동형암호를 이용한 금융 마이데이터의 개인정보 보호방안 관련 연구

- P2P 대출 시나리오를 기반으로 한 실증 분석 -

최현민*·곽문수**·김애지**·윤규환***

* **네이버클라우드 Security Dev1

*** 네이버클라우드 Digital Convergence Product Planning, 리더

요 약

2020년 신용정보의 이용 및 보호에 관한 법률(이하 신용정보법)의 본인신용정보관리업(이하 마이데이터 사업) 에 관련된 규정이 신설된 이후, 사용자들은 은행, 증권사 등에 흩어진 개인의 신용 정보들을 한 곳에 모아 카드 사용 내역, 주식 계좌 관리, 대출 금리 비교 등 다양한 서비스를 하나의 플랫폼에서 한 번에 확인할 수 있다.

하지만 2021년 12월에 금융마이데이터 서비스에서 마이데이터 API로 수집한 데이터가 불특정 다수에게 노출이 된 사건이 있었으며[1] 이 외에도 개인정보유출에 관한 여러 사건 및 사고가 일어났다. 따라서, 마이데이터 사업의 확장을 위해 마이데이터 사업자에게 전달되는 개인 정보를 안전하게 보호하는 방법에 대한 연구가 필요하다.

최근, 암호화된 상태에서 연산이 수행 가능한 동형암호 기술이 주목을 받고있다. 서비스 제공업체에서는 개인정보가 들어있는 평문 데이터 없이도 연산을 수행할 수 있기 때문에 개인정보유출을 막을 수 있다. 본 논문에서는 이러한 동형암호의 기술적인 특성을 분석한 뒤 동형암호를 적용한 Linear SVM 모델에서 P2P 대출 데이터의 채무불이행 여부 추론 실험을 통해 성능을 측정하고 금융 마이데이터에 동형암호를 안전하게 적용하기 위한 방법을 제안한다.

키워드

동형암호, 마이데이터, 개인정보 보호, 프라이버시 보존 머신러닝, 동형암호 기반 머신러닝, P2P대출

목 차

I. 서론

유럽연합의 General Data Protection Regulation 2016, 이하 GDPR 에서 개인정보 이동권(Data portability) 이 신설된 이후 우리나라에서는 2020년 신용정보법 의 개정을 통해 “개인신용정보의 전송요구”과 “본인신용정보관리업”에 관한 규정이 신설되었다. 개인신용정보의 전송요구는 “정보주체(이하 고객)가 본인 데이터에 대한 전송을 요청하면, 개인정보처리자는 보유한 데이터를 고객(요청자) 또는 고객이 지정한 제3자에게 전송하는 정보주체의 권리”인 GDPR의 “개인정보 이동권”에 해당하며 본인신용정보관리업 이하 “마이데이터 사업”은 정보주체의 신용관리를 지원하기 위하여 신용정보법에서 정의하는 신용정보(신용정보의 이용 및 보호에 관한 법률 시행령 제 2조)를 대통령령으로 정하는 방식으로 통합하여 그 신용정보주체에게 제공하는 행위를 영업으로 하는 것을 말한다.

신용정보법 제 22조의 9의 4에 의거하여 정보제공자는 고객이 본인에 관한 개인신용정보를 마이데이터사업자에게 전송할 것을 요구한 경우 API 방식으로 해당 고객의 개인신용정보를 해당 본인신용정보관리회사에 직접 전송하여야 한다.

마이데이터사업은 본인신용정보관리회사(이하 마이데이터사업자)에게 고객의 흩어진 개인신용정보 데이터를 한 곳에 모아서 단순히 정보를 제공하는 것 뿐만 아니라 데이터의 다양한 분석을 통해 자산관리, 지출분석 및 소비패턴을 고려한 금융상품 추천 등 고객에게 다양하고 편리한 서비스를 제공할 수 있다. 하지만 마이데이터사업자에서 개인정보 유출 및 해킹 등 침해사고가 일어날 경우 큰 피해가 생길 수 있다.[1] 또한, 마이데이터사업자가 제공하는 서비스에는 주문 내역 정보 등과 같이 소비자의 사생활과 관련된 데이터들을 사용할 수 있기 때문에 고객의 개인신용정보의 프라이버시 유출에 대한 이슈가 생길 수 있다.

따라서, 본 논문에서는 최신 암호화 기법인 동형암호 기술을 마이데이터 서비스에 적용하여 개인정보 유출에 대한 위험을 막고 이에 따라 더욱 다양한 개인정보의 활용이 가능하도록 기술적인 부분을 살펴볼 것이다. 동형암호(Homomorphic encryption, HE)란 암호화된 데이터간에 복호화 없이 동형연산이 가능하도록 하는 암호화 기법을 의미한다. 동형암호 기법을 적용하면 암호화된 상태에서 평균, 분산, 사분위수 구하기 등 기본적인 통계연산 뿐만

아니라 로지스틱 회귀 모델 등 머신러닝 기술을 적용할 수 있어서 보안을 유지한 채 다양한 데이터 분석이 가능해진다.

하지만 연산을 위한 연산키와 데이터 암호화를 위한 암호화키를 공유해야 하기때문에 안전하게 키 공유, 관리를 위한 아키텍처가 요구된다. 따라서, 본 연구에서는 앞서 말한 마이데이터 사업에서 사용자의 개인정보를 보호하기 위해 동형암호 기술을 분석하고 API에 동형암호를 적용하는 시나리오를 제안한다.

본 연구의 기여는 다음 3가지이다.

- 동형암호의 이론적인 분석과 머신러닝 기반 연구 동향을 살펴보고, 동형암호를 기반한 라이브러리 및 서비스 등을 살펴봄으로써 동형암호의 기술적인 특징과 동향을 조사
- P2P 대출심사 결과 데이터로 학습한 Linear SVM 모델을 만들어 동형암호화된 테스트 데이터로 채무불이행 여부를 추론하는 형태의 구체적인 실험을 통해 동형암호의 성능 측정
- 금융마이데이터에 동형암호를 안전하게 적용하기 위한 키, 데이터 공유 방법을 제안

Ⅱ. 마이데이터의 의의와 정책이 가져올 효과

1. 마이데이터의 의의

마이데이터산업이란 신용정보법 제 2의 9의 2에 의거하여 고객의 전송요구권 행사에 따라 분산되어 있는 개인신용 정보를 제공받아 해당 고객에게 통합조회 서비스를 제공하는 산업을 의미한다. 마이데이터 산업은 신용정보법 제4조에 의거하여 개인신용정보를 대량 집적하는 산업 특성상 엄격한 보안체계를 갖추도록 하고, 고객을 이해상충으로부터 보호하는 절차 등이 필요하여 허가산업으로 운영해야 한다. 허가신청은 금융위원회에서 매달 접수받고 있으며 2022년 7월 28일 기준 마이데이터 본허가 업체는 총 59개사 이고, 예비허가는 총 8개사 이다. [2]

2. 금융분야 마이데이터 정책이 가져올 효과

현재 은행이나 증권, 카드사 등에서 흩어져 있는 개인신용정보들은 한 곳에서 모아서 보기 어려운 상황이다. 고객은 자산 관리 등을 하기 위해서는 직접 은행에 발품을 팔아 정보들을 모아서 분석해야 한다. 일부 핀테크 업체 등에서 고객의 신용정보를 수집하고 분석하는 서비스를 제공하고 있지만 이는 “스크린 스크래이핑“ 방식을 이용하여 고객이 입력한 개별 금융기관의 아이디와 비밀번호, 공인인증서 등을 통해 해당 업체에서 대리인증을 받는 형식으로 진행된다. “스크린 스크래이핑“ 방식을 사용하면 고객의 로그인 정보등이 탈취당할 가능성이 있고, 효율적으로 고객의 신용정보를 수집하기 어렵다. 따라서, 마이데이터 정책이 활성화된다면 고객은 기존보다 더 다양한 금융 서비스들을 제공받을 기회가 제공된다.

3. 마이데이터 API 사용

금융 마이데이터는 2022년 1월 5일 부터 스크린 스크래이핑 방식이 전면 금지되고 마이데이터 사업자는 모든 이용자에게 API 방식으로만 서비스를 제공할 수 있다. [3]

마이데이터 API로 전달될 수 있는 데이터는 신용정보법 제33조의2(개인신용정보의 전송요구)에 의거하여 다음과 같다.신용정보법 시행령 제28조의3(개인신용정보의 전송요구) 의 6에 해당하는 아래의 정보들을 포함한다.

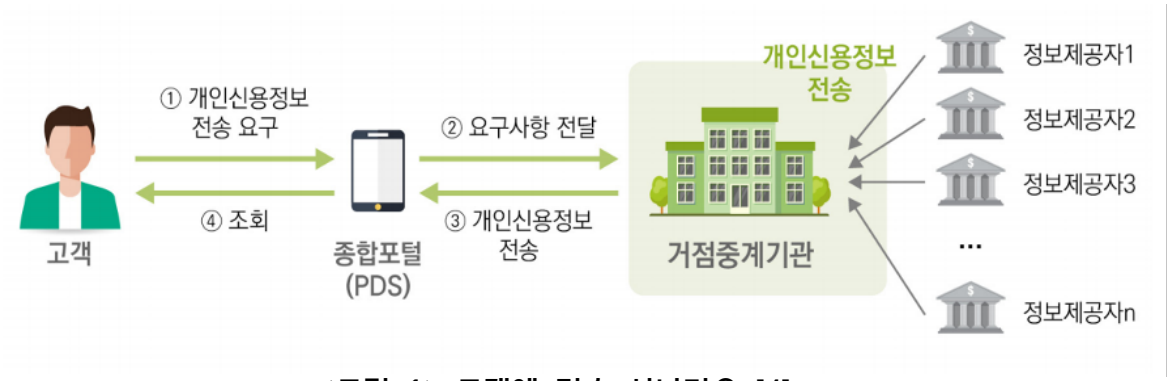
- 여수신정보
- 보험정보
- 카드정보
- 금융투자정보
- 개인형IRP 정보
- 통신업 정보
- 보증보험 정보
- 공공정보

4. 개인신용정보 전송 유형

금융분야 마이데이터 기술 가이드라인에 의하면 개인신용정보 전송 유형은 아래와 같이 크게 3가지로 나눌 수 있다. [4]

(1) 고객에 전송

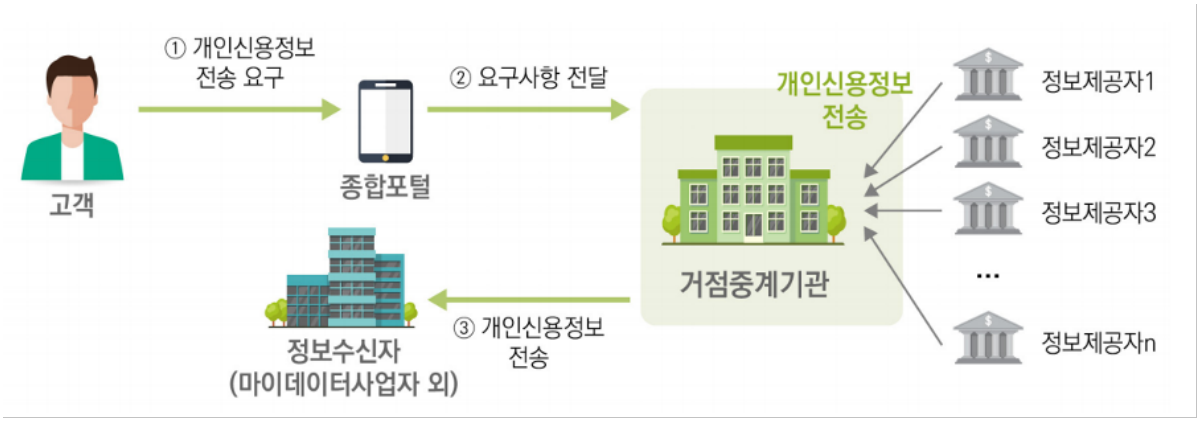
고객이 종합포털을 통해 정보제공자에 고객 본인에게 개인신용정보를 전송할 것을 요구하면, 정보제공자는 고객이 해당 개인신용정보를 조회·활용할 수 있도록 거점중계기관을 통해 고객의 PDS에 개인신용정보를 전송한다.



<그림 1> 고객에 전송 시나리오 [4]

(2) 마이데이터사업자 외 기관에 전송

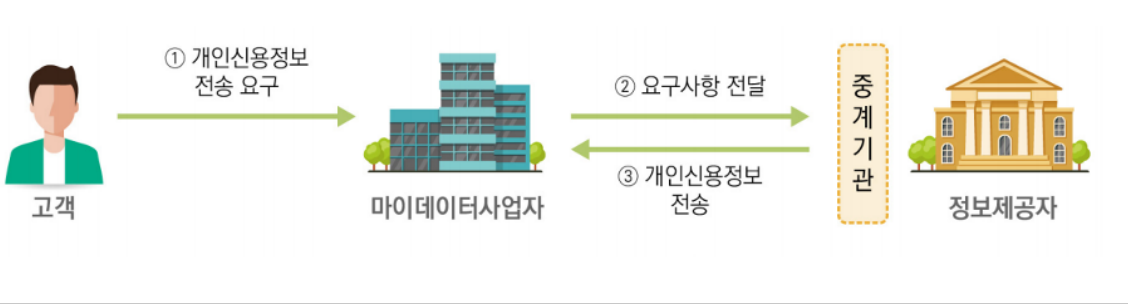
고객이 종합포털을 통해 마이데이터사업자 외 기관에 개인신용정보를 전송할 것을 요구하면, 정보제공자는 거점중계기관을 통해 개인신용정보를 전송한다.



<그림 2> 마이데이터사업자 외 전송 시나리오 [4]

(3) 마이데이터사업자에 전송

고객이 마이데이터사업자에게 개인신용정보를 전송할 것을 요구하면, 해당 정보제공자가 마이데이터사업자에게 API를 이용하여 개인신용정보를 전송한다.



<그림 3> 마이데이터사업자에 전송 시나리오 [4]

이 중, 고객의 자신의 정보를 조회하는 성격을 띄는 (1)을 제외하면 (2)와 (3)의 경우 고객이 아닌 정보수신자가 고객의 정보를 평문 그대로 전달받을 수 있다.

Ⅲ. 동형암호와 머신러닝

1. 동형암호의 소개

최근 머신러닝 및 딥러닝 기술의 발달과 스마트폰 보급으로 인하여 사용자들이 개인의 디바이스에서 사용 가능한 다양한 머신러닝 서비스들이 생겨나고 있다. 그러나 머신러닝 및 딥러닝 분석 기법은 학습 및 추론 과정에서 많은 양의 컴퓨팅 파워 및 메모리 등이 필요하기 때문에 일반적으로 사용자의 디바이스에서 수행되기 보다는 데이터센터의 서버나 클라우드 환경에서 수행되는 사례가 늘고 있다.

동형암호는 암호화된 상태에서 복호화 없이 데이터의 연산이 가능한 암호 알고리즘 또는 기술을 의미한다. 기존 공개키 암호 기술에서 평문을 연산하기 위해서는 반드시 암호문을 복호화를 수행해서 얻은 평문으로 연산을 수행해야 한다. 만일, 평문 데이터 소유주와 연산 주체가 서로 다른 경우 복호화 키의 노출 및 평문 데이터 노출 등으로 인한 데이터 프라이버시의 유출의 위험성이 있지만 동형암호를 사용하면 프라이버시 유출에 대한 걱정 없이 데이터의 연산 등이 가능하다.

여기서 말하는 데이터의 연산은 데이터가 숫자인 경우 덧셈과 곱셈, 반전 등을 의미하는데, 이러한 연산이 가능하다는 것은 다양한 통계연산 및 머신러닝 연산이 가능하다는 것을 의미한다. 이러한 동형암호의 특성으로 인해 동형암호화된 상태에서 머신러닝을 수행하기 위한 연구들이 활발히 진행되고 있다.

2. 동형암호의 역사

1978년 Rivest, Addelman, Dertouzous에 암호화된 상태에서 연산 수행이 가능한 동형암호의 개념이 최초로 제안된 이후, 꾸준히 동형암호에 대한 연구가 지속되었다. 그러나, 동형암호의 곱셈이 한계치를 넘어가게 되면 복호화한 값의 결과가 실제 결과와 일치하치 않게 되어 상용화에는 무리가 있었다.

이러던 중, 2009년 스탠포드 대학교의 박사과정 학생인 Craig Gentry 의 졸업논문에 횟수 제한 없이 곱셈 수행이 가능한 방법 [5]을 제시했지만, 실제 한 비트의 연산을 수행하기 위해 수 시간이 걸렸기 때문에 상용화에는 무리가 있었다. 2011년에는 기존의 비트수준의 연산에서 정수의 연산을 제공하는 BFV, BGV 등의 동형암호 알고리즘이 제안되었다. 2013년에는 작은 데이터 처리에 효율적인 CGGI 알고리즘이 제안되었고, 2016년에는 최초로 실수 연산이 가능한 CKKS [6] 알고리즘이 제안되었다. 머신러닝 및 통계 분석을 위해서는 실수 연산이 필수적이기 때문에 CKKS 알고리즘의 소개된 이후 동형암호를 머신러닝 및 통계 분석에 적용하고자 하는 다양한 연구가 활발히 진행되고 있다. [7,8,9,10]

3. 동형암호의 정의 및 용어 정리

동형암호는 다음과 같은 알고리즘으로 구성된다.

- $KeyGen(1^{\lambda})$ (키생성) : 안전성 파라미터 λ와 복호화키 sk, 암호화키 pk, 연산키 evk를 생성
- $Enc_{pk}(m)$ (암호화) : 암호화할 평문 m과 암호화키 pk를 통해 암호문 c를 생성
- $Dnc_{sk}(c)$ (복호화) : 복호화할 암호문 c와 복호화키 sk를 통해 복호화된

평문 m^{dec}를 생성

- $Eval_{evk}(f,c_1,c_2,\cdots,c_N)$ (연산) : 암호문 $c_1,c_2,\cdots,c_N$와 연산키 evk, 함수 f를 통해 연산이 완료된 암호문 c^{eval} 를 생성

일반적으로 동형암호는 공개키 알고리즘으로 정의되며 키 생성 알고리즘에서 생성되는 세 개의 키 중 복호화키는 비밀키, 나머지 암호화키와 연산키는 공개키이다.

<표 1> 동형암호 키의 종류별 용도와 공개 가능 여부

	암호화키	연산키	복호화키
용도	암호화 수행	연산 수행	복호화 수행
공개 가능 여부	가능	가능	불가능

안전성 파라미터는 암호화 알고리즘에서 제공하기 위한 안전성의 기준이 되는 파라미터를 의미하며 예를 들어 안전성 파라미터 λ가 128이라는 의미는 해당 암호 알고리즘을 공격하기 위하여 2^{128}번의 계산을 해야만 해당 암호 알고리즘의 취약성을 얻을 수 있다는 의미이다.

- 재부팅(Bootstrapping) : 대부분의 동형암호 알고리즘에서 곱셈 연산을 수행할 때, 동형암호의 안전성을 보장하기 위해 정의된 잡음(noise) 값이 커지게 되어, 한계치 이상 넘어가면 연산을 수행한 암호문을 복호화한 값이 실제 평문을 연산한 결과와 달라지게 된다. 따라서, 동형암호의 곱셈 횟수와 관계 없이 복호화를 잘 수행하기 위해 잡음을 줄이는 과정을 재부팅(Bootstrapping) 이라 한다.
- 제한 동형암호(Somewhat homomorphic encryption) : 가능한 연산의 횟수가 제한된 동형암호를 의미한다.
- 완전 동형암호(Fully homomorphic encryption) : 가능한 연산의 횟수의 제한이 없는 동형암호를 의미한다.
- 근삿값 동형암호(Approximate number homomorphic encryption) : 부동소숫점 연산이 가능한 동형암호를 의미한다.

4. 머신러닝 소개

머신러닝(기계학습)은 컴퓨터가 스스로 원본 데이터에서 패턴 등 지식을 학습하는 능력을 의미한다. 머신러닝은 크게 학습(training)과 추론(inference) 과정으로 나뉜다. 학습은 데이터를 입력으로 받아 예측 모형을 산출하는 과정이며, 추론은 얻어진 예측 모형에 데이터를 입력하여 결괏값을 얻어내는 과정이다. 머신러닝은 학습 방법에 따라 지도학습(supervised learning)과 비지도학습(unsupervised learning), 강화학습(reinforcement learning) 으로 크게 세 가지로 분류할 수 있다.

지도학습은 레이블(lable)이 있는 형태의 데이터를 통해 학습하는 방법이다. 여기서 레이블은 일종의 정답지로 볼 수 있는데, 예를 들어 사자와 호랑이를 구분하는 문제를 생각하면 레이블은 사자 또는 호랑이가 적힌 값으로 생각할 수 있다. 지도학습에는 대표적으로 회귀(regression)과 분류(classification) 기법이 있다. 비지도학습은 레이블이 없는 형태의 데이터를 통해 학습하는 방법이다. 보통 데이터에서 성질이 유사한 부분을 분류하거나 데이터 사이의 관계에 대한 특징을 추출해낸다. 대표적으로 클러스터링(clustering) 기법이 많이 쓰인다. 강화학습은 컴퓨터가 직접 시행착오를 통해 문제를 해결하는 방법을 학습하는 방법이다. 강화학습은 주로 컴퓨터 게임 문제를 풀거나 장기, 바둑 등을 해결하는 모델을 생성하는데 많이 사용된다.

5. 동형암호 기반 머신러닝

빅데이터 분석 및 머신러닝 기술의 발달로 인해 데이터의 프라이버시를 보호하면서 데이터를 분석하는 기술들의 필요성이 증대되고 있고, 대표적으로 동형암호 뿐만 아니라 차분 분석, 안전한 다자간 계산, 연합 학습 등이 있다. 하지만, 차분 분석의 경우 데이터에 노이즈가 더해지는 형태이기 때문에 프라이버시 보존이 보장될수록 데이터의 변형이 이루어져 결과가 정확하게 얻어지지 않을 수 있지만, 동형암호는 데이터 자체를 암호화하여 무시할 만한 오차를 제외하고 연산을 수행할 수 있다.

안전한 다자간 연산은 참가자들이 많아질수록 필요한 네트워크 트래픽 등이 많이 커지며 연합학습의 경우 Non-IID(Non-Independent and Identical Data) 문제 등을 해결해야 하는 이슈들이 있지만, 동형암호를 적용하면 암호화된 데이터 자체를 다루기 때문에 이러한 방법을 사용하지 않고도 안전하게 데이터를 수집 및 활용할 수 있다.

하지만, 동형암호는 다른 암호화 기법(대표적인 공개키 암호인 RSA)와 비교했을 때 키 사이즈 및 암호문 사이즈가 현저히 크며, 암호문의 연산 속도가 평문의 연산 속도가 최소 수십 배에서 수백 배 이상 차이가 날 수 있다. 그리고, 머신러닝에서 가장 많이 쓰이는 대표적인 활성화 함수 중 하나인 ReLU의 경우 평문 상태에서는 구현하기 쉽지만 동형암호화된 상태에서는 구현이 어렵기 때문에 단순히 머신러닝 모델에 동형암호를 접목하는 것이 아닌, 동형암호에 최적화된 머신러닝 모델에 대한 연구가 추가적으로 필요하다.

Ⅳ. 동형암호 기반 P2P 대출 데이터의 채무불이행 여부 추론실험

최근 각광받고 있는 P2P 대출(peer-to-peer lending, P2P Lending)은 온라인 상에서 채권자와 채무자를 연결해주는 대출 서비스이다. 전통적인 금융 서비스와는 달리 P2P 서비스는 온라인 방식으로 운영비를 줄일 수 있다. 이러한 장점으로 인해 P2P 대출을 통해 채권자는 은행 예금에 비해 비교적 높은 수익률을 낼 수 있고, 채무자들은 낮은 금리로 대출을 사용할 수 있다. 하지만, P2P 투자는 예금자보호법이 적용되지 않기 때문에 채권자 입장에서는 리스크가 낮은 투자를 선호할 수 밖에 없다. 따라서, P2P 대출 서비스는 채권자와 채무자를 연결 시켜주고, 채무자의 신용등급 및 채무불이행 가능성 등을 판단해주는 대가로 이익을 얻을 수 있다.

해외의 대표적인 P2P 대출 서비스 회사는 LendingClub[14] 이 있으며 국내에는 렌딧[15], 에잇퍼센트[16], 피플펀드[17] 등이 있다.

최근, HN핀코어에서 P2P 대출 업계 최초로 마이데이터 본 허가를 받았으며[21], 피플펀드는 예비허가를 받았다. [18] 이를 통해 알 수 있듯이, P2P 대출 업계에서는 마이데이터를 통해 채무자의 정보를 활용하여 심사평가모형을 인공지능 등을 활용하여 개발 및 활용하기 위한 준비를 하고 있다.

1. 실험 목적

본 논문에서는 P2P 대출 정보를 기반으로 학습한 머신러닝 모델과 동형암호를 활용하여 암호화된 상태에서 채무불이행 가능성을 예측할 수 있는 환경을 구현하고, 추론 성능과 데이터 사이즈를 기반으로 상용 가능성을 증명한다.

2. 실험 계획

다음과 같은 방식으로 실험을 수행하고, 동형암호화 추론 과정에서 각 단계 별 시간을 측정하여 실제 서비스에서 동형암호 기반 Linear SVM 추론이 합리적으로 활용 가능함을 증명한다. Linear SVM 알고리즘은 예측 정확성이 높고, 신경망 알고리즘에 비해 과잉적합(overfitting)에 강한 알고리즘이다.[19] 따라서, 본 논문에서는 SVM중 Linear 커널을 사용한 Linear SVM을 사용한 동형암호 추론 알고리즘을 구현하였다.
실험 내용을 간단히 정리하면 다음과 같다.

[데이터 전처리 및 가공]

P2P 대출거래 내역을 학습용 데이터와 추론용 데이터로 분리한 뒤, 데이터 분석 기법으로 가공하여 전처리한다.

[평문 모델 생성]

전처리한 데이터를 기반으로 학습용 데이터를 사용하여 Linear SVM 학습을 통해 평문 형태의 모델을 얻는다.

[동형암호화 추론]

동형암호화된 추론용 데이터를 위에서 얻은 평문 모델을 기반으로 Linear SVM 추론을 수행한다.

[추론결과 복호화 및 분석]

실험의 Correctness를 증명하기 위해 평문 상태에서 sklearn 라이브러리[11] 를 통해 Linear SVM 추론한 결과와 동형암호화 추론 결과를 복호화한 값이 서로 일치함을 보인다.

연구에 사용할 P2P 대출거래 정보를 얻기 위하여 kaggle [12] 에서 csv 형태를 gzip으로 제공하는 데이터를 사용할 예정이다. 해당 데이터는 세계적인 P2P 대출업체인 Lending Club에서 2007년부터 고객들의 채무상태와 대출 거래 정보 등을 포함하여 연도별로 제공하는 데이터를 연도별로 연결하였고, 151개의 속성으로 구성된 데이터 중 결측치가 90%이상인 9개의 속성을 제거하여 총 142개의 속성과 2925492개의 열로 구성된 데이터이다. 이 중, 우리는 2018년부터 2020년까지 최근 3개년 데이터를 사용한다.

대출거래 데이터에서 채무상태를 나타내는 'loan_status' 속성은 Fully Paid (상환완료)와 Charged off(공제), Default (채무불이행) 속성만 사용하고 나머지 속성의 데이터는 삭제하며, 해당 속성은 상환 능력이 있거나 없음을 나타내기 위해서 Charged off 를 Default 로 대체한다.

결측치 처리와 범주형 속성의 실수화, 불필요한 속성 삭제 등을 통해 최종적으로 10개 속성의 164216개의 데이터를 추출했다. 10개의 속성은 다음과 같다.

<표 2> 실험에 사용된 전처리된 데이터의 속성별 설명

속성	설명
int_rate	이자율
emp_length	고용기간, 1~10 범위이며 10년 이상은 10으로 나타냄
home_ownership	주택 소유 상태. RENT, OWN, MORTGAGE, OTHER 로 구성됨
annual_inc	대출자의 연간 소득
dti	주택담보대출과 요청받은 LC대출을 제외한 총 부채의무에 대한 대출자의 총 월 부채 상환액을 대출자의 자체 신고한 월 소득으로 나눈 비율
avg_cur_bal	모든 계좌의 평균 경상수지
mo_sin_old_rev_tl_op	가장 오래된 리볼빙 계정이 열린 후 개월 수
num_actv_rev_tl	현재 활성화된 거래 수
year	해당 자료의 년도 (2007~2020)
monthly_load	할부로 나가는 수입의 백분율. installment 와 annual_inc 을 통해 새로 얻은 속성

가공된 데이터의 80%는 학습용, 20%는 테스트용으로 분리한 뒤, sklearn 라이브러리의 svm. LinearSVC 모듈을 사용하여 선형 기저벡터 머신의 학습을 수행한 후, 평문 모델을 얻는다. 학습 데이터로 학습을 통해 평문 모델을 추출하고, 실제 평문 상태에서 테스트 데이터로 추론한 결과로 얻은 혼돈행

렬(Confusion matrix)은 다음과 같다.

<표 3> 채무불이행 데이터를 LinearSVM으로 추론한 성능 측정 결과

	Precision	recall	f1-score	support
0	0.84	0.65	0.73	24786
1	0.36	-	0.46	8058
accuracy	-	-	0.64	32844
macro avg	0.60	0.63	0.59	32844
weighted avg	0.72	0.64	0.67	32844

3. 실험 수행

(1)에서 얻은 테스트 데이터를 동형암호화 한 후, (2)에서 얻은 평문 모델을 통해 동형암호 기반 추론을 수행한다. 이때, 동형암호에 사용된 키/암호문 사이즈 및 추론 속도 등을 측정하여 실제 서비스에 적용하기 적합함을 증명한다. 그리고, 동형암호의 Correctness를 확인하기 위해 평문상태의 추론 결과와 동형암호 추론의 복호화 결과가 서로 일치함을 확인한다.

테스트를 위해 동형암호 라이브러리는 Microsoft사의 SEAL[13]을 사용한다. 본 테스트에 사용한 SEAL 버전은 4.0이다. 테스트 서버의 스펙은 NaverCloud의 Server 상품군 중 [CPU-Intensive] 를 사용하였으며, RAM은 64GB, 코어는 3.0GHz의 vCPU 32개로 구성되어 있다. LinearSVM 모델은 coefficient가 10개의 원소의 벡터로 구성되어 있고, 1개의 원소의 intercept로 구성되어 있다. 테스트를 위해 SEAL의 CKKS 알고리즘을 사용하고, 파라미터 구성은 다음과 같다.

$$\text{poly\_modulus\_degree} = 16{,}384$$

$$\text{scale} = 2^{30}$$

slot_size 는 8,192인데 이는 한 암호문에 최대 8,192개의 실수 값을 암호화할 수 있다는 의미이다. 암호화는 일반적으로 python의 기본 array나 numpy 라이브러리의 array, pandas 라이브러리의 DataFrame 형태의 데이터를 암호화할 수 있다. 예를들어, 고객이 10, 20, 30.5, 40, -62.5 라는 5개의 데이터를 암호화하기 위해서는 일반적으로 5개의 값을 하나의 벡터로 만든 후, 그 벡

터를 암호화하게 된다. 예를들면 다음과 같다.

plaintext_data = [10, 20, 30.5, 40, -62.5]

위의 plaintext_data를 암호화하면 암호화된 데이터의 크기는 8,192개가 되며 해당 암호문을 복호화하여 확인해보면 늘어난 길이 만큼의 데이터는 0으로 채워지게 된다. 즉, 복호화 하면 크기가 8,192의 아래와 같은 벡터가 나오게 된다.

decrypted_data = [10.0, 20.0, 30.5, 40.0, -62.5, 0, 0, ... , 0]

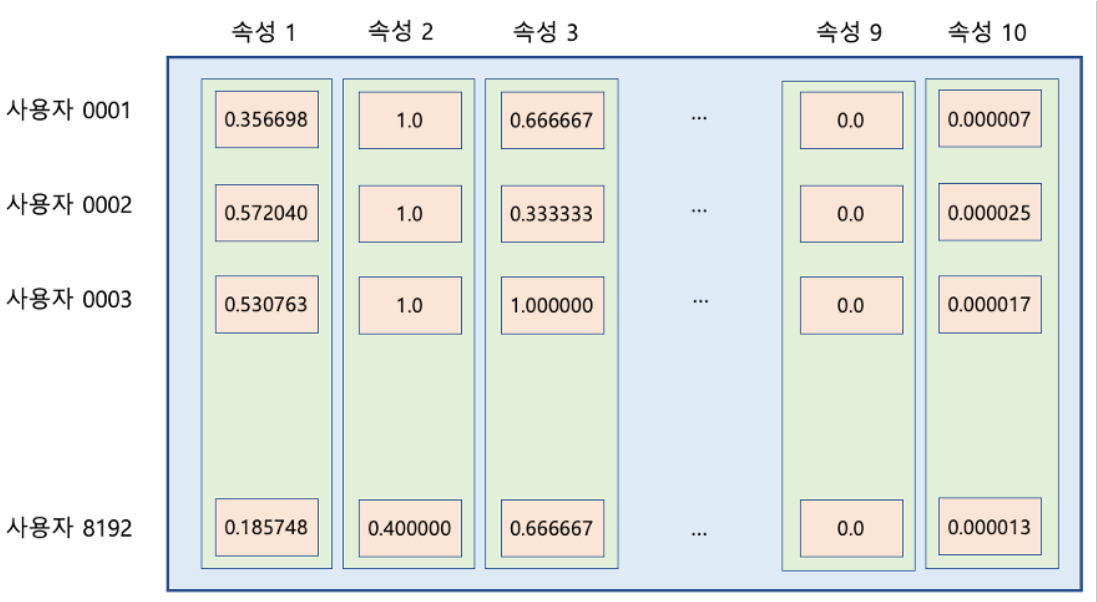
따라서, 연산을 수행하면 slot_size인 8,192개 이내의 데이터의 연산 속도는 동일하다.

본 실험에서 전처리된 클라이언트의 데이터는 10개의 속성으로 구성되는데, 최대의 효율을 내기 위해서는 slot_size 개수 만큼의 데이터를 한 번에 연산으로 사용하는게 가장 효율적이다. 이를 나타내면 다음과 같다.

<그림 4> 8,192명의 사용자에 대한 데이터의 예시

이를, 각 속성 별로 벡터화를 수행하면 크기가 8,192인 평문 벡터가 속성

개수인 10개 만큼 생성된다. 일반적으로, 효율적인 암호화를 위해 암호화할 데이터의 형태를 변경하는 작업을Packing이라 부른다.



<그림 5> 암호화를 수행하기 위한 데이터의 Packing 작업

이제, 각 속성의 벡터를 각각 암호화한 후, 평문 모델의 coefficient와 각각 InnerProduct를 수행한 후, intercept를 더한다. 실제로, 테스트데이터 8,192개의 데이터를 평문 상태에서 평문 모델로 추론한 값과 동형암호화된 상태에서 추론한 값을 복호화한 값이 서로 일치함을 확인하였다. 위 과정에서 8,192개의 데이터에 대한 연산에 걸린 시간을 측정하면 다음과 같다.

<표 4> 각 연산 과정에서의 시간

연산	수행 시간(단위 : 초)
Packing & 암호화	0.114781
추론	0.028898
복호화	0.002913
평가	0.005444

위 과정에서 사용된 키와 암호문 크기를 측정하면 다음과 같다.

<표 5> 실험에 사용된 평문데이터, 키와 암호문의 크기

객체	크기 (단위 : MB)
평문(csv)	0.679615
암호문	7.86545
Secret key	0.500084
Public key	1.00011
Galois key	78.13346
Relinear Key	3.00038
추론 결과 암호문	0.750108

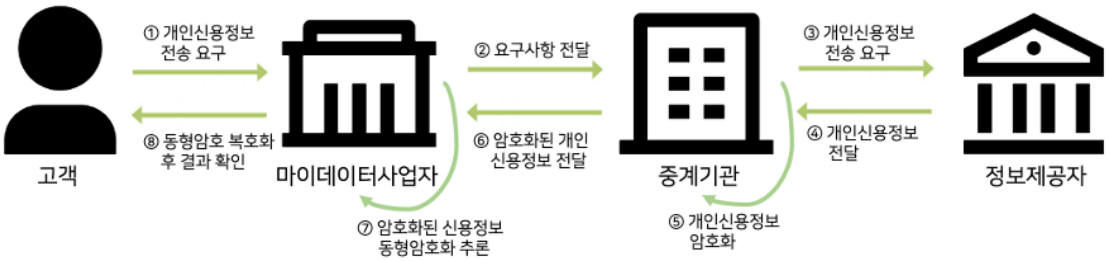
4. 실험 결과 분석

위의 결과를 보면 약 0.68MB 데이터의Packing 과정부터 복호화 과정까지 채0.15초가 걸리지 않는다는 것을 알 수 있다. 암호문 사이즈는 하나의 속성에 대한 평문 데이터를 암호화한 암호문의 크기와 속성 개수의 곱으로 정해지지만 추론결과 암호문은 추론 결과만 암호화된 결과이므로 Linear SVM 알고리즘의 경우 하나의 속성에 대한 평문을 암호화한 암호문의 크기와 동일하다. 동형암호화된 데이터를 제공하는 API를 가정했을 때, 한번의 API 호출 시 위 예제의 Linear SVM 알고리즘의 경우 Public Key와 암호문, 평문에 해당하는 약 9.55 MB의 메모리와 약0.11초의 연산 시간이 필요함을 알 수 있다. 16GB 메모리를 지원하는 서버에서 기본적인 어플리케이션에 사용되는 메모리와 CPU 성능을 고려하지 않고 동시에 약 1715 개의 암호문 생성이 가능함을 알 수 있다. 물론, CNN 등의 딥러닝 모델은 Layer가 깊고 사용해야 하는 연산들이 매우 많기 때문에 단순히 덧셈이나 곱셈 등의 연산을 동형암호 덧셈, 곱셈으로 치환하면 성능이 떨어질 수 있다. 연산의 성능을 높이기 위해 연산에 효율적인 형태로 packing을 적용하고, 활성화 함수 등은 ReLU와 같은 비다항식 함수를 사용하기 보다는 n차 이하의 근사화된 다항식을 사용하는 방법을 고려할 수 있다.[20]

Ⅴ. 마이데이터에 동형암호를 적용하는 시나리오 제안

1. 시나리오 제안

Ⅲ과 Ⅳ의 연구 결과를 통해 마이데이터 서비스에 동형암호를 적용하는 방법에 대한 시나리오를 제안한다. 제안하는 시나리오의 그림과 설명은 다음과 같다.



<그림 6> 동형암호 기반 고객의 개인신용정보의 머신러닝 추론 후 결과 조회 시나리오

본 시나리오는 고객이 생성한 키 쌍(암호화키, 연산키, 복호화키) 중, 공개키인 암호화키와 연산키는 중계기관에 이미 등록되어 있고, 복호화 키는 고객만이 소유하고 있다고 가정한다. 중계기관에서는 고객의 키를 업로드 받을 수 있는 인터페이스를 제공할 필요가 있다.

① 개인의 신용정보 전송 요구

고객은 마이데이터 사업자에 개인의 신용정보에 대한 전송을 요구한다.

② ~ ④ 고객의 요구사항 전달

마이데이터 사업자는 중계기관을 거쳐 정보제공자에게 고객의 개인신용정보를 마이데이터 API를 통해 요청하면, 정보제공자는 응답으로 요청한 고객의 개인신용정보를 전달한다. 이때, 제공하는 고객의 개인신용정보는 평문 형태로 전달된다. 이 과정은 기존의 개인신용정보 전송 유형 (3)과 동일하나, 마

이데이터 API에 동형암호에 필요한 파라미터 등이 옵셔널하게 추가될 수 있다.

⑤ 개인신용정보 암호화

중계기관은 정보제공자로부터 전달받은 고객의 개인신용정보를 고객의 동형암호의 암호화키를 통해 암호화를 수행한다. 이때, 암호화 과정에서 필요한 파라미터 등은 마이데이터 API의 요청시 마이데이터사업자로부터 전달받거나, 기본값으로 설정된 값을 사용하도록 한다.

⑥ 암호화된 개인신용정보 전달

중계기관에서 암호화된 개인신용정보와 고객의 동형암호 연산키를 마이데이터사업자로 전달한다. 이때, 마이데이터 API에 암호문과 연산키를 추가로 전달하거나, 다운로드 받는 링크 등으로 전달할 수 있다.

⑦ 암호화된 신용정보 동형암호화 추론

마이데이터 사업자는 암호화된 고객의 개인신용정보를 자신의 평문 모델과 동형 머신러닝 추론연산을 통해 연산을 수행하여 암호화된 추론 결과를 얻는다.

⑧ 동형암호 복호화 후 결과 확인

고객이 추론 결과를 마이데이터사업자에 요청하면 마이데이터사업자는 암호화된 추론결과를 고객에게 전달한다. 고객은 자신의 복호화키로 추론결과를 복호화하여 추론 평문결과를 얻는다.

2. P2P 대출 시나리오에 적용

위의 시나리오를 활용해 P2P 대출 서비스를 구축할 경우 채무자와 채권자 시나리오에 따른 고려사항은 각각 아래와 같다.

(1) 고객이 채무자인 경우

채무자의 데이터로 대출 가능성과 상한액을 조회할 수 있다. 이 경우, 금융마이데이터에서 마이데이터사업자로 전달되는 데이터는 채무자의 암호화키로 암호화되어 있기 때문에 복호화키를 가지고 있는 채무자가 아닌 대상에 유출이 되도 문제가 없다. 또한, 활용되는 데이터가 채무자의 데이터에

국한되기 때문에 가공없이 채무자에게 전달 가능하다.

(2) 고객이 채권자인 경우

채무자의 계약 불이행 예측 가능성을 조회할 수 있다. 이 경우, 금융마이데이터에서 마이데이터사업자로 전달되는 데이터는 채권자의 암호화키로 암호화되어 있기 때문에 복호화키를 가지고 있는 채권자가 아닌 대상에 유출이 되도 문제가 없는 것은 동일하다. 하지만, 위 케이스와 다르게 활용되는 데이터가 채권자가 아닌 다수의 채무자의 데이터이기 때문에 가공없이 채권자에게 전달될 경우 채무자의 데이터가 유출될 가능성이 있다. 때문에, 본 논문에서 예시로 든 추론 등의 데이터 가공을 통해 원본 데이터를 유추할 수 없는 형태의 데이터만 전달하는 것을 제안한다. 만약, 가공되지 않은 데이터가 유출될 경우에도 복호화키를 가지고 있는 채권자 외에는 영향이 없어 데이터의 유출에 대한 위험성을 최소화할 수 있다.

3 제안된 시나리오 분석

위에서 제안한 시나리오에 대하여 안전성을 살펴보면, 우리가 사용한 CKKS 알고리즘은 공개키 암호로 안전성은 RLWE(Ring Learning With Errors)에 의해 보장된다. 특히, 복호화키는 고객이 직접 생성하고 고객만 가지고 있기 때문에 위의 시나리오에서 고객이 복호화키를 유출하지 않고 잘 관리한다는 가정하에 안전하다.

마이데이터 API에서 제공하는 속성들을 가지고 있는 신용데이터를 직접 얻기 어렵기 때문에 실제 API의 예를 들어 연구를 수행할 수는 없었다. 하지만 Ⅳ의 실험에서 사용한 채무불이행 데이터는 실제 고객의 개인신용정보가 들어있는 데이터로 Kaggle 에서 꾸준히 연구된 데이터이다. 또한 많이 쓰이는 분류기 중 하나인 Linear SVM 에 적용하는 것은 실제 마이데이터사업자가 개인신용정보 분석하는 방법과 유사하며 추론 성능 또한 0.15초 이내로 합리적인 시간 내에 서비스가 가능하다는 점을 알 수 있었다. 특히, 해당 데이터들은 8,192개의 데이터를 동시에 추론에 사용한 것이기 때문에 8,192개의 데이터를 추론하면 실제 데이터는 0.018ms/건 의 속도로 추론을 수행한 것으로 볼 수 있다. 또한, 100만건의 데이터를 추론한다 하더라도 약 18.3초가 소요될 것으로 예상되기 때문에 배치 서비스에서도 충분히 효과적으로 동형암호 기반 머신러닝 서비스를 제공할 수 있을 것으로 보인다.

Ⅵ. 결론

본 논문에서는 최초로 마이데이터에 동형암호를 적용하는 아키텍처 관점에서 구체적인 시나리오를 제안하였다.

암호화된 상태에서 다양한 연산이 수행 가능한 동형암호의 특성으로 마이데이터사업자는 개인정보 유출 위험 없이 자유롭게 데이터를 활용할 수 있다. 특히, 실제 데이터를 기반으로 많이 쓰이는 Linear SVM 알고리즘에서 8,192개의 데이터를 0.15초 이내에 암호화된 추론이 이뤄질 수 있다는 사실과 암호문 및 키 사이즈가 일반적인 서버 스펙에서도 활용 가능한 합리적인 크기임을 확인하여 마이데이터 API에 실제 동형암호를 충분히 적용할 수 있다는 사실을 아키텍처 관점에서 확인하였다. 다만, 암호화된 데이터 또는 Galois key가 필요한 연산의 경우 사이즈가 큰 데이터나 키를 제공하기 위해 API 응답에 다운로드 URL을 제공하는 등의 고려가 필요하다.

앞으로는 본 연구를 바탕으로 금융 마이데이터 API에 동형암호를 적용하기 위하여 파라미터 및 필드 값 등에 대한 연구를 통해 기존의 마이데이터 API의 변경을 최대한 유지한 채 동형암호 기반 머신러닝을 적용할 수 있도록 하는 API 관점의 연구와, 동형암호를 기반으로 한 다양한 머신러닝 연산들에 대한 연구를 추가적으로 진행할 예정이다.

마이데이터에서 서비스에서 제공하는 각종 로직과 알고리즘은 각 사업자의 고유한 기술과 노하우가 들어가있고, 보안의 이슈 또한 있기 때문에 직접적으로 어떤 알고리즘을 사용하는지 알기 어렵다. 특히 개인신용데이터는 프라이버시 이슈로 인해 이상거래탐지시스템(FDS) 등 더 좋은 모델을 학계 및 업계에서 협업하여 만들기가 쉽지 않다. 하지만, 동형암호를 사용하면 암호화된 데이터의 연산뿐만 아니라 결합 등도 가능하기 때문에 이를 활용하면 더 좋은 모델을 개발하는데 큰 도움이 될 수 있다. 동형암호 기반 데이터 결합 및 머신러닝 학습 등에 대한 연구도 추후 진행할 예정이다.

[참고문헌]

[1] [Internet], Available : https://byline.network/2021/12/10-188, 2022.08.04
[2] [Internet], Available : https://www.cica.or.kr/14_mydata/mydata_05.jsp, 2022.08.04
[3] [Internet], Available : https://www.fsc.go.kr/no010101/77182?srchCtgry=&curPage=&srchKey=&srchText=&srchBeginDt=&srchEndDt=, 2022.08.04
[4] [Internet], Available : https://www.fsc.go.kr/po010101/76323?srchCtgry=1&curPage=&srchKey=&srchText=&srchBeginDt=&srchEndDt=, 2022.08.04
[5] Gentry, C. A fully homomorphic encryption scheme. Stanford university. 2009
[6] Cheon, J. H., Kim, A., Kim, M. et al, "Homomorphic Encryption for Arithmetic of approximate Numbers", International conference on the theory and application of cryptology and information security, pp 409-437, December 2017.
[7] Cheon, J. H., Han, K., Kim, A., Kim, M et al. Bootstrapping for approximate homomorphic encryption. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 360-384. Springer, Cham. April 2018
[8] Kim, A., Song, Y., Kim, M., Lee, K. et al, J. H. Logistic regression model training based on the approximate homomorphic encryption. BMC medical genomics, 11(4), pp 23-31, 2018
[9] Han, K., Hong, S., Cheon, J. H. et al. Logistic regression on homomorphic encrypted data at scale. In Proceedings of the AAAI conference on artificial intelligence, Vol. 33, No. 01, pp. 9466-9471. July 2019
[10] Ishiyama, T., Suzuki, T., & Yamana, H. Highly accurate CNN inference using approximate activation functions over homomorphic encryption. International Conference on Big Data (Big Data), pp. 3989-3995, IEEE. December 2020
[11] [Internet], Available: https://scikit-learn.org/stable, 2022.08.04
[12] [Internet], Available : https://www.kaggle.com/datasets/ethon0426/lending-club-20072020q1?select=Loan_status_2007-2020Q3.gzip, 2022.08.04
[13] [Internet], Available : https://www.microsoft.com/en-us/research/project/microsoft-seal, 2022.08.04
[14] [Internet], Available : https://www.lendingclub.com, 2022.08.04
[15] [Internet], Available : https://www.lendit.co.kr/loan, 2022.08.04

[16] [Internet], Available : https://8percent.kr, 2022.08.04
[17] [Internet], Available : https://www.peoplefund.co.kr, 2022.08.04
[18] [Internet], Available : https://www.ajunews.com/view/20211116102134913, 2022.08.04
[19] 김진우, 지원철. 신용카드 대손회원 예측을 위한 SVM 모형. 한국IT서비스학회지, 11(4), pp 233-250, 2012
[20] 최현민. 동형암호를 이용한 기계학습에서의 데이터 프라이버시 보존에 관한 연구, 한국정보보호학회 하계학술대회, 2022
[21] [Internet], Available : https://www.hankyung.com/it/article/202204129789i, 2022.08.04