ADPP: A NOVEL ANOMALY DETECTION AND PRIVACY-PRESERVING FRAMEWORK USING BLOCKCHAIN AND NEURAL NETWORKS IN TOKENOMICS

Wei Yao*, Jingyi Gu*, Wenlu Du*, Fadi P. Deek and Guiling Wang

New Jersey Institute of Technology, Newark, NJ, USA
*: These Authors contributed equally
wy95, jg95, wd48, fadi.deek, and gwang @njit.edu

ABSTRACT

The increasing popularity of crypto assets has resulted in greater cryptocurrency investor interest and more exposure in both industry and academia. Despite the substantial socioeconomic benefits, the anonymous character of cryptocurrency trading makes it prone to abuse and a magnet for illicit purposes, which cause monetary losses for individual traders and erosion in the standing of the tokenomics industry. To regulate the illicit behavior and secure users' privacy for cryptocurrency trading, we present an Anomaly Detection and Privacy-Preserving (ADPP) Framework integrating blockchain and deep learning technologies. Specifically, ADPP leverages blockchain technologies to build a user management platform that ensures anonymity and enhances the privacy-preservation of user information. Atop the user management system, an Anomaly Detection System adapts neural networks and imbalanced learning on topological cryptocurrency flow among users to identify anomalous addresses and maintain a sanction list repository. The experiments on the real-world dataset demonstrate the effectiveness and superior performance of ADPP. The flexible framework can be easily generalized to the crypto assets with public real-time transaction (e.g., Non-fungible Token), which takes up a significant proportion of market capitalization in the domain of tokenomics.

KEYWORDS

Cryptocurrency Anomaly Detection, Privacy-Preservation, Graph Neural Networks, Blockchain, Imbalanced Learning

1. INTRODUCTION

There are over 20,000 cryptocurrencies with over 1 trillion of total market capitalization, as of June 2022 [1], revealing the trend of technological innovation to digitize in the 21st century global economy. The continued evolution and popularity of cryptocurrency is happening for a reason: it enables efficient payment systems where trading is independent of any political interference or governmental regulatory bureaucracy, occurring rather through a decentralized distributed ledger. Yet there exists, for a long time, a discourse regarding the process in which cryptocurrency transactions can be regulated to prevent criminality. In fact, since its booming, a widespread illicit behavior in the cryptocurrency markets has been recorded as hackers engage in phishing attacks and attempt to steal credentials of traders and other sensitive privacy data, such as account information. These unlawful activities are especially prevalent in the Bitcoin market, with more than one-quarter of all users and close to one-half of bitcoin transactions being associated with illegal activities in 2017 and around 76 billion in illegal transactions annually [2]. Developing countermeasures to mitigate cybercriminality and the illicit use of cryptocurrencies is hence an exceptionally important task to ensure integrity of their transactions and continued evolution of the digital assets market.

As policymakers focus on regulatory matters, it seems also prudent for researchers and practitioners to explore technological remedies through user management platforms that can ultimately enhance the security and preserve the integrity of data and identity as well as to stave off any potential cybercriminality at the very beginning stages. Nonetheless, user management itself is a challenging task due to the diversification of end-use applications where one individual may have multiple virtual identities. Some solutions, like OpenID, can provide more simplicity allowing a single sign-on function to access different web service providers. In fact, lacking complete self-management for identities and privacy-preserving of sensitive information still remains an important open question to be addressed. One approach toward this end has been the use of traditional Public Key Infrastructure (PKI). However, the PKI may suffer from data loss and corruption due to the problem of centralized management, such as a Single-Point-of-Failure (SPOF). Others, like the Liberty Alliance Project [3], advocate for federated authentication techniques. Such federated methods, although breaking down the centralized management to several trusted circles, still do not fully resolve the issue when the management systems (i.e., service providers) shutdown and become unavailable. Blockchain technology, however, has proven to be effective in many fields yet has not been sufficiently applied to user management systems toward mitigating illicit behavior in cryptocurrency trading.

More diverse countermeasures can certainly be coupled with the above-mentioned security-enhanced and privacy-preserving identity management platform to increase effectiveness. As a primary preventative approach, detection-based countermeasures is a promising solution. These methods require historical transactions of cryptocurrencies (e.g., Bitcoin) and assume that illicit transactions happened in the first place. Through the training, the anomaly behavior will be identified accurately; hence, any future cybercriminality can be prevented. Machine learning algorithms and more recently neural networks, especially Graph Convolutional Networks (GCN) [4] and Gated Recurrent Units (GRU), are proposed to train an intelligent detection system that can evolve over time. These approaches can analyze based on various characteristics, e.g., topological networks and temporal interactions, yet they also easily suffer from an imbalanced class problem. However, findings reveal that they consider only part of these features and are not capable of learning in a comprehensive view.

Drawing on the above insights from the study, we propose an Anomaly Detection and Privacy-Preserving (ADPP) Framework which fully capitalizes on the advantages of blockchain and deep learning techniques for cryptocurrency trade. First, leveraging the Self-Sovereign Identity (SSI) framework that gives individuals control over their own identities, ADPP uses blockchain technologies to build a user management platform with a workflow of provision, issuance, and verification. It reduces the loss caused by unidentified users and enhances privacy-preserving of user identity information protection.

Besides, the Anomaly Detection System (ADS) is smoothly integrated into the platform to regulate the trade behavior of the crypto market. The nature of ADS is a binary classification deep model aiming to detect potential illicit addresses. It leverages GCN, GRU, and imbalanced learning to learn topological and dynamic cryptocurrency transaction networks. The suspicious illicit addresses detected by ADS are added to the sanction list repository. ADPP will trigger a warning if illicit addresses are detected during the verification process before two parties start the trade.

The experiments on a real-world dataset demonstrate the effectiveness of ADPP. The flexible framework can be generalized to those crypto assets which have public real-time transactions with addresses and features in the market, such as cryptocurrencies and Non-fungible Tokens (NFTs). Such crypto assets take a significant proportion of market capitalization in the domain of tokenomics. The main contributions of this paper can be summarized as follows:

1. The blockchain-based privacy-preserving authentication platform leverages the cryptographic technologies on blockchain ledger to authorize validated users for cryptocurrency trading without leaking users' privacy.
2. Anomaly Detection System based on GCN, GRU, and imbalanced learning captures topological patterns and dynamic changes of cryptocurrency flow between users. It aims to detect potential illicit addresses and maintain a Sanction List Repository to prevent illicit behaviors.
3. To the best of our knowledge, we are the first to integrate a privacy-preserving authentication platform with a cryptocurrency anomaly detection model based on deep learning. The resulting work effectively mitigates cybercriminality by enhancing privacy and identifying potential illicit addresses. This integration can similarly be generalized to other crypto assets.

The remaining parts of this paper are organized as follows: Section 2 provides an overview of related works. Section 3 introduces the technical background. Section 4 presents the architecture of the proposed platform, detailed system design with core internal workflow, algorithmic protocol, and a deep learning model-based anomaly detection system. Section 5 presents the experimental results and critical analysis. Finally, Section 6 offers concluding remarks and presents research challenges and opportunities to motivate future work.

2. RELATED WORKS

In this section, we introduce related work from three perspectives: identity provider-based platform, anomaly detection model and any studies that integrate blockchain-based privacy-preserved platform with deep learning technologies. The discussion discloses the dearth of prior research in privacy-preserving frameworks with functionality of detecting anomaly, and an urgent need to apply Blockchain technology for protecting user privacy.

2.1. Identity Provider Based Platform

Cross-platform and privacy-preserving are two major considerations in terms of designing a identity management system. Aiming toward cross-platform identity management, Microsoft experimented with Identity Management (IdM) on Microsoft.NET Passport [5], allowing users to log in to multiple service providers' websites with the same identity provided by Microsoft as the Identity Provider (IdP). The IdP allows users to access the different websites and redirects to different user profiles through the identity managed by the IdP. However, the drawbacks of centralized management are apparent, as all service providers and customers are affected if the Identity Provider (IdP) becomes unavailable or as a result of data loss and corruption. To tackle the problem of centralization, the Liberty Alliance Project [3] made another attempt at the IdP to build a federated network identity management. It advocates for a federated IdM method, which focuses on establishing trust circles between service providers and federated isolated customers. However, the problem with federated IdM is that once an authenticated service provider is unavailable, customers will not be able to access the resources of that provider. From the security perspective, OAuth (Open Authorization) [6] allows service providers' websites to safely access various identity information stored in the IdP with the user's authorization. In the authentication process, the customer can access the service provider's website without providing their account password but in the form of a token to the service provider's website. OAuth is widely used in the current internet environment since its usability and performance are improved. However, OAuth still suffers from some security vulnerabilities [7]. Another disadvantage of the OAuth framework is that the IdP serves as a centralized identity provider, which in some situations may yield data loss. For instance, if when accessing any website using OAuth provided by Google and it blocks that service provider, the customer will lose the access rights and data on that website.

2.2. Anomaly Detection

Artificial Intelligence has been increasingly playing an essential role in detecting and preventing cybercrime [8]. Standard machine learning algorithms (e.g., Random Forest [9]) can be applied to anomaly detection tasks which, by its nature, are binary classification problems. Such algorithms are demonstrated to be effective [10, 11]. These methods can extract phishing fraud features from the data collected and show a stable performance, however, they do not leverage any graph information. Since cryptocurrency trading can be naturally formulated as a transaction graph, a line of work [12, 13, 14] employed graph-based algorithms to detect illicit behavior. With the continued growth of deep learning, graph neural networks, such as GCN [4] and graph attention networks (GAT) [15] have been applied to detect malicious transactions. Based on an analysis of the weaknesses of attackers a heterogeneous graph neural network with an attention mechanism is also proposed [13]. The main drawback of the graph-based algorithms is the neglect of the temporal dependencies of multivariate time series data as identifying an anomaly in real time is a known challenge due to the frequency of seasonal behavior or a change in trend [16]. Statistical models, such as Autoregressive Integrated Moving Average (ARIMA) [17], and recurrent neural networks, such as Long Short-Term Memory (LSTM) [18] and GRU [19] are common approaches to address the temporal dependencies. Another challenge is the imbalance problem in the transaction dataset where illicit transactions are minority and only occupy a small portion of the whole. Some existing work leverages the imbalanced learning techniques: sampling methods drop majority samples or repeat minority samples randomly to control the imbalance ratio directly [20]; and re-weighting algorithms adaptively assign and adjust the weight on samples based on their classification performance [21]. The advantage of our work over existing approaches is that we consider both temporal dependence and the imbalance problem in anomaly detection.

2.3. Blockchain-Based Privacy-Preserved Platform Integrated with Machine Learning (ML)/Deep Learning (DL)

There exists some efforts toward integrating blockchain with ML/DL techniques. A privacy-preserving framework in smart power networks has been presented [22]. Specifically, a blockchain-based two-level privacy module is designed to verify data integrity and LSTM is employed for anomaly detection. A similar idea is proposed [23], where a two-level privacy-preservation approach with a blockchain module to transmit data and a standard machine learning algorithm (i.e., Principal Component Analysis) for data transformation is designed. Such integration of blockchain with the ML technique is applied into Internet of Things (IoT)-driven smart cities. However, we found no work that integrates blockchain with ML/DL in cryptocurrency trading. Additionally, the integration frameworks proposed are generally tailored for specific domains and thus difficult to migrate to other fields.

3. TECHNICAL BACKGROUND

In this section, the basic underlying knowledge regarding privacy in cryptocurrency trading and blockchain-based identity management framework will be described. In addition, the deep learning algorithms that we build upon are described.

3.1. Privacy Preserving

Privacy preserving involves two aspects. (1) **Transaction Privacy** refers to the identity information in transactions of cryptocurrency trading, which is essential since the transactions are published on a public blockchain. To achieve anonymity and protect transaction privacy, cryptocurrency platforms, such as Bitcoin, use the address as the presentation for the trading parties, usually called pseudonymous, to hide the real identity information of both parties. The addresses are essentially the hash values of the public keys of both parties. (2) **Identity Privacy**

refers to the identity information of the two parties of a cryptocurrency transaction in the real world. Although the user's identity in a cryptocurrency transaction is not associated with the identity in the real world, there are still some situations that may result in identity privacy leaking. For instance, by leveraging clustering and other technologies to analyze a transaction, find out the transaction relationship map between different accounts, and infer the input and output of the transaction, the real identity information of the two parties can finally be targeted. Since the blockchains for cryptocurrency trading use pseudonym addresses to protect transaction privacy, the framework proposed in this paper mainly focuses on identity privacy.

3.2. Self-Sovereign Identity

Self-Sovereign Identity (SSI) is a novel distributed identity management framework that gives individuals control over their digital and physical identities and is mostly built upon a blockchain. Traditionally users manage their digital identities either through their accounts on each identity provider, or by relying on a manager of different identity providers such as Google Sign-In. However, traditional methodologies result in (a) neither identity providers allowing users to control their information themselves; and (b) a limitation if users want to present their identity information to other individuals or service providers. In the latter case, the identity providers, who have full control, must provide approaches for the verification process. In other words, the users cannot control the information presented. In an SSI system, contrarily, if a particular identity provider authorizes the identities, these are maintained by the individuals, not the provider. To achieve individual control over identity information, SSI employs blockchain technology: the identity providers can publish their cryptographic authentication on the Distributed Ledger Technology (DLT); thus, other individuals can cryptographically verify the identities without interaction with the identity providers.

There are two major standard specifications. (1) **Decentralized Identifiers (DID)** is a W3C standard specification [24] for SSI. A DID can be resolved to a DID Document that provides the subject's identifying information (to whom the DID belongs). The DID Document contains the subject's authentication information, such as the public keys needed to validate the subject's signature. It also includes service endpoints that identify the URL where the subject's verification information can be retrieved. (2) **Verifiable Credential (VC)** is a W3C standard specification [25] for DID-based cryptographically verifiable digital credentials. It can represent the same information as a physical credential. With blockchain technologies, a VC can be issued by an *issuer* in a more tamper-evident and trustworthy method to a *holder*. A VC contains claims about the *holder*, certified by the *issuer*.

In addition, **Wallet** is an application that generates private and public key pairs and securely stores the key-pairs. The private keys can be used to prove ownership of DIDs and VCs cryptographically. The received credentials are also stored in the wallet.

3.3. Graph Convolutional Network

Traditional convolutional networks capture local information by a sliding filter on images or grids. GCN is the generalization form of convolutional network on graphs, which extracts the topological features to generate new node representations. It automatically learns not only the characteristics of central nodes but also the associated information from connected neighbors. Given an undirected graph $G = (V, E)$ with $N = |V|$ nodes and $M = |E|$ edges, $A \in \mathbb{R}^{N \times N}$ is the adjacency matrix of G where each element $A_{i,j}$ is 1 if node i and j is connected, 0 otherwise. $X \in \mathbb{R}^{N \times P}$ is a feature matrix. GCN is denoted as:

$$X'_t = \sigma\left(\widehat{A_t} X_t W_t\right), \widehat{A_t} = D_t^{-\frac{1}{2}}(A_t + I)D_t^{-\frac{1}{2}} \quad (1)$$

where $\hat{A}$ is the adjacency matrix normalized by the neighbors' degree. The self-loop constant matrix I considers the influence of the node itself. The degree matrix $D \in \mathbb{R}^{N \times N}$ is a diagonal

matrix composed of node degrees. W is a trainable weight matrix and $\sigma(\cdot)$ represents the activation function. We refer the reader to [4] for more details.

3.4. Gated Recurrent Units

GRU is simple, yet effective and fast to obtain time series patterns. Let X_t denote the training input. The key design of GRU consists of reset gate R_t takes account of short-term memory, determining how much the previous states are kept. Update gate Z_t is responsible for long-term memory, controlling how much new information to pass along into the future.

$$R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r) \tag{2}$$

$$Z_t = \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z) \tag{3}$$

$$\widetilde{H_t} = tanh(X_t W_{xh} + (R_t \odot H_{t-1}) W_{hh} + b_h) \tag{4}$$

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \widetilde{H_t} \tag{5}$$

where H_{t-1} is the hidden state at $t-1$, W_{xr}, W_{hr}, W_{xz} and W_{hz} are weight parameters, b_r and b_z are the bias parameters. Sigmoid function $\sigma(\cdot)$ maps the value to the interval of $(0,1)$.

To incorporate the effect of reset gate R_t, candidate hidden state $\widetilde{H_t}$ is constrained by the activation function in the interval $(-1,1)$. The element-wise product operator $\odot$ controls which matrix to remove from the previous time step. If entries in R_t are close to 0, then it ignores H_{t-1} and focuses on X_t only. Finally, the update gate Z_t determines how much information from H_{t-1} and $\widetilde{H_t}$ to be included in H_t. Suppose the update gates in all time steps are 1, then the hidden state of the beginning will be kept and passed to the final output, no matter how long the period is. We direct the readers to [18] if interested.

4. ADPP

4.1. The Architecture of ADPP

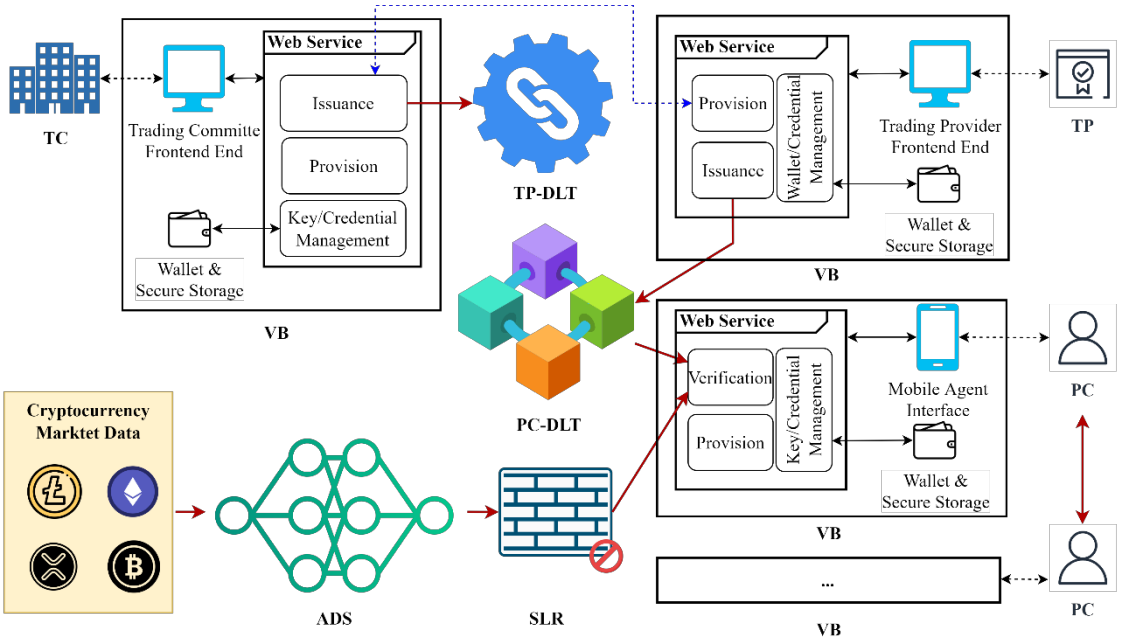


Figure 1. ADPP Architecture

Figure 1 presents the architecture of ADPP. ADPP defines three entities. (1) **Trusted Committees (TCs)** have the highest authority. They are pre-selected and pre-configured in the blockchain and host the other two consortium DLTs. An example TC can be the U.S. Securities and Exchange Commission (SEC). (2) **Trading Providers (TPs)** are authorized by a TC. An example TP can be Coinbase. It obtains licence from SEC and issues credentials to its customers. (3) **Peer customers (PCs)** are users who trade cryptocurrencies with each other in the ADPP after authorized by a TP.

ADPP has five components. (1) **Anomaly Detection System (ADS)** is responsible for detecting suspicious anomalous addresses. (2) **Sanction List Repository (SLR)** maintains addresses of anomalous customers that are caught by ADS. (3) **Peer Customer Distributed Ledger Technology (PC-DLT)** stores transactions that record the credentials issued by TPs to the PCs. (4) **Trading Providers Distributed Ledger Technology (TP-DLT)** is used to store transactions that a TC authorizes a TP. (5) **Virtual Broker (VB)** is a suite of middleware. It can be a web application containing some subsystems that leverage RESTful API and Web Service to provide the entities with certain features, such as provision, issuance, verification, and key management.

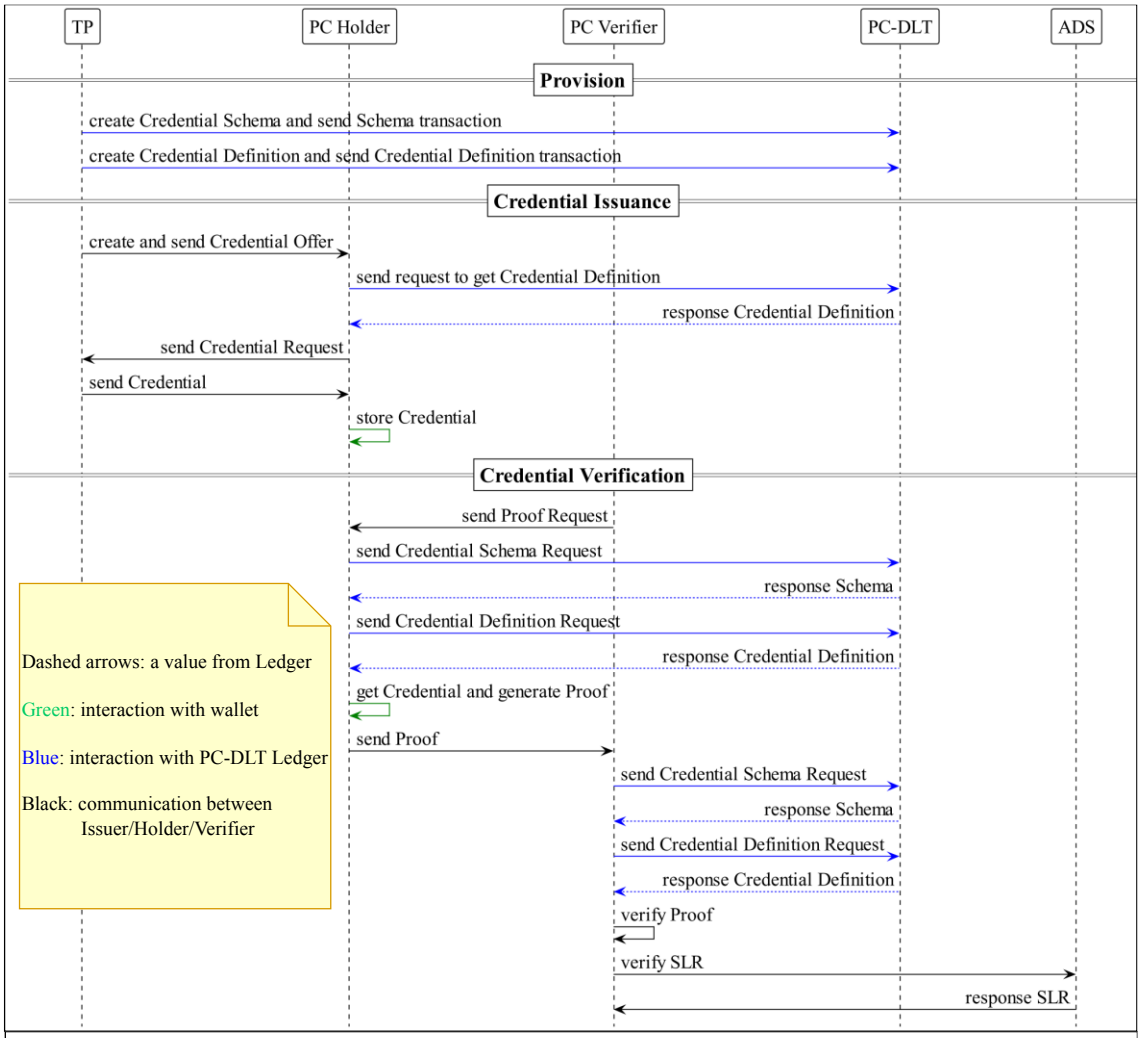
The workflow of ADPP is shown in Figure 1. TCs first complete their initialization and provision process. TCs can then evaluate and authorize TPs who apply to establish a trading platform. Once a TP is authorized, a corresponding blockchain transaction is published on the TP-DLT. Authorized TPs then issue VCs to their customer PCs through blockchain transactions on the PC-DLT. With more transaction data available, ADS learns anomalous behaviors from crypto transaction data. The suspicious illicit addresses detected by ADS are added into SLR. When two PCs want to trade cryptocurrency, they can verify each other's credentials and check whether the counterparty is in the SLR. Note that ADS can either be maintained by TC or the TP consortium.

4.2. Privacy-Preserving Authentication Platform

As shown in Figure 2, our blockchain-based privacy-preserving authentication platform allows users to regain control of identities and personal data to a certain extent. A decentralized key management system is integrated into the system to allow trading institutions to issue credentials to valid users. Only authenticated users are allowed cryptocurrency trading.

4.2.1. Provision

(1) **TC Provision**: All TC committee members are pre-selected, and the verifiable cryptographic information is stored in the genesis block of the TP-DLT. (2) **TP Provision**: A TP can use the tools provided by the TC to generate a key pair and a DID. The DID is linked to the key pair by using the public key as a unique identifier in the DID. The registration of the TP’s DID with the TC is completed off-chain by submitting the public key and other information directly to the TC. When the off-chain authorization is completed, the TC writes a transaction into the TP-DLT, indicating that the TP is authorized and can issue credentials to PCs. Then the TP creates and registers a credential schema in the PC-DLT. The credential schema describes the set of attributes for a particular certificate. The schema defines what information a PC requires to trade in the TP's platform. Based on the credential schema, the TP creates and registers a credential definition in the PC-DLT. The TP's identity information and the cryptographic attributes is bound to the credential schema and will be used to verify the credential. (3) **PC Provision**: A PC downloads a wallet and generates a key-pair and a DID. The key-pair is stored on the mobile device and must only be accessed by the PC’s password or biometric authentication.



ADPP Proposed Protocol Description

1. **Provision.**
   a. TC generates key pair $(pk^{TC}, sk^{TC}) \leftarrow KeyGen()$ and DID^{TC}
   b. TP generates key pair $(pk^{TP}, sk^{TP}) \leftarrow KeyGen()$ and DID^{TP}
   c. PC generates key pair $(pk^{PC}, sk^{PC}) \leftarrow KeyGen()$ and DID^{PC}
   d. TP creates schema based on attributes $sche \leftarrow ScheGen(attr, sk^{TP})$ and stores on PC-DLT.
   e. TP creates credential definition based on schema, binds with TP's public key $def \leftarrow DefGen(sche, pk^{TP})$ and stores on PC-DLT.
2. **Credential Issuance**
   a. PC sends request for credential to TP, with $(DID^{PC}, sche, attr^{PC})$,
   b. TP verifies attributes, generates a credential $cred^{PC}$
   c. TP signs the credential, gets a signature $\sigma_{cred} \leftarrow Sign(cred, sk^{TP})$
   d. TP issues $(cred, \sigma_{cred})$ to PC
3. **Credential Verification**
   a. The *verifier* PC sends request to check the required attribute $attr^X$
   b. The *holder* PC generates a proof $\Pi \leftarrow ProofGen(SigVerify(cred, \sigma_{cred}, pk^{PC} = 1) \wedge DID^{TP}$ is the *issuer* of cred $\wedge$ DID^{PC} is the *holder* of cred $\wedge$ $attr^X \in cred)$
   c. The *holder* PC sends $(\Pi, \sigma_{cred}, DID^{PC}, DID^{TP})$ to *verifier* PC
   d. The *verifier* PC checks whether $ProofVerify(\Pi) = 1$
   e. The *verifier* PC checks whether $attr^X \in$ SLR list

Figure 2. Protocol Description

4.2.2. Credential Issuance

A PC obtains a verifiable credential from a TP. Verifiable credentials must contain the PC's information affirming the PC is legal to perform a particular cryptocurrency trading. If a credential

schema for a particular kind of cryptocurrency has been registered on the PC-DLT and the PC wants to acquire a credential for trading that cryptocurrency, the PC will send a credential schema request to the PC-DLT to use the existing schema. Otherwise, the PC must contact the TP about registering a new schema. Once the PC receives the credential schema from PC-DLT, it can provide its DID and the value of required attributes in the schema and send a credential request to the TP. The validation process for the PC's information in the credential request is excluded from the scope. The TP ensures that the information provided by the PC is correct and places the

4.2.3. Credential Verification

Before two PCs trade cryptocurrency, mutual authentication is required. In a one-way authentication, the *verifier* PC sends a proof request for particular attributes to the *holder* PC. For example, in the case of Bitcoin trading, the proof request must contain the address attribute for Bitcoin transactions, along with a valid credential issued by a TP. After the *holder* PC receives the proof request from the *verifier* PC, it checks whether it has a corresponding schema containing the requested attributes. If yes, the *holder* PC can generate a presentation of the credential, which contains the requested attributes, and sends the presentation to the *verifier* PC. The *verifier* PC first checks whether a trustworthy TP has issued the credential contained in the presentation. Then it verifies whether the presented attributes in the proof suffice the request. In the Bitcoin case, the *verifier* will check whether a TP authorizes the holder's address for a Bitcoin transaction. Both parties will also check whether the counterparty is listed in the SLR.

4.3. Anomaly Detection System

The anonymous character of cryptocurrency trading is often abused for illicit purposes, such as ransomware, scam, terrorism, etc. PCs that initiate or receive cryptocurrency transactions with such illegal behaviors are categorized as anomalous users. To prevent malicious usage of cryptocurrency, ADS is designed as a binary classification model that formulates cryptocurrency transactions into graphs, aiming to identify anomalous addresses and add them to the SLR.

4.3.1. Graph Formulation

Information regarding cryptocurrency transactions between addresses is publicly available. Each transaction is associated with a sender address, a receiver address, a time stamp, and a number of features. The time stamp from raw data indicates the specific time when the cryptocurrency blockchain confirms the transaction. The confirmation time varies from minutes to hours, depending on the network. A time step in our model is defined as an interval of multiple days, which includes multiple time stamps so that various transactions in this interval can be graphically aggregated. The transaction features, such as transaction volume, cryptocurrency value, and transaction fee, are shared by both addresses.

At each time step t, cryptocurrency transactions in this step are used to construct an undirected graph $G_t = (V_t, E_t)$ with adjacency matrix A_t and node features X_t. Each edge $(v_{t,a}, v_{t,b})$ represents the flow of cryptocurrency in a specific transaction, the nodes $v_{t,a}$ and $v_{t,b}$ represent the sender address and receiver address in this transaction, respectively. If there are multiple transactions between two addresses in the time step t, the edges and the nodes can be differentiated by transaction id. There are two types of node features X_t: transaction features shared by the same edge, and the aggregated features derived from neighbors of node in the transaction network (e.g., statistics of neighboring transactions, the length of the transaction chain, etc.)

Therefore, given a set of graphs in the historical T time periods, we can learn an anomaly detection model F to classify whether each node in the graph at time step t is problematic or not. $\widehat{Y_t} = F(G_{t-T+1}, \dots, G_t)$ is a set of predictions for the nodes V_t. Each element $\widehat{Y_{v,t}} \in \{0,1\}$ is a binary value, 1 if v_t is anomalous, 0 otherwise.

4.3.2. Anomaly Detection Model

To extract topological dependence and temporal dynamics simultaneously from transaction networks, ADS integrates GCN with GRU, which are introduced in the previous section, into a sequence of layers, as shown in Figure 3. For a single layer from time step t to $t+1$, both feature matrix X_t and adjacency matrix A_t are first fed into GCN; then the output X_t' becomes the input of the GRU. H_t is the output of the current layer, and is fed as the input to the next layer. At the last time step, a function f concatenating Multilayer Perceptrons (MLPs) and activation functions maps H_t to the interval $(0,1)$. Each row of the output $H_t' = f(H_t) \in \mathbb{R}^{N\times 2}$ contains two values representing the probability of each node belonging to two classes. The final prediction $\widehat{Y}_t \in R^N$ denotes the class with higher probability.

The node classified by the model represents the address in a specific transaction. An address may appear in multiple nodes which corresponds to different transactions. If an address is detected as anomalous in any node, it is treated as an anomalous address and will be added into the SLR, considering an illicit address can also conduct legal transaction but a one-time good behavior won't make it a licit address.

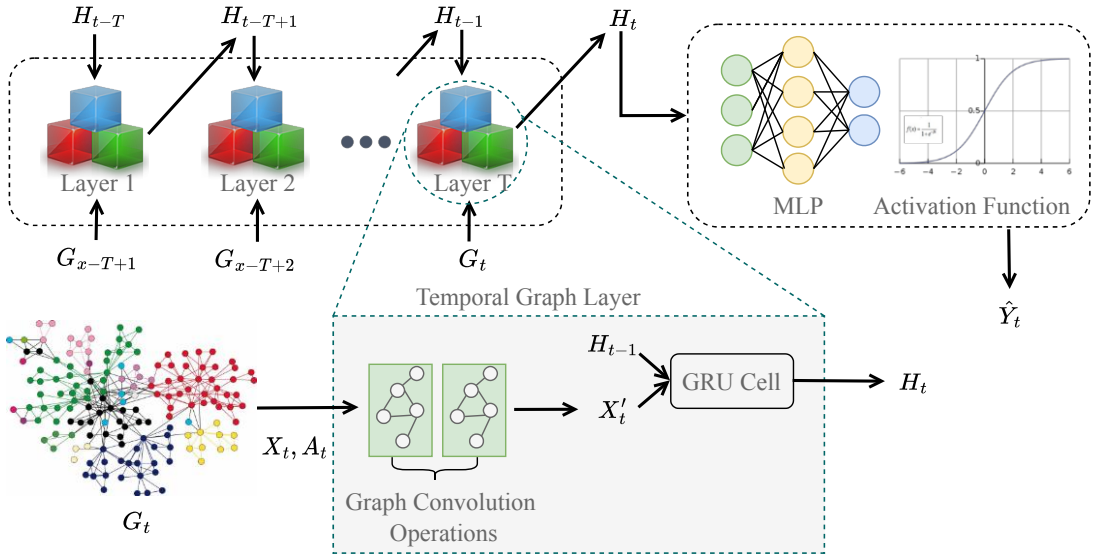


Figure 3. Overall Structure of Anomaly Detection Model

4.3.3. Enhancements Toward Imbalanced Class Problem

The anomaly detection task often suffers from an imbalanced class problem which shows a highly skewed class distribution. In our task, the proportion of malicious addresses is far less than the licit ones. To deal with this issue, ADS implements two enhancements in the proposed networks. For the simplicity of notation, the time step t is omitted.

Sampling Procedure is designed to balance the class of samples. Before training the networks, a subset of nodes is randomly sampled to construct the subgraph. The sampling probability for each node is determined by two aspects. (1) In bitcoin, the malicious node is the minority class, and the licit node is the majority class. The minority class is preferred to be chosen in the sampling procedure. (2) In the graph networks, the node with a larger degree is considered much more important. Such a node deserves to have a higher probability. Hence, the sampling probability P_v for the node v is calculated as follows:

$$P_v = \frac{P_v'}{\sum_v P_v'}, \quad P_v' = \frac{D_v}{c_v} \tag{6}$$

where D_v is the degree of node v, c_v is the total number of samples in the class that node v belongs to. The sampling probability is normalized so that the sum equals to 1. We randomly choose nodes with corresponding sampling probability by time steps to generate a set of subnodes and construct the subgraphs. ADS is trained by the subnodes.

Cost-Sensitive Loss Function is employed to reduce the influence of the imbalanced class. It assigns weights to classes during the training process. We employ cross entropy in this binary classification task to optimize the model. Let C_1, C_2 denote the majority (licit) and minority (anomalous) classes respectively, $|C_1|$ and $|C_2|$ denote the total samples in each class, the weights α_1 and α_2 are assigned to each class. The cost of missing an anomalous address is much larger than recognizing licit nodes as anomalous one. Hence, the weight for malicious samples α_2 is set to be larger than α_1 in the weighted entropy loss:

$$\mathcal{L}_{entropy} = \alpha_1 \frac{1}{|C_1|} \sum_{v \in C_1} Y_v \log(H'_v) + \alpha_2 \frac{1}{|C_2|} \sum_{v \in C_2} (1 - Y_v) \log(1 - H'_v), \quad \alpha_1 + \alpha_2 = 1 \tag{7}$$

$$\mathcal{L} = \lambda_1 \mathcal{L}_{entropy} + \lambda_2 \mathcal{L}_{L2}, \quad \mathcal{L}_{L2} = ||W_g||_2^2 \tag{8}$$

where $Y_v \in \{0,1\}$ is the true label of node v, H'_v is the probability of node v being anomalous.

Besides weighted entropy loss, L2 regularization $\mathcal{L}_{L2}$ is added to the loss function. It uses weights of GCN in all historical time steps as the constraint of model complexity. The L2 regularization helps prevent overfitting during the training process. λ_1 and λ_2 are penalty parameters to balance two types of loss.

5. EXPERIMENTS

Robust experiments were designed and carried out over real-world dataset to demonstrate the effectiveness of our proposed framework ADPP.

5.1. Datasets

Elliptic Data Set is released by Elliptic, a blockchain and cryptocurrency analytics company. It is the largest labeled cryptocurrency transaction data publicly available (https://www.kaggle.com/datasets/ellipticco/elliptic-data-set). The dataset is a bitcoin transaction network containing 203,769 nodes and 234,355 edges over 50-time steps. Each time step denotes an interval of 14 days. Each node is associated with 166 features, all of which are derived from public information and a label indicating whether it belongs to an illicit or licit class. The overall imbalance ratio of majority over minority class is approximately 10:1. Before the training process, the size of the dataset is shrunk to eighth of its original size by the sampling procedure. To tune the parameters and evaluate the performance, the dataset is split into training, validation, and test sets along time steps as the ratio of 31:5:13.

5.2. Implementation Setting

The two distributed ledgers in ADPP are deployed by leveraging Hyperledger Indy, an open-source project for consortium blockchain [26] and is funded by Linux Foundation. To fit the minimum number of nodes for a consensus model that can solve the Byzantine Fault Tolerance (BFT) problem [27]. The PC-DLT and TP-DLT are deployed on four TC nodes.

ADS is implemented in PyTorch. Hyper-parameters are tuned based on the validation set of *Elliptic Data Set*. We use Adam [28] as the optimizer with 0.001 of learning rate, 800 epochs. The historical length of transaction graph T is set to 5. The dimension of GCN and GRU layers are set to 128. The penalty parameters λ_1, λ_2, and weighted entropy parameters α_1, α_2 in the cost-sensitive loss function are set to 1, 0.001, 0.35 and 0.65, respectively.

5.3. Performance Analysis

We implement ADPP on *Elliptic Data Set* to evaluate performance regarding precision, recall, and F1. Precision evaluates the percentage of nodes classified as illicit are indeed illicit. Recall measures the percentage of illicit nodes are correctly identified. F1 is a combination of two metrics.

Figure 4 (a) presents the classification performance and the proportion of illicit samples over time. In the first six-time steps of the testing period, the performance of ADPP is steady at around 70% of precision, 50% of recall and 55% of F1 on average. It reveals that half of illicit addresses in the transactions are caught by the detection model. Note that the task is challenging given that the data is very imbalanced and only 10% of samples are illicit. Over time, the model achieves 80% of all metrics at 43rd step, demonstrating our model's effectiveness. Note that from the 44th to the 47th step, the percentage of anomalous samples sharply decreases to almost 0, probably due to the shutdown of some dark markets. Correspondingly, the performance of our model decreases. However, ADS can still catch half of the illicit nodes at the cost of misclassifying some good nodes as problematic. We argue that it is much more important to detect anomalous addresses even at the expense of recognizing legal addresses as illicit ones, considering the latter case is less harmful.

To quantify the efficacy of the enhancements towards imbalanced class problems, two variants of ADPP are implemented on the same test period of the *Elliptic Data Set*. Figure 4 (b) shows the variant without a sampling procedure in the training process. At the first period when there exists a dark market and a large number of anomalous addresses and transactions, 20% of them are missed by the variant compared with ADPP. During the shutdown of the dark market, the variant is not able to detect negative nodes. Compared with the variant in Figure 4 (c), which assigns equal weights on both classes in the loss function, i.e., $\alpha_1 = \alpha_2$, the cost-sensitive learning enables ADPP to focus much on minority samples. It largely boosts the performance regarding Precision and F1. This demonstrates that our proposed ADPP framework is effective for extracting illicit patterns and identify anomalous addresses.

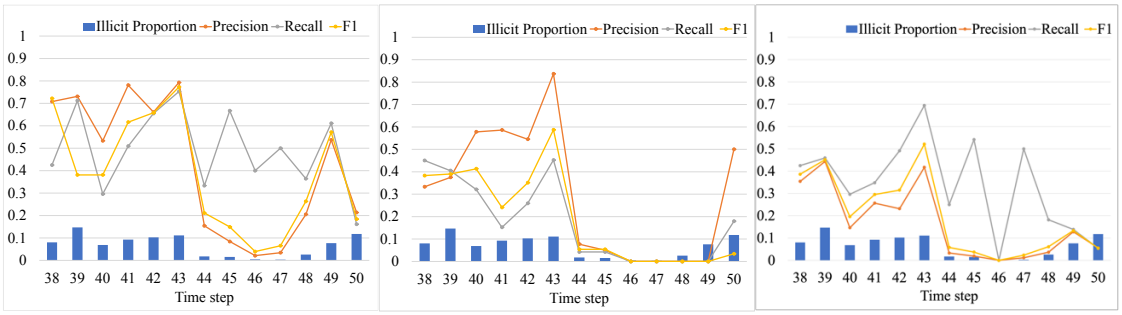


Figure 4. Anomaly detection performance results in testing periods over temporal dimension. The left subfigure (a) is the performance of ADPP. The middle subfigure (b) is the performance of variant of ADPP without sampling. The right subfigure (c) is the performance of variant without cost-sensitive learning.

6. CONCLUSION

This paper presents an anomaly detection and privacy-preserving platform (ADPP) integrating blockchain and deep learning techniques for crypto market trade. This blockchain-based privacy-preserving authentication platform enhances the security and privacy of user information. The Anomaly Detection System identifies illicit addresses in the transactions. The detected illicit users/addresses are added to the Sanction List Repository to help regulate the behavior in the crypto market.

The experiments are conducted on real-world cryptocurrency transaction data. Over half of the illicit addresses in the transactions can be detected by ADS, whether it's under the dark market or not. ADPP will trigger a warning if anomalous users in SLR are involved in a trade. It demonstrates the effectiveness of ADPP. It has a high potential for generalization ability to various crypto assets.

7. LIMITATIONS AND FUTURE WORKS

In the future, our work can be enhanced from three perspectives: (1) Our current work focuses on trade in the cryptocurrency market only. The improvement towards a universal framework for all types of crypto assets is a promising direction. (2) The performance of Create, Read, Update, and Delete (CRUD) operations in SLR can be further enhanced. For instance, the complexity of the searching operation can be reduced by adopting a probabilistic data structure such as bloom filter. (3) The training of ADS currently relies on external data. From our existing DLTs, additional data can be collected and applied to the federated learning framework to make it more robust and secure.

ACKNOWLEDGEMENTS

The research is partially supported by FHWA EAR 693JJ320C000021.

REFERENCES

[1] E. A. Akartuna, S. D. Johnson, and A. Thornton, "Preventing the money laundering and terrorist financing risks of emerging technologies: An international policy delphi study," *Technological Forecasting and Social Change*, vol. 179, p. 121632, 2022.

[2] S. Foley, J. R. Karlsen, and T. J. Putnin ̧š, "Sex, drugs, and bitcoin: How much illegal activity is financed through cryptocurrencies?," *The Review of Financial Studies*, vol. 32, no. 5, pp. 1798–1853, 2019.

[3] S. Cantor, J. Hodges, J. Kemp, and P. Thompson, "Liberty id-ff architecture overview," *Wason, Thomas (Herausgeber): Liberty Alliance Project Version*, vol. 1, 2003.

[4] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," *Advances in neural information processing systems*, vol. 28, 2015.

[5] R. Oppliger, "Microsoft. net passport: A security analysis," *Computer*, vol. 36, no. 7, pp. 29–35, 2003.

[6] B. Leiba, "Oauth web authorization protocol," *IEEE Internet Computing*, vol. 16, no. 1, pp. 74–77, 2012.

[7] S. Simpson and T. Groß, "A survey of security analysis in federated identity management," in *IFIP International Summer School on Privacy and Identity Management*, pp. 231–247, Springer, 2016.

[8] S. Dilek, H. Cakır, and M. Aydın, "Applications of Artificial Intelligence Techniques to Combating Cyber Crimes: A Review," *IJAIA*, vol. 6, no. 1, pp. 21–39, Jan. 2015, doi: 10.5121/ijaia.2015.6102.

[9] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, no. 2, pp. 197–227, 2016.

[10] E. Rabieinejad, A. Yazdinejad, and R. M. Parizi, "A deep learning model for threat hunting in ethereum blockchain," in *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 1185–1190, IEEE, 2021.

[11] M. Weber, G. Domeniconi, J. Chen, D. K. I. Weidele, C. Bellei, T. Robinson, and C. E. Leiserson, "Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics," *arXiv preprint arXiv:1908.02591*, 2019.

[12] X. Ao, Y. Liu, Z. Qin, Y. Sun, and Q. He, "Temporal high-order proximity aware behavior analysis on ethereum," *World Wide Web*, vol. 24, no. 5, pp. 1565–1585, 2021.

[13] Z. Liu, C. Chen, X. Yang, J. Zhou, X. Li, and L. Song, "Heterogeneous graph neural networks for malicious account detection," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 2077–2085, 2018.

[14] N. Pocher, M. Zichichi, F. Merizzi, M. Z. Shafiq, and S. Ferretti, "Detecting anomalous cryptocurrency transactions: an aml/cft application of machine learning-based forensics," *arXiv preprint arXiv:2206.04803*, 2022.

[15] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[16] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2017.

[17] J. D. Hamilton, *Time series analysis*. Princeton university press, 2020.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[20] Y. Liu, X. Ao, Z. Qin, J. Chi, J. Feng, H. Yang, and Q. He, "Pick and choose: a gnn-based imbalanced learning approach for fraud detection," in *Proceedings of the Web Conference 2021*, pp. 3168–3177, 2021.

[21] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, and J. Li, "Dice loss for data-imbalanced nlp tasks," *arXiv preprint arXiv:1911.02855*, 2019.

[22] M. Keshk, B. Turnbull, N. Moustafa, D. Vatsalan, and K.-K. R. Choo, "A privacy-preservingframework-based blockchain and deep learning for protecting smart power networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5110–5118, 2019.

[23] P. Kumar, R. Kumar, G. Srivastava, G. P. Gupta, R. Tripathi, T. R. Gadekallu, and N. N. Xiong, "Ppsf: a privacy-preserving and secure framework using blockchain-based machinelearning for iot-driven smart cities," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 3, pp. 2326–2341, 2021.

[24] M. Sporny, D. Longley, M. Sabadello, D. Reed, O. Steele, and C. Allen, "Decentralized identifiers (dids) v1.0."

[25] M. Sporny, D. Longley, and D. Chadwick, "Verifiable credentials data model v2.0."

[26] W. Yao, J. Ye, R. Murimi, and G. Wang, "A survey on consortium blockchain consensus mechanisms," *arXiv preprint arXiv:2102.12058*, 2021.

[27] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, p. 20, 1982.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.