ISSN 2383-6318(Print) / ISSN 2383-6326(Online)
KIISE Transactions on Computing Practices, Vol. 30, No. 12, pp. 633-647, 2024. 12
https://doi.org/10.5626/KTCP.2024.30.12.633

신용카드 이상거래 탐지를 위한 기계학습 방법의 성능 비교

(Comparative Study of Machine Learning Methods for Credit Card Fraud Detection)

김 미 숙 †　　황 규 백 ††
(Misuk Kim)　(Kyu-Baek Hwang)

요 약 신용카드 이상거래 탐지는 전형적인 분류 문제로 다양한 기계학습 방법들이 활용되고 있다. 본 논문에서는 공개되어 있는 실제 신용카드 거래 데이터인 Credit Card Fraud Detection 데이터와 IEEE-CIS Fraud Detection 데이터에 대해 9개의 기계학습 방법을 적용하고 그 결과를 분석했다. 실험 결과, extreme gradient boosting(XGBoost)과 light gradient boosting machine(LGBM)이 두 데이터 모두에 대해 가장 우수한 성능을 기록했다. 특히 본 논문의 Credit Card Fraud Detection 데이터에 대한 실험 결과는 기존 연구의 결과보다 우수했다. 이상거래 예측시간 측면에서는 LGBM이 XGBoost보다 30배 이상 빠른 속도를 기록했다. 본 논문에서는 이러한 결과에 기반해 gradient boosting 기반의 방법, 특히 LGBM이 신용카드 이상거래 탐지에 적합하다고 제안한다.

키워드: 분류, 신용카드 이상거래 탐지, gradient boosting, extreme gradient boosting, light gradient boosting machine

Abstract Credit card fraud detection is a typical classification problem. Previous studies have applied various machine learning methods to solve this problem. In this paper, we applied nine machine learning methods to two publicly available real credit card transaction datasets, i.e., the credit card fraud detection dataset and the IEEE-CIS fraud detection dataset. Our experimental results showed that extreme gradient boosting (XGBoost) and light gradient boosting machines (LGBMs) had the highest classification accuracy on both datasets. We also achieved better results than previous state-of-the-art results on the credit card fraud detection dataset. In terms of prediction time, the LGBM was more that 30 times faster than XGBoost. These results suggest that gradient boosting based methods, especially LGBM, are suitable for credit card fraud detection.

Keywords: classification, credit card fraud detection, gradient boosting, extreme gradient boosting, light gradient boosting machines

† 비 회 원 : 숭실대학교 대학원 컴퓨터학과 학생
misukcheol@gmail.com
†† 종신회원 : 숭실대학교 컴퓨터학부 교수(Soongsil Univ.)
kbhwang@ssu.ac.kr
(Corresponding author임)

논문접수 : 2024년 4월 22일
(Received 22 April 2024)
논문수정 : 2024년 8월 14일
(Revised 14 August 2024)
심사완료 : 2024년 8월 29일
(Accepted 29 August 2024)

Copyright©2024 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.
정보과학회 컴퓨팅의 실제 논문지 제30권 제12호(2024. 12)

1. 서 론

신용카드 이상거래 탐지(fraud detection) 문제는 기존의 정상거래 및 이상거래 데이터로부터 학습된 모델을 이용해 새로운 거래의 이상 여부를 예측하는 것으로, 전형적인 분류 문제에 해당한다. 신용카드 이상거래 탐지에는 거래 금액 및 시각을 비롯한 다양한 정보가 특징(features)으로 활용된다. 이러한 정보들 중 상당수는 민감한 개인정보와 관련되어 있기 때문에 이상거래 탐지 모델 구축을 위한 데이터를 공개하는 것은 용이하지 않다. 따라서 많은 논문들이 연구에 활용한 데이터를 공개하지 않거나[1-3], 실제 데이터가 아닌 합성(synthesized) 데이터를 사용하고 있다[4,5]. 그럼에도 불구하고 두 가지 공개된 데이터인 Credit Card Fraud Detection(CCFD) 데이터(https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud)와 IEEE-CIS Fraud Detection (IEEE-CIS FD) 데이터(https://www.kaggle.com/c/ieee-fraud-detection)가 있으며, 이 두 데이터의 경우 개인정보 유출 방지를 위해 특징의 의미는 대부분 공개되지 않고 있다. CCFD 및 IEEE-CIS FD 데이터를 활용한 다양한 기계학습 기반의 신용카드 이상거래 탐지 연구들이 있다[6-15]. 선행 연구에서는 기계학습 방법들 중 심층신경망(deep neural network, DNN)과 gradient boosting 기법들이 전반적으로 우수한 분류 정확도를 보였다. 다만 연구마다 적용 및 비교한 기계학습 방법, 정확도 평가지표, 실험 데이터, 테스트 성능 추정 방법 등이 달라 실제 현장에서 신용카드 이상거래 탐지를 위해 어떤 기계학습 방법을 적용해야 할지 명확하게 판단하는 것은 어렵다.

본 논문은 기존 연구들의 한계점을 보완하여 신용카드 이상거래 탐지문제에 가장 적합한 기계학습 방법을 제안하는 것을 목표로 한다. 이를 위해 CCFD 데이터 혹은 IEEE-CIS FD 데이터 중 하나만을 사용한 기존 연구들과 달리 두 데이터 모두를 사용했다. 또한 기계학습 분야에서 널리 활용되는 9가지 방법들을 적용 및 비교하여 두 데이터 모두에 우수한 성능을 보이는 공통적인 방법이 존재하는지 확인했다. 정확도 평가지표로는 불균형 데이터에 대한 분류 성능 측정에 효과적이라고 알려져 있는 area under the precision-recall curve (AUPRC)[16]와 일반적으로 널리 활용되고 있는 area under the receiver operating characteristics curve (AUROC)를 모두 사용했다. 또한 기계학습 방법의 학습 및 예측에 걸리는 시간도 측정 및 비교하여 실제 환경에서의 실행 가능성을 확인해 보았다.

본 논문의 구성은 다음과 같다. 1절의 서론에 이어, 2절에서는 관련 연구에 대해 설명하고, 3절에서는 실험 절차 및 결과를 기술한다. 마지막으로 4절에서 결론 및 향후 연구방향을 제시한다.

2. 관련 연구

2.1 신용카드 이상거래 탐지의 주요 이슈 및 적용 방법

신용카드 이상거래 탐지에 대한 연구동향을 분석한 [17,18]에 따르면 신용카드 이상거래 탐지 문제의 주요 이슈는 데이터 불균형(class imbalance), 컨셉 변화(concept drift), 데이터 스트림(data stream), 시간복잡도(time complexity), 데이터 용량[scalability], 공개된 데이터의 부족[data availability], 특징 선택(feature

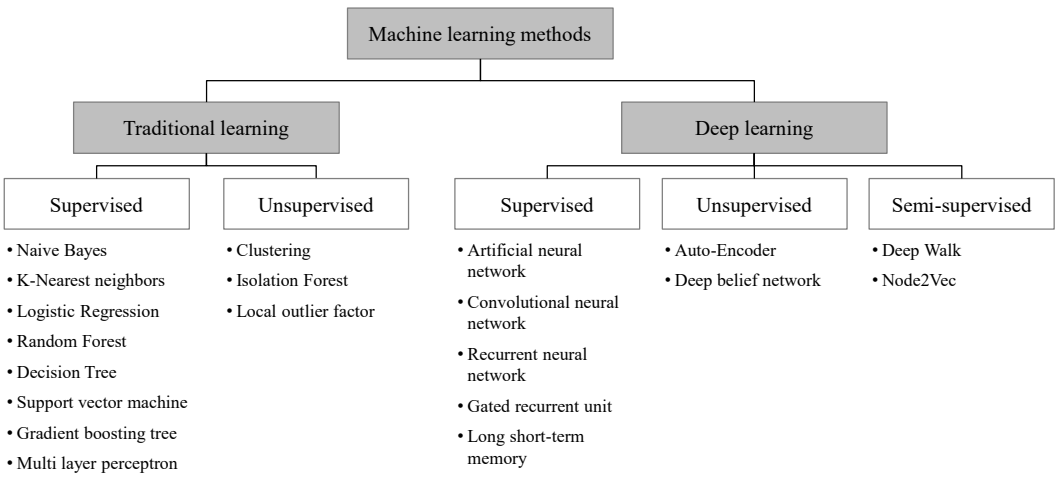


그림 1 신용카드 이상거래 탐지 문제에 활용되는 기계학습 방법
Fig. 1 Machine learning methods for credit card fraud detection

selection), 특징 구축(feature construction) 등이 있다.

신용카드 거래 데이터는 대다수가 정상이고 극히 일부만 이상거래인 불균형 데이터(imbalanced data)이다. 대표적인 공개데이터인 CCFD 데이터와 IEEE-CIS FD 데이터의 이상거래비율(이상거래수/전체거래수)은 각각 0.173%와 3.499%이다. 많은 연구에서 데이터 불균형으로 초래되는 문제를 완화하기 위해 과소표집(undersampling)이나 과대표집(oversampling)을 이용한 데이터 균형화(data balancing) 방법을 적용했다[6,8,9,11,12]. 한편 데이터 균형화는 분류 성능에 미치는 효과가 미미하거나 오히려 성능을 하락시킨다는 연구 결과도 있었다[19-21].

다음으로 컨셉 변화 문제가 있다[22]. 과거 데이터에서 학습된 모델의 성능이 새로운 데이터에 대해서 하락할 수 있다는 것이 컨셉 변화 문제이다. 실제 기업에서는 컨셉 변화 문제에 대응하기 위해 주기적으로 모델을 재학습하고 갱신해야 한다. 이런 측면에서 기계학습 방법을 평가할 때 주기적인 학습을 위한 시간복잡도를 고려할 필요가 있다.

한편, 신용카드 이상거래 탐지 문제에는 다양한 기계학습 방법이 활용되고 있다. 그림 1은 [18]에서 열거하고 있는 신용카드 이상거래 탐지에 활용된 기계학습 방법을 정리한 것이다. 그림에서 보이듯이 고전적인 기계학습과 딥러닝 방법 모두 이 문제에 적용되어 왔다. 2.2절과 2.3절에서는 이러한 방법들 중 본 논문에서 다루고 있는 CCFD 데이터와 IEEE-CIS FD 데이터에 적용되어 온 기계학습 방법들에 대해 상세하게 정리했다.

2.2 CCFD 데이터 관련 연구

여러 연구에서 CCFD 데이터에 대한 실험에 기반하여 적절한 신용카드 이상거래 탐지 방법을 제안하고 있다. [6]은 12개의 기계학습 방법을 이용하여, 하나의 방법을 단독으로 사용하는 경우, 동일한 방법 여러 개를 boosting하는 경우, 여러 방법의 결과를 다수결 방식으로 결합하는 경우의 매튜상관계수(Matthews' correlation coefficient) 값을 비교했다. 결과적으로 신경망과 나이브베이즈분류기(naive Bayes classifier)를 다수결 방식으로 결합한 경우의 매튜상관계수가 0.823으로 가장 높았다. [7]은 오토인코더(autoencoder)를 이용하여 데이터의 중첩 영역과 비중첩 영역을 구분하고, 중첩 영역에 있는 데이터만을 이용하여 인공신경망을 학습했다. 저자들의 제안 방법은 F1점수와 AUPRC 측면에서 random forest(RF)보다 더 우수한 성능을 보였다. [8]은 과대표집 방식 중 하나인 synthetic minority oversampling technique을 적용한 뒤 extreme gradient boosting (XGBoost), 로지스틱회귀(logistic regression), 나이브베이즈분류기, 선형판별분석(linear discriminant analysis, LDA)의 AUROC를 비교하였으며, 이 중 XGBoost의 AUROC가 0.99로 가장 우수했다. [9]는 이진클래스(binary class) 분류 방법과 단일클래스(one class) 분류 방법의 성능을 비교하여, 이진클래스 방법들이 평균적으로 더 우수한 결과를 보임을 밝혔다. 구체적으로 AUROC는 로지스틱회귀가 0.9794로, AUPRC는 categorical boosting(CatBoost)이 0.8567로 가장 우수했다. [10]은 신용카드 회사가 처리할 수 있는 이상거래 숫자를 고려하여 예측 이상거래 비율을 설정하고, 그 값에 따라 임계값이 조정되는 동적인 신경망[fraud detection neural network, FDNN]을 제안했다. FDNN의 F1점수는 0.83으로 선형회귀, RF, XGBoost, adaptive boosting (AdaBoost), light gradient boosting machine(LGBM)보다 우수했다. [11]은 오토인코더를 이용하여 데이터의 표현을 학습한 뒤 분류기로 인공신경망, convolutional neural network(CNN), recurrent neural network(RNN)와 support vector machine(SVM)을 적용하였다. 오토인코더와 분류기를 함께 적용한 경우와 분류기만 적용한 경우의 성능을 비교한 결과, 오토인코더와 분류기를 함께 적용한 경우의 재현율(recall), 정밀도(precision), F1점수, AUROC, AUPRC가 모두 향상되었다. AUROC는 오토인코더와 CNN 조합이 0.9217, AUPRC는 오토인코더와 인공신경망 조합이 0.7746으로 가장 높았다.

2.3 IEEE-CIS FD 데이터 관련 연구

IEEE-CIS FD 데이터를 이용한 연구는 주로 데이터 전처리 또는 기계학습 방법의 2가지 측면에서 수행되었다. [12]는 IEEE-CIS FD 데이터에 대한 분류 성능 향상을 위한 DNN 구조를 제안하였으며, accuracy 0.95, F1점수 0.95, 정밀도 0.95, 재현율 0.96, AUROC 0.99의 성능을 확인했다. [13]은 8개 기계학습 방법 중 가장 우수한 성능을 보이는 방법을 찾고, 그 방법을 다른 방법과 조합하여 추가적인 성능 개선을 보이는지 확인했다. AUROC 기준으로 AdaBoost의 성능이 0.71로 가장 우수했다. 이후 AdaBoost를 기본 방법으로 하고 나머지 방법들을 접목한 결과, 가장 우수한 성능을 보인 조합은 AdaBoost와 LGBM으로 AUROC가 0.82였다. [14]는 IEEE-CIS FD 데이터에 SVM, RF, XGBoost, LGBM을 적용하였으며 LGBM의 AUROC가 0.96, accuracy가 0.98로 다른 방법보다 우수했다. [15]에서는 사용자 개념을 도입하여 IEEE-CIS FD 데이터를 전처리했다. 학습데이터와 테스트데이터 모두에 존재하는 사용자는 기존 고객으로, 그 외는 신규 고객으로 정의했다. 기존 고객은 CatBoost를, 신규 고객은 DNN을 적용하여 이상거래 여부를 판단했다. CatBoost의 AUROC는 0.97이었고 DNN의 AUROC는 0.84였다.

2.4 관련 연구에 대한 고찰

CCFD 데이터와 IEEE-CIS FD 데이터를 이용한 기

존 연구에서는 대체적으로 gradient boosting 기반 방법과 DNN 기반 방법이 다른 기계학습 방법에 비해 우수한 성능을 보였다. 9개의 관련 연구 중 3개의 연구[8, 9,14]에서 gradient boosting 기반 방법이 가장 우수한 성능을 보였고, 2개의 연구[7,10]에서 DNN 기반 방법이 가장 우수한 성능을 보였다. 하지만 연구마다 상이한 정확도 평가지표를 활용하였고, 학습데이터 및 테스트데이터의 비율, 실험의 반복횟수 등 실험 방식의 차이가 존재한다. [12,15]는 각각 DNN과 CatBoost를 실험에 활용하였으나, 다른 방법과의 성능 비교는 하지 않았다. [11]도 오토인코더가 데이터의 적절한 표현을 학습하는 효과가 있음을 입증하였으나, 다른 기계학습 방법과의 성능 비교는 수행하지 않았다. 또한, 각 연구마다 CCFD 혹은 IEEE-CIS FD 데이터 중 하나의 데이터에만 실험을 수행했다는 한계가 있다. 본 논문에서는 두 데이터 모두에 다양한 기계학습 방법을 적용하여 신용카드 이상거래 탐지문제에 적합한 기계학습 방법을 제안하고자 한다. 또한 데이터 불균형, 컨셉 변화 및 시간복잡도와 같은 신용카드 이상거래 탐지 분야의 주요 이슈들과 관련하여, 선택된 기계학습 방법이 실제 기업에서 활용되기에 적합한지에 대해서도 논의한다.

3. 실험 절차와 결과

3.1 정확도 평가지표

본 논문에서는 분류 정확도 평가를 위해 AUROC와 AUPRC를 사용했다. AUROC는 (재현율과 동일한) 민감도(sensitivity)와 1-특이도(specificity) 값을 각각 y축과 x축으로 하는 ROC 곡선(receiver operating characteristics curve)의 아래면적값을 의미한다. 반면, AUPRC는 y축은 정밀도, x축은 재현율로 하는 PR 곡선(precision-recall curve)의 아래면적값을 의미한다. 분류 모델은 어떤 거래가 특정 클래스에 속할 확률값을 예측한 뒤, 사전에 정의한 임계값을 기준으로 0 혹은 1 클래스로 분류한다. 민감도, 특이도, 정밀도, 재현율은 임계값을 어떤 값으로 정하는지에 따라 값이 달라지는 지표이다. 반면 ROC 곡선과 PR 곡선은 임계값을 0부터 1까지 변화시키면서 y축과 x축의 성능지표를 측정하여 그려진다. 따라서, AUROC와 AUPRC는 특정 임계값에 국한되지 않고 전반적으로 좋은 성능을 보이는 방법을 찾을 수 있다. 특히 AUPRC는 불균형 데이터의 분류 성능 측정에 효과적이라고 알려진 성능지표이다[16]. AUROC와 AUPRC의 차이점은 각각 1-특이도와 정밀도를 사용한다는 점이다. AUROC의 1-특이도는 실제 다수클래스 거래 중 소수클래스로 오분류한 거래의 비율을 측정하는 지표이다. 즉, 다수클래스를 정확히 분류하는 것에 초점을 맞춘 지표이다. 반면, AUPRC에 사용되는 정밀도는 분류 모델이 소수클래스로 예측한 거래가 실제 소수클래스와 얼마나 일치하는지를 측정하는 지표이다. 정밀도는 소수클래스를 정확하게 탐지하는 확률을 계산하게 된다. 따라서, 신용카드 이상거래 탐지와 같이 소수클래스의 판별이 중요한 문제에서는 AUPRC가 적합한 지표라고 할 수 있다. AUROC와 AUPRC는 Python scikit-learn package(version 0.24.1)(https://scikit-learn.org)을 이용해 계산했다.

3.2 기계학습 방법

본 논문에서는 기존 연구에서 우수한 성능을 보였던 방법을 포함한 9개의 선형 및 비선형 기계학습 방법을 적용 및 비교했다. 선형 방법으로는 LDA와 3개의 정규화된(regularized) 로지스틱회귀 방법인 lasso, ridge, elastic net를, 비선형 방법으로는 XGBoost, LGBM,

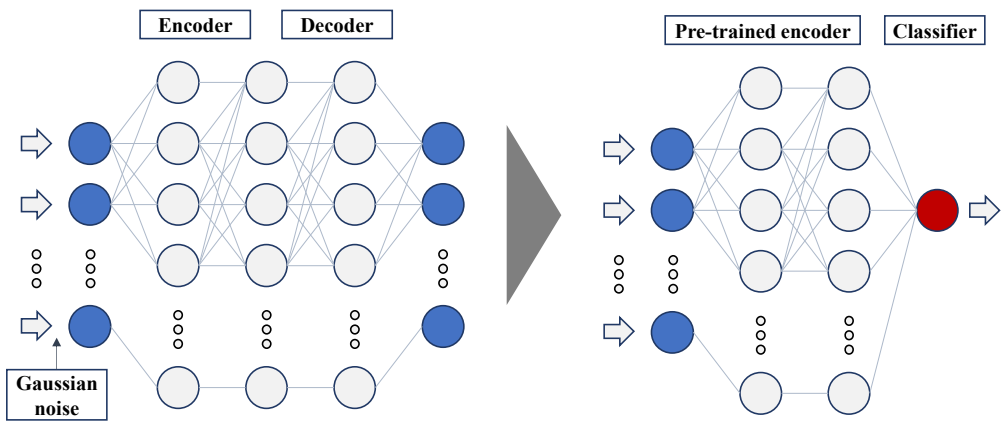


그림 2 오토인코더로 사전학습된 신경망 기반 분류기 구조도
Fig. 2 Structure of the classifier based on autoencoders

RF, SVM 및 오토인코더로 사전학습된 신경망 기반 분류기를 적용했다.

선형 방법 중 LDA[23]는 데이터를 1차원 공간에 사영(projection)한 후, 클래스를 잘 구분하는 결정 경계를 찾는 방법이다. Lasso[24], ridge[25], elastic net[26]는 각각 L1정규화, L2정규화, L1과 L2정규화를 혼합 적용한 로지스틱회귀 방법이다.

비선형 방법 중 XGBoost[27]와 LGBM[28]은 gradient boosting 기반의 기계학습 방법으로 의사결정나무(decision tree)의 앙상블 모델이다. RF[29]는 bootstrap 기반의 의사결정나무 앙상블 모델이다. SVM[30]은 클래스 사이의 간격인 마진(margin)을 최대화하는 경계를 찾는 분류 방법이다. SVM은 커널트릭(kernel trick)이라는 개념을 통해 고차원 영역에서의 선형 분리를 수행한다. 본 연구에서는 SVM에 RBF(radial basis function) 커널을 적용했다. 오토인코더[31]는 인코더(encoder)와 디코더(decoder)로 이루어진 신경망 기반의 비지도기계학습(unsupervised learning) 모델이며, 입력데이터를 암호화하고 데이터의 특징을 추출하여 다시 본래의 입력 형태로 복호화한다. 본 논문에서는 사전학습된 오토인코더의 인코더에 sigmoid 함수로 구성된 한 개의 출력층(output layer)을 분류기로 추가하여 최종 모델을 만들었다(그림 2). 입력층(input layer)과 출력층 사이에 총 3개의 은닉층(hidden layer)이 존재한다. 학습데이터에는 평균 0, 표준편차 1인 가우스 잡음(gaussian noise)을 추가했다. 은닉층의 은닉노드(hidden node) 개수, dropout rate, 활성함수(activation function), optimizer, 학습률(learning rate), epoch 설정에 대해서는 3.3절에 기술한다.

XGBoost는 XGBoost Python package(version 2.0.2)(https://xgboost.readthedocs.io)를, LGBM은 LGBM Pythonpackage(version 3.3.2)(https://lightgbm.readthedocs.io)를 이용했다. 오토인코더 기반의 분류기는 Python의 Keras 라이브러리(version 2.70)(https://keras.io)를 이용하여 구현했다. 다른 기계학습 방법은 Python의 scikit-learn package(version 0.24.1)를 활용했다.

3.3 실험 절차

3.3.1 신용카드 이상거래 탐지 데이터

본 연구에서는 CCFD 데이터와 IEEE-CIS FD 데이터를 실험에 활용했다. CCFD 데이터는 Worldline사와 브뤼셀자유대학교의 기계학습 그룹에 의해 연구목적으로 사용할 수 있도록 공개되어 있다. 이 데이터는 이틀에 걸친 실제 신용카드 거래 데이터로, 전체 284,807 거래 중 비정상거래가 492건(0.173%)인 불균형 데이터이다. 총 30개의 특징으로 구성되어 있는데, 별도의 설명을 제공하지 않는 28개 특징은 원 자료에 주성분 분석을 적용하여 변환된 것이다. 거래 시각과 금액 특징은 주성분 분석에 의해 변환되어 있지 않으며, 시각은 최초 거래로부터의 초 단위 경과 시간을 의미한다.

IEEE-CIS FD 데이터는 IEEE Computational Intelligence Society가 주최한 신용카드 이상거래 탐지 모델 구축 대회에 Vesta사가 제공한 것이다. 이 데이터는 전체 590,540거래와 433개의 특징으로 이루어진 대용량 데이터로, 비정상거래는 전체 거래의 3.499%인 20,663개이다. 데이터는 크게 transaction 데이터와 identification 데이터로 구성되어 있으며, 두 데이터는 TransactionID라는 특징으로 병합될 수 있다.

실험에 활용한 CCFD 데이터는 https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud에서 2020년 9월 13일에 내려 받았으며, IEEE-CIS FD 데이터는 https://www.kaggle.com/c/ieee-fraud-detection에서 2022년 3월 11일에 내려 받았다.

3.3.2 데이터 전처리

기계학습 방법의 테스트 정확도 추정 방식으로는, 데이터가 불균형 데이터인 점을 고려해 층화5겹교차검증(stratified 5-fold cross validation)을 이용했다. CCFD 데이터의 경우, 교차검증의 각 반복에서 학습데이터는 Python scikit-learn의 standardscaler 함수를 이용해 평균 0, 분산 1로 표준화했고, 학습데이터에서 계산된 평균과 분산으로 테스트데이터를 표준화했다.

IEEE-CIS FD 데이터는 층화5겹교차검증 수행 전, 별도의 데이터 전처리 과정을 거쳤다. 전처리는 IEEE-CIS FD 데이터를 이용한 경진대회의 1위팀이 제안한 방식(https://www.kaggle.com/code/cdeotte/xgb-fraud-with-magic-0-9600)을 적용했다. IEEE-CIS FD 데이터의 transaction 데이터와 identification 데이터는 표 1과 같이 구성되어 있다.

우선 transaction 데이터의 V특징(V1-V339) 중 결측거래의 수가 동일한 V특징들을 15개의 집합으로 구성한다({V1-V11}, {V12-V34}, {V35-V52}, {V53-V74}, ..., {V322-V339}). 집합 내에서 서로 상관도가 높은(0.75이상) V특징들을 부분집합으로 묶는다. 예를 들어, V1-V11의 특징은 {{V1}, {V2, V3}, {V4, V5}, {V6, V7}, {V8, V9}, {V10, V11}}의 부분집합으로 구분된다. 각 부분집합에 속한 특징 중 유일한 값을 가장 많이 보유한 특징 1개를 대표 특징으로 선정한다(V1, V3, V4, V6, V8, V11). 이런 방식으로 339개의 V특징 중 219개를 제거하고 120개의 V특징을 사용했다.

TransactionDT는 데이터가 시작된 2017년 12월 1일부터 해당 거래가 발생한 시점까지의 초 단위 시간차이 값을 의미한다. TransactionDT를 (24X60X60)으로 나누어 날짜로 변환한 Transaction_Day 특징을 만든다.

표 1 IEEE-CIS FD 데이터 구성
Table 1 Compositions of the IEEE-CIS FD dataset

Subdatasets	Features	Description	Type	# of Features	Example values
Key	TransactionID	Transaction ID	Integer	1	3663549
Transaction	ProductCD	Product code	Categorical	1	C, H, R, S, W
	TransactionDT	Time difference from a give reference datetime	Integer	1	18403224
	TransactionAmt	Transaction amount in USD	Numerical	1	31.95
	card1-card6	Card information	Categorical	6	10409 111 150 visa 226 debit
	addr1-addr2	Address	Categorical	2	170 87
	P_email domain	Purchaser email domain	Categorical	1	gmail.com
	R_email domain	Recipient email domain	Categorical	1	hotmail.com
	dist1-dist2	Country distance	Numerical	2	1 37
	C1-C14	Anonymous features	Numerical	14	6 6 0 ...
	D1-D15	Time difference	Numerical	15	419 27 398 ...
	M1-M9	Anonymous features	Categorical	9	T, F T, F T, F T, F M0, M1, M2 ...
	V1-V339	Anonymous features	Numerical	339	1 1 0 47.95 ...
Identification	id1-id38	Anonymous features	Categorical or numerical	38	-45 280290 0 0 ...
	DeviceType	Device type	Categorical	1	mobile, desktop
	DeviceInfo	Device information	Categorical	1	MYA-L13 iOS Device

D특징(D1-D15)은 시간과 관계된 특징이며, Transaction_Day에서 D특징을 빼는 방식으로 D1n-D15n을 생성했다. D1-D15 대신 D1n-D15n을 활용하고, TransactionDT와 Transaction_Day는 제외했다.

범주형(categorical) 특징들은 n개의 범주를 0부터 n-1까지의 연속적 수치값으로 변환했다. 또한, TransactionAmt의 소수점 값인 cents와 D1과 D15의 차이가 3이상인 경우를 구분하는 outsider15라는 신규 특징을 만들었다.

표 2 IEEE-CIS FD 데이터 전처리에 따른 73개의 신규 파생 특징
Table 2 New features (n=73) derived from preprocessing of the IEEE-CIS FD dataset

Methods	New features
Mean (36 features)	TransactionAmt_card1_mean, TransactionAmt_card1_addr1_mean, TransactionAmt_card1_addr1_P_emaildomain_mean, D9_card1_mean, D9_card1_addr1_mean, D9_card1_addr1_P_emaildomain_mean, D11_card1_mean, D11_card1_addr1_mean, D11_card1_addr1_P_emaildomain_mean, TransactionAmt_uid_mean, D4_uid_mean, D9_uid_mean, D10_uid_mean, D15_uid_mean, C1_uid_mean, C2_uid_mean, C4_uid_mean, C5_uid_mean, C6_uid_mean, C7_uid_mean, C8_uid_mean, C9_uid_mean, C10_uid_mean, C11_uid_mean, C12_uid_mean, C13_uid_mean, C14_uid_mean, M1_uid_mean, M2_uid_mean, M3_uid_mean, M4_uid_mean, M5_uid_mean, M6_uid_mean, M7_uid_mean, M8_uid_mean, M9_uid_mean
Standard deviation (15 features)	TransactionAmt_card1_std, TransactionAmt_card1_addr1_std, TransactionAmt_card1_addr1_P_emaildomain_std, D9_card1_std, D9_card1_addr1_std, D9_card1_addr1_P_emaildomain_std, D11_card1_std, D11_card1_addr1_std, D11_card1_addr1_P_emaildomain_std, TransactionAmt_uid_std, D4_uid_std, D9_uid_std, D10_uid_std, D15_uid_std, C14_uid_std
Frequency (8 features)	addr1_FE, card1_FE, card1_FE, card3_FE, card1_addr1_FE, P_emaildomain_FE, card1_addr1_P_emaildomain_FE, uid_FE
Unique count (12 features)	uid_P_emaildomain_ct, uid_dist1_ct, uid_TransactionDT_ct, uid_id_02_ct, uid_cents_ct, uid_C13_ct, uid_V314_ct, uid_V127_ct, uid_V136_ct, uid_V309_ct, uid_V307_ct, uid_V320_ct
Aggregation (2 features)	card1_addr1, card1_addr1_P_emaildomain

데이터 전처리 과정 중 가장 특별한 단계는 신용카드를 구분할 수 있는 UID라는 특징을 도입한 것이다. Vesta사의 설명에 따르면, IEEE-CIS FD 데이터는 어떤 신용카드의 거래 중 하나가 비정상으로 판별되면, 해당 신용카드로 발생된 모든 거래는 설령 최초에는 정상거래로 분류되었더라도 최종적으로는 비정상거래로 변경하여 저장되었다.

따라서, 동일한 신용카드에서 발생한 거래들을 묶을 수 있는 공통 특징을 생성하고, 공통 특징을 기반으로 신용카드를 판별할 수 있도록 기계학습 방법을 학습했다. 전체 데이터에 LGBM을 적용하여 특징의 중요도(variable importance)를 계산한 후, 중요도가 높은 특징 중 card1, addr1, D1을 연결(concatenation)하여 UID를 만들었다. 기계학습 방법이 UID에 과적합(overfitting)하는 것을 막기 위해, UID를 직접적으로 학습에 사용하지는 않았다. 대신, 신용카드를 구분할 수 있도록 중요 특징들을 이용하여 73개의 파생 특징을 생성했다. 전술한 것처럼 LGBM에 의해 계산된 특징의 중요도가 높은 특징 중, 42개의 특징을 활용하여 파생 특징을 만들었다. 42개의 중요 특징은 addr1, card1, card2, card3, P_emaildomain, dist1, id_02, cents, TransactionAmt, TransactionDT, D4, D9, D10, D15, C1, C2, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, M1, M2, M3, M4, M5, M6, M7, M8, M9, V127, V136, V309, V307, V314, V320이다.

단일 중요 특징 혹은 두개 이상의 중요 특징을 연결하여 그룹을 만들고, 그 그룹에 대한 평균, 표준편차, 빈도수(frequency), 유일값의 수(unique count), 조합된 값(aggregation)을 기준으로 파생 특징을 생성했다. 예를 들어, TransactionAmt_uid_mean과 TransactionAmt_uid_std는 동일한 UID를 가진 예제들의 TransactionAmt의 평균과 표준편차를 의미한다. 중요 특징들의 조합으로 생성된 파생 특징 중에서, 전체 데이터의 75%를 학습데이터, 25%를 테스트데이터로 나누고 LGBM을 학습시켜 테스트데이터의 AUROC가 향상되는 73개의 파생 특징만 활용했다. 73개의 파생 특징은 표 2에 정리하였다.

다음으로는 시간이 경과하여도 특징의 성능이 일관되게 유지되는지 확인하기 위하여, IEEE-CIS FD 데이터의 처음 1달의 데이터와 마지막 1달의 데이터를 각각 학습데이터와 테스트데이터로 나누고 기계학습 방법의 성능을 측정했다. 특징 1개로만 구성된 XGBoost를 학습데이터에 적용하여 학습시키고 테스트데이터의 AUROC를 측정했다. 이 테스트 AUROC 값이 낮다는 것은 해

당 특징은 미래 거래의 특징을 제대로 나타내지 못함을 의미한다. 따라서, 테스트 AUROC가 0.5미만인 17개의 특징을 제외했다.

마지막으로 결측치의 비율이 87% 이상인 7개의 D특징(D6, D7, D8, D9, D12, D13, D14)도 제외했다. Key 값인 TransactionID도 최종 특징에서 제외하였다.

결과적으로 최초 433개의 특징에서 출발하여, 219개의 V특징 제외, TransactionDT 제외, 신규 특징인 cents 및 outsider15 추가, UID를 설명할 수 있는 73개의 파생 특징 추가, 시간 경과에 따라 성능이 떨어지는 17개의 특징 제외, 결측치 비율이 높은 7개의 D특징 제외, TransactionID 제외라는 과정을 거쳐 최종적으로 263개의 특징을 선정했다.

전처리 과정을 거친 IEEE-CIS FD 데이터는 CCFD 데이터와 마찬가지로 층화5겹교차검증을 통하여 학습데이터와 테스트데이터로 분리하고 학습데이터는 평균 0, 분산 1로 테스트데이터는 학습데이터의 평균과 분산을 기준으로 표준화했다.

3.3.3 하이퍼파라미터(Hyperparameter) 최적화 및 분류 모델의 정확도 평가

기계학습 모델의 테스트 정확도를 추정하기 위해 전술한 바와 같이 층화5겹교차검증을 이용했으며 교차검증의 각 반복의 학습데이터에 대한 층화3겹교차검증을 통해 하이퍼파라미터를 최적화했다. 하이퍼파라미터 최적화에는 모든 경우의 조합을 전수 조사하는 그리드 탐색을 이용했다. 각 기계학습 방법별 하이퍼파라미터 및 탐색 범위는 표 3과 같다.

표 3 기계학습 방법별 하이퍼파라미터 및 탐색 범위
Table 3 Hyperparameters and their search ranges per machine learning method

Machine learning methods	Hyperparameters and their search ranges
LDA	N/A
Lasso	'C': {0.1, 1, 10}
Ridge	'C': {0.1, 1, 10}
Elastic net	'C': {0.1, 1, 10}
XGBoost	'n_estimators': {3000, 5000} 'max_depth': {10, 12} 'colsample_bytree': {0.4, 1} 'subsample': {0.8, 1} 'learning_rate': {0.02, 0.1}
LGBMs	'n_estimators': {3000, 5000} 'reg_alpha': {0, 0.03} 'num_leaves': {31, 100} 'min_child_samples': {15, 20} 'learning_rate': {0.02, 0.1}
RFs	'n_estimators': {100, 500, 1000, 3000, 5000}
SVMs	'C': {0.1, 1}
Autoencoder -based classifier	'hidden node': {3000, 5000, 10000} 'dropout rate': {0.05, 0.5, 0.8} 'activation function': {leakyrelu, tanh} 'optimizer': {Adam, Adadelta} 'learning_rate': {0.001, 0.0001, 0.00001}

기계학습 방법 중 LDA는 하이퍼파라미터가 없다. Lasso, ridge, elastic net과 SVM의 'C'는 학습데이터의 오분류를 허용하는 정도를 의미한다. 낮은 'C'는 의사결정표면(decision surface)를 단순하게 하고, 높은 'C'는 모든 학습데이터를 올바르게 분류하기 위하여 의사결정표면을 복잡하게 한다. XGBoost, LGBM, RF의 'n_estimators'는 앙상블 구성에 사용하는 의사결정나무의 개수를 의미한다. XGBoost의 'max_depth'는 의사결정나무의 최대 깊이, 'colsample_bytree'는 각 나무 구성에 사용되는 특징의 표본추출 비율, 'subsample'은 각 나무 학습에 이용되는 학습데이터의 표본추출 비율에 대한 하이퍼파라미터이다. 'learning_rate'는 과적합을 방지하기 위하여 각 학습단계에서 얻어진 가중치의 영향도를 축소하는 값이다. LGBM의 'reg_alpha'는 L1 정규화 제어를 위한 값, 'num_leaves'는 하나의 나무가 가질 수 있는 단말노드의 최대 개수, 'min_child_samples'는 단말노드가 가져야 하는 데이터 예제의 최소 개수를 가리키며, 'learning_rate'는 XGBoost의 'learning_rate'의 의미와 동일하다. 오토인코더로 사전학습된 신경망 기반 분류기의 경우 'hidden_node'는 은닉노드 개수를 의미하며 모든 은닉층에 대해 동일하게 설정했다. 'dropout rate'는 dropout rate를, 'activation function'은 은닉층의 활성화 함수를 가리킨다.

한편 오토인코더 및 오토인코더로 사전학습된 신경망 기반 분류기의 optimizer, 학습률, epoch 값은 그리드 탐색이 아닌 학습곡선(learning curve) 관찰 방식으로 선정하였다. 이를 위하여 하이퍼파라미터 최적화를 위한 층화3겹교차검증의 내부 학습데이터 중 10%를 검증데이터(validation data)로 사용했다. 내부 학습데이터에 대해 그리드 탐색 대상인 하이퍼파라미터 조합을 고정하고, optimizer와 학습률을 변경시키며 학습곡선을 그린다. 학습곡선을 그릴 때, x축인 epoch는 0부터 300까지 적용했으며 y축은 평균제곱오차(mean squared error)를 측정했다. 학습곡선에서 평균제곱오차가 감소하는 경향성을 보고 epoch를 정했다. 구체적으로 검증데이터의 평균제곱오차가 가장 작은 값을 보이는 경우의 optimizer, 학습률, epoch 조합을 선정했다. 이 과정을 3개 폴드(fold)에 대해 각각 수행하고 3개의 optimizer, 학습률, epoch 조합 중 다수결에 따라 최종 1개의 조합을 선정했다. 그리드 탐색 대상인 하이퍼파라미터 조합을 바꿔

표 4 오토인코더 및 오토인코더 기반 분류기의 optimizer, 학습률 및 epoch 설정값

Table 4 Selected optimizer, learning rate, and epoch values for autoencoders and autoencoder-based classifier

CCFD dataset			
Models	Optimizer	Learning rate	Epoch
Autoencoder pretraining	Adam	0.0001	200
	Adadelta	0.001	300
Autoencoder-based classifier learning	Adam	0.00001	50
IEEE-CIS FD dataset			
Models	Optimizer	Learning rate	Epoch
Autoencoder pretraining	Adam	0.0001	300
Autoencoder-based classifier learning	Adam	0.00001	200

동일한 과정을 반복했다. 최종 optimizer, 학습률, epoch 조합의 결과는 표 4와 같다. 학습의 배치(batch) 크기는 모두 2048로 설정했다.

전술한 바와 같이 그리드 탐색을 통해 층화3겹교차검증의 3개 내부 테스트데이터에 대한 평균 성능이 가장 좋은 하이퍼파라미터 값의 조합을 선정했다. 이후 해당 하이퍼파라미터 조합으로 기계학습 모델 평가를 위한 층화5겹교차검증의 학습데이터를 학습한 뒤 해당 테스트데이터에 대한 분류 성능을 계산했다. 이러한 층화5겹교차검증을 데이터의 분할을 무작위로 달리하여 10번 반복하고 그 평균 성능값을 비교의 기준으로 삼았다. 단, SVM의 경우 학습에 소요되는 시간의 제약으로 인하여 CCFD 데이터에 대해서만 실험하였다.

3.4 실험 결과

3.4.1 기계학습 방법별 성능 결과

표 5와 6은 기계학습 방법별로 층화5겹교차검증의 10회 반복을 통해 얻어진 50개의 성능지표 결과값의 평균과 표준편차를 나타내고 있다. 표에서 P-값은 가장 우수한 성능을 보이는 기계학습 방법과 해당 방법의 결과의 차이에 대한 대응표본 t 검정(paired t-test)으로 계산된 것이다.

CCFD 데이터에 대한 결과를 보면 XGBoost가 AUPRC와 AUROC 모두에서 가장 좋은 성능을 보였으며 다른 기계학습 방법과의 비교에서도 통계적으로 유의미한 차이가 있다는 것을 알 수 있다. AUPRC의 경우 선형 방법보다는 비선형 방법이 좋은 성능을 보이고 비선형 방법 중에서도 gradient boosting 계열인 XGBoost와 LGBM이 가장 우수하며 또다른 의사결정나무 기반 앙상블 방법인 RF도 크게 떨어지지 않는 성능을 보이고 있다. SVM은 비선형 방법 중 가장 낮은 성능을 보였으며, 오토인코더로 사전학습된 신경망 기반 분류기는 그보다 조금 더 우수한 성능을 보였다. 선형 방법의 AUPRC는 0.73~0.75 수준으로 XGBoost의 0.86과 비교하여 큰 차이가 난다. 한편 AUROC의 경우 모든 방법의 성능이 0.9 이상이었다. AUROC는 AUPRC의 결과와는 다르게 선형 방법의 성능이 RF, SVM 및 오토인코더로 사전학습된 신경망 기반 분류기보다 우수한 결과를 보였다.

실제 기업에서 혹은 연구를 수행하며 AUROC만을 측정한다면, AUROC 성능이 우수한 선형 방법을 선택할 가능성도 있다. 그러나 표 5에서 확인할 수 있듯이 선형 방법들의 AUPRC는 비선형 방법의 AUPRC에 비

표 5 CCFD 데이터에 대한 기계학습 방법별 성능

Table 5 Performances of nine machine learning methods on the CCFD dataset

Machine learning methods	AUPRC			AUROC		
	Mean	Standard deviation	P-value	Mean	Standard deviation	P-value
LDA	0.7315	0.0478	<2.2e-16	0.9807	0.0061	0.0030
Lasso	0.7582	0.0473	<2.2e-16	0.9805	0.0068	0.0120
Ridge	0.7594	0.0455	<2.2e-16	0.9761	0.0089	8.635e-10
Elastic net	0.7588	0.0472	<2.2e-16	0.9808	0.0067	0.0132
XGBoost	**0.8612**	0.0358	N/A	**0.9827**	0.0077	N/A
LGBMs	0.8579	0.0382	0.0046	0.9805	0.0083	1.057e-05
RFs	0.8556	0.0364	0.0145	0.9682	0.0134	0.0056
SVMs	0.8036	0.0473	<2.2e-16	0.9449	0.0228	<2.2e-16
Autoencoder-based classifier	0.8336	0.0351	6.315e-14	0.9716	0.0101	4.376e-12

표 6 IEEE-CIS FD 데이터에 대한 기계학습 방법별 성능
Table 6 Performances of eight machine learning methods on the IEEE-CIS FD dataset

Machine learning methods	AUPRC			AUROC		
	Mean	Standard deviation	P-value	Mean	Standard deviation	P-value
LDA	0.3950	0.0076	<2.2e-16	0.8373	0.0037	<2.2e-16
Lasso	0.3851	0.0095	<2.2e-16	0.8444	0.0040	<2.2e-16
Ridge	0.4304	0.0076	<2.2e-16	0.8543	0.0033	<2.2e-16
Elastic net	0.3851	0.0095	<2.2e-16	0.8444	0.0040	<2.2e-16
XGBoost	0.9228	0.0034	<2.2e-16	**0.9836**	0.0011	N/A
LGBMs	**0.9249**	0.0032	N/A	0.9801	0.0014	<2.2e-16
RFs	0.8722	0.0035	<2.2e-16	0.9726	0.0013	<2.2e-16
Autoencoder-based classifier	0.7737	0.0094	<2.2e-16	0.9494	0.0043	<2.2e-16

해 좋지 않다. 또한, [32]에서는 AUROC의 성능이 높다고 하여 AUPRC의 성능이 높은 것을 반드시 보장하지 않는다고 주장한다. 즉, 상황에 적절하지 못한 평가지표를 사용한다면 잘못된 의사결정을 할 수도 있다. 3.1절에서 언급한 바와 같이 신용카드 이상거래 탐지 문제와 같이 소수클래스의 판별이 더 중요한 문제에 적합한 지표는 AUPRC이다. 따라서 AUPRC가 우수한 비선형 방법을 선택해야 한다.

IEEE-CIS FD 데이터에 대해서는 AUPRC는 LGBM이 AUROC는 XGBoost가 가장 좋은 성능을 보였다. CCFD 데이터와 마찬가지로 두 방법의 성능은 다른 방법과 통계적으로 유의미한 차이가 있었다. AUPRC는 CCFD 데이터의 결과처럼 선형 방법보다는 비선형 방법이, 비선형 방법 중에서도 LGBM과 XGBoost가 우수한 성능을 보였다. 다만 CCFD 데이터에 대한 결과와는 달리 선형 방법의 AUPRC는 0.3~0.4 수준, 비선형 방법의 AUPRC는 0.7~0.9로 성능 차이가 컸다. 또한 AUROC의 경우에도 비선형 방법이 선형 방법보다 높은 AUROC 값을 보였다. XGBoost와 LGBM이 가장 좋은 성능을 보였고 다음으로는 RF 및 오토인코더로 사전학습된 신경망 기반 분류기의 성능이 좋았다. 선형 방법은 비선형 방법의 0.9 수준의 AUROC 대비 0.8 수준의 낮은 결과를 보였다. 두 데이터에 대한 실험 결과를 종합해 보면 데이터의 종류 및 성능지표와 관계없이 보편적으로 좋은 분류 성능을 보여주는 방법은 gradient boosting 기반의 방법임을 알 수 있다.

3.4.2 기존 연구와의 비교

본 논문과 기존 연구 [9,11]은 CCFD 데이터에 대한 기계학습 방법의 AUPRC와 AUROC를 측정하였다. 다만 [9]는 본 논문과 동일하게 층화5겹교차검증 방식을 10번 반복한 성능의 평균값이고, [11]은 층화5겹교차검증 방식을 1회만 수행한 결과이다. 표 7에 각 연구의 결과를 정리했다.

본 연구와 [9]에서 공통적으로 사용한 기계학습 방법인 XGBoost와 RF를 비교한 결과, [9]에서는 XGBoost의 AUPRC가 0.8549, AUROC가 0.9786이었고, RF의 AUPRC는 0.8069, AUROC는 0.9620이었다. 본 논문에서는 XGBoost의 AUPRC가 0.8612, AUROC가 0.9827이었고, RF의 AUPRC는 0.8556, AUROC는 0.9682로 본 연구의 결과가 더 좋았다. 특히 본 논문의 XGBoost 성능은 [9]에서 가장 우수한 성능을 보인 CatBoost의 AUPRC 0.8567과 로지스틱 회귀분석의 AUROC 0.9794보다 뛰어났다.

본 논문과 유사하게 오토인코더 기반의 분류기를 활용하여 CCFD 데이터의 AUPRC와 AUROC를 측정한 [11]의 결과와 비교해도 본 논문의 성능이 더 우수했다. 본 논문에서는 분류기로 sigmoid 함수로 구성된 한 개의 출력층을 활용하였으나, [11]에서는 인공신경망, CNN, RNN, SVM을 분류기로 활용하였다. [11]에서는 인공신경망, CNN, RNN, SVM 순서로 AUPRC는 0.7705, 0.7532, 0.7553, 0.6117의 결과를, AUROC는 0.9277, 0.9254, 0.9278, 0.8566의 결과를 보였다. 본 논문의 분류기는 AUPRC 0.8336, AUROC 0.9716로 [11]의 결과보다 우수했다.

동일한 기계학습 방법이 연구마다 성능 차이를 보이는 이유는 하이퍼파라미터 설정 때문인 것으로 보인다. 표 8은 본 논문과 [9]에서 사용한 하이퍼파라미터 값을 비교하고 있다. [9]에서는 과적합 방지를 위해 XGBoost와 RF의 하이퍼파라미터 중 'max_depth'의 숫자를 각각 1과 4로 제한하고 그 외 다른 하이퍼파라미터들은 기본 설정(default)으로 유지하였다. 반면 본 논문에서는 층화3겹교차검증을 통해 XGBoost의 'max_depth'를 폴

표 7 본 연구와 [9,11]의 기계학습 방법별 성능
Table 7 Performances of nine machine learning methods in our experiments and previous studies [9,11]

Machine learning methods	AUPRC			AUROC		
	Ours	[9]	[11]	Ours	[9]	[11]
XGBoost	**0.8612**	0.8549		**0.9827**	0.9786	
RFs	0.8556	0.8069		0.9682	0.9620	
CatBoost		0.8567			0.9751	
Logistic Regression		0.7490			0.9794	
Autoencoder-based classifier	**0.8336**		(DNN)0.7705 (CNN)0.7532 (RNN)0.7553 (SVM)0.6117	**0.9716**		(DNN)0.9277 (CNN)0.9254 (RNN)0.9278 (SVM)0.8566

드별로 10과 12 중 선택하였다. RF의 'max_depth'는 기본 설정인 '제한없음'으로 두었다. 또한 본 논문에서는 'n_estimators'와 같은 주요 하이퍼파라미터들도 그리드 탐색을 통해 선정하였다.

표 9는 본 논문과 [11]에서 사용한 오토인코더의 하이퍼파라미터를 비교하고 있다. 오토인코더의 경우 설정이 필요한 하이퍼파라미터의 종류와 범위가 많다. 표 9에서 볼 수 있듯이, 본 논문과 [11]에서 사용한 하이퍼파라미터 중 같은 값으로 지정된 것은 '# of bottleneck layers' 뿐이다.

본 논문과 [9,11]과의 비교 결과는 동일한 기계학습 방법이라 해도 하이퍼파라미터 탐색을 통해 최적의 조합을 찾는 단계를 반드시 수행해야 함을 의미한다. 특히 본 논문과 [11]의 성능 차이는 각 연구에서 사용한 분류기가 서로 다르다는 점에도 기인할 수 있으나, 오토인코더 자체의 하이퍼파라미터 설정 차이도 영향을 미쳤을 것으로 판단된다.

3.4.3 기계학습 방법별 학습 및 예측시간 결과

한편 기계학습 모델의 학습에 소요되는 시간과 1개의 거래에 대해 이상거래를 판단하는 예측시간도 중요하게 고려해야 할 요소이다. 2.1절에서는 신용카드 이상거래 탐지 문제의 주요 이슈 중 하나로 컨셉 변화를 언급했다. 신용카드 이상거래의 분포는 주기적으로, 또는 새로운 공격방식의 등장 등에 의해 시간에 따라 변화한다 [33]. 데이터의 분포가 변화하므로 한번 구축한 시스템을 오랜 기간 사용한다면 분류성능이 점차 저하될 것이다. 이러한 데이터의 특성을 고려한 이상거래 탐지 시스템의 주기적인 학습 계획이 필요하다. 따라서 실제 상황

표 8 본 연구와 [9]에서의 XGBoost 및 RF의 하이퍼파라미터 설정값
Table 8 Hyperparameter values of XGBoost and RFs in our experiments and a previous study [9]

XGBoost		
Hyperparameters	Ours	[9]
max_depth	{10, 12}	1
n_estimators	{3000, 5000}	default(100)
colsample_bytree	{0.4, 1}	default(1)
subsample	{0.8, 1}	default(1)
learning_rate	{0.02, 0.1}	default(0.1)
RFs		
Hyperparameters	Ours	[9]
max_depth	default (until all leaves are pure)	4
n_estimators	{100, 500, 1000, 3000, 5000}	default(100)

표 9 본 연구와 [11]에서의 오토인코더의 하이퍼파라미터 설정값
Table 9 Hyperparameter values of autoencoders in our experiments and a previous study [11]

Hyperparameters	Ours	[11]
# of layers in Encoder and Decoder	{1}	{2, 3, 4, 5}
# of nodes in Encoder and Decoder	{3000, 5000, 10000}	{1-512}
# of bottleneck layers	{1}	{1}
# of nodes in bottleneck	{3000, 5000, 10000}	{10, 12, 14, 16, 18, 20, 22, 24}
dropout rate	{0.05, 0.5, 0.8}	N/A
activation function	{leakyrelu, tanh}	{relu, sigmoid, linear'}
optimizer	{Adam, Adadelta}	{Adamax}
learning rate	{0.001, 0.0001, 0.00001}	{0.01, 0.001}

에서는 기계학습 방법의 평균적인 학습 소요시간도 중요한 고려 요소이다. 또한 성능이 우수한 기계학습 방법이라 해도, 실시간 거래 데이터에 대한 예측시간이 오래 소요되면 그 방법은 활용할 수 없다. 실제 상황에서는 1초에 수천에서 수만 건에 이르는 거래가 발생한다. 특히, 부정사용에 따른 이상거래는 단시간에 연속적으로 발생하는 경향이 있다. 따라서 거래의 이상여부를 빠르게 예측하고 추가적인 거래가 발생하지 못하도록 조기에 차단하는 것이 중요하다.

표 10 CCFD 데이터에 대한 기계학습 방법별 학습 및 예측 시간

Table 10 Training and prediction time for the CCFD dataset

Machine learning methods	Average training time (seconds)	Average prediction time for one test example (seconds)
LDA	0.0010	0.0009
Lasso	8.5176	0.0010
Ridge	0.7807	0.0009
Elastic net	8.6136	0.0010
XGBoost	10.2791	0.0668
LGBMs	11.7136	0.0016
RFs	88.1600	0.0737
SVMs	1459.9928	0.0012
Autoencoder-based classifier	Pretraining: 613.0124	0.0678
	Classification: 86.1102	

표 11 IEEE-CIS FD 데이터에 대한 기계학습 방법별 학습 및 예측 시간

Table 11 Training and prediction time for the IEEE-CIS FD dataset

Machine learning methods	Average training time (seconds)	Average prediction time for one test example (seconds)
LDA	16.5587	0.0038
Lasso	109.0125	0.0154
Ridge	12.0001	0.0228
Elastic net	110.0257	0.0399
XGBoost	58.3158	0.1248
LGBMs	82.4253	0.0037
RFs	1392.3987	0.3140
Autoencoder-based classifier	Pretraining: 6934.7440	0.0614
	Classification: 2389.6004	

표 10과 11은 각 방법별로 학습과 예측에 걸리는 시간을 비교하고 있다. 본 논문에서 사용한 컴퓨터 환경은 AMD Ryzen 9 5950X 16-Core CPU와 128GB RAM, NVIDIA GeForce RTX 3090, Windows 11 Pro이다. 평균 학습시간은 층화5겹교차검증을 10회 반복한 총 50회의 학습시간의 평균을 의미하고, 평균 예측시간은 1개의 시험 예제의 이상거래 여부를 예측하는 데 소요된 시간의 평균이다. 학습에 가장 많은 시간이 소요되는 방법은 SVM이며 이는 SVM의 학습알고리즘 시간 복잡도가 $O(n^2)$임에 기인하는 것으로 보인다(단, n은 학습데이터 크기). 선형 방법인 LDA, lasso, ridge, elastic net은 비선형 방법에 비해 학습시간이 짧은 편이다. Gradient boosting 기반의 XGBoost와 LGBM 학습에는 전체 방법들 중 중간정도의 시간이 소요되었다. 주기적인 재학습이 용이하며 동시에 우수한 성능을 확보하는 측면에서 실제 상황에서 gradient boosting 기반 방법을 활용하는 것이 적합하다고 판단된다.

평균 예측시간을 통해 실제 거래 1건의 이상여부 판정에 걸리는 시간을 예상해볼 수 있다. XGBoost와 LGBM 두 방법 모두 AUPRC와 AUROC 결과는 우수했지만 예측시간에서 큰 차이가 났다. LGBM은 두 데이터 모두에서 예측시간이 짧은 방법 중 하나였다. XGBoost는 CCFD 데이터 0.0668초, IEEE-CIS FD 데이터 0.1248초의 예측시간을 보였고, LGBM은 CCFD 데이터 0.0016초, IEEE-CIS FD 데이터 0.0037초의 예측시간을 보였다. 즉, LGBM이 XGBoost에 비하여 약 34~42배 빠른 예측시간을 보였다. 결과적으로 성능과 시간 효율성을 모두 고려하여 가장 우수한 방법은 LGBM이라고 볼 수 있다.

4. 결론 및 논의

본 논문은 신용카드 이상거래 탐지문제에 가장 적합한 기계학습 방법을 제안한다. 선형 방법으로는 LDA와 정규화된 로지스틱회귀 방법인 lasso, ridge, elastic net을, 비선형 방법으로는 XGBoost, LGBM, RF, SVM 및 오토인코더로 사전학습된 신경망 기반의 분류기를 적용했다. 총 9개의 기계학습 방법을 실제 신용카드 데이터인 CCFD 데이터와 IEEE-CIS FD 데이터에 적용하여 AUPRC와 AUROC의 결과를 비교하였다.

CCFD 데이터에서는 AUPRC 기준으로 XGBoost, LGBM 순으로 좋은 결과를 보였다. AUROC 기준으로는 XGBoost, elastic net, LDA, LGBM 순으로 좋은 결과를 보였다. 다만 CCFD 데이터의 AUROC 성능은 모든 기계학습 방법의 성능이 우수한 편이었고, 방법간 큰 차이는 없었다. 한편, 본 논문의 CCFD 데이터에 대한 XGBoost의 분류 정확도는 선행 연구에서 가장 우수

한 성능을 보인 CatBoost와 로지스틱회귀보다 좋은 결과였다. IEEE-CIS FD 데이터에서는 AUPRC 기준으로 LGBM, XGBoost 순으로 좋은 결과를 보였다. AUROC 기준으로는 XGBoost, LGBM 순으로 높은 분류 정확도를 보였다. AUPRC가 불균형 데이터의 정확도 측정에 더 의미 있는 평가지표라는 점을 고려하면, 성능 측면에서는 XGBoost와 LGBM이 신용카드 이상거래 탐지문제에 가장 적합하다고 볼 수 있다.

Gradient boosting 기반의 방법이 기업에서 활용하기에도 적합한지에 대해 이상거래 탐지 문제의 주요 이슈들을 중심으로 확인해보았다. 먼저, 기업과 많은 연구에서는 신용카드 거래데이터의 불균형 이슈를 해결하기 위하여 데이터 균형화를 통해 클래스간 균형을 맞춘 뒤 기계학습 방법을 학습한다. 그러나 [19]에서 주장한 바처럼 불균형 데이터를 균형화하는 것은 효과가 없거나, 오히려 분류성능을 저하시킬 수 있다. [19]에서 CCFD 데이터와 IEEE-CIS FD 데이터에 8개의 기계학습 방법(AdaBoost, XGBoost, RF, SVM, LDA, Lasso, Ridge, ElasticNet)과 7개의 표집화(sampling) 기법을 조합한 결과를 보면, CCFD 데이터는 표집화 방식 적용에 의해 유의미한 성능(AUPRC, AUROC) 향상의 사례가 없었다. 오히려 표집화 방식 적용에 의해 성능이 저하되는 사례가 있었다. IEEE-CIS FD 데이터도 역시 표집화 방식 적용이 성능 향상에 도움이 되지 못했으나, 일부 표집화 방식과 선형 기계학습 방법 조합 시 유의미한 성능 향상이 발견되기도 했다. 그러나 CCFD 데이터 결과와 마찬가지로 성능 저하의 사례가 훨씬 많았다. 본 논문에서 가장 좋은 성능을 보인 XGBoost의 경우, 표집화를 적용하지 않은 경우보다 유의미하게 성능이 향상되는 표집화 방식이 존재하지 않았다. 이러한 과거 연구결과를 고려할 때 표집화는 신용카드 이상거래 탐지에는 적합하지 않거나 혹은 매우 엄정한 평가를 거쳐야 한다고 판단된다.

본 논문에서는 하나의 거래에 대해 이상거래 여부를 예측하는 시간이 가장 짧은 LGBM을 이상거래 탐지에서 활용하기에 가장 적합한 방법으로 제안하였다. 즉 LGBM은 실제 신용카드 거래의 비정상여부를 빠르고 정확하게 판단할 수 있는 방법으로 볼 수 있다. 학습시간 기준으로 보면, 실험에 사용한 모든 기계학습 방법 중 LGBM이 가장 빠른 학습시간을 기록한 것은 아니다. 그러나 선형 기계학습 방법에 비해 상대적으로 성능이 우수한 비선형 방법들 중에서는 gradient boosting 기반 방법의 학습시간이 짧은 편이다. 컨셉 변화를 고려한 효율적인 재학습이 가능하며 동시에 우수한 성능을 확보할 수 있다는 측면에서 gradient boosting 기반 방법을 사용하는 것이 실제 상황에 적합하다고 판단된다. 본 논문의 이러한 결과는 신용카드 이상거래 탐지문제를 해결하기 위한 기업의 의사결정과정에서 중요한 판단근거가 될 수 있을 것으로 기대한다.

향후에는 특징에 대한 정확한 의미를 확인할 수 있는 실제 신용카드 거래 데이터를 이용하여, 데이터 및 특징 분석을 기반으로 좀 더 높은 정확도를 보이는 기계학습 방법을 찾는 연구로의 확장이 필요하다. 또한, 거래의 종류(온라인/오프라인, 국내/해외 등)에 따라 각각 적합한 방법이 다를 수 있으므로, 이를 고려하여 종합적인 정확도를 높이는 연구도 가능하다.

References

[1] J. Jurgovsky, M. Granitzer, K. Ziegler, S. Calabretto, P.-E. Portier, L. He-Guelton, and O. Caelen, “Sequence classification for credit-card fraud detection,” *Expert Systems with Applications*, Vol. 100, pp. 234-245, 2018.

[2] S. Nami and M. Shajari, “Cost-sensitive payment card fraud detection based on dynamic random forest and k-nearest neighbors,” *Expert Systems with Applications*, Vol. 110, pp. 381-392, 2018.

[3] Y. Wang, S. Adams, P. Beling, S. Greenspan, S. Rajagopalan, M. Velez-Rojas, S. Mankovski, S. Boker, and D. Brown, “Privacy preserving distributed deep learning and its application in credit card fraud detection,” *2018 17th IEEE International Conference on Trust Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*, IEEE, pp. 1070-1078, 2018.

[4] E. Lopez-Rojas, A. Elmir, and S. Axelsson, “PaySim: A financial mobile money simulator for fraud detection,” *28th European Modeling and Simulation Symposium*, EMSS, pp. 249-255, 2016.

[5] L. Zheng, G. Liu, W. Luan, Z. Li, Y. Zhang, C. Yan, and C. Jiang, “A new credit card fraud detecting method based on behavior certificate,” *2018 IEEE 15th International Conference on Networking Sensing and Control (ICNSC)*, IEEE, pp. 1-6, 2018.

[6] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, “Credit card fraud detection using AdaBoost and majority voting,” *IEEE Access*, Vol. 6, pp. 14277-14284, 2018.

[7] Z. Li, M. Huang, G. Liu, and C. Jiang, “A hybrid method with dynamic weighted entropy for handling the problem of class imbalance with overlap in credit card fraud detection,” *Expert Systems with Applications*, Vol. 175, 114750, 2021.

[8] A. Q. Abdulghani, O. N. Ucan, and K. M. A. Alheeti, “Credit card fraud detection using XGBoost algorithm,” *2021 14th International Conference on Developments in eSystems Engineering (DeSE)*,

IEEE, pp. 487-492, 2021.
[9] J. L. Leevy, J. Hancock, and T. M. Khoshgoftaar, "Comparative analysis of binary and one-class classification techniques for credit card fraud data," *Journal of Big Data*, Vol. 10, No. 118, 2023.
[10] C.-T. Ting, C. Lee, S.-H. Huan, and W.-C. Peng, "Credit card fraud detection via intelligent sampling and self-supervised learning," *ACM Transactions on Intelligent Systems and Technology*, Vol. 15, No. 2, pp. 1-29, 2024.
[11] H. Fanai and H. Abbasimehr, "A novel combined approach based on deep Autoencoder and deep classifiers for credit card fraud detection," *Expert Systems with Applications*, Vol. 271, 119562, 2023.
[12] K. I. Alkhatib, A. I. Al-Aiad, M. H. Almahmoud, and O. N. Elayan, "Credit card fraud detection based on deep neural network approach," *2021 12th International Conference on Information and Communication Systems (ICICS)*, IEEE, pp. 153-156, 2021.
[13] E. F. Malik, K. W. Khaw, B. Belaton, W. P. Wong, and X. Chew, "Credit card fraud detection using a new hybrid machine learning architecture," *Mathematics*, Vol. 10, No. 9, 1480, 2022.
[14] D. Ge and S. Chang, "Credit card fraud detection using lightgbm model," *2020 International Conference on E-commerce and Internet Technology (ECIT)*, IEEE, pp. 232-236, 2020.
[15] N. Nguyen, T. Duong, T. Chau, V.-H. Nguyen, T. Trinh, D. Tran, and T. Ho, "A proposed model for card fraud detection based on Catboost and deep neural network," *IEEE Access*, Vol. 10, pp. 96852-96861, 2022.
[16] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS One*, Vol. 10, No. 3, p. e0118432, 2015.
[17] A. Cherif, A. Badhib, H. Ammar, S. Alshehri, M. Kalkatawi, and A. Imine, "Credit card fraud detection in the era of disruptive technologies: A systematic review," *Journal of King Saud University-Computer and Information Sciences*, Vol. 35, No. 1, pp. 145-174, 2023.
[18] C. Bockel-Rickermann, T. Verdonck, and W. Verbeke, "Fraud analytics: A decade of research: Organizing challenges and solutions in the field," *Expert Systems with Applications*, Vol. 232, 120605, 2023.
[19] M. Kim and K.-B. Hwang, "An empirical evaluation of sampling methods for the classification of imbalanced data," *PLoS One*, Vol. 17, No. 7, p. e0271260, 2022.
[20] R. Goorbergh, M. Smeden, D. Timmerman, and B. V. Calster, "The harm of class imbalance corrections for risk prediction models: illustration and simulation using logistic regression," *Journal of the American Medical Informatics Association*, Vol. 29, No. 9, pp. 1525-1534, 2022.
[21] G. Velarde, M. Weichert, A. Deshmunkh, S. Deshmane, A. Sudhir, K. Sharma, and V. Joshi, "Tree boosting methods for balanced and imbalanced classification and their robustness over time in risk assessment," *Intelligent Systems with Applications*, Vol. 22, 2024.
[22] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys (CSUR)*, Vol. 46, No. 4, pp. 1-37, 2014.
[23] G. J. McLachlan, *Discriminant analysis and statistical pattern recognition*, John Wiley & Sons, New Jersey, 2005.
[24] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, Vol. 58, No. 1, pp. 267-288, 1996.
[25] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, Vol. 12, No. 1, pp. 55-67, 1970.
[26] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, Vol. 67, No. 2, pp. 301-320, 2005.
[27] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785-794, 2016.
[28] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems*, Vol. 30, 2017.
[29] L. Breiman, "Random forests," *Machine Learning*, Vol. 45, pp. 5-32, 2001.
[30] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, Vol. 20, pp. 273-297, 1995.
[31] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE Journal*, Vol. 37, No. 2, pp. 233-243, 1991.
[32] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," *Proc. of the 23rd International Conference on Machine Learning*, pp. 233-240, 2006.
[33] D. Malekian and M. R. Hashemi, "An adaptive profile based fraud detection framework for handling concept drift," *2013 10th International ISC Conference on Information Security and Cryptology (ISCISC)*, pp. 1-6, IEEE, 2013.



김 미 숙

2007년 8월 연세대학교 산업공학과 졸업(공학사). 2010년 2월 서울대학교 산업공학과 졸업(공학석사). 2024년 8월 숭실대학교 컴퓨터학과 졸업(공학박사). 2010년 1월~현재 비씨카드 재직. 관심분야는 기계학습, 인공지능, 데이터마이닝



황 규 백

2005년 8월 서울대학교 전기·컴퓨터공학부 졸업(공학박사). 2012년 3월~2013년 2월 미국 보스턴어린이병원 의학과 객원연구원. 2006년 3월~현재 숭실대학교 컴퓨터학부 교수. 관심분야는 기계학습, 생물정보학