ISSN 2383-6318(Print) / ISSN 2383-6326(Online)
KIISE Transactions on Computing Practices, Vol. 25, No. 1, pp. 1-8, 2019. 1
https://doi.org/10.5626/KTCP.2019.25.1.1

오토인코더를 활용한 효율적인 신용카드 사기 탐지 지도 기법

(Efficient Supervised Credit Card Fraud Detection Technique using Autoencoder)

이 용 현 † 구 해 모 † 김 형 주 ††
(YongHyun Lee) (HeyMo Kou) (Hyoung-Joo Kim)

요 약 신용카드 결제 이상 거래 탐지는 카드의 사용이 실시간으로 이루어지고, 탐지가 즉각적으로 이루어져야 한다는 점에서 스트리밍 데이터 분석으로 볼 수 있고, 이는 배치 분석보다 더 빠른 실시간 분석을 요구한다. 데이터에서 핵심 부분만을 추출하여 분석하는 방법은 이러한 연산 속도의 요구사항을 잘 만족시킬 수 있을 것이고, 주성분 분석 등의 기법을 통해 이루어져 왔다. 본 논문에서는 인공신경망을 활용한 차원 축소 기법인 오토인코더로 데이터를 전처리하여 데이터의 차원을 축소한 후 데이터 마이닝 기법을 적용하는 방법을 제안한다. 오토인코더는 데이터 차원들 간의 비선형적인 결합 관계도 포착할 수 있기에 보다 효과적인 차원 축소 방법이다. 또한 이를 데이터베이스 내에서의 이상 탐지 분석에 어떻게 사용할 지에 대하여 CQL과의 연동 방법론을 제시하고자 한다.

키워드: 오토인코더, 이상 거래 탐지, 차원 축소, 데이터 정리

Abstract Credit card fraud detection can be viewed as Streaming Data Analysis in which a card is used in real time and the detection must be done immediately. This requires a real time analysis that is faster than the batch analysis. The method of extracting and analyzing only the core part of the data will satisfy the requirements of this computation speed. This has been done through techniques such as principal component analysis. In this paper, we propose a method that applies data mining techniques after reducing the dimension of data by preprocessing the data with Autoencoder. This method is known as the dimension reduction method and it uses a Neural Network. Autoencoder is a very efficient method of dimension reduction because it can capture nonlinear associations between data feature dimensions. We also propose a methodology to combine Autoencoder with CQL for fraud detection analysis in the database.

Keywords: autoencoder, fraud detection, dimentionality reduction, data reduction

· 이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.R0113-15-0005, 대규모 트랜잭션 처리와 실시간 복합 분석을 통합한 일체형 데이터 엔지니어링 기술 개발)

† 비 회 원 : 서울대학교 컴퓨터공학과(Seoul Nat'l Univ.)
leeyh@idb.snu.ac.kr
(Corresponding author임)
†† 비 회 원 : 서울대학교 컴퓨터공학과
hmkou@idb.snu.ac.kr
††† 종신회원 : 서울대학교 컴퓨터공학과 교수
hjk@snu.ac.kr

논문접수 : 2018년 2월 8일
(Received 8 February 2018)
논문수정 : 2018년 9월 21일
(Revised 21 September 2018)
심사완료 : 2018년 11월 20일
(Accepted 20 November 2018)

정보과학회 컴퓨팅의 실제 논문지 제25권 제1호(2019. 1)

1. 서 론

1920년대에 처음 도입되었고, 1950년대 후반에 미국 주요 은행들이 참여한 이후로, 신용카드는 현금과 함께 사람들의 주요 거래 혹은 결제 수단이 되어왔다. 한국은행이 조사한 <2016년 지급수단 이용 행태 조사 >에 따르면 대한민국 개인의 신용카드 및 체크, 직불카드 보유 비율은 각각 93.3%, 98.3%로 성인 거의 대부분이 카드를 보유하고 있다. 수많은 이점 및 편의성을 제공해주는 신용카드이지만, 탄생한 이후로 신용카드와 관련하여 수많은 이상 거래(Fraud) 방법들이 생겨났다. 이러한 이상 거래를 어떻게 예방할 것인지, 그리고 이상 거래가 발생하였을 때 이를 어떻게 탐지할 것인가는 금융권의 가장 중요한 이슈 중 하나이다. 처음에는 단순히 장부에 기록되는 형태의 신용카드였지만, 정보통신 기술 및 전자기술의 발전으로 전산화되었고, 이는 이상 거래 탐지를 위하여 관련 도메인 지식만을 활용하던 초기의 탐지 방식을 넘어서, 사용된 카드의 데이터를 축적하고 이를 전산적으로 분석하는 데이터 기반 분석을 가능하게 해주었다.

신용카드 사용 관련 이상 거래 혹은 이상 패턴 탐지(Fraud Detection)는 카드 사용이 실시간으로 이루어지고, 이상 거래가 발생하자마자 이를 탐지해야 한다는 점에서 스트리밍 분석이고, 이는 일반적인 배치 분석과는 차이점을 가진다. 카드 사용 데이터가 실시간으로 스트림 되고, 즉각적인 이상 거래 탐지가 필요하기 때문에 배치 분석보다는 보다 빠른 분석이 필요하고, 새로 들어온 데이터로부터의 재학습에도 보다 빠른 학습이 요구된다. 데이터에서 필요한 혹은 중요한 정보만을 추출 혹은 압축함으로써 처리해야 할 데이터의 크기을 줄일 수 있고[1], 이는 스트림 데이터 분석이 필요로 하는 빠른 연산 속도 성취에 도움이 될 것이다.

본 논문에서는 인공신경망을 활용한 데이터 축소 방식인 오토인코더를 사용하여, 이상 거래 탐지를 위한 탐지 성능은 유지하면서, 학습 및 탐지 속도를 향상시키는 방법을 제안하고자 한다. 데이터의 특성 혹은 변수들간의 선형 결합만을 포착할 수 있는 주성분 분석(Principle Component Analysis - PCA)에 비하여 오토인코더는 Sigmoid, Relu, Tanh 등의 Activation Function을 사용하여 변수들간의 비선형적인 관계를 추출할 수 있을 뿐만 아니라, 주성분 분석을 사용하여 압축된 변수들 간에서도 관계를 추출할 수 있는 유용한 데이터 축소 방법이다. 이상 거래 탐지를 위한 기계학습 방법 중에서는 Random Forest 방식을 사용하였다. Random Forest는 이상 거래 탐지에 대하여 다른 기계학습 기법들보다 높은 정밀도 및 재현율을 보여주었다[2,3].

2. 관련 연구

2.1 Fraud Detection Techniques

도메인 지식을 통한 이상 거래 탐지가 아닌, 데이터를 이용한 이상 거래 탐지를 위하여 다양한 기법들이 제안되어 왔는데, 크게 2가지 접근법으로 볼 수 있다. 첫 번째는 데이터의 분포에 대한 방법으로, 이는 정상 카드 거래 결제 건수에 비해 이상 거래 건수의 비율이 압도적으로 적은 클래스 간의 쏠림 현상에 초점을 맞추는 방식이다. 이러한 쏠림 현상을 해결하기 위해 소수 클래스 인스턴스의 개수를 늘리는 Over-sampling 기법들과 다수 클래스의 인스턴스 개수를 줄이는 Under-sampling 기법들이 있으며, 이들을 조합한 Hybrid-over and under sampling 기법들도 있다. 두 번째 방식은 알고리즘 측면에서 접근하는 방식인데, 알고리즘 학습 단계에서 각 카드 결제 건의 Fraud 해당 유무를 알고 분석하는 지도학습(Supervised Learning) 방식과 대부분의 결제 건과 다른 유형을 보이는 결제 건을 찾아내는 자율 학습(Unsupervised Learning) 방식이 있다. 본 논문에서는 지도 학습의 속도를 향상시킬 수 있는 방법을 찾는 것에 목적을 둔다.

이상 거래 탐지를 위해 지도학습을 사용하는 방식은 오랫동안 연구되어 왔다. 의사결정나무는 트리 형식으로 표현될 수 있고, If then-else 로 변환이 가능한 데이터 마이닝 모델이다. ID3, CART, C4.5 등의 의사결정나무 모델로 고객분류 및 이상 패턴 탐지를 하는 연구가 진행되어 왔다[2,4].

Support Vector Machine (SVM)은 클래스 간의 마진이 최대화 되도록 하는 초평면을 찾는 기계 학습 기법이다. 주어진 변수들만으로는 두 클래스가 선형적으로 나뉘어지지 않을 수 있으므로, 커널 함수를 사용하여 더 높은 차원의 공간으로 변환시킨 후에 클래스들을 선형적으로 구분할 수 있는 초평면을 만들도록 한다.

Random Forest는 1개의 트리만을 생성하여 분류를 하는 의사결정나무 방식의 불안정성을 보완하고자 다수의 의사결정나무를 생성하여, 이들의 분류 결과들 중 가장 많이 도출되는 클래스를 선택하는 방식이다[5]. 다수의 트리가 생성되기 때문에 과적합 및 데이터 내부의 잡음에 강한 모습을 보인다. Random Forest는 이상 패턴 감지, 스팸 메일 탐지, 네트워크 침입 탐지 등 데이터의 클래스들이 편향된 형태를 가진 다양한 분야에서 다른 데이터 마이닝 기법에 비해 높은 정확성을 보인다[6]. 또한 Random Forest가 이상 거래 탐지 분야에서 다른 데이터 마이닝 기법에 비해 높은 성능을 보인다는 것은 다른 여러 논문에서도 실험되었다. Whitrow et al.은 Random Forest가 SVM, 로지스틱 회귀분석 및 K-Nearest Neighbors (KNN)보다 더 좋은 성능을 보

인다고 하였고[3], Bhattacharyya et al.도 Random Forest가 SVM과 로지스틱 회귀분석보다 높은 정확성을 보임을 보였다[2]. 본 논문에서도 클래스를 구분할 모델로써 Random Forest 모델을 사용할 것이다. 인공신경망도 Random Forest와 비슷한 성능을 보였음에도 불구하고 인공신경망은 Random Forest에 비해 설정해야 할 파라미터들이 압도적으로 많고, 각 파라미터 설정에 따라 성능의 차이가 매우 심하기에 Random Forest 모델을 선택하였다.

인공신경망은 사람의 뇌 신경 회로를 묘사하여 만든 데이터 마이닝 기법이다. 데이터를 입력 받는 입력층, 클래스 분류 결과를 출력하는 출력층, 그리고 입력층과 출력층 사이의 은닉층, 3 종류의 층들로 구성되어 있다. 학습과정에서 각 층들간의 Weight 및 Bias가 학습데이터의 클래스를 예측할 수 있도록 조정되고, 분류 과정에서는 분류하고자 하는 데이터가 학습된 인공신경망의 입력층과 은닉층들을 통과하면서 출력층에서 클래스가 분류된다. Patidar et al.은 신용카드 이상 거래 탐지에서 인공신경망이 높은 정확도를 가짐을 보였다[7].

2.2 Autoencoder

데이터 마이닝에서 기법의 정확도 향상, 연산 속도 향상 및 시각화 등을 위한 적절한 차원축소는 필수적이다. 도메인 지식을 미리 가지고 있는 분야라면, 분석가가 직접 비교적 중요성이 떨어지는 변수를 배제할 수 있다. 그렇지만 도메인 지식에 기반한 방식은 변수의 내재된 중요성을 판단하지 못할 수 있으며, 변수 간의 결합이 유의미한 의미를 지닐 경우, 이를 쉽게 포착하지 못한다.

관련 도메인 지식에 의존하지 않고, 가장 많이 사용되는 고차원의 데이터를 저차원으로 축소시키는 방법으로는 주성분 분석이 있다. 주성분 분석은 데이터 변수 축들의 선형 결합으로 이루어진 새로운 축들 중 데이터를 사상시켰을 때 분산이 가장 커지는 축, 즉 데이터의 주성분을 찾아내는 방식이다. 주성분 분석을 사용함으로써 데이터를 가장 잘 설명할 수 있는 새로운 변수 혹은 특성을 추출할 수 있다[8]. 그러나 주성분 분석은 기존 변수들간의 선형 결합만을 추출할 수 있다는 단점이 있다. 만약 변수들 간의 관계가 지수, 혹은 Sigmoid 형식이라면 주성분 분석은 이러한 특징을 정확히 잡아낼 수 없다.

오토인코더는 출력층의 결과물이 입력층에 입력되는 데이터를 완벽하게 복구하는 것이 목적인 인공신경망이다. 그림 1은 오토인코더의 형태를 보여준다. 5개의 입력 노드들은 입력 데이터가 총 5개의 변수를 가지고 있음을 의미하는데, 이는 5차원의 데이터로 볼 수 있다. 5차원의 데이터들은 그림 1의 A부분을 통과하면서 3개의 은닉 노드들을 가진 은닉층에서 3차원으로 압축된다.

고차원의 입력 데이터가 저차원의 은닉층에서 압축되는

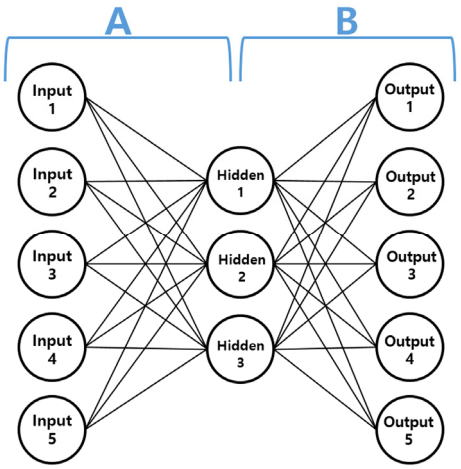

그림 1 단순 오토인코더 구조
Fig. 1 Simple Autoencoder Architecture

이러한 부분을 '인코더'라고 한다. 그림 1의 B부분에서는 3차원으로 압축된 데이터가 5차원으로 차원이 다시 확장된다. 은닉층에서 저차원으로 압축된 데이터를 원래 차원의 데이터로 복구시키는 이러한 작업을 '디코더'라고 한다. 이렇게 차원이 압축되었다가 다시 확장된 데이터가 원본 데이터와 동일하도록 만드는 것이 오토인코더의 목적이다. 인공신경망의 Activation Function으로 Sigmoid, Tanh, Rectified Linear Unit(ReLU) 등을 사용함으로써, 변수들 간의 조합을 선형조합 이상의 복잡한 조합을 만들 수 있다.

Hinton et al.은 가로 및 세로가 각각 28 pixel인 784차원의 숫자 및 사람 얼굴 사진 표본들을 PCA를 사용하여 차원을 압축하였다가 복구시킨 사진들과 오토인코더를 사용하여 차원을 압축하였다가 복구시킨 사진들을 비교하였는데, PCA방식에 비해 오토인코더 방식이 더 원본 이미지에 가깝게 복구됨을 보여 주었다[9].

3. Autoencoder를 사용한 전처리 기법

클래스를 예측하기 위한 분류기의 생성은 미리 정의된 규칙을 바탕으로도 가능하지만, 대부분은 데이터와 예측하고자 하는 클래스 혹은 값 사이의 관계를 우리가 구할 수 있는 학습 데이터로부터 모델링하는 방식이다. 이러한 모델링을 하기 전에 각 데이터들이 어떻게 표현되는지를 알아야 하는데, 이것이 해당 데이터의 변수 혹은 특성이 된다. 어떠한 변수를 선택하느냐에 따라서 동일한 기계 학습 알고리즘에서도 모델의 성능이 확연히 달라지므로 예측을 잘 할 수 있는 좋은 변수를 찾아내는 것은 데이터 마이닝에서 매우 중요하다. 이러한 변수들은 대부분 해당 분야에 대한 전문지식을 필요로 한다.

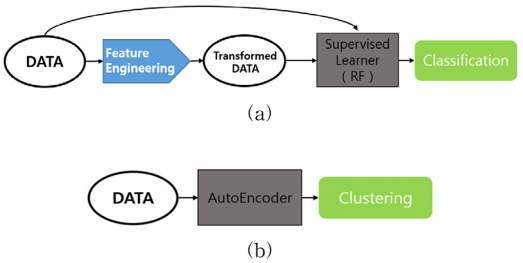

그림 2 기존의 지도 학습(a), 기존의 오토인코더(b)
Fig. 2 Conventional Supervised Learning (a), Conventional Autoencoder (b)

실제로 이미지 프로세싱이나 컴퓨터 비전 등 특정 도메인을 대상으로 한 변수 선택 혹은 추출 대한 연구들도 진행되어 왔다[10]. 도메인에 무관하게 어떤 분야에도 적용할 수 있는 기법들도 많이 연구되었는데, 대표적으로 표준화, 정규화, 주성분 분석 등이 있다. 더 좋은 특성를 선택하기 위한 이러한 작업들은 Feature Engineering이라고 불리며, 데이터 마이닝 모델링 직전에 전처리 과정으로써 사용된다[10]. 이러한 과정은 그림 2(a)와 같다.

Hinton et al.은 오토인코더를 문헌 검색과 얼굴 인식 등에 적용시켰고, 주성분 분석 및 LSA를 사용하여 2차원으로 축소시킨 결과보다 오토인코더를 사용하여 2차원으로 축소시킨 결과가 각 클래스의 특성을 더 잘 보여주고 있음을 보였다[9]. 이 과정의 흐름도는 그림 2의 (b) 그림과 같다. 본 논문에서는 오토인코더를 그림 2의 (b)가 아니라 (a) 그림처럼 지도 학습 모델링을 위한 전처리 과정으로 사용하는 방법을 제안하고자 한다.

Deep Neural Network 구성을 위하여 인공신경망 앞에 오토인코더를 배치하는 방안은 이전부터 제안되어 왔다[9]. 그럼에도 불구하고 오토인코더를 통한 차원 축소를 통하여 인공신경망이 아닌 의사결정나무, SVM 및 Random Forest 등의 기타 지도 학습과의 결합은 앞의 연구만큼 활발하지는 않았다. 본 논문에서는 오토인코더에서 인코더 부분까지만을 진행하여 데이터의 차원을 축소시킨 후, 지도 학습을 사용하여 데이터 마이닝 모델을 학습 및 새로운 데이터를 분류하는 방법을 제안한다. 그림 3은 이러한 과정을 보여준다. 이는 그림 2(a)와 매우 비슷한데, 오토인코더를 사용하여 Feature Engineering을 한다고 볼 수 있다. 오토인코더를 사용하여 차원을 축소하더라도, 축소하지 않은 원본 데이터만큼의 정밀도 및 재현율이 확보된다면 지도 학습의 분류 성능은 유지하면서 학습 시간 및 분류 시간의 단축을 기대할 수 있다.

오토인코더는 주성분 분석에 비해 몇 가지 중요한 이점을 보인다. 첫 번째로 오토인코더는 변수들의 선형조합보다 복잡한 조합이 가능하므로 주성분 분석보다 변수

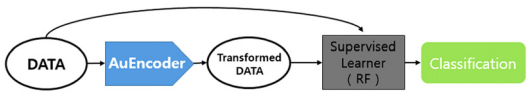

그림 3 오토인코더를 전처리기로 활용한 지도 학습
Fig. 3 Supervised Learning using Autoencoder as a Preprocessor

들간의 비선형 조합을 더 잘 반영할 수 있다. 두 번째 이점은 주성분 분석으로 만들어진 새로운 변수들의 축들은 서로 직교하기 때문에 반복적인 주성분 분석이 불가능하다. 그러나 오토인코더는 각 변수들의 축들이 직교하지 않으므로 이미 주성분 분석이 된 변수들을 대상으로 추가적인 차원 축소가 가능할 뿐만 아니라, 오토인코더를 적용한 후에 주성분 분석을 적용하거나 중첩적인 오토인코더를 적용하는 등 다양한 변형이 가능하다. 보다 나아가서, 그림 1처럼 1개의 은닉층이 아니라 다수의 은닉층들을 사용한 오토인코더를 활용한 방법도 제안해보고자 한다.

이러한 오토인코더를 이용한 방식은 빠른 처리가 필요한 스트림 데이터를 데이터베이스 안에서 실시간으로 처리하는 데에 도움이 되리라 예상된다. 지도 학습 모델을 CQL(Continuous Query Language)로 변환하여 DSMS(Data Stream Management System)에 탑재함으로써 DSMS 내부에서의 이상 거래 탐지가 가능할 것이다. 그림 4는 이를 위한 흐름도를 보여준다.

이상패턴 분석기는 기존의 데이터로부터 미리 학습해 둔 데이터 마이닝 모델을 CQL로 변환하여 데이터베이스 내부에 저장해 둔 것이다. 실시간 스트림으로 들어오는 데이터를 오토인코더를 사용하여 차원을 축소시키고, 이상패턴 분석기가 CQL을 사용하여 이상 거래 유무를 판단하는 것이다. 알고리즘1은 데이터 마이닝 모델 중 의사결정 나무를 CQL을 활용하여 구현한 예시이다.

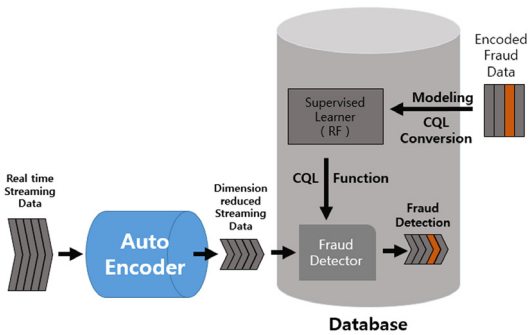

그림 4 오토인코더를 전처리기로 활용한 데이터베이스 내에서의 스트림 데이터 분석
Fig. 4 Stream data analysis in DBMS using Autoencoder as preprocessor

알고리즘 1 의사결정나무 CQL 예제
Algorithm 1. Decision Tree CQL Example

```
CASE WHEN Column1 <= 3.4324565 THEN
    'Normal'
ELSE
    CASE WHEN Column2 <= 1.867473 THEN
        CASE WHEN Column5 <= 21.48375 THEN
            'Fraud'
        ELSE
            'Normal'
        END
    ELSE
        'Normal'
    END
END
```

4. 실 험

4.1 Data set & Environment

본 데이터는 2013년 9월 유럽에서의 2일간의 신용카드 사용 기록이다. 492건의 이상 거래, 284,315건의 정상 거래로 구성된 총 284,807건의 사용 기록이며, 구성 비에서 볼 수 있듯이 매우 불균형한 데이터이다. 개인 신상 정보 기밀로 인하여 원본 데이터가 아닌, 주성분 분석이 적용된 28개의 변수를 가진 데이터이다[11].

실험 환경으로는 Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz × 2 (8 × 2 cores), 80GB DDR4 RAM, Ubuntu 16.03.2 LTS를 사용하였고, 언어로는 Python 2.7을 사용하였다. TensorFlow 1.2 버전을 오토인코더 구현을 위하여[12], Scikit-learn 0.18.1을 Random Forest 모델링을 위하여 사용하였다[13].

Random Forest의 각 트리에서 불순도를 측정하는 방법으로는 지니 인덱스(Gini Index)를 사용하였고, 입력 변수의 개수를 n 이라고 하였을 때, 트리의 성장을 위해 고려하는 최대 변수의 개수는 $\sqrt{n}$개로 설정하였다. 트리의 개수는 5, 10, 30, 50, 100, 300, 500, 1000, 5000으로 다양하게 실험하였다.

4.2 1 Layer Autoencoder

1 Layer 오토인코더에서는 그림 1과 같이 1개의 은닉층을 가진 오토인코더를 다음의 2가지 조건에 따라 총 9 종류 생성하였다. Activation Function으로는 가장 많이 쓰이는 3가지를 선택하였다[14].

- 3가지 Activation Function : Sigmoid, ReLU, Tanh
- 28 차원(변수)을 축소 : 3, 5, 10

원본 데이터를 8 : 2의 비율로 학습 데이터와 검증 데이터로 나누었고, 각각을 9가지 오토인코더로 전처리한 후에 Random Forest로 학습 및 검증한 결과와 원본 데이터를 차원 축소 없이 Random Forest로 학습 및 검증한 결과를 비교하였다. 교차타당화를 5회 시행하여

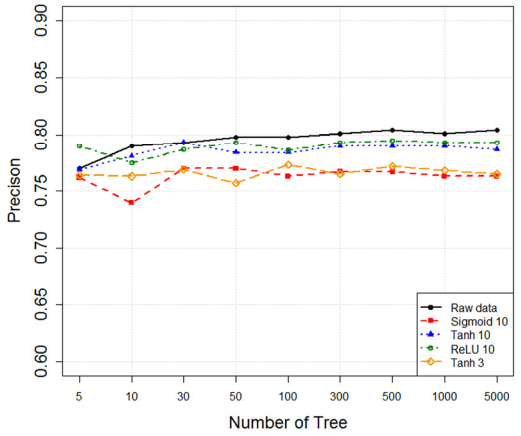

그림 5 1 Layer 오토인코더 및 Random Forest 정밀도
Fig. 5 Random Forest precision with 1 Layer Autoencoder

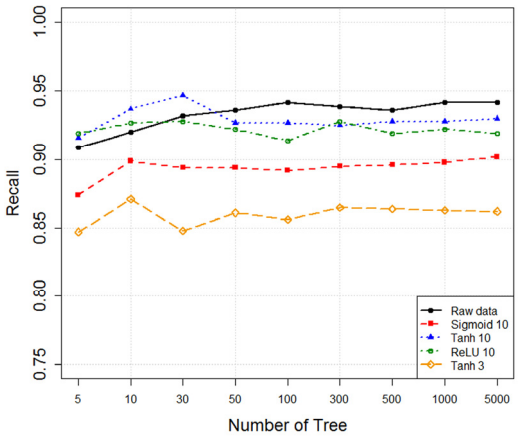

그림 6 1 Layer 오토인코더 및 Random Forest 재현율
Fig. 6 Random Forest recall with 1 Layer Autoencoder

결과를 통합하였고, 원본 데이터와 각 오토인코더로 전처리된 데이터의 학습 시간, 분류 시간, 정밀도 및 재현율을 계산 및 비교하였다.

그림 5와 그림 6은 9개의 오토인코더 중에서 정밀도 및 재현율이 가장 높은 3개의 오토인코더(Sigmoid 10, Tanh 10, ReLU 10)를 적용한 데이터와 3차원으로 축소시키는 오토인코더(Tanh 3)을 적용한 데이터와 원본 데이터를 비교한 실험 결과이다. 9개의 오토인코더 중에서 10차원으로 압축되고 Action Function이 각각 ReLU, Tanh인 ReLU 10과 Tanh 10이 가장 좋은 성능을 보이는데, 원본 데이터와의 정밀도 및 재현율 차이가 1~2% 내외이다. 트리의 개수가 적을 경우 원본 데이터보다 더 높은 재현율을 보이기도 한다. ReLU의 경우 저차원으로 압축할수록 정밀도와 재현율 성능이 많이 떨어

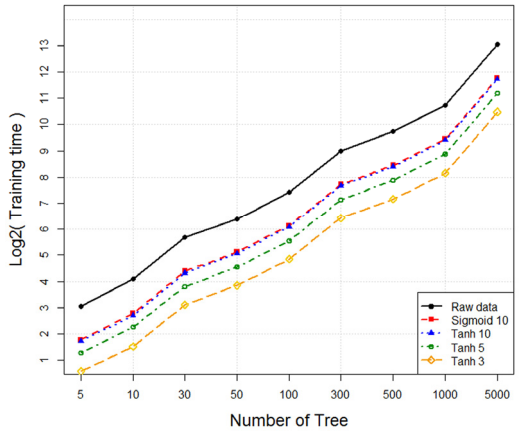

그림 7 1 Layer 오토인코더 및 Random Forest의 학습 시간
Fig. 7 Random Forest training time with 1 Layer Autoencoder

지는데, 5차원의 경우 68%로, 3차원의 경우 40%정도까지 떨어진다. 반면에 Tanh의 경우 3차원으로까지 축소하였음에도 불구하고 정밀도는 2~3%, 재현율은 7%정도 밖에 차이 나지 않는다. 따라서 Tanh가 ReLU나 Sigmoid보다 본 데이터의 특징을 잘 함축할 수 있다고 볼 수 있을 것이다.

그림 7은 원본 데이터와 1 Layer 오토인코더로 전처리한 데이터에 대한 Random Forest 학습 시간의 비교 그래프이다. 오토인코더로 출력되는 변수의 개수가 적을수록 학습시간이 급감함을 볼 수 있다. 28개 변수를 가진 원본 데이터에서 10개의 변수로 축소할 경우 약 2배만큼 빨라졌으며, 3개의 변수로 축소할 경우 약 4배만큼 빨라졌음을 볼 수 있다. 10개의 변수를 추출하기 위한 오토인코더의 학습시간이 약 160초였으므로 트리의 개수가 100개 이상일 경우, 오토인코더 학습 및 Random Forest 학습 시간이 원본 데이터의 Random Forest 학습보다 빠르다. 최근 분류결과를 반영하여 모델을 재학습하는 경우가 많은데, 이 경우 오토인코더를 사용한 차원 축소의 효과는 더욱 극대화 된다.

표 1은 원본 데이터와 오토인코더로 전처리한 후의 Random Forest의 분류 시간 비교이다. 새로 들어온 하나의 데이터 인스턴스가 오토인코더를 통과하여 차원 축소되는 시간은 0.001초로 표 1의 분류시간에 비해 무시할 수 있을 정도로 작다. 평균적으로 약 10% 정도의 분류시간이 단축되었으며, 이러한 시간의 감소는 실시간으로 스트림되는 데이터에서의 이상 패턴 감지를 처리하는데 더욱 용이할 것이다.

4.3 2 Layer Autoencoder

2 Layer 오토인코더에서는 2개의 은닉층을 가진 오토

표 1 1 Layer 오토인코더 및 Random Forest의 분류 시간
Table 1 Random Forest classification time with 1 Layer Autoencoder (단위 : 초)

Num of Tree	Autoencoder				Original Data
	Tanh 10	ReLU 10	Tanh 5	Tanh 3	
5	0.026	0.03	0.026	0.026	0.037
10	0.047	0.054	0.051	0.053	0.063
30	0.14	0.15	0.145	0.155	0.180
50	0.233	0.249	0.242	0.26	0.292
100	0.463	0.487	0.484	0.519	0.601
300	1.423	1.515	1.483	1.659	1.780
500	2.402	2.403	2.508	2.675	3.024
1000	4.869	4.81	5.106	5.412	5.961
5000	24.82	24.16	25.95	26.45	29.82

인코더를 다음의 2가지 조건에 따라 총 15 종류 생성하였다. Random Forest 적용 및 성능 비교 기준은 1 Layer 오토인코더와 동일하다.

- 3가지 Activation Function : Sigmoid, ReLU, Tanh
- 28 차원을 5가지 차원으로 축소 : (10, 3), (10, 5), (20, 3), (20, 5), (20, 10)

그림 8 및 그림 9는 15개의 오토인코더 중에서 정밀도 및 재현율이 가장 높은 4개의 오토인코더와 원본 데이터의 비교이다. 20차원으로 압축 후, 다시 10차원으로 압축하고 Action Function이 Tanh인 Tanh 20-10이 가장 좋은 정밀도와 재현율을 보인다. 2 Layer 오토인코더에서도 Tanh는 다른 Activation Function들보다 저차원에서도 높은 정밀도 및 재현율을 보인다.

그림 10은 원본 데이터와 2 Layer 오토인코더로 전처리한 후의 Random Forest 학습 시간의 비교 그래프이다. 1 Layer 오토인코더와 마찬가지로 오토인코더에서

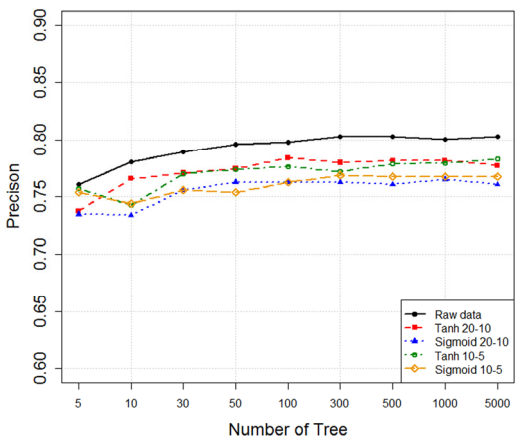

그림 8 2 Layer 오토인코더 및 Random Forest 정밀도
Fig. 8 Random Forest precision with 2 Layer Autoencoder

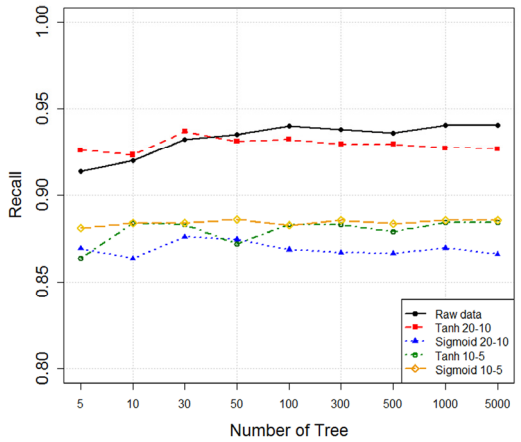

그림 9 2 Layer 오토인코더 및 Random Forest 재현율
Fig. 9 Random Forest recall with 2 Layer Autoencoder

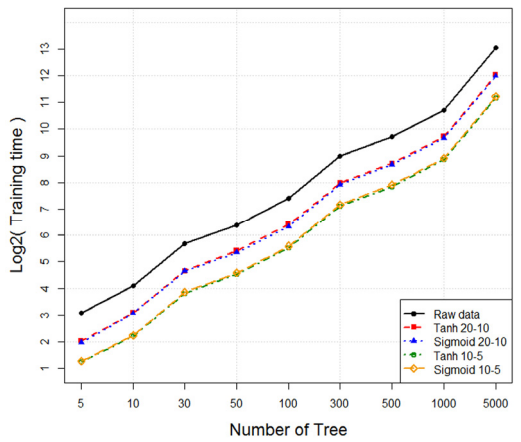

그림 10 2 Layer 오토인코더 및 Random Forest의 학습 시간
Fig. 10 Random Forest training time with 2 Layer Autoencoder

서 출력되는 변수의 개수가 적을수록 학습시간이 급감함을 볼 수 있고, 학습 시간의 감소 비율도 비슷하다. 20-10개의 변수를 추출하기 위한 오토인코더의 학습시간이 약 230초, 10-5개의 변수를 추출하는 데는 약 190초 정도로 1 Layer 오토인코더보다는 학습시간이 길었다.

2 Layer 오토인코더에서도 새로 들어온 하나의 데이터 인스턴스가 오토인코더를 통과하여 차원 축소되는 시간은 0.001초로 매우 작기에 무시가능 한 수준이다. 표 2에서도 원본 데이터보다 평균적으로 약 10% 정도의 분류시간이 짧아졌음을 확인할 수 있다. 덧붙여 3 Layer 이상의 오토인코더에서는 학습시간만 추가적으로 길어질 뿐, 정밀도나 재현율이 나아지진 않았다.

표 2 2 Layer 오토인코더 및 Random Forest의 분류 시간
Table 2 Random Forest classification time with 2 Layer Autoencoder (단위 : 초)

Num of Tree	Autoencoder				Original Data
	Tanh 20-10	Tanh 10-5	Sig 20-10	Sig 10-5	
5	0.027	0.025	0.026	0.026	0.037
10	0.057	0.049	0.053	0.051	0.063
30	0.155	0.141	0.151	0.149	0.180
50	0.259	0.252	0.247	0.247	0.292
100	0.516	0.467	0.487	0.491	0.601
300	1.562	1.465	1.485	1.488	1.780
500	2.651	2.443	2.537	2.535	3.024
1000	5.356	4.943	5.117	5.127	5.961
5000	27.21	25.01	26.00	26.13	29.82

오토인코더를 전처리기로 사용한 방법의 정밀도 및 재현율 향상을 보여주기 위하여 주성분 분석 처리된 데이터에서 주성분 칼럼들만을 3, 5, 10개를 각각 선택하여 Random Forest를 만들어 보았다. 해당 방식은 정밀도가 최고 0.33, 재현율이 최고 0.795으로, 이는 ReLu 10 오토인코더를 사용하였을 경우의 평균 정밀도인 0.79, 평균 재현율인 0.92에 비해 매우 좋지 않은 성능을 보였다. 이를 통하여 적절한 오토인코더를 사용한 Feature engineering 기법은 주성분 분석을 사용한 변수 선택보다 비교적 뛰어난 전처리 기법이라 볼 수 있다.

5. 결 론

본 논문에서는 실시간 스트림으로 들어오는 데이터에 대해 지도 학습의 속도를 개선하고자 인공신경망을 활용한 차원 축소 기법인 오토인코더를 전처리기로 사용하는 방법을 제안하였다. 오토인코더는 데이터의 변수들 간의 비선형적인 관계까지 포착할 수 있고, 주성분 분석과의 조합 및 오토인코더의 반복적인 적용도 가능한 이점이 있다. 이상 거래 탐지를 위하여 오토인코더로 신용카드 결제 데이터의 변수들에 대해 차원 축소를 시행한 후, 대표적인 지도 학습 중 하나인 Random Forest로 분류를 시도하였는데, 정밀도와 재현율의 큰 소실 없이 학습 속도를 확연히 올릴 수 있었다. 또한 새로운 인스턴스의 분류 속도를 약 10% 정도 빠르게 할 수 있었다.

차후 계획으로는 그림 4에서 보인 Random Forest 모델을 실제로 CQL로 변환하여 DSMS 내부에서의 탐지를 구현해보고자 한다. 본 논문에서는 이상 거래 탐지를 위한 기법으로 Random Forest를 사용하였지만, SVM이나 로지스틱 회귀, 인공신경망 등 기타 여러 데이터 마이닝 기법들과 오토인코더의 조합도 시도해보고자 한다. 더 나아가 그림 4에서는 입력으로 들어오는 스

트림 데이터가 DSMS 외부의 오토인코더에서 인코딩되는 방식을 택하였지만, CQL로 변환된 오토인코더 모델을 직접 DSMS 내부에 삽입함으로써 전처리 및 분류의 모든 과정을 DSMS에서도 해결하는 방법도 시도해보고자 한다.

References

[1] M. H. Ur Rehman, C. S. Liew, A. Abbas, P. P. Jayaraman, T. Y. Wah, and S. U. Khan, "Big Data Reduction Methods: A Survey," *Data Science and Engineering*, Vol. 1, No. 4, pp. 265-284, Dec. 2016.

[2] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data Mining for Credit Card Fraud: A Comparative Study," *Decision Support Systems*, Vol. 50, No. 3, pp. 602-613, Feb. 2011.

[3] C. Whitrow, D. J. Hand, P. Juszczak, D. Weston, and N. M. Adams, "Transaction Aggregation as a Strategy for Credit Card Fraud Detection," *Data Mining and Knowledge Discovery*, Vol. 18, No. 1, pp. 30-55, Jul. 2009.

[4] H. Shao, H. Zhao, and G. R. Chang, "Applying Data Mining to Detect Fraud Behavior in Customs Declaration," *International Conference on Machine Learning and Cybernetics*, Vol. 3, pp. 1241-1244, Nov. 2002.

[5] L. Breiman, *Machine Learning*, Vol. 45. No. 1, pp. 5-32, Springer Press, Berlin, 2001.

[6] A. Verikas, A. Gelzinis, and M. Bacauskiene, "Mining Data with Random Forests: A Survey and Results of New Tests," *Pattern Recognition*, Vol. 44, No. 2, pp. 330-349, Feb. 2011.

[7] R. Patidar, and L. Sharma, "Credit Card Fraud Detection using Neural Network," *International Journal of Soft Computing and Engineering (IJSCE)*, Vol. 1, pp. 32-38, Jun. 2011.

[8] H. Abdi, and L. J. Williams, "Principal Component Analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, Vol. 2, No. 4, pp. 433-459, Jul. 2011.

[9] G. E. Hinton, and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, Vol. 313, No. 5786, pp. 504-507, Jul. 2006.

[10] M. S. Nixon, and A. S. Aguado, *Feature Extraction & Image Processing for Computer Vision*, 3rd Ed., pp. 235-242, Academic Press, Cambridge, 2012.

[11] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating Probability with Undersampling for Unbalanced Classification," *2015 IEEE Symposium Series on Computational Intelligence*, pp. 159-166, Dec. 2015.

[12] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, and M. Kudlur, "Tensorflow: A System for Large-scale Machine Learning," *Operating Systems Design and Implementation(OSDI)*, Vol. 16, pp. 265-283, Nov. 2016.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. WSeiss, V. Dubourg, and J. Vanderplas, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, Vol. 12, pp. 2825-2830, Oct. 2011.

[14] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus, "Regularization of Neural Networks using Dropconnect," *International Conference on Machine Learning*, Vol. 28, No. 3, pp. 1058-1066, Feb. 2013.

이 용 현

2015년 성균관대학교 컴퓨터공학부 학사 2015년~현재 서울대학교 컴퓨터공학부 석박사통합과정 재학 중. 관심분야는 데이터 마이닝, 빅데이터, 데이터베이스

구 해 모

2014년 서울대학교 컴퓨터공학부 학사 2014년~현재 서울대학교 컴퓨터공학부 석박사통합과정 재학 중. 관심분야는 데이터베이스, 빅데이터, 데이터 마이닝

김 형 주

1982년 서울대학교 전산학과 학사. 1985년 Univ. of Texas at Austin 석사. 1988년 Univ. of Texas at Austin 박사. 1988년~1990년 Georgia Institute of Technology 부교수. 1991년~현재 서울대학교 컴퓨터공학부 교수. 관심분야는 데이터베이스, XML, 시맨틱웹, 빅데이터