컬 학습 단계, 글로벌 조정 단계, 동적 조정 단계로 구성된다.

먼저 글로벌 모델 G를 빈 트리 집합으로 초기화한다. 이 단계에서는 각 클라이언트가 로컬 데이터셋을 이용해 독립적으로 학습을 시작하기 전의 준비 과정을 포함한다. 이후 각 클라이언트 C_i는 로컬 랜덤 포레스트 모델 RF_i를 T 개의 트리로 초기화하고, 로컬 데이터셋 D_i를 사용해 학습을 진행한다. <그림 1>과 같이 학습이 완료되면 각 트리의 정확도를 계산하여 상위 k 개의 트리를 선택하여 *selected_trees*를 구성한다. 이 선택된 트리들은 바이트 형식으로 변환되어 서버 S로 전송된다.



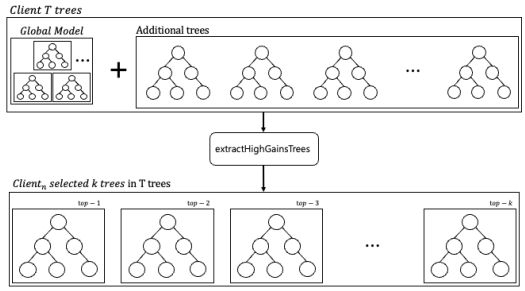
〈그림 1〉 클라이언트 측 트리 선택



〈그림 2〉 서버 측 집계 및 트리 선택

서버는 <그림 2>와 같이 각 클라이언트로부터 받은 *selected_trees*를 수신하여, 이를 트리 객체로 변환한 후 *G_trees*집합에 추가한다. 모든 트리가 수집되면 불순도를 계산하여 트리를 정렬하고 상위 N 개의 트리를 선택하여 글로벌 모델 G를 구성한다. 글로벌 모델 G는 바이트 형식으로 변환되어 클라이언트에 송신한다.

첫 라운드 이후, <그림 3>과 같이 각 클라이언트는 서버로부터 글로벌 모델 G를 다운로드 받아 트리 객체로 변환한다. 이후, 추가적인 T 개의 트리를 수신한 글로벌 모델 G에 추가하고 로컬 데이터셋을 사용해 학습한다. 모든 트리의 정확도를 계산한 후, 상위 k 개의 트리를 선택하여 *new_selected_trees*를 구성하고, 이를 서버로 전송한다. 또한, 추가 학습된 트리를 포함한 로컬 모델 G_i를 로컬 용도로 저장한다.



〈그림 3〉 클라이언트 측 추가 트리 학습 및 선택

각 클라이언트는 로컬 테스트셋을 사용하여 로컬 모델 G_i를 평가하고 결과를 수집하여 통합한다. 서버는 평가를 위해 준비된 데이터셋을 사용하여 글로벌 모델 G를 사용하여 예측을 수행한 후 평가 지표를 계산하여 *globalmodel_metrics*에 저장한다.

FedRFBagging 알고리즘은 각 클라이언트의 로컬 데이터를 활용하여 개별 모델을 학습하고, 이를 중앙 서버에서 통합하여 최종 글로벌 모델을 구축하는 과정을 통해 데이터 프라이버시를 보호하면서도 높은 성능의 예측 모델을 달성할 수 있다. 초기화 단계에서 로컬 랜덤 포레스트의 트리 수, 서버로 전송하기 위한 로컬 모델의 트리 수, 서버 측에서 글로벌 모델을 구성할 때의 트리 수를 직접 제어하여 라운드 수, 클라이언트 수, 데이터의 특성에 따라 조정이 가능하다.

상위 성능 트리만을 전송하는 최적화된 방안을 통해 통신 오버헤드를 효과적으로 줄일 수 있으며, 데이터의 크기가 줄어들어 네트워크 부하가 감소하고, 서버 측에서 처리해야 하는 데이터의 양이 줄어들어 모델 통합 과정이 더 신속해지고 효율성이 향상된다는 이점을 가진다.

Ⅳ. 실험

본 연구는 FedRFBagging 알고리즘의 성능을 평가하기 위해 <표 2>와 같이 5개의 이진 분류 데이터셋을 사용했다.[16] 해당 데이터셋은 머

신러닝 알고리즘의 성능을 평가 및 비교하기 위해 사용되는 데이터셋이며[20-25], 내용은 다음과 같다.

a9a데이터셋은 성인의 인구 통계학적 정보를 바탕으로 연간 소득이 50,000달러를 초과하는지를 예측하는 이진 분류 문제를 다루며 데이터셋은 총 32,561개의 데이터 포인트, 17개의 변수로 구성되어 있다. cod-rna데이터셋은 유전자 시퀀스 데이터를 기반으로 특정 유전자 서열이 주어진 클래스에 속하는지를 예측하는 이진 분류 문제를 다루며 데이터셋은 총 59,535개의 데이터 포인트, 10개의 변수로 구성되어 있다. ijcnn1 데이터셋은 국제공동학습 기술 컨퍼런스 (International Joint Conference on Neural Networks)에서 제공하는 데이터셋으로 49,990개의 데이터 포인트, 14개의 변수로 구성되어 있다. HIGGS 데이터셋은 고에너지 물리학 실험에서 발생하는 입자 충돌 데이터를 기반으로 힉스 보손의 존재를 예측하는 이진 분류 문제를 다루며 11,000,000개의 데이터 포인트, 28개의 변수로 구성되어 있다. SUSY데이터셋은 초대칭성 (Supersymmetry)을 검출하기 위한 고에너지 물리학 실험 데이터를 기반으로 입자 충돌 이벤트를 분류하는 이진 분류 문제를 다루며 총 5,000,000개의 데이터 포인트, 20개의 변수로 구성되어 있다. 사용한 모든 데이터셋은 공개되어 있으며 LIBSVM 데이터 웹사이트에서 확인할 수 있다.[1)]

a9a, cod-rna 데이터셋의 결측 값은 0으로 채웠으며 HIGGS, SUSY 데이터셋은 1,000,000개의 데이터 포인트만 선택해 실험을 진행했다.[16] 2개, 5개, 10개의 클라이언트 상황에서 글로벌 모델과 클라이언트 모델의 성능을 평가했으며 이를 위해 모든 데이터셋은 각 데이터 포인트에 대해 무작위로 파티션 ID를 할당하여 글로벌 모델을 평가하기위한 데이터셋을 포함하여 총 11개의 파티션으로 나누었다.

FedRFBagging 알고리즘을 사용한 연합 학습 환경에서의 랜덤 포레스트 모델과 중앙 집중화 기반 환경의 랜덤 포레스트 모델의 성능을 평가하기 위해 정확도 (Accuracy, acc)와 ROC-AUC (Receiver Operating Characteristic, auc)를 평가 지표로 사용했다. 정확도는 다음과 같이 정의된다.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

여기서, TP (True Positive)는 실제로 양성인 데이터를 양성으로 예측한 경우, TN (True Negative)은 실제로 음성인 데이터를 음성으로 예측한 경우, FP (False Positive)는 실제로 음성인 데이터를 양성으로 예측한 경우, FN (False Negative)은 실제로 양성인 데이터를 음성으로 예측한 경우를 의미한다. 정확도는 전체 데이터에서 모델이 올바르게 예측한 데이터 포인트의 비율을 나타내며 1에 가까울 수록 모델의 성능이 좋음을 의미한다.

ROC-AUC는 TPR (True Positive Rate)과 FPR (False Positive Rate) 간의 관계를 나타내는 ROC곡선 아래의 면적을 의미하며 다음과 같이 정의된다.

$$TPR = \frac{TP}{TP + FN}$$

TPR은 모델이 실제로 양성인 데이터를 얼마나 잘 예측하는지를 나타내는 지표로 양성 샘플 중에서 모델이 얼마나 잘 맞추었는지를 비율로 나타낸다.

$$FPR = \frac{FP}{FP + TN}$$

FPR은 모델이 실제로 음성인 데이터를 얼마나 잘못 예측했는지를 비율로 나타낸다.

1) https://www.csie.ntu.edu.tw/～cjlin/libsvmtools/datasets/binary.html

ROC-AUC는 TPR과 FPR을 사용하여 다음과 같은 적분으로 계산한다.

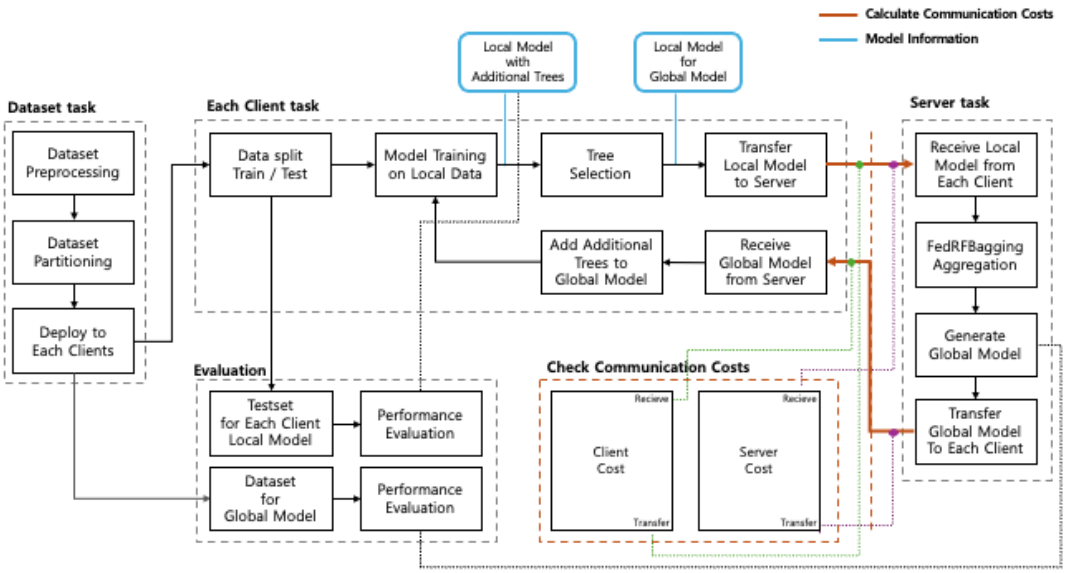
$$AUC = \int_0^1 TPR(FPR)d(FPR)$$

ROC-AUC 값은 0에서 1사이의 값을 가지며, 0.5인 경우는 무작위 추측 수준의 성능을, 1에 가까울 수록 모델의 성능이 좋음을 의미한다. 정확도와 ROC-AUC를 함께 사용함으로써, 이진 분류 문제에 대한 모델의 예측 성능과 판별 능력을 종합적으로 평가했다.

모든 데이터셋은 학습, 테스트 비율을 0.75 : 0.25로 했으며 정확도와 ROC-AUC값을 평가 지표로 하여 성능을 확인했다. 학습 라운드는 3, 로컬 랜덤 포레스트 모델의 트리 수(T)는 250, 상위 트리 수(k)는 250, 1라운드 이후 추가 학습 트리 수(T)는 100, 글로벌 모델(G)을 구성하는 트리 수(N)는 400으로 고정하여 <그림 4>와 같이 데이터셋 전처리, 분할 후 클라이언트에 이를 배포하는 데이터셋 작업과 학습 및 로컬 모델을 생성하는 클라이언트 작업, 로컬 모델을 수신한 후 FedRFBagging을 사용하여 글로벌 모델을 생성하는 서버 작업, 마지막으로 클라이언트와 서버에서 발생하는 통신 비용을 계산하는 실험을 진행했다. 학습에 참여한 클라이언트 모두가 모든 라운드에 참여했으며 실험 결과는 <표 3>과 같으며, 이는 중앙 집중식 방식과 비교한 FedRFBagging을 사용한 랜덤 포레스트 모델의 성능 평가 결과를 보여준다. <표 4>는 데이터셋과 클라이언트 수(2,10)에 따른 클라이언트와 서버에 대한 각각의 통신 비용, 전송 및 수신 데이터 용량을 보여준다.

〈표 2〉 데이터셋 요약

Dataset	Task Type	Data No.	Dimension
a9a	classification	32561	17
cod-rna	classification	59535	10
ijcnn1	classification	49990	14
HIGGS	classification	1000000	28
SUSY	classification	1000000	20



〈그림 4〉 본 연구에서 FedRFBagging 알고리즘의 연합 학습 절차

먼저, 각기 다른 클라이언트 수(2,5,10)에서의 성능을 비교하였으며, 표는 중앙 집중식 모델과 FedRFBagging 알고리즘을 사용한 글로벌 모델 및 클라이언트 모델의 acc와 auc 점수를 포함하고 있다. 중앙 집중식 모델은 데이터를 중앙에서 수집하여 학습한 모델로, 각 데이터셋에 대해 단일한 성능 지표를 나타낸다. FedRFBagging 알고리즘은 분산된 클라이언트 환경에서 각각의 클라이언트가 로컬 데이터를 사용하여 모델을 학습한 후, 이를 중앙 서버에서 집계하여 글로벌 모델을 형성한다.

2개의 클라이언트 환경에서는 a9a, cod-rna, HIGGS, SUSY 데이터셋에 대해 글로벌 모델은 중앙 집중식 모델보다 각각 1.1%, 2.9%, 6.6%, 0.5% 더 높은 성능을 보였으며, ijcnn1 데이터셋에서는 유사한 성능(0.2% 더 낮음)을 보였다. 클라이언트 모델의 경우 a9a, cod-rna, ijcnn1, HIGGS, SUSY 데이터셋에서 중앙 집중식 모델과 글로벌 모델보다 각각 2.7%, 3.1%, 0.8%, 6.8%, 0.6% 더 높은 성능을 보였다.

5개의 클라이언트 환경에서는 a9a, ijcnn1 데이터셋에서 글로벌 모델의 성능이 중앙 집중식 모델보다 각각 0.4%, 0.1% 더 낮은 성능을 보였으며, 나머지 데이터셋에서는 cod-rna 3.4%, HIGGS 0.6%, SUSY 1.5% 더 높은 성능을 보였다. 클라이언트 모델의 경우 모든 데이터셋에서 중앙 집중식 모델과 글로벌 모델보다 각각 2.0%에서 4.2% 사이의 높은 성능을 보이는 것을 확인할 수 있다.

10개의 클라이언트 환경에서는 a9a, cod-rna, ijcnn1, HIGGS, SUSY 데이터셋에서 글로벌 모델의 성능이 중앙 집중식 모델보다 각각 0.3%, 0.6%, 0.1%, 0.3%, 0.4% 낮은 성능으로 모든 데이터셋에서 글로벌 모델의 성능이 하락했음을 확인할 수 있는데, 클라이언트 모델의 경우 중앙 집중식 모델과 글로벌 모델과 비교했을 때, 2.8%에서 4.1% 더 높은 성능을 보이는 것을 확인할 수 있다.

<표 3>의 실험 결과를 종합해보면, FedRFBagging 알고리즘을 적용한 연합 학습 환경에서의 랜덤 포레스트 모델의 성능은 기존의 방법론에서 발생한 문제와 같이 클라이언트 수가 증가함에 따라 성능이 저하되는 문제가 발생하는데[11][13], 추가 학습된 트리를 포함한 로컬 모델을 활용함으로써, 클라이언트 모델의 성능이 중앙 집중식 모델과 글로벌 모델보다 항상 높게

〈표 3〉 중앙 집중식 방식과 비교한 FedRFBagging의 성능 평가

Dataset		Centralized Model	FedRFBagging					
			2 Clients		5 Clients		10 Clients	
			Global Model	Client Model	Gobal Model	Client Model	Global Model	Client Model
a9a	acc	0.8296	0.8387	0.8526	0.8418	0.8470	0.8316	0.8581
	auc	0.8932	0.8922	0.9100	0.8898	0.9110	0.8756	0.9102
cod-rna	acc	0.9053	0.9351	0.9352	0.9390	0.9315	0.9228	0.9307
	auc	0.9734	0.9824	0.9800	0.9825	0.9805	0.9678	0.9789
ijcnn1	acc	0.9562	0.9504	0.9637	0.9480	0.9628	0.9489	0.9645
	auc	0.9791	0.9723	0.9854	0.9662	0.9863	0.9580	0.9852
HIGGS	acc	0.6901	0.7568	0.7174	0.6966	0.7166	0.6711	0.7120
	auc	0.7618	0.8389	0.7919	0.7681	0.7907	0.7344	0.7882
SUSY	acc	0.7893	0.7995	0.7974	0.7951	0.7975	0.7842	0.7965
	auc	0.8567	0.8700	0.8624	0.8646	0.8687	0.8523	0.8676

나타났다. 이는 FedRFBagging 알고리즘이 로컬 데이터 특성에 맞춘 학습을 통해 각 클라이언트의 데이터 특성을 반영한 모델을 생성할 수 있음을 보여준다.

<표 4>의 통신 비용 결과를 보면 첫 번째 라운드에서는 각 클라이언트가 학습을 하고 난 후에 글로벌 모델이 생성되므로 클라이언트의 수신 비용이 없음을 확인할 수 있다.

데이터셋의 크기에 따라 통신 비용이 증가하는 경향이 있지만, 이 비용이 기하급수적으로 증가하지 않고 일정 수준에서 유지된다. 이는 HIGGS 데이터셋(654.5MB)의 경우에서 확인할 수 있듯이 2개의 클라이언트 환경에서 라운드 2의 클라이언트의 통신 비용은 전송 시 123.26MB, 수신 시 203.47MB이며, 3라운드에서는 각각 121.69MB, 200.37MB로 일정 수준을 유지하는 것을 확인할 수 있으며, 서버 또한 일정 수준을 유지하는 것을 알 수 있다. 또한 작은 크기의 a9a 데이터셋(1.6MB)의 경우에도 라운드 2에서 전송과 수신 각각 11.73MB, 18.94MB, 라운드 3에서 각각 11.76MB, 19.01MB로 일정 수준을 유지하고 있다.

클라이언트 수가 10개로 증가할 때 통신비용이 커지기는 하지만, 클라이언트 수가 2개인 경우와 비교했을 때 클라이언트 수 증가에 따라 자연스럽게 증가하는 수준이며, 기하급수적 증가는 보이지 않는다.

결론적으로, FedRFBagging 알고리즘은 분산된 연합 학습 환경에서 중앙 집중식 방식과 유사한, 또는 더 높은 성능을 달성할 수 있는 효과적인 방법임을 확인할 수 있다. 이는 클라이언트 간의 데이터의 Non-IID에 의한 편차에도 우수한 모델 성능을 유지하고, 분산된 환경에서 개별 학습 모델의 성능을 최적화하는 데 기여함을 시사한다. 또한 <표 4>에서 확인할 수 있듯 클라이언트와 서버 간의 통신 오버헤드 문제도 효과적으로 해결하였다. 데이터셋의 크기와 클라이언트 수의 증가에 따른 통신 비용을 관리

〈표 4〉 FedRFBagging 알고리즘의 통신 비용 결과

전송 및 수신 데이터 용량 (MB)

Dataset	Round	2 Clients		10 Clients	
		Client (전송 / 수신)	Server (전송 / 수신)	Client (전송 / 수신)	Server (전송 / 수신)
a9a (1.6MB)	Round 1	11.85 / 0	11.78 / 11.85	11.85 / 0	165.81 / 109.71
	Round 2	11.73 / 18.94	37.88 / 23.38	11.91 / 19.41	192.84 / 120.02
	Round 3	11.76 / 19.01	38.00 / 23.56	11.97 / 19.31	190.10 / 119.61
cod-rna (3.8MB)	Round 1	10.93 / 0	10.55 / 10.93	10.93 / 0	154.31 / 99.24
	Round 2	10.25 / 17.22	34.02 / 20.51	10.40 / 17.72	170.23 / 105.05
	Round 3	9.97 / 16.65	32.88 / 19.96	9.93 / 16.75	162.68 / 100.50
ijcnn1 (5.8MB)	Round 1	4.79 / 0	4.82 / 4.79	4.79 / 0	66.78 / 43.22
	Round 2	4.59 / 7.76	15.49 / 9.18	4.60 / 7.84	76.96 / 46.21
	Round 3	4.54 / 7.59	15.15 / 9.10	4.50 / 7.65	74.37 / 45.69
HIGGS (654.5MB)	Round 1	127.96 / 0	126.61 / 127.96	127.96 / 0	1758.93 / 1152.97
	Round 2	123.26 / 203.47	404.18 / 256.94	123.73 / 205.00	2021.62 / 1238.82
	Round 3	121.69 / 200.37	397.49 / 242.56	122.53 / 201.33	1980.85 / 1232.95
SUSY (2.26GB)	Round 1	92.64 / 0	92.74 / 93.89	92.74 / 0	1298.11 / 851.90
	Round 2	89.11 / 149.66	296.21 / 178.64	89.16 / 150.14	1455.24 / 894.27
	Round 3	86.57 / 144.93	288.97 / 173.39	86.96 / 144.32	1440.83 / 869.62

가능한 수준에서 유지하여 데이터셋의 크기, 클라이언트 수가 많은 환경에서도 효과적으로 관리할 수 있음을 알 수 있다. 따라서 FedRFBagging 알고리즘은 성능 최적화와 통신 효율성 두 가지 측면에서 모두 실용적으로 적용 가능함을 보여준다.

Ⅴ. 결론

본 연구에서는 안정적인 예측 성능을 제공하는 배깅을 활용하여 기존의 중앙 집중식 환경에서 사용되는 랜덤 포레스트 모델을 연합학습 환경으로 효과적으로 전환할 수 있는 FedRFBagging 알고리즘을 제안한다. 이는 분산 환경에서 데이터 프라이버시와 보안을 유지하는 연합 학습 환경에서 효과적으로 기존의 랜덤 포레스트 모델을 사용할 수 있도록 하며 해당 알고리즘을 사용하여 중앙 집중식 환경과 연합 학습 환경에서의 랜덤 포레스트 모델을 정확도, ROC-AUC 지표를 사용해 오픈 데이터의 이진 분류 문제에 대한 성능을 평가하고 비교했다.

그 결과, FedRFBagging 알고리즘은 각 클라이언트의 로컬 데이터 특성에 기반하여 로컬 랜덤 포레스트 모델의 트리를 동적으로 조정하고, 서버와 클라이언트에서 성능이 높은 트리만을 선택해 서버로 전송함으로써 통신 비용을 줄여 클라이언트와 서버 간의 통신 오버헤드 문제를 효과적으로 해결했다. 이는 데이터셋의 크기와 클라이언트 수가 증가하는 환경에서도 관리 가능한 수준에서 유지하여 대규모 분산 연합 학습 환경에서 효율적으로 운영될 수 있음을 시사한다.

또한, 다양한 데이터 분포에서도 중앙 집중화된 랜덤 포레스트 모델과 유사하거나 더 나은 성능을 보였다. 클라이언트 측에서 추가 트리를 학습하는 단계를 통해 로컬 데이터에 맞춘 최적화를 수행함으로 다양한 클라이언트 데이터 조건에 적응하여 모델의 안정성과 높은 예측 성능을 유지했다. 이는 FedOps와 같은 실제 연합 학습 환경[26-27]에서 실용적으로 적용 가능함을 의미하며, 신약 개발과 같은 랜덤 포레스트 모델을 사용하는 분야[7-11]에서 기존의 모델을 연합 학습 환경으로 전환할 때 효과적으로 활용할 수 있는 알고리즘을 시사한다.

다만, 본 연구의 한계점은 서버 측 글로벌 모델이 테스트 데이터셋을 가지고 있는 상황을 기준으로 진행되었으며 실제 연합 학습 환경에서는 글로벌 모델의 성능을 평가할 때 클라이언트 측의 테스트 데이터를 사용해야 하거나 글로벌 모델을 위한 데이터셋을 준비해야 할 필요가 있다. 또한, 본 연구에서는 통신 비용 감소와 모델 성능 향상에 초점을 맞추었으나, 실시간 데이터 업데이트나 네트워크 지연 등의 동적 환경에서의 평가 및 분석이 부족하다. 이러한 실시간 요소는 연합 학습 환경에서 중요한 변수로 작용할 수 있으며, 이를 고려한 추가 연구가 필요하다. 마지막으로, 본 연구에서 사용된 데이터셋은 특정 분야에 한정된 공개 데이터셋으로, 다양한 도메인에서의 일반화 가능성을 확보하기 위해 보다 다양한 데이터셋과 실제 산업 데이터를 활용한 추가적인 검증이 필요하다.

결론적으로, 본 연구는 기존의 중앙 집중화 기반 환경에서 연합 학습 환경으로 랜덤 포레스트 모델을 전환하는 데 있어 중요한 기여를 하며 향후 연구에서는 다양한 도메인과 데이터셋을 대상으로 한 추가적인 검증과 함께, 실시간 데이터 업데이트와 네트워크 지연을 고려한 동적 환경에서의 연구를 진행하여 FedRFBagging 알고리즘의 실용성을 더욱 강화할 것이다.

참 고 문 헌

[1] S. Shen, T. Zhu, D. Wu, W. Wang, W. Zhou, “From distributed machine learning to federated

learning: In the view of data privacy and security", Concurrency and Computation: Practice and Experience, Vol.34, pp.27-38, 2020.

[2] F. Sattler, S. Wiedemann, K. Müller, W. Samek, "Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data", IEEE Transactions on Neural Networks and Learning Systems, Vol.31, No.2, pp.3400-3413, 2019.

[3] J. Ma, S. Naas, S. Sigg, X. Lyu, "Privacy-preserving federated learning based on multi-key homomorphic encryption", International Journal of Intelligent Systems, Vol.37, No.3, pp.5880- 5901, 2022.

[4] E. Şahin, "Assessing the predictive capability of ensemble tree methods for landslide susceptibility mapping using XGBoost, gradient boosting machine, and random forest", SN Applied Sciences, Vol.2, pp.1-17, 2020.

[5] M. A. Afrianto, M. Wasesa, "The impact of tree-based machine learning models, length of training data, and quarantine search query on tourist arrival prediction's accuracy under COVID-19 in Indonesia", Current Issues in Tourism, Vol.25, pp.3854-3870, 2022.

[6] S. Premanand, S. Narayanan, "A Tree Based Machine Learning Approach for PTB Diagnostic Dataset", Journal of Physics: Conference Series, Vol.2115, 2021.

[7] S. Saju, S.Vasantha Swami Nathan, G. Kavitha, A. R. Mahendran, A. Amudha, "Random Forest in Closed-Loop Control of Anesthesia", Journal on Electronic and Automation Engineering, pp.107-112, 2023.

[8] Yiran Zhao, Houbao xu, "Prediction of Anti-Breast Cancer Drugs Activity Based on Bayesian Optimization Random Forest", 42nd Chinese Control Conference (CCC), pp.3471-3475, 2023.

[9] Chrobak, D., Kołodzieczak, M., Kozlovska, P., Krzemińska, A., & Miller, T., "Leveraging random forest techniques for enhanced microbiological analysis: a machine learning approach to investigating microbial communities and their interactions", Scientific Collection InterConf, pp. 386-398, 2023.

[10] Raposo, L.M., Rosa, P.T.C.R., Nobre, F.F, "Random Forest Algorithm for Prediction of HIV Drug Resistance", Pattern Recognition Techniques Applied to Biomedical Problems, 2020.

[11] S. Mehta, S. Sharma, S. Anand, S. Sharma, A. Goel, "Random Forest Algorithm for Enhanced Prediction of Drug Target Interactions," International Journal of Innovative Technology and Exploring Engineering, Vol.9, No.4, pp.208-212, 2020.

[12] K. V. Sarma, S. Harmon, T. Sanford, H. Roth, Z. Xu, J. Tetreault, D. Xu, M. G. Flores, A. Raman, R. Kulkarni, B. Wood, P. Choyke, A. Priester, L. Marks, S. Raman, D. Enzmann, B. Turkbey, W. Speier, C. Arnold, "Federated learning improves site performance in multicenter deep learning without data sharing", Journal of the American Medical Informatics Association, Vol.28, No.6, pp.1259-1264, 2021.

[13] Micah J. Sheller, Brandon Edwards, G. A. Reina, Jason Martin, Sarthak Pati, Aikaterini Kotrotsou, Mikhail Milchenko, Weilin Xu, D. Marcus, R. Colen, S. Bakas, "Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data", Scientific Reports, Vol.10, 2020.

[14] O. A. Wahab, A. Mourad, H. Otrok, T. Taleb, "Federated Machine Learning: Survey, Multi-Level Classification, Desirable Criteria and Future

Directions in Communication and Networking Systems", IEEE Communications Surveys & Tutorials, Vol.23, No.3, pp.1342-1397, 2021.

[15] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. Bhagoji, K. Bonawitz, Z. B. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, S. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konecný, A. Korolova, F. Koushanfar, O. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, S. Zhao, "Advances and Open Problems in Federated Learning", Foundations and Trends in Machine Learning, Vol.14, No.1-2, pp.1-210, 2021.

[16] Ma, C., Qiu, X., Beutel, D. J., & Lane, N, "Gradient-less Federated Gradient Boosting Tree with Learnable Learning Rates", Journal of Federated Learning, Vol.3, No.2, pp.27-38, 2023

[17] M. Gençtürk, A. A. Sinaci, N. Cicekli, "BOFRF: A Novel Boosting-Based Federated Random Forest Algorithm on Horizontally Partitioned Data", IEEE Access, Vol.10, pp.89835-89851, 2022.

[18] H. Yao, J. Wang, P. Dai, L. Bo, Y. Chen, "An Efficient and Robust System for Vertically Federated Random Forest", arXiv preprint arXiv:2201.10761, 2022.

[19] Y. Wang, H. Wu, D. Nettleton, "Stability of Random Forests and Coverage of Random-Forest Prediction Intervals," arXiv preprint arXiv:2310. 18814, 2023.

[20] Y. Wu and B. Wang, "A Framework Using Absolute Compression Hard-Threshold for Improving The Robustness of Federated Learning Model", 6th International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp. 1106-1111, 2023.

[21] Q. Li, C. Xie, X. Xu, X. Liu, C. Zhang, B. Li, B. He, D. Song, "Effective and Efficient Federated Tree Learning on Hybrid Data," in Proceedings of the International Conference on Learning Representations (ICLR), 2024.

[22] Q. Li, W. Zhaomin, Y. Cai, Y. Han, C. Yang, T. Fu, B. He, "FedTree: A Federated Learning System For Trees," in Proceedings of Machine Learning and Systems (MLSys), Vol.5, 2023.

[23] Kwatra, S., Varshney, A.K., Torra, V., "Integrally Private Model Selection for Support Vector Machine", Computer Security. ESORICS 2023 International Workshops, vol.14398, pp.249-259, 2024.

[24] M. Mohammadi, A. A. Atashin, D. A. Tamburri, "From ℓ1 Subgradient to Projection: A Compact Neural Network for ℓ1 -Regularized Logistic Regression," Neurocomputing, Vol.526, pp.30-38, 2023.

[25] I. J. Mouri, M. Ridowan, M. A. Adnan, "Data Poisoning Attacks and Mitigation Strategies on Federated Support Vector Machines," SN Computer Science, Vol.5, Article No.241, 2024.

[26] J. Moon, S. Yang and K. Lee, "FedOps: A Platform of Federated Learning Operations With Heterogeneity Management", IEEE Access, vol. 12, pp. 4301-4314, 2024.

[27] S. Yang, J. Moon, J. Kim, K. Lee and K. Lee, "FLScalize: Federated Learning Lifecycle Management Platform", IEEE Access, vol. 11, pp. 47212-47222, 2023.

저 자 소 개

송 인 서(InSeo Song)

・2020년~현재: 가천대학교 컴퓨터공학과(학사과정)

<관심분야> 인공지능, 연합학습, IoT, 디지털 트윈

이 강 윤(KangYoon Lee)

・1986년: 연세대학교 전자공학과 (공학사)

・1996년 : 연세대학교 전자계산학과 (공학석사)

・2010년 : 숭실대학교 IT정책경영(공학박사)

・2016년~현재 : 가천대학교 컴퓨터공학과 교수

<관심분야> 인공지능, IoT, 빅데이터 활용, 솔루션, 디지털 트윈