

## 명세서

### 발명의 명칭: 딥러닝 기반의 부정거래 탐지 방법

#### 기술분야

- [1] 본 발명은 딥러닝 기반의 부정거래 탐지 방법에 관한 것이다. 구체적으로, 본 발명은 블록체인 네트워크 내에서 발생하는 데이터를 이용하여 부정거래 여부를 탐지하는 방법에 관한 것이다.

[2]

#### 배경기술

- [3] 대부분의 금융 관련 기업들은 전자금융 사기에 대응하기 위해 부정거래 탐지 시스템(Fraud Detection System; 이하 FDS)을 구축 및 운영하고 있다. 부정거래 탐지 시스템은 임의의 주소(계좌) 또는 거래가 부정거래 패턴과 유사한지 여부를 판별한다. 하지만 종래의 대부분의 부정거래 탐지 시스템은 미리 정해진 규칙을 이용하여 부정거래 여부를 판단하기에, 지속적으로 변화하는 부정거래 패턴을 사람의 개입 없이는 업데이트 하지 못한다는 한계가 있었다.
- [4] 또한, 최근 다수의 연구에서는 종래의 규칙 기반 접근 보다 머신러닝 기반 접근을 이용할 때 더 좋은 성능의 부정거래 탐지 시스템을 구축할 수 있다고 소개하고 있다. 더욱이 암호화폐에 대한 부정거래 탐지 시스템은 종래의 금융권에서 부정거래 탐지 시스템을 구축할 때 다루는 데이터와는 다른 형태의 데이터를 갖고 있다. 예를 들어, 이더리움에 대한 트랜잭션 데이터에는 이더리움의 발신자가 트랜잭션을 발생시킬 때 생성되는 다양한 데이터가 포함되는데, 이러한 데이터는 기존의 금융거래에서는 발생되지 않던 데이터이다.
- [5] 따라서, 종래의 부정거래 탐지 시스템에 대한 연구 및 발명에서는 상기 암호화폐에 특화된 부정거래 탐지에 대한 고려가 상대적으로 부족한 실정이다.

[6]

#### 발명의 상세한 설명

##### 기술적 과제

- [7] 본 발명의 목적은, 암호화폐 거래에 특화된 지도학습을 이용한 딥러닝 기반의 부정거래 탐지 방법을 제공하는 것이다.
- [8] 또한, 본 발명의 목적은, 서로 다른 이종의 데이터를 전처리하고, 전처리된 데이터를 기초로 특징벡터를 각각 추출한 뒤, 추출된 특징벡터들을 이용하여 부정거래 탐지 성능을 향상시키는 것이다.
- [9] 또한, 본 발명의 목적은, 학습을 통하여 부정거래 탐지 성능을 향상시킬 수 있는 딥러닝 모듈을 기반으로 동작하는 부정거래 탐지 방법을 제공하는 것이다.
- [10] 본 발명의 목적들은 이상에서 언급한 목적으로 제한되지 않으며, 언급되지 않은 본 발명의 다른 목적 및 장점들은 하기의 설명에 의해서 이해될 수 있고, 본

발명의 실시예에 의해 보다 분명하게 이해될 것이다. 또한, 본 발명의 목적 및 장점들은 특허 청구 범위에 나타낸 수단 및 그 조합에 의해 실현될 수 있음을 쉽게 알 수 있을 것이다.

[11]

### 과제 해결 수단

- [12] 상기 기술적 과제를 달성하기 위한 본 발명의 몇몇 실시예에 따른 딥러닝 기반의 부정거래 탐지 방법은, 특정 지갑주소에 대한 트랜잭션 데이터를 기초로 복수의 통계적 특징을 도출하여 제1 데이터셋을 생성하는 단계, 상기 지갑주소와 관련된 스마트 컨트랙트를 실행시키며, 상기 트랜잭션 데이터에 포함된 입력 코드(input code)를 이용하여 제2 데이터셋을 생성하는 단계, 상기 스마트 컨트랙트의 배포시 사용되는 컨트랙트 주소의 바이트 코드를 이용하여 제3 데이터셋을 생성하는 단계, 미리 학습된 딥러닝 모듈을 이용하여, 상기 제1 내지 제3 데이터셋에 대해 각각 제1 내지 제3 특징벡터를 추출하는 단계, 및 추출된 상기 제1 내지 제3 특징벡터를 이용하여, 상기 지갑주소에 대한 부정거래여부를 추론하는 단계를 포함한다.
- [13] 또한, 상기 제1 내지 제3 데이터셋 중 적어도 하나의 데이터셋을 대상으로 데이터의 수를 증가시키는 오버샘플링을 수행하는 단계를 더 포함하되, 상기 오버샘플링을 수행하는 단계는, 상기 제1 내지 제3 데이터셋에 포함되고, 서로 인접한 복수의 데이터를 선정하는 단계와, 상기 선정된 복수의 데이터에 대해 SMOTE 기법을 이용하여 추가 데이터를 생성하는 단계를 포함할 수 있다.
- [14] 또한, 상기 오버샘플링을 수행하는 단계는, 상기 제2 데이터셋 및 상기 제3 데이터셋을 대상으로 상기 오버샘플링을 수행하는 것을 포함할 수 있다.
- [15] 또한, 상기 제1 데이터셋을 생성하는 단계는, 상기 지갑주소에 대한 복수의 트랜잭션 데이터를 수신하는 단계와, 상기 복수의 트랜잭션 데이터를 기초로 서로 다른 통계적 특징을 도출하는 단계와, 상기 도출된 통계적 특징의 값이, 0 내지 1 사이에 분포되도록, 정규화를 수행하는 단계를 포함할 수 있다.
- [16] 또한, 상기 통계적 특징은, 상기 지갑주소에 수신된 트랜잭션 데이터를 기초로 생성된 제1 통계적 특징과, 상기 지갑주소에서 송신된 트랜잭션 데이터를 기초로 생성된 제2 통계적 특징을 포함할 수 있다.
- [17] 또한, 상기 제1 데이터셋을 생성하는 단계는, 상기 수신된 트랜잭션 데이터 또는 상기 송신된 트랜잭션 데이터가 존재하지 않아, 상기 제1 통계적 특징 또는 상기 제2 통계적 특징에 결측치(none)가 생기는 경우, 상기 결측치의 값을 미리 정해진 값으로 대체하는 단계를 더 포함할 수 있다.
- [18] 또한, 상기 제2 데이터셋을 생성하는 단계는, 상기 지갑주소에 대한 복수의 트랜잭션 데이터를 수신하는 단계와, 상기 복수의 트랜잭션 데이터에 포함된 복수의 입력 코드를 추출하는 단계와, 상기 추출된 복수의 입력 코드를 연결하는 단계와, 상기 연결된 입력 코드를 미리 정해진 단위로 분리하는 단계와, 상기

- 분리된 입력 코드를 대상으로 정수 인코딩을 수행하는 단계를 포함할 수 있다.
- [19] 또한, 상기 제2 데이터셋을 생성하는 단계는, 상기 정수 인코딩된 각 데이터의 문자열이 미리 정해진 문자열 최대 길이보다 작은 경우, 해당 문자열에 미리 정해진 문자를 추가하고, 상기 정수 인코딩된 각 데이터의 문자열이 상기 문자열 최대 길이보다 큰 경우, 해당 문자열에서 상기 문자열 최대 길이를 초과하는 문자열을 삭제하는 것을 포함할 수 있다.
- [20] 또한, 상기 제3 데이터셋을 생성하는 단계는, 상기 컨트랙트 주소와 관련된 바이트 코드를 추출하는 단계와, 상기 추출된 바이트 코드를 미리 정해진 단위로 분리하는 단계와, 상기 분리된 바이트 코드를 대상으로 정수 인코딩을 수행하는 단계를 포함할 수 있다.
- [21] 또한, 상기 제3 데이터셋을 생성하는 단계는, 상기 정수 인코딩된 각 데이터의 문자열이 미리 정해진 문자열 최대 길이보다 작은 경우, 해당 문자열에 미리 정해진 문자를 추가하고, 상기 정수 인코딩된 각 데이터의 문자열이 상기 문자열 최대 길이보다 큰 경우, 해당 문자열에서 상기 문자열 최대 길이를 초과하는 문자열을 삭제하는 것을 포함할 수 있다.
- [22] 또한, 상기 제1 내지 제3 특징벡터를 추출하는 단계는, 상기 제1 데이터셋을 미리 학습된 컨볼루션 신경망에 인가하는 단계와, 상기 컨볼루션 신경망을 통과한 값에 맥스풀링을 적용하는 단계와, 상기 맥스풀링을 통과한 값에 소프트맥스 함수를 적용하는 단계를 포함할 수 있다.
- [23] 또한, 상기 제1 내지 제3 특징벡터를 추출하는 단계는, 상기 제2 데이터셋 또는 상기 제3 데이터셋을 미리 학습된 제1 컨볼루션 신경망에 인가하는 단계와, 상기 제1 컨볼루션 신경망을 통과한 값에 제1 맥스풀링을 적용하는 단계와, 상기 제2 데이터셋을 미리 학습된 제2 컨볼루션 신경망에 인가하는 단계와, 상기 제2 컨볼루션 신경망을 통과한 값에 제2 맥스풀링을 적용하는 단계와, 상기 제2 맥스풀링이 적용된 값에 소프트맥스 함수를 적용하는 단계와, 상기 제1 맥스풀링이 적용된 제1 중간값과, 상기 소프트맥스 함수가 적용된 제2 중간값의 원소곱을 계산하는 단계를 포함할 수 있다.
- [24] 또한, 상기 원소곱의 결과값에 선형 인공신경망(ANN), 정류 선형 유닛 함수(ReLU), 및 배치 정규화를 순차적으로 적용하는 단계를 더 포함할 수 있다.
- [25] 또한, 상기 원소곱의 결과값에 선형 인공신경망(ANN) 및 누설 정류 선형 유닛 함수(Leaky ReLU)를 순차적으로 적용하는 단계를 더 포함할 수 있다.
- [26] 또한, 상기 부정거래여부를 추론하는 단계는, 상기 제1 내지 제3 특징벡터를 연결(concatenation)하는 단계와, 상기 연결된 제1 내지 제3 특징벡터에 레이어 정규화(Layer Normalization)를 수행하는 단계와, 상기 레이어 정규화를 수행한 값을 피드-포워드 네트워크에 인가하는 단계와, 상기 피드-포워드 네트워크를 통과한 값과, 상기 연결된 제1 내지 제3 특징벡터를 더한 중간값을 도출하는 단계와, 상기 중간값에 시그모이드 함수를 적용하여, 상기 부정거래에 대한 판단 결과를 출력하는 단계를 포함할 수 있다.

[27]

**발명의 효과**

[28] 이상에서 설명한 바와 같이, 본 발명의 부정거래 탐지 방법은, 서로 다른 종류의 데이터를 전처리하고, 전처리된 데이터를 기초로 특징벡터를 각각 추출하여 병합하는 과정을 수행함으로써, 부정거래 여부를 판단할 수 있다. 이를 통해, 본 발명은 부정거래 탐지에 대한 정확도를 높이고, 암호화폐와 관련된 금융사고의 위험성을 낮출 수 있다.

[29] 또한, 본 발명의 부정거래 탐지 방법은, 부정거래 판단에 이용되는 데이터의 종류 및 양을 확장시키고, 확장된 데이터를 기초로 딥러닝 모델을 계속해서 학습시킴으로써, 부정거래 탐지에 대한 정확도를 높이고, 암호화폐 거래에 대한 신뢰성을 향상시킬 수 있다.

[30] 상술한 내용과 더불어 본 발명의 구체적인 효과는 이하 발명을 실시하기 위한 구체적인 사항을 설명하면서 함께 기술한다.

[31]

**도면의 간단한 설명**

[32] 도 1은 본 발명의 몇몇 실시예에 따른 딥러닝 기반의 부정거래 탐지 시스템을 나타내는 개략도이다.

[33] 도 2는 도 1의 부정거래 탐지서버에서 부정거래 탐지 방법을 수행하는 구성요소를 나타내는 블록도이다.

[34] 도 3은 도 2의 통계적 특징 전처리부의 동작을 나타내는 순서도이다.

[35] 도 4는 도 2의 통계적 특징 전처리부에서 이용하는 트랜잭션 데이터의 구성을 나타내는 도면이다.

[36] 도 5는 도 2의 통계적 특징 전처리부에서 도출하는 통계적 특징들을 나타내는 도면이다.

[37] 도 6 내지 도 8은 도 3의 통계적 특징 전처리부의 동작을 예시적으로 설명하기 위한 도면이다.

[38] 도 9는 도 2의 입력 코드 전처리부의 동작을 나타내는 순서도이다.

[39] 도 10 내지 도 12는 도 9의 입력 코드 전처리부의 동작을 예시적으로 설명하기 위한 도면이다.

[40] 도 13은 도 2의 바이트 코드 전처리부의 동작을 나타내는 순서도이다.

[41] 도 14 및 도 15는 도 2의 오버샘플링부의 동작을 예시적으로 설명하기 위한 도면이다.

[42] 도 16은 도 2의 제1 특징벡터 추출부의 구성을 나타내는 블록도이다.

[43] 도 17은 도 2의 제1 특징벡터 추출부의 동작을 예시적으로 설명하기 위한 도면이다.

[44] 도 18은 도 2의 제2 특징벡터 추출부의 구성을 나타내는 블록도이다.

[45] 도 19는 도 2의 제2 특징벡터 추출부의 동작을 예시적으로 설명하기 위한

도면이다.

- [46] 도 20은 도 2의 제3 특징벡터 추출부의 구성을 나타내는 블록도이다.
- [47] 도 21은 도 2의 특징벡터 추론부의 구성을 나타내는 블록도이다.
- [48] 도 22는 본 발명의 몇몇 실시예에 따른 부정거래 탐지 방법을 수행하는 시스템의 하드웨어 구현을 설명하기 위한 도면이다.

[49]

### 발명의 실시를 위한 형태

- [50] 본 발명의 이점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 상세하게 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 그러나 본 발명은 이하에서 개시되는 실시예들에 한정되는 것이 아니라 서로 다른 다양한 형태로 구현될 것이며, 단지 본 실시예들은 본 발명의 개시가 완전하도록 하며, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명은 청구항의 범주에 의해 정의될 뿐이다. 명세서 전체에 걸쳐 동일 참조 부호는 동일 구성 요소를 지칭한다.
- [51] 본 명세서에서 사용된 용어는 실시예들을 설명하기 위한 것이며 본 발명을 제한하고자 하는 것은 아니다. 본 명세서에서, 단수형은 문구에서 특별히 언급하지 않는 한 복수형도 포함한다. 명세서에서 사용되는 "포함한다(comprises)" 및/또는 "포함하는(comprising)"은 언급된 구성요소, 단계, 동작 및/또는 소자는 하나 이상의 다른 구성요소, 단계, 동작 및/또는 소자의 존재 또는 추가를 배제하지 않는다.
- [52] 다른 정의가 없다면, 본 명세서에서 사용되는 모든 용어(기술 및 과학적 용어를 포함)는 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 공통적으로 이해될 수 있는 의미로 사용될 수 있을 것이다. 또한, 일반적으로 사용되는 사전에 정의되어 있는 용어들은 명백하게 특별히 정의되어 있지 않는 한 이상적으로 또는 과도하게 해석되지 않는다.
- [53] 이하에서는, 도 1 내지 도 22를 참조하여, 본 발명의 몇몇 실시예에 따른 딥러닝 기반의 부정거래 탐지 방법에 대해 설명하도록 한다.
- [54]
- [55] 도 1은 본 발명의 몇몇 실시예에 따른 딥러닝 기반의 부정거래 탐지 시스템을 나타내는 개략도이다.
- [56] 도 1을 참조하면, 본 발명의 몇몇 실시예에 따른 부정거래 탐지 방법은, 부정거래 탐지 서버(100)에서 수행된다. 부정거래 탐지 서버(100)는 통신망(1200)을 통해 사용자 단말(1100) 및 블록체인 네트워크(1300)와 연계되어 동작할 수 있다.
- [57] 부정거래 탐지 서버(1000)는 사용자 단말(1100)에게 다양한 암호화폐(또는, 가상화폐) 관련 거래 서비스를 제공할 수 있다. 이때, 암호화폐 관련 거래

서비스에는 암호화폐에 대한 구매(buy), 판매(sell), 입금(deposit), 출금(withdraw), 이체(transfer) 등의 서비스를 포함할 수 있다.

- [58] 한편, 블록체인을 이용한 암호화폐는 블록체인 네트워크(1300) 내에서 채굴되고, 기록되고, 전송될 수 있다. 여기에서, 블록체인을 이용한 암호화폐에는 비트코인(bitcoin), 이더리움(ethereum; 이하, ETH) 등이 포함될 수 있다.
- [59] 또한, 블록체인 네트워크(1300) 내에서 이루어지는 암호화폐의 수신, 송신, 교환 등은 블록체인 트랜잭션(blockchain transaction)으로 정의되거나 처리될 수 있다. 블록체인 트랜잭션은 해쉬 함수(Hash function)을 통해 변환되어 임의의 길이의 값으로 생성되어 블록에 저장되거나 처리될 수 있다.
- [60] 한편, '스마트 컨트랙트(smart contract)'란 블록체인 기반으로 동작하며, 서면으로 이루어지던 계약을 코드로 구현하여 특정 조건이 충족되었을 때 해당 계약이 이행되도록 하는 스크립트(script)라고 볼 수 있다.
- [61] 본 발명의 부정거래 탐지 서버(1000)는 사용자가 보유한 지갑주소와 관련된 블록체인 트랜잭션(이하, 트랜잭션)과, 각각의 트랜잭션에 포함되어 스마트 컨트랙트를 실행시키는 인자의 역할을 하는 입력 코드(Input Code), 및 스마트 컨트랙트의 배포시 이용되는 컨트랙트 바이트 코드(contract bytecode; 이하 바이트 코드)를 기초로, 사용자의 부정거래 여부를 탐지할 수 있다.
- [62] 구체적으로, 부정거래 탐지 서버(1000)는 지도학습 방법론을 이용한 딥러닝 기반의 부정거래 탐지 방법을 수행한다. 예를 들어, 가상화폐(예를 들어, 이더리움)의 지갑주소가 입력되는 경우, 부정거래 탐지 서버(1000)는 미리 학습된 머신러닝 모델이 해당 지갑주소와 관련된 데이터가 부정거래패턴과 얼마나 유사한지를 판단한 뒤, 해당 지갑주소의 위험도 평가 점수를 반환한다.
- [63] 부정거래 탐지 서버(1000)는 지도된 데이터셋(supervised dataset)(즉, 각 데이터에 대해 이상여부에 대한 라벨값이 부여된 데이터셋)을 이용하여 부정거래 탐지 모델을 학습시킬 수 있다. 이때, 부정거래 여부 탐지를 목적으로 한 지도된 데이터셋은 아래 같이 구성될 수 있다. 아래 [표 1] 내의 라벨 열(column)에서 '1'은 부정거래인 경우를 의미하고, '0'은 정상거래인 경우를 의미할 수 있다.

[64]

[65] [표1]

| 가상화폐 지갑주소                                      | 통계적<br>특징 | 입력<br>코드 | 바이트<br>코드 | 라벨<br>값 |
|------------------------------------------------|-----------|----------|-----------|---------|
| 0xD854526EA2E285666C9d35D2Da3C07D514DB<br>90a7 | ...       | ...      | ...       | 0       |
| 0x395604F1Db081376D0FE5dC9F154d0946065Cc<br>Ab | ...       | ...      | ...       | 1       |
| 0xd6Cfd617103649A2F72DF3C7A89ff246836bFC<br>08 | ...       | ...      | ...       | 1       |

[66] 부정거래 탐지 서버(1000)에서 부정거래를 탐지하는데 이용되는 데이터셋을 전처리하는 방법이나, 부정거래여부를 판단하는 방법에 대한 자세한 내용은 이하에서 후술하도록 한다.

[67] 한편, 본 발명에서 부정거래 탐지 서버(1000)와 사용자 단말(1100)은 서버-클라이언트 시스템으로 구현될 수 있다. 부정거래 탐지 서버(1000)는 각 고객계정에 사용자의 가입명의정보, 지갑주소, 트랜잭션 정보, 스마트 컨트랙트 정보를 저장 및 관리할 수 있고, 암호화폐와 관련된 다양한 서비스를 사용자 단말(1100)에 설치된 암호화폐 거래 어플리케이션을 통해 제공할 수 있다.

[68] 이때, 암호화폐 어플리케이션은 암호화폐 서비스를 제공하기 위한 전용 어플리케이션이거나, 웹 브라우징 어플리케이션일 수 있다. 여기에서, 전용 어플리케이션은 사용자 단말(1100)에 내장된 어플리케이션이거나, 어플리케이션 배포 서버로부터 다운로드 되어 사용자 단말(1100)에 설치된 어플리케이션일 수 있다.

[69] 한편, 사용자 단말(1100)은 유무선 통신 환경에서 복수의 어플리케이션을 동작시킬 수 있는 통신 단말기를 의미한다. 도 1에서 사용자 단말(1100)은 휴대용 단말기의 일종인 스마트폰(smart phone)으로 도시되었지만, 본 발명이 이에 한정되는 것은 아니며, 상술한 바와 같이 암호화폐 어플리케이션 또는 SNS 어플리케이션을 동작시킬 수 있는 장치에 제한없이 적용될 수 있다. 예를 들어, 사용자 단말(1100)은 퍼스널 컴퓨터(PC), 노트북, 태블릿, 휴대폰, 스마트폰, 웨어러블 디바이스(예를 들어, 위치형 단말기) 등의 다양한 형태의 전자 장치를 포함할 수 있다.

[70] 한편, 통신망(1200)은 부정거래 탐지 서버(1000), 사용자 단말(1100) 및 블록체인 네트워크(1300)를 연결하는 역할을 수행한다. 통신망(1200)은 예컨대 LANs(Local Area Networks), WANs(Wide Area Networks), MANs(Metropolitan Area Networks), ISDNs(Integrated Service Digital Networks) 등의 유선 네트워크나, 무선 LANs, CDMA, 블루투스, 위성 통신 등의 무선 네트워크를 망라할 수 있으나, 본 발명의 범위가 이에 한정되는 것은 아니다.

- [71] 이하에서는, 본 발명의 몇몇 실시예에 따른 딥러닝 기반의 부정거래 탐지 방법에 대해 자세히 설명하도록 한다.
- [72]
- [73] 도 2는 도 1의 부정거래 탐지서버에서 부정거래 탐지 방법을 수행하는 구성요소를 나타내는 블록도이다.
- [74] 도 2를 참조하면, 본 발명의 몇몇 실시예에 따른 부정거래 탐지서버(1000)는, 데이터 전처리부(100), 특징벡터 추출부(200), 및 특징벡터 추론부(300)를 포함한다.
- [75] 데이터 전처리부(100)는 특정 사용자의 지갑주소에 대한 트랜잭션 데이터와, 상기 트랜잭션 데이터에 포함된 입력 코드와, 상기 사용자가 이용하는 스마트 컨트랙트에 포함된 바이트 코드를 수신한다. 데이터 전처리부(100)는 수신된 트랜잭션 데이터, 입력 코드, 및 바이트 코드를 각각 전처리함으로써, 제1 내지 제3 데이터셋(D1', D2', D3')을 출력한다.
- [76] 구체적으로, 데이터 전처리부(100)는 통계적 특징 전처리부(110), 입력 코드 전처리부(120), 및 바이트 코드 전처리부(130)를 포함한다. 또한, 본 발명의 일 실시예에서, 데이터 전처리부(100)는 오버샘플링부(140)를 더 포함할 수 있다. 다만, 본 발명이 이에 한정되는 것은 아니며, 본 발명의 다른 실시예에서 데이터 전처리부(100)는 오버샘플링부(140)를 미포함한 구성으로 수행될 수 있음은 물론이다. 다만, 설명의 편의를 위하여 이하에서는 데이터 전처리부(100)에 오버샘플링부(140)가 포함된 것으로 예로 들어 설명하도록 한다.
- [77] 이하에서는, 우선, 통계적 특징 전처리부(110)의 동작에 대해 자세히 설명하도록 한다.
- [78]
- [79] 도 3은 도 2의 통계적 특징 전처리부의 동작을 나타내는 순서도이다. 도 4는 도 2의 통계적 특징 전처리부에서 이용하는 트랜잭션 데이터의 구성을 나타내는 도면이다. 도 5는 도 2의 통계적 특징 전처리부에서 도출하는 통계적 특징들을 나타내는 도면이다. 이하에서는 설명의 편의를 위하여, 암호화폐가 이더리움(ETH; 이하, 이더)인 것을 예로 들어 설명하도록 한다.
- [80] 먼저, 도 3을 참조하면, 통계적 특징 전처리부(110)는 특정 지갑주소에 대한 트랜잭션 데이터를 수신한다(S111). 이때, 통계적 특징 전처리부(110)는 트랜잭션 데이터(transaction data)를 이용하여 복수 개의 통계적 특징을 포함하는 데이터셋을 생성한다. 트랜잭션 데이터는 이더(ETH)를 송신인(sender)으로부터 수신인(recipient)에게 전송할 때 발생하는 데이터이며, 이더(ETH)를 1회 전송시, 1개의 트랜잭션 데이터가 생성될 수 있다.
- [81] 예를 들어, 도 4를 참조하면, 트랜잭션 데이터는 송신인(sender), 수신인(recipient), 전송량(Amount), 입력 코드(Input Code), 가스 리미트(Gas limit), 가스 가격(Gas Price) 등의 항목을 포함할 수 있다.
- [82] 이어서, 통계적 특징 전처리부(110)는 수신된 트랜잭션 데이터를 기초로

통계적 특징을 도출한다(S113). 이때, 통계적 특징 전처리부(110)에서 생성하는 통계적 특징은 특정 지갑주소에서 주고받은 트랜잭션 데이터를 기초로 생성될 수 있다.

- [83] 예를 들어, 통계적 특징 전처리부(110)는 특정 지갑주소에서 수신한 트랜잭션 데이터를 기초로 도 5에 개시된 16가지의 통계적 특징을 산출할 수 있다. 또한, 특정 지갑주소에서 송신한 트랜잭션 데이터를 기초로 도 5에 개시된 16가지의 통계적 특징을 산출할 수 있다. 즉, 특정 지갑주소에 수신된 트랜잭션 데이터를 기초로 16개의 통계적 특징이 산출되고, 특정 지갑주소에서 송신한 트랜잭션 데이터를 기초로 16개의 통계적 특징이 산출되어, 특정 지갑주소에서는 총 32개의 통계적 특징이 산출될 수 있다.
- [84] 이하에서는, 도 6 내지 도 8을 참조하여 통계적 특징 전처리부(110)에서 도출되는 각각의 통계적 특징의 각 항목에 대해, 예를 들어 설명하도록 한다.
- [85]
- [86] 도 6 내지 도 8은 도 3의 통계적 특징 전처리부의 동작을 예시적으로 설명하기 위한 도면이다.
- [87] 도 6을 참조하면, 제1 사용자(USER1)는 3명의 다른 사용자들(USER2, USER3, USER4)에게 4번의 트랜잭션을 수행하여 이더(ETH)를 송금하였다. 이하에서는, 제1 사용자(USER1)는 제2 사용자(USER2)에게 1회, 제3 사용자(USER3)에게 1회, 제4 사용자(USER4)에게 2회, 총 4회 송금을 수행한 것을 예로 들어 설명하도록 한다. 이때, 통계적 특징 전처리부(110)는 제1 사용자(USER1)의 송금 트랜잭션 데이터를 기초로 아래의 16개의 통계적 특징(도 5의 각 항목 참조)을 도출할 수 있다.
- [88]
- [89] 1) 트랜잭션 개수: 제1 사용자(USER1)는 4번의 송금 트랜잭션을 발생시켰으므로, 트랜잭션의 개수는 '4'이다.
- [90] 2) 지갑주소의 수명(lifetime): 특정 지갑주소가 만들어진 이후, 처음으로 거래된 시점과 가장 최근에 거래된 시점 간 차이를 나타낸다. 도 6에서 제1 사용자(USER1)가 첫 거래한 시점은 제2 사용자(USER2)에게 제1 트랜잭션(TD1)을 보낸 "2021-07-11-21:01:32" 이고, 마지막으로 거래한 시점은 제4 사용자(USER4)에게 제4 트랜잭션(TD4)을 보낸 "2021-11-11-12:01:12" 이다. 지갑주소의 수명은 도출된 시간차를 분(minute)으로 환산한 값이다.
- [91] 3) (중복을 제거한) 상대주소의 총 개수: 제1 사용자(USER1)는 제2 사용자(USER2), 제3 사용자(USER3), 및 제4 사용자(USER4)를 포함하는 3명의 사용자에게 송금하였으므로, 상대주소의 총 개수는 3이다.
- [92] 4) 이더(ETH) 전송량 총 합계: 제1 사용자(USER1)는 각 트랜잭션 마다 0.2 ETH, 0.3 ETH, 0.1 ETH, 0.2 ETH를 보냈으므로, 전송한 이더의 총 합계는 '0.8'이다.
- [93] 5) 최대 이더(ETH) 전송량: 제1 사용자(USER1)는 각 트랜잭션 마다 0.2 ETH, 0.3 ETH, 0.1 ETH, 0.2 ETH를 보냈으므로, 이들 중 가장 큰 값은 '0.3' 이므로, 최대

이더 전송량은 '0.3'이다.

- [94] 6) 최소 이더(ETH) 전송량: 제1 사용자(USER1)는 각 트랜잭션마다 0.2 ETH, 0.3 ETH, 0.1 ETH, 0.2 ETH를 보냈으며, 이들 중 가장 작은 값은 '0.1' 이므로, 최소 이더 전송량은 '0.1'이다.
- [95] 7) 평균 이더(ETH) 전송량: 제1 사용자(USER1)는 각 트랜잭션마다 0.2 ETH, 0.3 ETH, 0.1 ETH, 0.2 ETH를 보냈으며, 이들 값의 평균값은 '0.2'이므로, 평균 이더 전송량은 '0.2'이다.
- [96] 8) 이더(ETH) 전송량 표준편차: 제1 사용자(USER1)는 각 트랜잭션마다, 0.2 ETH, 0.3 ETH, 0.1 ETH, 0.2 ETH를 보냈으며, 이들 값의 표준편차는 약 0.071 이므로, 이더 전송량 표준편차는 '0.071'이다.
- [97] 9) 최대 트랜잭션 수수료: 제1 사용자(USER1)는 각 트랜잭션마다, 0.0001 ETH, 0.0011 ETH, 0.005 ETH, 0.011 ETH의 트랜잭션 수수료(transaction fee)를 지불했으며, 이들 값 중 가장 큰 값은 0.011 이므로, 최대 트랜잭션 수수료는 '0.011'이다.
- [98] 10) 최소 트랜잭션 수수료: 제1 사용자(USER1)는 각 트랜잭션마다, 0.0001 ETH, 0.0011 ETH, 0.05 ETH, 0.011 ETH의 트랜잭션 수수료(transaction fee)를 지불했으며, 이들 값 중 가장 작은 값은 0.0001 이므로, 최소 트랜잭션 수수료는 '0.0001'이다.
- [99] 11) 평균 트랜잭션 수수료: 제1 사용자(USER1)는 각 트랜잭션마다, 0.0001 ETH, 0.0011 ETH, 0.05 ETH, 0.011 ETH의 트랜잭션 수수료(transaction fee)를 지불했으며, 이들 값의 평균값은 0.0043 이므로, 평균 트랜잭션 수수료는 '0.0043'이다.
- [100] 12) 트랜잭션 수수료 표준편차: 제1 사용자(USER1)는 각 트랜잭션마다, 0.0001 ETH, 0.0011 ETH, 0.05 ETH, 0.011 ETH의 트랜잭션 수수료(transaction fee)를 지불했으며, 이들 값의 표준편차는 약 0.004 이므로, 트랜잭션 수수료 표준편차는 '0.004'이다.
- [101] 13) 인접 트랜잭션 간 최대 시간 차이: 특정 지갑주소의 모든 트랜잭션 중, 임의의 트랜잭션과 그 직전에 발생한 트랜잭션 간 시간 차이 중 최대값을 의미한다. 따라서, 제1 사용자(USER1)가 송금한 4개 트랜잭션에서 3개의 인접 트랜잭션 간 시간차를 구할 수 있으며, 이들 값 중 가장 큰 값이 인접 트랜잭션 간 최대 시간 차이가 된다. 인접 트랜잭션 간 최대 시간 차이는 도출된 최대 시간 차이를 분(minute)으로 환산한 값이다.
- [102] 14) 인접 트랜잭션 간 최소 시간 차이: 특정 지갑주소의 모든 트랜잭션 중, 임의의 트랜잭션과 그 직전에 발생한 트랜잭션 간 시간 차이 중 최소값을 의미한다. 따라서, 제1 사용자(USER1)가 송금한 4개 트랜잭션에서 3개의 인접 트랜잭션 간 시간차를 구할 수 있으며, 이들 값 중 가장 작은 값이 인접 트랜잭션 간 최소 시간 차이가 된다. 인접 트랜잭션 간 최소 시간 차이는 도출된 최소 시간 차이를 분(minute)으로 환산한 값이다.

- [103] 15) 인접 트랜잭션 간 평균 시간 차이: 특정 지갑주소의 모든 트랜잭션 중, 임의의 트랜잭션과 그 직전에 발생한 트랜잭션 간 시간 차이값들의 평균값을 의미한다. 따라서, 제1 사용자(USER1)가 송금한 4개 트랜잭션에서 3개의 인접 트랜잭션 간 시간차를 구할 수 있으며, 이들 값들에 대한 평균값이 인접 트랜잭션 간 평균 시간 차이가 된다. 인접 트랜잭션 간 평균 시간 차이는 도출된 평균 시간을 분(minute)으로 환산한 값이다.
- [104] 16) 인접 트랜잭션 시간 차이 표준 편차: 특정 지갑주소의 모든 트랜잭션 중, 임의의 트랜잭션과 그 직전에 발생한 트랜잭션 간 시간 차이값들의 표준편차를 의미한다. 따라서, 제1 사용자(USER1)가 송금한 4개 트랜잭션에서 3개의 인접 트랜잭션 간 시간차를 구할 수 있으며, 이들 값들에 대한 표준편차가 인접 트랜잭션 간 시간 차이 표준 편차가 된다. 인접 트랜잭션 간 시간 차이 표준 편차는 도출된 평균 시간을 분(minute)으로 환산한 값이다.
- [105]
- [106] 위의 과정을 통하여, 통계적 특징 전처리부(110)에서 도출된 통계적 특징은 도 7의 <T1> 테이블과 같이 해당 지갑주소와 연계되어 하나의 레코드로 그룹핑될 수 있다. 통계적 특징 전처리부(110)는 도출된 통계적 특징을 해당 지갑주소와 연계하여 데이터베이스에 저장할 수 있다.
- [107] 이어서, 다시 도 3을 참조하면, 통계적 특징 전처리부(110)는 해당 지갑주소에 연계된 통계적 특징의 값들을 정규화(normalization) 시킨다(S117). 예를 들어, 통계적 특징 전처리부(110)는 최대-최소 정규화(Min-Max Normalization)를 이용하여 통계적 특징의 값들을 0에서 1 사이의 값으로 변경할 수 있다.
- [108] 구체적으로, 통계적 특징 전처리부(110)는 아래 [수학식 1]을 이용하여 통계적 특징의 값들을 정규화할 수 있다.
- [109]
- [110] [수학식 1]
- [111] 
$$x'_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$
- [112] 여기에서, x는 지갑주소에 연계된 통계적 특징의 값들의 집합일 수 있다.
- [113] 정규화를 수행함에 따라, 지갑주소에 연계된 통계적 특징의 값들(예를 들어, 도 7의 T1)은 0 내지 1사이의 값(예를 들어, 도 7의 T2)으로 변경되어 이용될 수 있다.
- [114] 추가적으로, 통계적 특징 전처리부(110)는 특정 지갑주소에 송금 트랜잭션만 존재하건, 수신 트랜잭션만 존재하는 경우, 존재하지 않는 항목에 대해 결측치가 존재할 수 있다.
- [115] 예를 들어, 도 8의 T31를 참조하면, 특정 지갑주소에 송금 트랜잭션만 존재하고 수신 트랜잭션이 존재하지 않는 경우, 수금특징1 내지 수금특징5에 할당된 값은 결측치(none)로 처리된다. 이 경우, 통계적 특징 전처리부(110)는 해당

지갑주소에 결측치(none)로 처리된 값들을 '0'으로 대체하는 결측치 처리를 수행할 수 있다(도 3의 S115). 이를 통해, 통계적 특징 전처리부(110)는 특정 지갑주소에 할당된 통계적 특징의 크기를 동일하게 설정할 수 있다.

- [116] 통계적 특징 전처리부(110)에서 수행되는 결측치 처리 단계(S115)는, 통계적 특징 도출 단계(S113)와, 정규화 수행 단계(S117) 사이에 수행될 수 있다. 다만, 본 발명이 이에 한정되는 것은 아니며, 결측치 처리 단계(S115)는 정규화 수행 단계(S117) 이후에 수행되거나, 생략되어 실시될 수 있음은 물론이다.
- [117] 즉, 통계적 특징 전처리부(110)는 특정 사용자의 지갑주소에 대한 트랜잭션 데이터를 이용하여 32개의 통계적 특징을 포함하는 데이터셋을 생성할 수 있다.
- [118] 이하에서는, 데이터 전처리부(100)에 포함된 입력 코드 전처리부(120)의 동작에 대해 살펴보도록 한다.
- [119]
- [120] 도 9는 도 2의 입력 코드 전처리부의 동작을 나타내는 순서도이다. 도 10 내지 도 12는 도 9의 입력 코드 전처리부의 동작을 예시적으로 설명하기 위한 도면이다.
- [121] 도 9를 참조하면, 입력 코드 전처리부(120)는 특정 지갑주소에 대한 트랜잭션에 포함된 입력 코드(Input Code)를 이용하여 데이터셋을 생성한다.
- [122] 이를 위해, 입력 코드 전처리부(120)는 우선 특정 지갑주소에 대한 복수의 트랜잭션 데이터를 수신한다(S121).
- [123] 이어서, 입력 코드 전처리부(120)는 복수의 트랜잭션 데이터에 포함된 입력 코드를 추출한다(S122). 여기에서, 입력 코드는 트랜잭션에 스마트 컨트랙트(smart contract)를 실행시키는 인자로서 포함될 수 있는 코드를 의미한다. 입력 코드는 스마트 컨트랙트를 배포시킬 때 사용되는 컨트랙트 바이트 코드(contract bytecode; 이하, 바이트 코드)와는 달리 매 트랜잭션마다 존재할 수 있다.
- [124] 구체적으로, 입력 코드는, 외부 소유 주소(Externally Owned Address, EOA) 또는 컨트랙트 주소(Contract Address; 이하, CA)를 가진 소유자가, 메시지 송신자(message sender)로서 임의의 컨트랙트 주소에 메시지(즉, 트랜잭션)를 보내어, 해당 주소에 작성된 컨트랙트를 실행시킬 목적으로, 컨트랙트 내에서 어떤 함수를 사용할지, 어떤 인자를 입력하여 그 함수를 실행시킬지를 트랜잭션 내 데이터 필드(Input data)에 작성한 뒤, 일련의 인코딩 과정을 거쳐 얻은 문자열을 의미한다.
- [125]
- [126] 예를 들어, 입력 코드는 다음과 같은 방식으로 생성될 수 있다.
- [127] 1) 송신자가 컨트랙트 주소(CA)로 트랜잭션을 보낼 때, 해당 컨트랙트 내 어떤 함수를 이용할지, 그 함수에 어떤 인자를 전달할지를 입력한다.
- [128] 2) 어떤 함수를 이용할지를 특정하기 위해 입력된 문자열 타입의 정보를 암호화 해시 알고리즘(Keccak-256)을 이용하여 변환한다.

- [129] 3) 변환된 코드에서 앞의 4 바이트만 추출한다.
- [130] 4) 선택된 함수에 전달할 인자가 이더(ETH)의 양(amount)인 경우, 그 값을 웨이(Wei)로 단위 변환한다. 만약, 입력된 인자가 수치인 경우 그 값을 그대로 사용하고, 입력된 인자가 문자열인 경우, 해당 값을 바이트 코드로 변환한다
- [131] 5) 앞선 4)에서 구한 값을 16진수로 변환하되, 직렬화된 부호없는 빅엔디언(Big Endianness) 256비트 정수로 인코딩한다.
- [132] 6) 3)과 5)의 결과를 연결함으로써 입력 코드가 생성된다.
- [133] 다만, 본 발명에서 이용하는 입력 코드를 생성하는 방법이 위의 예시에 한정되는 것은 아니며, 다양한 방식으로 변형되어 이용될 수 있음은 물론이다.
- [134] 입력 코드 전처리부(120)는 위와 같은 방법으로 생성되어 트랜잭션 데이터에 포함된 입력 코드를 추출하며, 복수의 트랜잭션 데이터가 존재하는 경우, 각 트랜잭션 데이터마다 입력 코드를 추출한다.
- [135]
- [136] 이어서, 입력 코드 전처리부(120)는 추출된 복수의 입력 코드를 연결한다(S123).
- [137] 예를 들어, 도 10을 참조하면, 제1 사용자(USER1)의 4개의 송신 트랜잭션(TD1, D2, D3, D4)에는 3개의 입력 코드가 존재할 수 있다. 이때, 트랜잭션 중에 입력 코드가 존재하지 않는 제4 트랜잭션(TD4)의 경우, 제외하고 입력 코드를 추출할 수 있다.
- [138] 입력 코드 전처리부(120)는 복수 개의 트랜잭션에서 추출된 입력 코드를 연결한다. 예를 들어, 도 11의 <T41>을 참조하면, 입력 코드 전처리부(120)는 제1 트랜잭션(TD1)의 제1 입력 코드(Oxasdf)와, 제2 트랜잭션(TD2)의 제2 입력 코드(Ox123)와, 제3 트랜잭션(TD3)의 제3 입력 코드(Ox454e34)를 연결한다. 이때, 입력 코드 전처리부(120)는 복수의 입력 코드 사이에 미리 정해진 문자(예를 들어, '0')를 삽입하여 복수의 입력 코드를 하나의 문자열로 합칠 수 있다. 이때, 입력 코드를 연결하는 순서는, 미리 정해진 기준(예를 들어, 트랜잭션 생성 시간)으로 정렬되어 결정될 수 있다.
- [139] 이어서, 입력 코드 전처리부(120)는 미리 정해진 단위로 연결된 입력 코드를 분리한다(S124). 여기에서, 미리 정해진 단위(n)는 2 이상의 자연수로 설정될 수 있다. 예를 들어, 도 11의 <T42>를 참조하면, 미리 정해진 단위가 4일 때(즉, n=4), 연결된 입력 코드는 'Oxas', 'df00', 'x123', '00x4', '54e3', '4'로 구분될 수 있다.
- [140] 이어서, 입력 코드 전처리부(120)는 분리된 입력 코드에 정수 인코딩을 수행한다(S125). 여기에서, 정수 인코딩은 미리 정해진 단위로 구분된 문자열(이하, 단위 문자열)들을 특정 정수로 사상(또는, 맵핑(mapping))하는 것을 의미한다. 정수 인코딩을 통해, 각각의 단위 문자열은 하나의 정수로 사상될 수 있다. 이때, 단위 문자열에 사상되는 정수는 임의로 결정될 수 있다. 다만, 동일한 단위 문자열에는 동일한 정수가 사상될 수 있다.
- [141] 예를 들어, 도 11의 <T43>를 참조하면, 제1 단위 문자열(Oxas)은 3으로

사상되고, 제2 단위 문자열(dfda)은 1로 사상되고, 제3 단위 문자열(01f)은 2로 사상되고, 제4 단위 문자열(0xfa)는 4로 사상될 수 있다. 이때, 서로 다른 지갑주소에 대하여 분리된 입력 코드에 동일한 단위 문자열이 포함되는 경우(즉, 두번째 지갑주소와 세번째 지갑주소에 공통 포함된 제4 단위 문자열(0xfa)), 해당 단위 문자열은 동일한 정수(즉, 4)로 사상되도록 처리될 수 있다.

- [142] 이어서, 입력 코드 전처리부(120)는 인코딩된 입력 코드에 미리 설정된 문자열 최대 길이에 맞추어 제로 패딩(Zero-padding) 동작을 수행함으로써 각각의 인코딩된 입력 코드의 길이를 통일한다(S126).
- [143] 여기에서, 제로 패딩(Zero-padding) 동작은 서로 다른 길이의 문자열의 크기를 맞추기 위해, 상대적으로 길이가 짧은 문자열의 끝에 무정보를 뜻하는 '0'을 추가하는 동작을 의미한다. 예를 들어, 도 12의 <T44>를 참조하면, 첫번째 지갑주소와 세번째 지갑주소의 입력 코드는 2개의 정수로 구성되고, 두번째 지갑주소의 입력 코드는 3개의 정수로 구성되는 될 수 있다. 이 경우, 입력 코드 전처리부(120)는 첫번째 지갑주소와 세번째 지갑주소의 입력 코드 뒤에 '0'을 1개씩 추가함으로써 지갑주소의 인코딩된 입력 코드의 문자열의 길이를 동일하게 맞춰줄 수 있다.
- [144] 한편, 가장 긴 길이의 입력 코드에 맞추어 제로 패딩을 수행하는 경우, 불필요하게 데이터 처리량이 증가될 수 있다. 따라서, 입력 코드 전처리부(120)는 미리 설정된 문자열 최대 길이에 맞추어 입력 코드의 길이를 통일시킨다. 예를 들어, 인코딩된 입력 코드의 길이가 문자열 최대 길이보다 긴 경우, 문자열 최대 길이를 초과하는 값들은 삭제함으로써 고려하지 않을 수 있다. 이때, 문자열 최대 길이는 작업자 또는 사용자에게 의해 미리 설정되어 이용될 수 있다.
- [145] 전술한 S121 단계 내지 S126 단계를 통하여, 입력 코드 전처리부(120)는 입력 코드에 대한 데이터셋을 생성할 수 있다.
- [146] 이하에서는, 입력 코드 전처리부(120)와 마찬가지로, 바이트 코드 전처리부(130)에서 바이트 코드에 대한 데이터셋을 생성하는 방법에 대해 설명하도록 한다.
- [147]
- [148] 도 13은 도 2의 바이트 코드 전처리부의 동작을 나타내는 순서도이다.
- [149] 실질적으로 바이트 코드 전처리부(130)의 동작은 전술한 입력 코드 전처리부(120)의 동작과 유사하므로, 이하에서는 중복되는 설명은 생략하거나 간단히 설명하도록 한다.
- [150] 도 13을 참조하면, 바이트 코드 전처리부(130)는 컨트랙트 주소(CA)에 대응되는 바이트 코드를 추출한다(S132). 여기에서, 바이트 코드는 컨트랙트 주소(CA)에서 작성된 컨트랙트 바이트 코드(contract bytecode; 이하, 바이트 코드)를 의미하며, 하나의 컨트랙트 주소(CA)에는 하나의 바이트 코드가 생성된다.
- [151] 조금 더 구체적으로, 스마트 컨트랙트가 프로그래밍되고, 해당 스마트

컨트랙트에 대한 소스 코드가 컴파일되는 경우 바이트 코드가 생성되며, 생성된 바이트 코드는 블록에 포함되어 EVM(Ethereum Virtual Machine)에 의해 실행될 수 있다. 바이트 코드에 대한 구체적인 내용은 이미 공개되었으므로 여기에서 자세한 설명은 생략하도록 한다.

- [152] 이어서, 바이트 코드 전처리부(130)는 추출된 바이트 코드를 미리 정해진 단위로 분리한다(S134). 여기에서, 미리 정해진 단위(n)는 2 이상의 자연수로 설정될 수 있으며, S134 단계의 동작은 도 9에서 전술한 S124 단계와 실질적으로 동일하게 수행될 수 있다.
- [153] 이어서, 바이트 코드 전처리부(130)는 분리된 바이트 코드에 정수 인코딩을 수행한다(S135). 여기에서, 정수 인코딩은 미리 정해진 단위로 구분된 문자열(이하, 단위 문자열)들을 특정 정수로 사상(mapping)하는 것을 의미한다. 마찬가지로, S135 단계의 동작은 도 9에서 전술한 S125 단계와 실질적으로 동일하게 수행될 수 있다.
- [154] 이어서, 바이트 코드 전처리부(130)는 인코딩된 바이트 코드에 미리 설정된 문자열 최대 길이에 맞추어 제로 패딩(Zero-padding) 동작을 수행함으로써 각각의 인코딩된 바이트 코드의 길이를 통일한다(S136). 마찬가지로, S135 단계의 동작은 도 9에서 전술한 S125 단계와 실질적으로 동일하게 수행될 수 있다.
- [155] 전술한 S132 단계 내지 S136 단계를 통하여, 바이트 코드 전처리부(130)는 바이트 코드에 대한 데이터셋을 생성할 수 있다.
- [156]
- [157] 이어서, 다시 도 2를 참조하면, 오버샘플링부(140)는, 통계적 특징 전처리부(110)에서 생성된 통계적 특징에 대한 데이터셋(이하, 통계적 특징 데이터셋(D1)과, 입력 코드 전처리부(120)에서 생성된 입력 코드에 대한 데이터셋(이하, 입력 코드 데이터셋(D2), 및 바이트 코드 전처리부(130)에서 생성된 바이트 코드(이하, 바이트 코드 데이터셋(D3)에 대한 데이터셋을 수신한다.
- [158] 이어서, 오버샘플링부(140)는 통계적 특징 데이터셋(D1), 입력 코드 데이터셋(D2), 및 바이트 코드 데이터셋(D3)의 데이터량을 확장시킬 수 있다.
- [159] 구체적으로, 통계적 특징 데이터셋(D1), 입력 코드 데이터셋(D2), 및 바이트 코드 데이터셋(D3)에는 각 데이터마다 라벨값(즉, 부정거래인지 정상거래인지를 나타내는 값)이 포함될 수 있다. 다만, 일반적으로 정상거래사례는 부정거래사례보다 훨씬 많기에, 효과적인 머신러닝 모델의 학습을 위해서는 부정거래사례를 증가시키는 작업이 필요하다. 오버샘플링부(140)는 오버샘플링 동작을 수행함으로써 소수 클래스의 개수를 증가시킬 수 있다.
- [160] 이하에서는 도 14 및 도 15를 참조하여, 오버샘플링부(140)의 동작을 예를 들어 설명하도록 한다.

[161]

[162] 도 14 및 도 15는 도 2의 오버샘플링부의 동작을 예시적으로 설명하기 위한 도면이다.

[163] 도 14를 참조하면, 원본 데이터셋(Original Dataset)에서, 정상 거래 사례(사각형 아이콘)는 부정 거래 사례(원형 아이콘)와 비교하여 비중이 높게 형성되어 있다. 이에 오버샘플링부(140)는 부정 거래 사례(원형 아이콘)를 증가시키기 위해 SMOTE(Synthetic Minority Oversampling Technique; 이하, SMOTE 기법)를 이용하여 서로 인접한 데이터를 기초로 추가 데이터를 생성한다(Generating samples). 이를 통해, 오버샘플링부(140)는 부정 거래 사례(원형 아이콘)가 증가된 데이터셋(resampled Dataset)을 생성할 수 있다.

[164] 도 15를 참조하면, 오버샘플링부(140)에서 수행되는 SMOTE 기법은, 서로 인접한 2개의 데이터에 대하여, 하기 [수학식 2]를 이용하여 추가 데이터를 생성한다.

[165]

[166] [수학식 2]

[167] 
$$x' = x_0 + \text{random}(0, 1) * (x_0 - x_m)$$

[168] 여기에서,  $x'$ 는 새로 생성되는 추가 데이터를 의미하고,  $x_0$ 와  $x_m$ 는 서로 인접한 2개의 데이터를 의미한다.

[169] 위 [수학식 2]을 통해, 오버샘플링부(140)는 서로 인접한 2개의 데이터 중간에 위치한 새로운 추가 데이터를 생성함으로써, 데이터셋에 포함된 데이터의 양을 증가시킬 수 있다. 이를 통해, 오버샘플링부(140)는 상대적으로 비중이 낮은 소수 클래스 데이터의 개수와 상대적으로 비중이 높은 다수 클래스 데이터의 개수를 동등하게 맞출 수 있다.

[170] 다만, 본 발명에서 오버샘플링부(140)가 데이터를 증가시키기 위해 사용하는 방법이 위의 방법으로 한정되는 것은 아니며, 다양한 데이터 증가를 위한 방법이 사용될 수 있음은 물론이다.

[171] 위와 같은 오버샘플링을 수행한 경우와, 수행하지 않은 경우를 나누어 학습 모델의 성능을 실험해보았으며, 이에 대한 결과는 다음 [표 2]과 같다.

[172]

[173] [표2]

| Case                    | 정확도(%) | 정밀도   | 재현율   | F1 스코어 |
|-------------------------|--------|-------|-------|--------|
| 오버샘플링을<br>수행한 경우        | 91.1   | 0.703 | 0.853 | 0.762  |
| 오버샘플링을<br>수행하지 않은<br>경우 | 90.8   | 0.854 | 0.574 | 0.689  |

- [174] 여기에서, 정확도는 데이터셋에 라벨값과 본 발명의 부정거래 탐지 모델(이하, 탐지 모델)의 예측치와의 일치율을 의미하고, 정밀도(Precision)는 '탐지 모델이 “부정거래”라고 판정한 모든 경우' 중에서 '해당 모든 경우 중 실제로 “부정거래”라고 라벨링 된 경우'에 대한 비율을 의미하고, 재현율(Recall)은 '실제로 “부정거래”라고 라벨된 모든 경우' 중에서 '탐지 모델이 “부정거래”라고 예측한 경우'에 대한 비율을 의미하고, F1 스코어(F1 Score)는 재현율과 정밀도 간의 조화평균의 값을 의미한다.
- [175] 이때, F1 스코어는 재현율과 정밀도의 절대적인 크기와 재현율과 정밀도가 얼마나 상이한지 여부를 측정하기 위해 조화평균을 이용하여 탐지 모델의 성능을 판단할 수 있다.
- [176] 따라서, 위 [표 2]의 결과를 참고할 때, 오버샘플링을 수행하는 경우에, 탐지 모델의 부정거래에 대한 탐지능력이 향상되는 것을 확인할 수 있다.
- [177]
- [178] 다시 도 2를 참조하면, 오버샘플링부(140)는 통계적 특징 데이터셋(D1)을 오버샘플링함으로써 제1 데이터셋(D1')을 출력하고, 입력 코드 데이터셋(D2)을 오버샘플링함으로써 제2 데이터셋(D2')을 출력하고, 바이트 코드 데이터셋(D3)을 오버샘플링함으로써 제3 데이터셋(D3')을 출력할 수 있다.
- [179] 결과적으로, 데이터 전처리부(100)는 특정 사용자의 지갑주소에 대한 트랜잭션 데이터를 이용하여 생성한 통계적 특징들을 기초로 제1 데이터셋(D1')를 생성하여 출력하고, 상기 트랜잭션 데이터에 포함된 입력 코드를 기초로 제2 데이터셋(D2')을 생성하여 출력하고, 상기 사용자가 이용하는 스마트 컨트랙트에 포함된 바이트 코드를 기초로 제3 데이터셋(D3')을 생성하여 출력할 수 있다.
- [180] 이어서, 데이터 전처리부(100)에서 출력된 제1 내지 제3 데이터셋(D1', D2', D3')은 특징벡터 추출부(200)에 입력된다.
- [181] 특징벡터 추출부(200)는 제1 특징벡터 추출부(210), 제2 특징벡터 추출부(220) 및 제3 특징벡터 추출부(230)를 포함한다.
- [182] 이하에서는, 도 16 내지 도 20을 참조하여, 제1 특징벡터 추출부(210), 제2 특징벡터 추출부(220) 및 제3 특징벡터 추출부(230)의 각각의 구성 및 동작방법에 대해 구체적으로 설명하도록 한다.
- [183]
- [184] 도 16은 도 2의 제1 특징벡터 추출부의 구성을 나타내는 블록도이다. 도 17은 도 2의 제1 특징벡터 추출부의 동작을 예시적으로 설명하기 위한 도면이다.
- [185] 도 16 및 도 17을 참조하면, 제1 특징벡터 추출부(210)는 제1 데이터셋(D1')(즉, 트랜잭션 데이터를 이용하여 생성한 통계적 특징들을 기초로 생성된 데이터셋)을 입력받고, 이에 대한 출력으로 제1 특징벡터(CV1)를 출력한다.
- [186] 제1 특징벡터 추출부(210)는 1차원 컨볼루션 신경망(1-Dimensional Convolutional Neural Network; 이하, 1D-CNN)을 이용하여 제1 데이터셋(D1')에

대한 새로운 특징을 추출한다(S11). 다만, 본 발명에서 사용하는 컨볼루션 신경망이 1D-CNN에 한정되는 것은 아니며, 다차원으로 구성된 CNN 모듈을 이용할 수 있다. 다만 설명의 편의를 위하여 본 명세서에서는 1D-CNN을 예로 들어 설명하도록 한다.

- [187] 구체적으로, 1D-CNN은 통계적 특징들에 대한 공간적 정보(spatial information) 및 각 요소 간 공간적 상관관계를 포착(capture)할 목적으로 사용될 수 있다.
- [188] 제1 데이터셋(D1')의 동일한 행 내의 통계적 특징값은 서로 연관되어 있다. 예를 들어, 하나의 지갑주소에서 보낸 이더(ETH)의 총량, 최대값, 최소값, 평균값, 표준편차 등은 서로 관련이 있는 값들일 수 있다. 컨볼루션 신경망을 사용할 때 커널 크기(kernel size)가 4인 경우, 1D-CNN 모델은 “총량, 최대값, 최소값, 평균값”의 4 가지 특징을 고려하여 새로운 특징값을 얻을 수 있다.
- [189] 즉, 제1 특징벡터 추출부(210)에서 1D-CNN을 적용하는 경우, 설정된 커널 크기(kernel size)만큼 한 행 내의 통계적 특징값에 대한 새로운 특징을 추출할 수 있다.
- [190] 이어서, 제1 특징벡터 추출부(210)는 1D-CNN를 통과한 값에 글로벌 맥스풀링(global maxpooling)(또는, 맥스 풀링(maxpooling))을 적용하여 은닉 상태 벡터의 차원을 감소시킬 수 있다(S12). 이때, 글로벌 맥스풀링은 풀링 크기(pooling size) 내에서 최대값을 도출하므로, 벡터의 크기(예를 들어, width size)를 감소시킬 수 있고, 통계적 특징값 중에서 상대적으로 중요도가 낮은 요소들을 고려하지 않게 하므로 계산 복잡도 측면에서 효율적인 학습이 가능해진다. 글로벌 맥스풀링에 대한 내용은 이미 공지되어 있으므로, 여기에서 자세한 설명은 생략하도록 한다.
- [191] 이어서, 제1 특징벡터 추출부(210)는 글로벌 맥스풀링을 통과한 값에 소프트맥스(softmax) 함수를 적용한다(S13). 여기에서, 소프트맥스 함수는 입력값을 모두 확률 형태(Probability)로 변경하며, 소프트맥스 함수의 출력값을 모두 더하는 경우 총합이 1이 되도록 각각의 입력값을 0 내지 1의 범위 내로 압축시킨다. 즉, 제1 특징벡터 추출부(210)는 소프트맥스 함수를 적용함으로써, 각 특징값이 부정거래 여부를 탐지하는데 얼마나 영향을 미치는지를 수치화할 수 있다.
- [192] 여기에서, 소프트 함수( $\sigma(x)_i$ )는 하기 [수학식 3]으로 계산될 수 있다.
- [193]
- [194] [수학식 3]
- [195]
- $$\sigma(x_i) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}}$$
- [196] 여기에서, K는 구성요소(element)의 개수를 의미한다.
- [197] 소프트 맥스 함수는 다른 구성요소들에 대한 특정 요소의 상대적 중요도를 구할 수 있는 활성화 함수라는 점에서 이미 공지된 시그모이드 함수와는 차이가

있다.

- [198] 전술한 S11 단계 내지 S13 단계는 통계적 특징을 포함하는 제1 데이터셋(D1')에 중요도 가중치 게이트를 적용하는 과정으로 해석될 수 있다.
- [199] 한편, 제1 특징벡터 추출부(210)는 제1 데이터셋(D1')에 선형 인공신경망(Linear Artificial Neural Network; 이하, ANN)을 적용한다(S14).
- [200] 이어서, 제1 특징벡터 추출부(210)는 S13 단계의 결과값과, S14 단계의 결과값의 원소곱(element-wise product)을 계산한다(S15). 이때, 원소곱은 초기 통계적 특징값과 각 통계적 특징이 얼마나 가중될 것인지에 대한 수치를 곱함으로써, 부정거래를 탐지하는 데 중요한 역할을 하는 통계값에 높은 가중치를 부여할 수 있다.
- [201] 이어서, 제1 특징벡터 추출부(210)는 S15 단계의 결과값에 선형 인공신경망(예를 들어, ANN; 이하, Linear)과, 정류 선형 유닛(Rectified Linear Unit; 이하, ReLU) 함수를 순차적으로 적용함으로써 제1 특징벡터(CV1)를 출력한다(S16, S17).
- [202] 이때, 제1 특징벡터 추출부(210)에서 '중요도 가중치 게이트'를 적용하는 경우(즉, S11 단계 내지 S15 단계를 적용하는 경우)와, 중요도 가중치 게이트를 적용하지 않는 경우(즉, S11 단계 내지 S13 단계를 생략하고 S14 단계만을 수행하는 경우)를 나누어 학습 모델의 성능을 실험해보았으며, 이에 대한 결과는 다음 [표 2]과 같다. [표 3]의 정확도, 정밀도, 재현율, F1 스코어에 대한 설명은 전술하였으므로 이하에서 중복되는 설명은 생략하도록 한다.

[203]

[204] [표3]

| Case                    | 정확도(%) | 정밀도   | 재현율   | F1 스코어 |
|-------------------------|--------|-------|-------|--------|
| 중요도 가중치 게이트를 적용한 경우     | 91.1   | 0.703 | 0.853 | 0.762  |
| 중요도 가중치 게이트를 적용하지 않은 경우 | 84.2   | 0.528 | 0.934 | 0.674  |

- [205] 따라서, 위 [표 3]의 결과를 참고할 때, 중요도 가중치 게이트를 적용하는 경우(즉, S11 단계 내지 S15 단계를 적용하는 경우)에, 탐지 모델의 부정거래에 대한 탐지성능이 향상되는 것을 확인할 수 있다.
- [206]
- [207] 도 18은 도 2의 제2 특징벡터 추출부의 구성을 나타내는 블록도이다. 도 19는 도 2의 제2 특징벡터 추출부의 동작을 예시적으로 설명하기 위한 도면이다.

이하에서는 전술한 제1 특징벡터 추출부(210)에서 설명한 내용과 중복되는 내용은 생략하거나 간단히 하고, 차이점을 위주로 설명하도록 한다.

- [208] 도 18 및 도 19를 참조하면, 제2 특징벡터 추출부(220)는 제2 데이터셋(D2')(즉, 트랜잭션 데이터에서 추출된 입력 코드를 기초로 생성된 데이터셋)을 입력받고, 이에 대한 출력으로 제2 특징벡터(CV2)를 출력한다.
- [209] 우선, 제2 특징벡터 추출부(220)는 1차원 컨볼루션 신경망(1-Dimensional Convolutional Neural Network; 이하, 1D-CNN)과 글로벌 맥스풀링을 순차적으로 통과시킨 제1 중간값을 도출한다(S21, S22).
- [210] 또한, 제2 특징벡터 추출부(220)는 1차원 컨볼루션 신경망(1-Dimensional Convolutional Neural Network; 이하, 1D-CNN), 글로벌 맥스풀링, 및 소프트맥스 함수를 순차적으로 통과시킨 제2 중간값을 도출한다(S23, S24, S25).
- [211] 이어서, 제2 특징벡터 추출부(220)는 제1 중간값과 제2 중간값의 원소곱(element-wise product)을 계산한다(S26).
- [212] 이어서, 제2 특징벡터 추출부(220)는 선형 인공신경망(Linear; S27a), ReLU 함수(S27b), 배치 정규화(Batch Normalization)(S27c)를 순차적으로 적용한다(S27). 여기에서, 배치 정규화는 신경망에 배치 정규화 레이어를 삽입함으로써, 여러 입력값들을 모은 배치에 대하여, 각 레이어의 입력값의 평균과 표준편차를 다시 조절함으로써, 입력값이 한쪽으로 쏠리거나 너무 퍼지거나 너무 좁아지지 않도록 하는 인공신경망 기법을 의미한다. 배치 정규화를 수행하는 방법은 이미 공지되어 있으므로, 여기에서 자세한 설명은 생략하도록 한다.
- [213] 이어서, 제2 특징벡터 추출부(220)는 S27 단계의 결과값에 제2 데이터셋(D2')을 더한다(S28).
- [214] 이어서, 제2 특징벡터 추출부(220)는 선형 인공신경망(Linear; S28a)과 누설 ReLU(Leaky ReLU; S28b) 함수를 순차적으로 적용하여 제2 특징벡터(CV2)를 출력한다(S28). 여기에서, Leaky ReLU 함수는 이미 공지되어 있으므로, 여기에서 자세한 설명은 생략하도록 한다.
- [215] 이때, 제2 특징벡터 추출부(220)에서 CNN을 포함하는 중요도 가중치 게이트를 적용하는 경우와, RNN 계열 모델(예를 들어, LSTM 또는 GRU)을 적용한 경우를 나누어 학습 모델의 성능을 실험해보았으며, 이에 대한 결과는 다음 [표 3]과 같다. [표 4]의 정확도, 정밀도, 재현율, F1 스코어에 대한 설명은 전술하였으므로 이하에서 중복되는 설명은 생략하도록 한다.
- [216]

[217] [표4]

| Case                                  | 정확도(%) | 정밀도   | 재현율   | F1 스코어 |
|---------------------------------------|--------|-------|-------|--------|
| CNN을 포함한<br>중요도 가중치<br>게이트를<br>적용한 경우 | 91.1   | 0.703 | 0.853 | 0.762  |
| GRU                                   | 88.5   | 0.618 | 0.902 | 0.741  |
| LSTM                                  | 89.9   | 0.667 | 0.853 | 0.742  |

[218] 따라서, 위 [표 4]의 결과를 참고할 때, 중요도 가중치 게이트를 적용하는 경우, RNN 계열 모델만 사용한 경우보다 탐지 모델의 부정거래에 대한 탐지성능이 향상되는 것을 확인할 수 있다. 또한, 중요도 가중치 게이트에 포함된 CNN은 RNN을 사용하는 경우에 비해 계산 복잡도를 낮추고 모델 학습의 속도를 향상시키는 장점을 지닌다.

[219]

[220] 도 20은 도 2의 제3 특징벡터 추출부의 구성을 나타내는 블록도이다. 이하에서는 전술한 제1 특징벡터 추출부(210) 및 제2 특징벡터 추출부(220)에서 설명한 내용과 중복되는 내용은 생략하거나 간단히 하고, 차이점을 위주로 설명하도록 한다.

[221] 도 20을 참조하면, 제3 특징벡터 추출부(230)는 제3 데이터셋(D3')(즉, 컨트랙트 주소(CA)에서 작성된 바이트 코드를 기초로 생성된 데이터셋)을 입력받고, 이에 대한 출력으로 제3 특징벡터(CV3)를 출력한다.

[222] 우선, 제3 특징벡터 추출부(230)는 1차원 컨볼루션 신경망(1-Dimensional Convolutional Neural Network; 이하, 1D-CNN)과 글로벌 맥스풀링을 순차적으로 통과시킨 제1 중간값을 도출한다(S31, S32).

[223] 또한, 제3 특징벡터 추출부(230)는 1차원 컨볼루션 신경망(1-Dimensional Convolutional Neural Network; 이하, 1D-CNN), 글로벌 맥스풀링, 및 소프트맥스 함수를 순차적으로 통과시킨 제2 중간값을 도출한다(S33, S34, S35).

[224] 이어서, 제3 특징벡터 추출부(230)는 제1 중간값과 제2 중간값의 원소곱(element-wise product)을 계산한다(S36).

[225] 이어서, 제3 특징벡터 추출부(230)는 선형 인공신경망(Linear; S37)과 누설 ReLU(Leaky ReLU; S38) 함수를 순차적으로 적용하여 제3 특징벡터(CV3)를 출력한다(S37, S38).

[226] 이때, 제3 특징벡터 추출부(230)는 제2 특징벡터 추출부(220) 보다 더 많은 레이어(layer)를 사용한다. 이는 실험적으로 동일한 모델을 사용하는 경우, 입력 코드를 이용하는 모델이 바이트 코드를 이용하는 모델보다 탐지성능이 낮게 도출되었다. 따라서, 본 발명에서는 이를 보완하기 위해 제2 특징벡터

추출부(220)에서 제3 특징벡터 추출부(230)보다 많은 레이어를 사용하도록 설계하였다.

- [227] 이때, 제2 특징벡터 추출부(220)에서 사용된 배치 정규화는 모델 학습 시 학습 흐름(flow)이 활성화 함수(ex. ReLU 등)를 지날 때마다 공변량 변화(Covariate shift) 현상이 나타나는 문제를 해소해준다. 즉, 배치 정규화는 학습 과정을 안정화해주는 기능을 수행한다. [표 5]는 제2 특징벡터 추출부(220)와 제3 특징벡터 추출부(230)의 성능을 비교한 결과를 나타낸다. [표 5]의 정확도, 정밀도, 재현율, F1 스코어에 대한 설명은 전술하였으므로 이하에서 중복되는 설명은 생략하도록 한다.

[228]

[229] [표5]

| Case            | 정확도(%) | 정밀도   | 재현율   | F1 스코어 |
|-----------------|--------|-------|-------|--------|
| 바이트 코드를 이용하는 모델 | 87.9   | 0.602 | 0.918 | 0.726  |
| 입력 코드를 이용하는 모델  | 91.1   | 0.703 | 0.853 | 0.762  |

- [230] 이어서, 다시 도 2를 참조하면, 제1 특징벡터 추출부(210)에서 출력된 제1 특징벡터(CV1), 제2 특징벡터 추출부(220)에서 출력된 제2 특징벡터(CV2), 및 제3 특징벡터 추출부(230)에서 출력된 제3 특징벡터(CV3)는 특징벡터 추론부(300)에 전달된다.

- [231] 이하에서는, 도 21을 참조하여 특징벡터 추론부(300)의 구성 및 동작에 대해 살펴보도록 한다.

[232]

- [233] 도 21은 도 2의 특징벡터 추론부의 구성을 나타내는 블록도이다.

- [234] 도 21을 참조하면, 특징벡터 추론부(300)는 제1 내지 제3 특징벡터(CV1, CV2, CV3)을 연결(concatenation)하고, 레이어 정규화(Layer Normalization), 피드-포워드 네트워크(feed-forward network), 및 시그모이드(sigmoid) 함수를 이용하여 부정거래 여부에 대한 예측값(Prediction)을 출력한다.

- [235] 우선, 특징벡터 추론부(300)는 제1 내지 제3 특징벡터(CV1, CV2, CV3)을 연결한다(S41). 여기에서, 연결(concatenation)은, 둘 이상의 특징벡터들을 서로 연결하는 것을 의미한다.

- [236] 특징벡터 추론부(300)는 통계적 특징, 입력 코드, 및 바이트 코드의 서로 다른 정보의 흐름을 결합하는 방식으로 복수 개의 흐름을 연결시킬 수 있다. 특징벡터 추론부(300)는 제1 내지 제3 특징벡터(CV1, CV2, CV3)을 연결함으로써 탐지 모델이 부정거래 여부를 탐지할 때, 앞서 전처리한 통계적 특징, 입력 코드, 및 바이트 코드를 모두 고려할 수 있도록 한다.

- [237] 이어서, 특징벡터 추론부(300)는 연접된 특징벡터에 대해 레이어 정규화(Layer Normalization)를 수행한다(S42). 이 과정에서 사용된 레이어 정규화는 학습 안정화에 도움을 준다. 이때, 레이어 정규화의 방법은 이미 공지된 내용이므로, 여기에서 자세한 설명은 생략하도록 한다.
- [238] 이어서, 특징벡터 추론부(300)는 선형 인공신경망(Linear)과 누설 ReLU 함수를 교번하여 적용함으로써 피드-포워드 네트워크 구조를 형성하고, 레이어 정규화된 데이터를 통과시킨다(S43). 피드-포워드 네트워크 구조는 더 깊은 모델을 만들게 하면서 탐지 모델의 학습시 추상화 수준을 높이는데 도움을 준다.
- [239] 이어서, 특징벡터 추론부(300)는 S41 단계의 제1 출력값과, S43 단계의 제2 출력값을 더한다(S44).
- [240] 이어서, 특징벡터 추론부(300)는 S44 단계에서 더해진 값에 대해 선형 인공신경망(Linear)과, 시그모이드 함수를 순차적으로 통과시켜 부정거래 여부에 대한 예측값(Prediction)을 출력한다(S45, S46). 여기에서, 시그모이드 함수 ( $\sigma(x)$ )는 하기 [수학식 4]로 계산되는 활성화 함수이다.
- [241]
- [242] [수학식 4]
- [243]
- $$\sigma(x) = \frac{1}{1 + e^{-x}}$$
- [244] 시그모이드 함수는 부정거래 탐지 여부에 대해 “부정거래” 또는 정상거래 중 하나를 분류하는 이진 분류 모델에 해당한다. 따라서, 시그모이드 함수는 S45 단계의 출력값이 미리 정한 기준치(예를 들어, 0.5)보다 큰 경우, “부정거래”로 판단하고, 미리 정한 기준치(예를 들어, 0.5)보다 작은 경우, “정상거래”로 판단할 수 있다.
- [245] 특징벡터 추론부(300)에서 S42 단계 내지 S45 단계는 트랜스포머(transformer)의 인코더 구조에 해당하며, 해당 구조를 적용하는 경우와, 해당 구조를 적용하지 않는 경우를 나누어 학습 모델의 성능을 실험해본 결과는 다음 [표 6]과 같다. [표 6]의 정확도, 정밀도, 재현율, F1 스코어에 대한 설명은 전술하였으므로 이하에서 중복되는 설명은 생략하도록 한다.
- [246]
- [247] [표6]

| Case                  | 정확도(%) | 정밀도   | 재현율   | F1 스코어 |
|-----------------------|--------|-------|-------|--------|
| 트랜스포머의 인코더 구조를 적용한 경우 | 91.1   | 0.703 | 0.853 | 0.762  |
| 위 구조를 적용하지 않은 경우      | 90.8   | 0.699 | 0.837 | 0.760  |

- [248] 따라서, 위 [표 6]의 결과를 참고할 때, 트랜스포머의 인코더 구조를 적용하는 경우, 부정거래에 대한 탐지성능이 향상되는 것을 확인할 수 있다.
- [249]
- [250] 도 22는 본 발명의 몇몇 실시예에 따른 부정거래 탐지 방법을 수행하는 시스템의 하드웨어 구현을 설명하기 위한 도면이다.
- [251] 도 22를 참조하면, 본 발명의 몇몇 실시예들에 따른 부정거래 탐지 방법을 수행하는 서버(100)는 전자 장치(1000)로 구현될 수 있다. 전자 장치(1000)는 컨트롤러(1010, controller), 입출력 장치(1020, I/O), 메모리 장치(1030, memory device), 인터페이스(1040, interface) 및 버스(1050, bus)를 포함할 수 있다. 컨트롤러(1010), 입출력 장치(1020), 메모리 장치(1030) 및/또는 인터페이스(1040)는 버스(1050)를 통하여 서로 결합될 수 있다. 이때, 버스(1050)는 데이터들이 이동되는 통로(path)에 해당한다.
- [252] 구체적으로, 컨트롤러(1010)는 CPU(Central Processing Unit), MPU(Micro Processor Unit), MCU(Micro Controller Unit), GPU(Graphic Processing Unit), 마이크로프로세서, 디지털 신호 프로세서, 마이크로컨트롤러, 어플리케이션 프로세서(AP, application processor) 및 이들과 유사한 기능을 수행할 수 있는 논리 소자들 중에서 적어도 하나를 포함할 수 있다.
- [253] 입출력 장치(1020)는 키패드(keypad), 키보드, 터치스크린 및 디스플레이 장치 중 적어도 하나를 포함할 수 있다.
- [254] 메모리 장치(1030)는 데이터 및/또는 프로그램 등을 저장할 수 있다.
- [255] 인터페이스(1040)는 통신 네트워크로 데이터를 전송하거나 통신 네트워크로부터 데이터를 수신하는 기능을 수행할 수 있다. 인터페이스(1040)는 유선 또는 무선 형태일 수 있다. 예컨대, 인터페이스(1040)는 안테나 또는 유무선 트랜시버 등을 포함할 수 있다. 도시하지 않았지만, 메모리 장치(1030)는 컨트롤러(1010)의 동작을 향상시키기 위한 동작 메모리로서, 고속의 디램 및/또는 에스램 등을 더 포함할 수도 있다. 메모리 장치(1030)는 내부에 프로그램 또는 어플리케이션을 저장할 수 있다.
- [256] 본 발명의 실시예들에 따른 서버(100) 및 사용자 단말(1100)은 각각 복수의 전자 장치(1000)가 네트워크를 통해서 서로 연결되어 형성된 시스템일 수 있다. 이러한 경우에는 각각의 모듈 또는 모듈의 조합들이 전자 장치(1000)로 구현될 수 있다. 단, 본 실시예가 이에 제한되는 것은 아니다.
- [257] 추가적으로, 서버(100)는 워크스테이션(workstation), 데이터 센터, 인터넷 데이터 센터(internet data center(IDC)), DAS(direct attached storage) 시스템, SAN(storage area network) 시스템, NAS(network attached storage) 시스템, RAID(redundant array of inexpensive disks, or redundant array of independent disks) 시스템, 및 EDMS(Electronic Document Management) 시스템 중 적어도 하나로 구현될 수 있으나, 본 실시예가 이에 제한되는 것은 아니다.
- [258] 또한, 서버(100)는 사용자 단말(1100)을 이용하여 네트워크를 통해서 데이터를

전송할 수 있다. 네트워크는 유선 인터넷 기술, 무선 인터넷 기술 및 근거리 통신 기술에 의한 네트워크를 포함할 수 있다. 유선 인터넷 기술은 예를 들어, 근거리 통신망(LAN, Local area network) 및 광역 통신망(WAN, wide area network) 중 적어도 하나를 포함할 수 있다.

- [259] 무선 인터넷 기술은 예를 들어, 무선랜(Wireless LAN: WLAN), DMNA(Digital Living Network Alliance), 와이브로(Wireless Broadband: Wibro), 와이맥스(World Interoperability for Microwave Access: Wimax), HSDPA(High Speed Downlink Packet Access), HSUPA(High Speed Uplink Packet Access), IEEE 802.16, 롱 텀 에볼루션(Long Term Evolution: LTE), LTE-A(Long Term Evolution-Advanced), 광대역 무선 이동 통신 서비스(Wireless Mobile Broadband Service: WMBS) 및 5G NR(New Radio) 기술 중 적어도 하나를 포함할 수 있다. 단, 본 실시예가 이에 제한되는 것은 아니다.
- [260] 근거리 통신 기술은 예를 들어, 블루투스(Bluetooth), RFID(Radio Frequency Identification), 적외선 통신(Infrared Data Association: IrDA), UWB(Ultra-Wideband), 지그비(ZigBee), 인접 자장 통신(Near Field Communication: NFC), 초음파 통신(Ultra Sound Communication: USC), 가시광 통신(Visible Light Communication: VLC), 와이 파이(Wi-Fi), 와이 파이 다이렉트(Wi-Fi Direct), 5G NR (New Radio) 중 적어도 하나를 포함할 수 있다. 단, 본 실시예가 이에 제한되는 것은 아니다.
- [261] 네트워크를 통해서 통신하는 서버(100)는 이동통신을 위한 기술표준 및 표준 통신 방식을 준수할 수 있다. 예를 들어, 표준 통신 방식은 GSM(Global System for Mobile communication), CDMA(Code Division Multi Access), CDMA2000(Code Division Multi Access 2000), EV-DO(Enhanced Voice-Data Optimized or Enhanced Voice-Data Only), WCDMA(Wideband CDMA), HSDPA(High Speed Downlink Packet Access), HSUPA(High Speed Uplink Packet Access), LTE(Long Term Evolution), LTEA(Long Term Evolution-Advanced) 및 5G NR(New Radio) 중 적어도 하나를 포함할 수 있다. 단, 본 실시예가 이에 제한되는 것은 아니다.
- [262] 이상 첨부된 도면을 참조하여 본 발명의 실시예를 설명하였지만, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자는 본 발명이 그 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 실시될 수 있다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다.

## 청구범위

- [청구항 1] 암호화폐의 트랜잭션 데이터를 수집하는 부정거래 탐지 서버에서 수행되는 부정거래 탐지 방법에 있어서,  
특정 지갑주소에 대한 상기 트랜잭션 데이터를 기초로 복수의 통계적 특징을 도출하여 제1 데이터셋을 생성하는 단계;  
상기 지갑주소와 관련된 스마트 컨트랙트를 실행시키며, 상기 트랜잭션 데이터에 포함된 입력 코드(input code)를 이용하여 제2 데이터셋을 생성하는 단계;  
상기 스마트 컨트랙트의 배포시 사용되는 컨트랙트 주소의 바이트 코드를 이용하여 제3 데이터셋을 생성하는 단계;  
미리 학습된 딥러닝 모듈을 이용하여, 상기 제1 내지 제3 데이터셋에 대해 각각 제1 내지 제3 특징벡터를 추출하는 단계; 및  
추출된 상기 제1 내지 제3 특징벡터를 이용하여, 상기 지갑주소에 대한 부정거래여부를 추론하는 단계를 포함하는 부정거래 탐지 방법.
- [청구항 2] 제1 항에 있어서,  
상기 제1 내지 제3 데이터셋 중 적어도 하나의 데이터셋을 대상으로 데이터의 수를 증가시키는 오버샘플링을 수행하는 단계를 더 포함하되,  
상기 오버샘플링을 수행하는 단계는,  
상기 제1 내지 제3 데이터셋에 포함되고, 서로 인접한 복수의 데이터를 선정하는 단계와,  
상기 선정된 복수의 데이터에 대해 SMOTE 기법을 이용하여 추가 데이터를 생성하는 단계를 포함하는 부정거래 탐지 방법.
- [청구항 3] 제2 항에 있어서,  
상기 오버샘플링을 수행하는 단계는,  
상기 제2 데이터셋 및 상기 제3 데이터셋을 대상으로 상기 오버샘플링을 수행하는 것을 포함하는 부정거래 탐지 방법.
- [청구항 4] 제1 항에 있어서,  
상기 제1 데이터셋을 생성하는 단계는,  
상기 지갑주소에 대한 복수의 트랜잭션 데이터를 수신하는 단계와,  
상기 복수의 트랜잭션 데이터를 기초로 서로 다른 통계적 특징을 도출하는 단계와,  
상기 도출된 통계적 특징의 값이, 0 내지 1 사이에 분포되도록, 정규화를 수행하는 단계를 포함하는 부정거래 탐지 방법.
- [청구항 5] 제4 항에 있어서,  
상기 통계적 특징은,  
상기 지갑주소에 수신된 트랜잭션 데이터를 기초로 생성된 제1 통계적

특징과,

상기 지갑주소에서 송신된 트랜잭션 데이터를 기초로 생성된 제2 통계적 특징을 포함하는 부정거래 탐지 방법.

[청구항 6]

제5 항에 있어서,

상기 제1 데이터셋을 생성하는 단계는,

상기 수신된 트랜잭션 데이터 또는 상기 송신된 트랜잭션 데이터가 존재하지 않아, 상기 제1 통계적 특징 또는 상기 제2 통계적 특징에 결측치(*none*)가 생기는 경우, 상기 결측치의 값을 미리 정해진 값으로 대체하는 단계를 더 포함하는 부정거래 탐지 방법.

[청구항 7]

제1 항에 있어서,

상기 제2 데이터셋을 생성하는 단계는,

상기 지갑주소에 대한 복수의 트랜잭션 데이터를 수신하는 단계와,  
상기 복수의 트랜잭션 데이터에 포함된 복수의 입력 코드를 추출하는 단계와,

상기 추출된 복수의 입력 코드를 연결하는 단계와,

상기 연결된 입력 코드를 미리 정해진 단위로 분리하는 단계와,

상기 분리된 입력 코드를 대상으로 정수 인코딩을 수행하는 단계를 포함하는

부정거래 탐지 방법.

[청구항 8]

제7 항에 있어서,

상기 제2 데이터셋을 생성하는 단계는,

상기 정수 인코딩된 각 데이터의 문자열이 미리 정해진 문자열 최대 길이보다 작은 경우, 해당 문자열에 미리 정해진 문자를 추가하고,

상기 정수 인코딩된 각 데이터의 문자열이 상기 문자열 최대 길이보다 큰 경우, 해당 문자열에서 상기 문자열 최대 길이를 초과하는 문자열을 삭제하는 것을 포함하는 부정거래 탐지 방법.

[청구항 9]

제1 항에 있어서,

상기 제3 데이터셋을 생성하는 단계는,

상기 컨트랙트 주소와 관련된 바이트 코드를 추출하는 단계와,

상기 추출된 바이트 코드를 미리 정해진 단위로 분리하는 단계와,

상기 분리된 바이트 코드를 대상으로 정수 인코딩을 수행하는 단계를 포함하는 부정거래 탐지 방법.

[청구항 10]

제9 항에 있어서,

상기 제3 데이터셋을 생성하는 단계는,

상기 정수 인코딩된 각 데이터의 문자열이 미리 정해진 문자열 최대 길이보다 작은 경우, 해당 문자열에 미리 정해진 문자를 추가하고,

상기 정수 인코딩된 각 데이터의 문자열이 상기 문자열 최대 길이보다 큰 경우, 해당 문자열에서 상기 문자열 최대 길이를 초과하는 문자열을

- 삭제하는 것을 포함하는 부정거래 탐지 방법.
- [청구항 11] 제1 항에 있어서,  
 상기 제1 내지 제3 특징벡터를 추출하는 단계는,  
 상기 제1 데이터셋을 미리 학습된 컨볼루션 신경망에 인가하는 단계와,  
 상기 컨볼루션 신경망을 통과한 값에 맥스풀링을 적용하는 단계와,  
 상기 맥스풀링을 통과한 값에 소프트맥스 함수를 적용하는 단계를  
 포함하는 부정거래 탐지 방법.
- [청구항 12] 제1 항에 있어서,  
 상기 제1 내지 제3 특징벡터를 추출하는 단계는,  
 상기 제2 데이터셋 또는 상기 제3 데이터셋을 미리 학습된 제1 컨볼루션  
 신경망에 인가하는 단계와,  
 상기 제1 컨볼루션 신경망을 통과한 값에 제1 맥스풀링을 적용하는  
 단계와,  
 상기 제2 데이터셋을 미리 학습된 제2 컨볼루션 신경망에 인가하는  
 단계와,  
 상기 제2 컨볼루션 신경망을 통과한 값에 제2 맥스풀링을 적용하는  
 단계와,  
 상기 제2 맥스풀링이 적용된 값에 소프트맥스 함수를 적용하는 단계와,  
 상기 제1 맥스풀링이 적용된 제1 중간값과, 상기 소프트맥스 함수가  
 적용된 제2 중간값의 원소곱을 계산하는 단계를 포함하는 부정거래 탐지  
 방법.
- [청구항 13] 제12 항에 있어서,  
 상기 원소곱의 결과값에 선형 인공신경망(ANN), 정류 선형 유닛  
 함수(ReLU), 및 배치 정규화를 순차적으로 적용하는 단계를 더 포함하는  
 부정거래 탐지 방법.
- [청구항 14] 제12 항에 있어서,  
 상기 원소곱의 결과값에 선형 인공신경망(ANN) 및 누설 정류 선형 유닛  
 함수(Leaky ReLU)를 순차적으로 적용하는 단계를 더 포함하는 부정거래  
 탐지 방법.
- [청구항 15] 제1 항에 있어서,  
 상기 부정거래여부를 추론하는 단계는,  
 상기 제1 내지 제3 특징벡터를 연접(concatenation)하는 단계와,  
 상기 연접된 제1 내지 제3 특징벡터에 레이어 정규화(Layer  
 Normalization)를 수행하는 단계와,  
 상기 레이어 정규화를 수행한 값을 피드-포워드 네트워크에 인가하는  
 단계와,  
 상기 피드-포워드 네트워크를 통과한 값과, 상기 연접된 제1 내지 제3  
 특징벡터를 더한 중간값을 도출하는 단계와,

상기 중간값에 시그모이드 함수를 적용하여, 상기 부정거래에 대한 판단 결과를 출력하는 단계를 포함하는 부정거래 탐지 방법.