


# Machine learning in/for blockchain: Future and challenges

Fang CHEN<sup>1</sup>, Hong WAN<sup>2</sup>, Hua CAI<sup>1</sup>, and Guang CHENG<sup>3\*</sup> 

<sup>1</sup>Department of Industrial Engineering, Purdue University, West Lafayette, IN, 47906 U.S.A.

<sup>2</sup>Department of Industrial and System Engineering, North Carolina State University, Raleigh, NC, 27695 U.S.A.

<sup>3</sup>Department of Statistics, Purdue University, West Lafayette, IN, 47906 U.S.A.

*Key words and phrases:* Bitcoin, blockchain, deep learning, machine learning, reinforcement learning.

*MSC 2010:* Primary 42A61, secondary 62M10.

*Abstract:* Machine learning and blockchain are two of the most notable technologies of recent years. The first is the foundation of artificial intelligence and big data analysis, and the second has significantly disrupted the financial industry. Both technologies are data-driven, and thus there are rapidly growing interests in integrating both for more secure and efficient data sharing and analysis. In this article, we review existing research on combining machine learning and blockchain technologies and demonstrate that they can collaborate efficiently and effectively. In the end, we point out some future directions and expect more research on deeper integration of these two promising technologies. *The Canadian Journal of Statistics* 49: 1364–1382; 2021 © 2021 Statistical Society of Canada

*Résumé:* L'apprentissage machine et les chaînes de blocs sont deux technologies récentes et remarquables. Alors que la première constitue les assises de l'intelligence artificielle et de l'analyse des mégadonnées, la deuxième a perturbé substantiellement l'industrie financière. Les deux technologies étant axées sur les données, il existe un intérêt croissant pour leur intégration afin d'améliorer l'efficacité et la sécurité du partage et de l'analyse de données. Les auteurs font une revue de la recherche portant sur la combinaison de l'apprentissage machine avec les chaînes de blocs, puis constatent que ces technologies s'harmonisent de façon efficace. Ils concluent en identifiant de futurs sujets de recherche et s'attendent à davantage de recherche pour une intégration plus complète des deux technologies prometteuses. *La revue canadienne de statistique* 49: 1364–1382; 2021 © 2021 Société statistique du Canada

## 1. INTRODUCTION

A blockchain is a shared, distributed public ledger that stores transaction data in a chain of sequential blocks (Dinh & Thai, 2018). The data (block) are time-stamped and validated before being added to the chain. Each block contains information from previous blocks. The mathematical structure of the blockchain for storing data makes it nearly impossible to fake (MIT Technology Review Editors, 2018). Thanks to the legacy of cryptocurrency, the term “blockchain” has transformed from a term in cryptography to a buzz word. Many people believe that cryptocurrency is blockchain. This is incorrect. While blockchain is the foundation of cryptocurrency, the applications of blockchain technology are much wider. Scenarios involving the validation, auditing, and sharing of data can all consider applying blockchains.

---

\* Author to whom correspondence may be addressed.

Email: [chengg@purdue.edu](mailto:chengg@purdue.edu)

In this article, we review existing research on combining blockchain and machine learning and demonstrate that they can collaborate efficiently and effectively. “Machine learning” is a general terminology that includes a variety of methods, such as machine learning, deep learning, and reinforcement learning. These methods are the core technology in big data analysis. As a distributed and append-only ledger system, blockchain is a natural tool for sharing and handling big data from various sources through the incorporation of smart contracts (i.e., a piece of code that will execute automatically in certain conditions). More specifically, blockchain can preserve data security and encourage data sharing when we train and test machine learning models. Also, blockchain can be utilized for distributing computing powers, building Internet of Things (IoT) networks, and developing online predictive models with various sources of data. This is especially important for deep learning procedures that require tremendous amounts of computational power. On the other hand, blockchain systems generate huge amounts of data from different sources, and the corresponding distributed systems are harder to monitor and control than centralized ones. Efficient data analysis and forecasting of system behaviours are critical for optimal blockchain mechanism designs. In addition, machine learning can facilitate the data verification process and the identification of malicious attacks and dishonest transactions in a blockchain. Interdisciplinary research on combining the two technologies is of great potential.

In this article, we review articles that either use machine learning techniques to study the blockchain system or structure or implement blockchain techniques to improve machine learning, e.g., through collaborative or distributed learning. The reviewed articles are summarized in Table 1. Articles that apply machine learning and blockchain techniques separately are not included in this article but are listed in Table 2. We first review basic blockchain structure and terminology in Section 2. This article is by no means exhaustive, but sufficient for Sections 3–5, which introduce how different machine learning methods can be incorporated into blockchain systems. Our work is concluded in Section 6 with a discussion of potential research directions and challenges that may arise from ongoing and future fusion of machine learning and blockchain.

## 2. REVIEW OF BLOCKCHAIN

A blockchain, literally speaking, is just a chain of digital blocks. Each block contains a certain amount of data, and the chain connects these data to form a distributed database. A node is a device that stores a full copy of the transaction history of the blockchain. A newly created block includes multiple transactions collected from nodes and broadcasts to every node on the network. The new block can be accepted and added to the blockchain by nodes that have the same consensus protocol. Each added block includes the information of the previous block in the chain. Hence, if a block is changed, all blocks before this block will be invalid as well. The strategies used to reach agreement with the new block (consensus) vary between different types of blockchain. The mathematical structure of a blockchain implies two essential properties: (i) the data (in a block) are immutable (MIT Technology Review Editors, 2018) and (ii) the distributed network, through consensus, allows users to communicate directly with each other and download a copy of the current ledger, which means that there is continuous monitoring and redundancy of the data in the network. Therefore, the blockchain is more robust to individual outage and attacks.

Depending on who can access the blockchain and who can validate the data, the blockchain can be classified into public chains, private chains, and consortium chains (Zheng et al., 2018a, 2018b). A comparison of these three different types of blockchains is shown in Table 3.

In what follows, we use the Bitcoin system, which is the most well-known blockchain application, as an example to demonstrate how blockchain works in detail. Typically, an end-to-end blockchain-based transaction needs to be validated at two different levels: the node level and the block level. The transaction is first verified between two nodes (Zheng et al.,

TABLE 1: Summary of articles in this article.

Model	Application	Article
Supervised/unsupervised learning without deep methods	Transaction entity classification	Yin & Vatraru (2017), Jourdan et al. (2018), Akcora et al. (2020)
	Bitcoin price prediction	Jourdan et al. (2018), Shah & Zhang (2014), Akcora et al. (2019), Abay et al. (2019), Dey et al. (2020)
Supervised learning with deep methods	Privacy and security preservation	Harris & Waggoner (2019), Chen et al. (2018), Zhu, Li & Yu (2019b)
	Computational power allocation	Luong et al. (2018)
	Cryptocurrency power prediction	McNally Roche & Caton (2018), Lahmiri & Bekiros (2019), Alessandretti et al. (2018)
Reinforcement learning	IoT networks	Liu, Lin & Wen (2018)
	Bitcoin mining	Eyal & Sirer (2014), Sapirshstein, Sompolinsky & Zohar (2017), Wang, Liew & Zhang (2019)

TABLE 2: Summary of some less relevant articles.

Area	Exemplary sources
Healthcare	Mamoshina et al. (2017), Juneja & Marefat (2018), Okalp et al. (2018), Zheng et al. (2018a, 2018b), Firdaus et al. (2018), Wang et al. (2018), Vyas, Gupta & Yadav (2019), Bhattacharya et al. (2019), Agbo, Mahmoud & Eklund (2019), Khezzr et al. (2019).
IoT Related	Liu, Lin & Wen (2018), Xiong & Xiong (2019), Lee & Ryu (2018), Qin et al. (2019), Ozyilmaz, Dogan & Yurdakul (2018), Singla, Bose & Katariya (2018), Shen et al. (2019), Li et al. (2019), Rathore, Pan & Park (2019), Ferrag & Maglaras (2020).

2018a, 2018b). Then, a unique digital signature, which is a hash wrapping all information of the transaction, is created. The digital signature that represents the transaction is submitted to the transaction pool and is waited to be added to a new block. Before the new block is accepted by the blockchain network, it is required to be validated by other miners on the network through the Proof-of-Work (PoW) consensus protocol. The PoW process includes aggregating a set of transactions to the new block and finding a hash value, i.e., lower than a target value (Ghimire & Selvaraj, 2018). The new block is only accepted by the network if the transactions are valid and unspent. Other nodes continue working on creating the new block using the hash from the previous block (Nakamoto, 2008).

Since the probability of finding a new, valid block is extremely low and the PoW process requires a huge amount of computing power and a high consumption of electricity, miners tend to collaborate with each other by forming mining pools. After participating in a mining pool, individual miners could receive a steady reward and significantly lower the risk of not getting any rewards constantly. On the other hand, mining pools usually charge membership fees

TABLE 3: Comparison of three types of blockchains.

Attribute	Public	Private	Consortium
Who runs/manages the chain	All miners	One organization/user	Selected users
Nodes need permission to access	No	Yes	Yes
Security	Nearly impossible to fake	Could be tampered with	Could be tampered with
Efficiency	Low	High	High
Centralized	No	Yes	Partial
Example	Bitcoin, Ethereum	IBM HyperLedger	Quorum

to each participant and allocate rewards to each miner according to their own reward-sharing mechanisms (Bhaskar & Lee, 2015). Some common reward allocation mechanisms in practice are Pay-Per-Last-N-Shares (PPLNS) (Qin, Yuan & Wang, 2019) and Full-Pay-Per-Share (Zhu et al., 2019a). One popular public chain is Ethereum (Wood, 2014), which allows users to send not only digital coins, but also smart contracts (Wohrer & Zdun, 2018). In order to reduce the energy consumption required for validation, Ethereum plans to switch its consensus protocol from PoW to Proof-of-Stake (PoS) gradually (Saleh, 2021).

### 3. SUPERVISED/UNSUPERVISED LEARNING WITHOUT DEEP METHODS

In this section, we review several applications of machine learning to blockchain. Specifically, Section 3.1 reviews three studies regarding transaction entity classification (Yin & Vatrappu, 2017; Jourdan et al., 2018; Akcora et al., 2020) with different purposes. One focuses on the recognition of cybercriminal entities using supervised learning (Yin & Vatrappu, 2017) as well as topological data analysis (TDA) (Akcora et al., 2020), while another focuses on the recognition of common categories of entities for most transactions (Jourdan et al., 2018). Section 3.2 reviews Bitcoin price prediction from different perspectives such as probabilistic graphical models (Jourdan et al., 2018), Bayesian regression (Shah & Zhang, 2014), and feature selection on blockchain topological structure using Granger causality and TDA (Akcora et al., 2019; Abay et al., 2019; Dey et al., 2020).

#### 3.1. Transaction Entity Classification

In a Bitcoin network, it is crucial to recognize entities behind potentially illegal nodes. The study of identifying entities behind addresses of nodes is called address clustering (Harrigan & Fretter, 2016). Yin & Vatrappu (2017) apply supervised learning to classify entities of transactions that may involve cybercriminal activities. The classification model is trained using 854 observations with categorical identifiers and then applied to study 10,000 uncategorized observations that comprise 31.62% of unique addresses and 28.99% of total coins in the overall Bitcoin blockchain. The categorical identifiers represent 12 classes of entities, five of which are related to cybercriminal activities. Thirteen classifiers from the Python machine learning package “scikit-learn” are applied. By comparing the accuracy scores of all the classifiers, it is found that random forests (77.38%), extremely randomized forests (76.47%), bagging (78.46%) and gradient boosting (80.76%) stand out as the four best classifiers. After further comparing the precision, recall, and F1 scores of these classifiers, bagging and gradient boosting stand out, which are then both applied to analyze the 10,000 uncategorized observations. The classification

TABLE 4: Classification performance (Jourdan et al., 2018).

Category	Accuracy	$F_1$	Precision
Exchange	0.94	0.92	0.91
Gambling	0.95	0.97	1.00
Mining	0.50	0.67	1.00
Service	0.95	0.88	0.83
Overall	0.92	0.91	0.92

outcome suggests that 5.79% (3.16%) of addresses and 10.02% (1.45%) of coins are from cybercriminal entities according to the bagging (gradient boosting) method.

Bitcoins have been found to be a common way to make ransomware payments. In order to detect addresses related to ransomware payments, Akcora et al. (2020) apply TDA to generate a Bitcoin address graph by grouping similar addresses into nodes and then representing common addresses between two nodes as an edge. TDA is an approach commonly used for dimension reduction. It represents the data set in a graph by first dividing data into sub-samples based on different filtration criteria and then clustering similar points within each sub-sample. The Bitcoin transaction graph model is a directed graph  $G = (V, E, B)$ , where  $V$  is the set of vertices,  $E$  is a set of edges, and  $B = \{\text{Address, Transaction}\}$  is a set of node types. By using six graph features extracted from each address, a TDA Mapper method is applied to create six filtered cluster tree graphs. After calculating the number of ransomware addresses in each cluster, denoted as  $N$ , a suspicion score is assigned to a new address. The suspicion scores of addresses in a cluster are set to be 0 initially. A suspicion score is incremented by one if inclusion and size thresholds are satisfied: (i) the inclusion threshold, denoted by  $\epsilon_1$ , times the total number of labelled ransomware addresses is less than  $N$  and (ii) the size threshold, denoted by  $\epsilon_2$ , times the number of labelled ransomware addresses in the cluster is greater than the number of all addresses in the cluster. Suspicious addresses are then filtered by a quantile threshold, denoted by  $q$ , specifically on whether their suspicion scores are higher than the quantile threshold. The result suggests that the best TDA model, with  $\epsilon_1 = 0.05$ ,  $\epsilon_2 = 0.35$ , and  $q = 0.7$ , outperforms random forest (RF), and XGBoost models in new ransomware address identification.

Jourdan et al. (2018) classify entities of transactions into four most common categories—exchange, service, gambling, and mining pool, based on data collected from 97 sources (Ermilov, Panov & Yanovich, 2017). The goal of this classification task is to assist in selecting an appropriate predictive model built on transaction category (Jourdan et al., 2018). The applied classification method is a gradient boosted decision tree algorithm fit with a Gaussian process-based optimization procedure for determining optimal hyperparameter values. Table 4 suggests that accuracy for the exchange, gambling, and service categories is high. However, accuracy for the mining pool category is poor. This may indicate that mining activities may not be appropriate as an independent label.

### 3.2. Bitcoin Price Prediction

Unspent transaction outputs (UTXOs) record the number of Bitcoins involved in a transaction and enable the tracking of buying and selling information for the Bitcoin price prediction. Another contribution of Jourdan et al. (2018) is the development of probabilistic graphical models for forecasting the value of UTXOs. The first model is called the block-transaction address model (BT-A), a stationary graphical model of a Bitcoin block with conditional dependence

TABLE 5: BT-A and BT-EA performance (Jourdan et al., 2018).

Metric	BE-TA	BE-TA	BE-TA	BE-TA	BT-A
	E	S	G	M	All
MSE	1.22	-0.30	-0.02	0.06	1.12
RMSE	125	53.3	1.15	5.19	90.5
MAE	15.6	0.94	0.20	2.42	7.47
RMAE	1.82	1.74	1.86	1.93	1.69
NRMSE	1.34	1.28	1.42	1.22	1.29

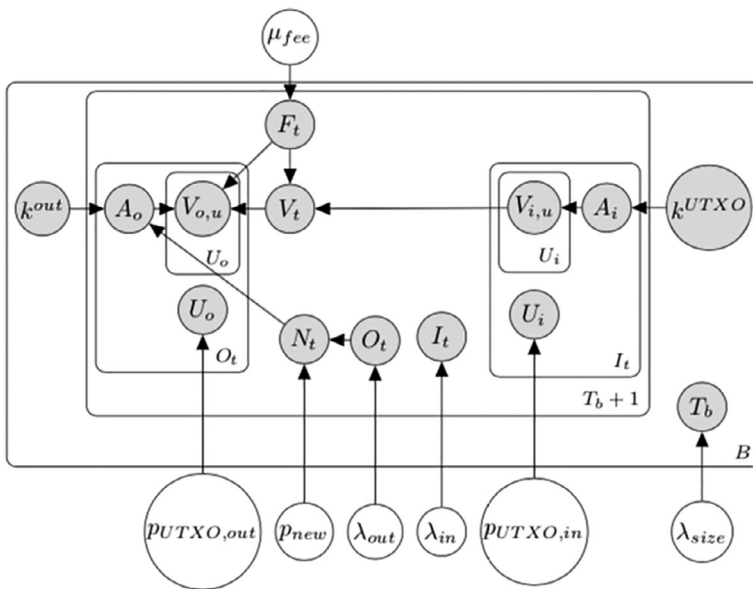


FIGURE 1: BT-A model (Jourdan et al., 2018).

structures. As an extension of the BT-A, a block-transaction entity-address model (BT-EA) adds a categorical entity to each address. In terms of mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), root mean absolute error (RMAE), the simulation results in Table 5 suggest that this extension significantly outperforms the BT-A model in all categories except for exchange.

The dependence structure of the BT-A model for predicting output UTXO value  $V_{o,u}$  is illustrated in Figure 1. The BT-A model starts by computing the number of available UTXOs for the  $i$ th input address  $A_i$ , denoted by  $k_{A_i}^{UTXO}$ . For each input address, the number of UTXOs used in a transaction is uniform randomly drawn from  $\{1, k_{A_i}^{UTXO}\}$  with the corresponding UTXO value denoted by  $V_{i,u}$ . The total input value of a transaction is calculated by summing the input UTXO value of each input address, denoted by  $V_t = \sum V_{i,u}$ . The value of an output UTXO is uniform randomly drawn between one and the total transaction value minus the validation fee.

To predict Bitcoin price, Shah & Zhang (2014) apply Bayesian regression in a latent source model, i.e., a nonparametric model for the binary classification of time series. The latent source



model framework is described in Chen, Nikolov & Shah (2013) and Bresler, Chen & Shah (2014), where the latent sources are time series with binary labels. Specifically, the model describes  $K$  distinct, unknown latent sources,  $s_1, \dots, s_K$  generated from a latent distribution over  $\{1, \dots, K\}$  with probabilities  $\{\mu_1, \dots, \mu_k\}$ , and  $K$  latent distributions, denoted by  $P_1, \dots, P_K$ . Each labelled data  $(x, y)$  is generated using a sample index  $T \in \{1, \dots, K\}$  with  $P(T = k) = \mu_k$  and  $x = s_T + \epsilon$ , where  $\epsilon$  follows the Gaussian distribution and  $y$  is generated from  $\mathbb{R}$  as per  $P_T$ . The model that predicts  $y$  given  $x$  is

$$P(y|x) = \sum_{k=1}^T P(y|x, T = k)P(T = k|x) = \sum_{k=1}^T P_k(y)\exp\left(-\frac{1}{2}\|x - s_k\|_2^2\right)\mu_k.$$

Due to a lack of information regarding latent parameters, empirical data are used as a proxy for estimating  $P(y|x)$ . The expectation of  $y|x$  can be estimated as

$$E[y|x] = \frac{\sum_{i=1}^n y_i \exp(-\frac{1}{4}\|x - x_i\|_2^2)}{\sum_{i=1}^n \exp(-\frac{1}{4}\|x - x_i\|_2^2)}. \tag{1}$$

The future average price change is determined by price changes over three periods of historical data: over the previous 30, 60, and 120 min. These price changes are denoted by  $\Delta p^j$  for  $j = 1, 2, 3$ . Each  $\Delta p^j$  is calculated by Equation (1). Then,  $\Delta p$  over a 10s period is formulated as

$$\Delta p = w_0 + \sum_{j=1}^3 w_j \Delta p^j + w_4 r, \tag{2}$$

where  $w_0, w_1, w_2, w_3$ , and  $w_4$  are weights to be estimated and  $r = (v_b - v_a)/(v_b + v_a)$ , where  $v_b$  and  $v_a$  are the top 60 of total buying and selling volumes.

We would like to point out that in order to apply Equation (2), it is crucial to verify the stationarity of the price data, which was unfortunately not done in the referenced article. The trading strategy for each user is designed as “buy one Bitcoin when  $\Delta p > t$ , sell one Bitcoin when  $\Delta p < -t$ , and otherwise hold the current number of Bitcoins when  $-t \leq \Delta p \leq t$ .” Here,  $t$  is a pre-specified threshold. The designed prediction model is trained by data gathered from Okcoin before May 2014 and is tested by data after that period. It is found that increasing  $t$  leads to an increase in the average profit per trade.

Beside using Bayesian regression to predict Bitcoin price, the selection of input features is also important to predictive performance. To better characterize input features, Akcora et al. (2019) introduce the concept of a graph chainlet, which describes the local topological features of a Bitcoin blockchain, and also explores the impacts of Bitcoin blockchain structure on Bitcoin price formation and dynamics. A transaction-address graph representation of a Bitcoin blockchain is shown in Figure 2. The circular vertices  $a_1, a_2, a_3, a_4$ , and  $a_5$  are addresses of UTXOs, the square vertex represents a single transaction, and the edges denote UTXOs (a transfer of Bitcoins). A chainlet model represents  $x$  input UTXOs and  $y$  output UTXOs involved in a transaction, denoted by  $C_{x \rightarrow y}$ . All chainlets and chainlet clusters formed by various criteria are evaluated by the Granger causality test (Granger, 1969). The results suggest that the split chainlet cluster defined by  $y < x < 20$ , individual chainlets ( $C_{1 \rightarrow 7}$ ,  $C_{6 \rightarrow 1}$ , and  $C_{3 \rightarrow 3}$ ), extreme chainlets ( $C_{20 \rightarrow 2}$ ,  $C_{20 \rightarrow 3}$ ,  $C_{20 \rightarrow 12}$ , and  $C_{20 \rightarrow 17}$ ), and clusters defined by cosine similarity ( $C_{9 \rightarrow 11}$ ,  $C_{3 \rightarrow 17}$ ,  $C_{8 \rightarrow 14}$ , and  $C_{1 \rightarrow 1}$ ) are significant in Bitcoin price formation and dynamics. A price prediction model is further developed using significant chainlets.

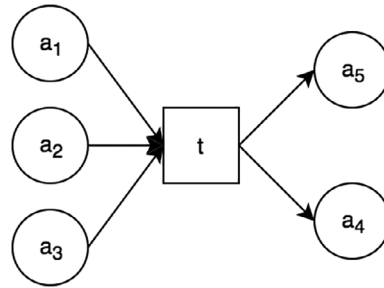


FIGURE 2: A transaction-address graph.

Chainlet models study the topological features of a single transaction and only take the number of input and output UTXOs into account. Abay et al. (2019) extend the chainlet model to a new graphical model, ChainNet, that further considers topological features based on the number of distinct chainlets and the amount of coins transferred by the chainlets. More specifically, from the perspective of all transactions, an occurrence matrix is created to count the number of transactions between distinct chainlets. An amount matrix records the sum of Bitcoins transferred between distinct chainlets. By considering both the occurrence and the amount of Bitcoins transferred in a transaction, an  $\epsilon$ -threshold occurrence matrix, denoted by  $O^\epsilon$ ,  $\epsilon \in \{0, 10, 20, 30, 40, 50\}$ , is created to count the number of distinct chainlets larger than  $\epsilon$ . Different thresholds result in different values of  $O^\epsilon$ , which are considered as filtration features inputs in the predictive model. Betti sequences and Betti derivatives for the blockchain network are also considered as features in the model. A sliding prediction approach associated with parameters controlling the prediction horizon, window length, and training length is applied to train the time series predictive model. According to simulation results, ChainNet's adoption of Betti model features and filtration features for short- and long-term prediction results in better performance.

Beside considering the effects of features of the Bitcoin network's topological structure on Bitcoin price formation and dynamics, topological features of other types of cryptocurrencies may also affect Bitcoin price. Dey et al. (2020) evaluate Bitcoin price formation and dynamics using the chainlet model and joint topological features of Bitcoin and Litecoin. Specifically, the occurrence of distinct chainlets in Bitcoin and Litecoin networks, denoted by  $O_{x \rightarrow y}^B$  and  $O_{x \rightarrow y}^L$ , respectively, are considered. The amount of coins transferred in Bitcoin and Litecoin, denoted by  $A_{x \rightarrow y}^B$  and  $A_{x \rightarrow y}^L$ , respectively, are also included. Granger causality tests (Granger, 1969) with one to five lag effects are applied to assess chainlet significance. The results suggest that the occurrence of chainlets in the Litecoin network ( $O_{3 \rightarrow 3}^L$ ,  $O_{4 \rightarrow 4}^L$ ,  $O_{4 \rightarrow 5}^L$ , and  $O_{3 \rightarrow 6}^L$ ) is significant to Bitcoin price for all five lag effects based on Granger causality. Also, the occurrence and value of chainlets in the Bitcoin network ( $O_{20 \rightarrow 2,3,12}^B$ ,  $O_{1 \rightarrow 7}^B$ ,  $A_{20 \rightarrow 12,20}^B$ , and  $A_{3 \rightarrow 4}^B$ ) are also important to Bitcoin price for all five lag effects.

Although there are other studies related to Bitcoin price prediction using machine learning methods, e.g., Greaves & Au (2015), Jiang & Liang (2017), Jang & Lee (2018), and Sun, Liu & Sima (2020), it is not feasible to include all these articles in this article. We move on to review more articles related to the prediction of cryptocurrency price using deep learning in Section 4.

#### 4. SUPERVISED LEARNING WITH DEEP METHODS

In this section, we turn to the application of deep learning. In Section 4.1, three privacy-preserving collaborative learning frameworks (Harris & Waggoner 2019; Chen et al., 2018; Zhu, Li & Yu, 2019b) are reviewed. In Section 4.2, we review a deep learning approach (Luong et al., 2018)



that allocates computational resources to assist mobile blockchain mining. In Section 4.3, we focus on cryptocurrency price prediction (McNally, Roche & Caton, 2018; Lahmiri & Bekiros, 2019) and digital portfolio management using recurrent neural network (RNN) and long-short term memory (LSTM) models (Alessandretti et al., 2018).

#### 4.1. Decentralized, Privacy-Preserving Collaborative Learning

Harris & Waggoner (2019) build a decentralized collaborative learning framework with blockchain. This new framework extends two previous frameworks (Abernethy & Frongillo, 2011; Waggoner, Frongillo & Abernethy, 2015) and is designed to collaboratively build a data set and train a predictive model. The framework starts by letting the provider define a loss function and upload 10 out of 100 partial data sets with corresponding hashes. By using a smart contract that initially contains a model, other participants add their own data or upload an update along with a deposit of one unit of currency, kept until an end condition set by the provider is met. The provider uploads the remaining 90 partial data sets to evaluate participants' models. The better model tends to receive more rewards in the end.

Chen et al. (2018) propose a framework called learning chain to preserve users' privacy by applying a decentralized version of the stochastic gradient descent (SGD) algorithm and a differential privacy mechanism. The proposed framework contains three phases: blockchain initialization, local gradient computation, and global gradient aggregation. In the first phase, a peer-to-peer network is set up with computing nodes and data holders. The second phase involves each data holder  $P_k$  retrieving the current model from the block  $t$ , denoted by  $w_t$ , and computing its own local gradient. A differential privacy mechanism is then applied to generate a hidden local gradient, denoted by  $\nabla g_k(w_t)^*$ , by adding a noise factor to the local gradient. The message broadcasts a pseudo-identity of  $P_k$  and a normalized, hidden, local gradient, denoted by  $\nabla \hat{g}_k(w_t)^*$ , together with the norm of its unnormalized version, to computing nodes on the network. In the final phase, after solving a PoW problem, the winning node selects top  $l$ -nearest local normalized gradients according to the cosine distance between each normalized local gradient and the sum of the  $\nabla g_k(w_t)^*$ s to update the global gradient. The predictive model is updated as  $w_{t+1} = w_t + \eta \nabla J(w_t)$ , where  $\nabla J(w_t)$  is the updated global gradient.

Learning chain is trained and tested using three different data sets: a synthetic data set, a Wisconsin breast cancer data set, the MNIST data set, and the Ethereum blockchain framework. There exists a trade-off between privacy and accuracy in the sense that decreasing the privacy budget leads to an increase in test errors on all data sets. The proposed model is further compared with the learning chainEX model, i.e., implemented with a lower privacy budget. The similar test error between learning chain and learning chainEX suggests the differential privacy scheme in learning chainEX is effective and efficient.

Zhu, Li & Yu (2019b) develop a blockchain-based privacy-preserving framework to secure a share of updates in federated learning. The federated learning algorithm, developed by McMahan et al. (2016), allows each mobile device to compute and upload updates to the global predictive model based on their local data set. A security issue arises in federated learning when there exist Byzantine devices on the network. In this case, a blockchain transaction mechanism is adopted to ensure the security of sharing and updating. Specifically, model updates are written in a blockchain transaction. A transaction that contains the information including changes to hyperparameters and weights and public keys (participants' addresses) broadcasts to other nodes. Then, other nodes can validate the transaction and test updates according to their local data sets. If most nodes confirm that the performance score of the updated model is higher than that of the existing model under their local data sets, then the updates are implemented into the current model.

## 4.2. Computing Power Allocation

Luong et al. (2018) develop a deep learning-based auction algorithm for edge computing resource allocation to support mobile mining activities. The designed framework enables mobile device miners to submit their bid valuation profiles to one edge computing service provider (ECSP) for buying additional computing power. The valuation profile for miner  $i$ , denoted by  $v_i$ , is drawn from a distribution that assigns a higher value  $v_i$  when its block size, divided by its initial computing capacity, is larger. The ECSP evaluates all valuation profiles and maximizes its revenue in the following steps.

An allocation rule is applied to map transformed valuation profiles, defined as  $\bar{v}_i = \phi_i(v_i)$ , to assignment probabilities using a softmax function. The winner miner  $i$  will pay the price  $p_i = \phi_i^{-1}(\text{ReLU}(\max_{i \neq j} \bar{v}_j))$ . In the end, ECSP loss function is defined as

$$\hat{R}(\mathbf{w}, \beta) = - \sum_{i=1}^N g_i^{(\mathbf{w}, \beta)}(\mathbf{v}^s) p_i^{(\mathbf{w}, \beta)}(\mathbf{v}^s),$$

to which SGD is applied. Here,  $g_i$  is the assignment probability and  $N$  is the number of miners. This deep learning-based auction mechanism is empirically compared to a regular auction mechanism. It is found that the deep learning-based auction achieves higher revenue and converges to an optimal value faster than other mechanisms.

## 4.3. Cryptocurrency Price Prediction

For forecasting Bitcoin price, McNally, Roche & Caton (2018) compare the performances of two deep learning algorithms, i.e., an RNN and an LSTM. It is interesting to note that two hidden layers with 20 nodes per layer are sufficient in both models. Specifically, the RNN model adopts the tanh function as its activation function, while the LSTM applies tanh and sigmoid functions for different gates, which results in a longer training time. The data set used to train and test the LSTM and RNN models contains Bitcoin prices from August 2013 to July 2016. Features used in the model include the opening price, daily high, daily low, closing price, hash rate, and mining difficulty. Feature importance is evaluated by the Boruta algorithm, which is a wrapper built around the RF classification algorithm. A traditional autoregressive integrated moving average (ARIMA) time series model is empirically compared with these deep learning models. The simulation results suggest that the LSTM, RNN, and ARIMA models have similar accuracies, at 52.78%, 50.25%, and 50.05%, respectively. However, the deep learning models have much lower RMSE values. In addition, the LSTM model is capable of recognizing long-term dependencies, in contrast to the RNN model.

In contrast with other studies for predictive models, Lahmiri & Bekiros (2019) instead conduct a chaotic time series analysis before building deep learning models. Hence, the first step is to calculate the largest Lyapunov exponent (LLE) and then apply detrended fluctuation analysis to detect chaotic characteristics of the cryptocurrency price data without the assumption of stationarity. Then, a deep neural network model with an LSTM implementation (Hochreiter & Schmidhuber, 1997) (DLNN) and a generalized regression neural network (GRNN) model (Specht, 1991) are built to predict the price of three cryptocurrencies: Bitcoin, Digital Cash, and Ripple. The number of data samples obtained for the model is 3,006, 1,704, and 1,357 for Bitcoin, Digital Cash, and Ripple, respectively. The authors create a many-to-many sequence predictive model, which utilizes the first 90% of the observations for training and the last 10% of the observations for testing and out-of-sample forecasting. According to Table 6 and Figure 3, whose  $x$ -axis represents the time horizon and the  $y$ -axis represents the price, the positive Hurst exponent indicates the presence of long-term features in the data and the negative LLE indicates that the training data is chaotic. As a result, a short-term predictive model would be suitable for

TABLE 6: Chaotic analysis (Lahmiri & Bekiros, 2019).

	LLE		HE	
	Training sub-sample	Testing sub-sample	Training sub-sample	Test sub-sample
Bitcoin	0.1250	-7.8711	1.0087	0.9776
Digital cash	0.3205	-10.7333	0.9559	1.0901
Ripple	0.8181	-0.0065	1.0741	0.8715

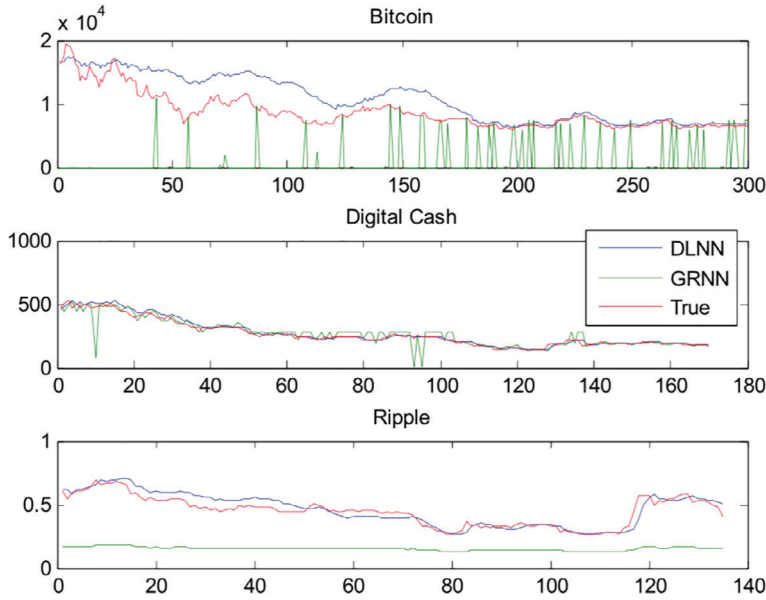


FIGURE 3: Prediction results (Lahmiri & Bekiros, 2019).

the data. The simulation results suggest that the LSTM model outperforms the GRNN model in predicting all three cryptocurrency prices. Although the RMSE of the LSTM model is still high, the model demonstrates a trend similar to real price changes for all three cryptocurrencies.

Beside cryptocurrency price prediction, Alessanderti et al. (2018) explore a portfolio analysis by forecasting daily prices of 1,681 types of cryptocurrencies. Three models are developed to predict the price of every kind of cryptocurrency. For each type  $c$ , the target is the return of investment at each time  $t_i \in \{0, \dots, 895\}$ , expressed as

$$ROI(c, t_i) = \frac{\text{price}(c, t_i) - \text{price}(c, t_i - 1)}{\text{price}(c, t_i - 1)}.$$

The features considered are price, market capitalization, market share, rank, and volume. The first model is an ensemble of regression trees fit using XGboost and pairs the features and prices of each type of cryptocurrency. The second model is a regression model that considers features of all kinds of cryptocurrency as a whole, paired with prices of each type of cryptocurrency. The third model adopts an RNN with an LSTM implementation and uses the second model's feature-target pairing strategy. All models are developed for one-step ahead forecasting.

A portfolio is constructed based on the predicted prices. Model hyperparameters are optimized by maximizing either the sharp ratio or the geometric mean of total returns. The results suggest that all three models generate profits, and that optimization using the sharp ratio metric achieves a higher return. Another conclusion is that the first two models, implementing gradient boosting with decision trees, have higher accuracy in the short-term (5–10 days), while the third model adopting LSTM has better predictive performance in the long term (around 50 days).

## 5. REINFORCEMENT LEARNING

In this section, we first review a framework that incorporates reinforcement learning into blockchain to ensure the security of data collection, storage, and processing in an IoT network (Liu, Lin & Wen, 2018). Second, we review two types of frameworks that study Bitcoin blockchain mining activities. The first explores the potential of Bitcoin mining through mobile networks (Nguyen et al., 2020), while the second formulates a Markov decision process (MDP) for modelling blockchain mining activity (Eyal & Sirer, 2014; Sapirshtein, Sompolinsky & Zohar 2017). The last work in this article applies a new reinforcement learning algorithm to find an optimal mining strategy (Wang, Liew & Zhang, 2019).

### 5.1. Internet of Things

Liu, Lin & Wen (2018) propose a framework to secure data collection and sharing among mobile terminals (MTs) on an IoT network. The framework consists of two phases: data collection and data sharing. In the first phase, each MT, denoted by  $m$ , adopts multi-agent deep reinforcement learning to maximize the efficacy of data collection. The state space is defined as  $S = \{S_1, S_2, S_3\}$ . Here,  $S_1 = \{(x^k, y^k), (x^c, y^c)\}$  is a set of states representing the coordinates of  $k$  point of interest (PoIs) and  $c$  obstacles in the environment. The environment is a map of size  $E_x \times E_y$ , where  $x \in [0, E_x], y \in [0, E_y]$ .  $S_2$  contains the MTs' coordinates; and  $S_3$  represents the sensing time  $h_i(k) \in [0, t]$  for the  $k$  PoI. The action space consists of a movement direction, denoted by  $\theta_t^m$ , and a movement distance, denoted by  $l_t^m$ . Thus, the action space is  $A = \{(\theta_t^m, l_t^m) \mid \theta_t^m \in [0, 2\pi), l_t^m \in [0, l_{max}]\}$ . The reward  $r_t^m$  is

$$r_t^m = \frac{w_t b_t^m}{\alpha b_t^m + \kappa l_t^m},$$

where  $b_t^m$  is the amount of collected data;  $\alpha$  and  $\kappa$  are the energy consumption per collected data and per unit distance travelled; and  $w_t$  is the achieved geographic fairness, calculated by  $w_t = \frac{(\sum_{k=1}^K h_t(k))^2}{K \sum_{k=1}^K h_t(k)^2}$ . Each MT is implemented by four deep neural networks and an actor-critic algorithm is applied to maximize the reward.

After the MTs finish data collection, they share the data through an Ethereum blockchain network. Before this, the data are sent to a certificate authority (CA) for verification. Once the CA verifies the ownership of the MTs' data and checks the consistency of the received data with the original data, a digital signature is generated and sent back to the MTs. As a result, each MT is able to broadcast its transaction request consisting of the digital signature from CA, the original data, and MT's public key to other nodes on the blockchain network to be further validated. Relative to randomly moving MTs, the MTs implementing deep reinforcement learning collect much more data but consume more energy. The blockchain-based data sharing framework can still store all the data sent by the MTs even under a denial-of-service attack.

### 5.2. Bitcoin Mining

As discussed in earlier sections, blockchain mining requires a huge amount of computing power, so it is nearly impossible to apply blockchain to a mobile system. Nguyen et al. (2020)

propose a mobile edge computing (MEC)-based blockchain network to assist mobile users (MUs) offloading mining tasks to a MEC server. Specifically, the state space is defined as  $s^t = \{D_1^t, D_0^t, g^t\}$ , where  $D_1^t$  and  $D_0^t$  are new and buffered transaction data at time  $t$ , respectively, and  $g^t$  is the power gained when a miner  $n$  offloads a task  $m$  to the MEC server. The action space is expressed as  $a^t = x_{nm}^t$ , where  $x_{nm}^t \in \{0, 1\}$  indicates whether the  $n$ th MU processes  $m$  mining tasks locally or offloads the  $m$  tasks to the MEC server, respectively. The goal of a miner is to maximize the privacy level  $P^t$ , defined in He et al. (2017), and minimize the total cost of energy and time consumption. The system reward is formulated as  $r^t(s, a) = P^t(s, a) - C^t(s, a)$ , where  $P(s, a)$  is the privacy level and  $C(s, a)$  is the total cost of electricity and computing power. A value-based method, Q-learning, and deep Q learning are applied to update the Q value. The results suggest that, although the convergence speed for Q-learning and deep Q learning are almost the same, agents trained by deep Q learning receive higher total rewards.

Although the mining and selling of Bitcoins could generate a large revenue, the cost of mining is also high due to the high consumption of electricity. There is now interest in finding an optimal mining strategy to maximize profits. Since Bitcoin mining can be modelled as a Markov decision process (MDP) that contains an enormous number of states, reinforcement learning can be applied to study this MDP. An MDP for Bitcoin mining was first proposed in Eyal & Sirer (2014) and extended in Sapirshstein, Sompolinsky & Zohar (2017). The environment assumes that the block generation times independently follow a Poisson distribution. A new block is created by an honest agent with probability  $1 - \alpha$  while a new block is obtained by an adversarial agent, also known as an attacker, with probability  $\alpha$ . The adversary may hide some blocks on its own private chain, but the blockchain is always the longest public chain. The state space of the MDP is defined as  $(a, h, fork)$ , where  $a$  represents the number of blocks on the adversary's private chain;  $h$  represents the number of blocks on the public chain;  $fork$  is an environment variable that has three values, one of  $\{irrelevant, relevant, active\}$ . The state  $(a, h, irrelevant)$  denotes the case when the previous state is  $(a - 1, h)$  and the *match* action is feasible, i.e., the last mined block accepted by the chain was mined by the adversary;  $(a, h, relevant)$  denotes the case when previous state is  $(a, h - 1)$  and the *match* action is infeasible, i.e., the last mined block was mined by the honest miner; and the *active* refers to the case that the network is broken into two branches containing the same number of blocks. When  $fork = active$ , the probability that the next block is generated from the honest block is  $\gamma$  and the probability that the next book is generated from the adversarial block is  $1 - \gamma$ . The action space, defined as  $A = \{adopt, override, match, wait\}$ , contains four actions. The *adopt* action refers to an agent always mining the last block on the public chain without any blocks on its private chain. The *override* action becomes feasible when the number of blocks on the private chain is more than the number of blocks on the public chain. In other words, all blocks on the private chain are published to replace the existing public chain. The *match* action refers to the adversary releasing the same number of blocks as the current public chain, which creates a fork on the public chain. The *wait* action, where the adversary keeps mining on its private chain without releasing any new blocks to the public chain, is always feasible. The transition probability and reward matrices are shown in Table 7. Since the honest agent is considered as a part of the environment, the focus is on finding an optimal strategy for adversarial agents. The number of blocks on the public chain is considered as a reward. The reward is formulated in two dimensions: the number of blocks mined by honest agents and adversarial agents, respectively. The reward function then considers a relative reward instead of an absolute reward. The objective function is

$$f(s, a) = \frac{q^a(s, a)}{q^a(s, a) + q^h(s, a)}. \quad (3)$$

Wang, Liew & Zhang (2019) plan to apply off-policy Q-learning to solve this problem. Unfortunately, the reason why Q-learning is used there is not mentioned in the original article.

TABLE 7: The state transitions and reward matrices (Sapirshtein, Sompolinsky & Zohar, 2017).

State at time $t$ , action	State at time $t + 1$	Transition probability	Reward
$(a, h, \cdot), adopt$	$(1, 0, irrelevant)$	$\alpha$	$(0, h)$
	$(0, 1, irrelevant)$	$1 - \alpha$	$(0, h)$
$(a, h, \cdot), override$	$(a - h, 0, irrelevant)$	$\alpha$	$(h + 1, 0)$
	$(a - h - 1, 1, relevant)$	$1 - \alpha$	$(h + 1, 0)$
$(a, h, irrelevant), wait$ $(a, h, relevant), wait$	$(a + 1, h, irrelevant)$	$\alpha$	$(0, 0)$
	$(a, h + 1, relevant)$	$1 - \alpha$	$(0, 0)$
$(a, h, active), wait$ $(a, h, relevant), match$	$(a + 1, h, active)$	$\alpha$	$(0, 0)$
	$(a - h, 1, relevant)$	$\gamma(1 - \alpha)$	$(h, 0)$
	$(a, h + 1, relevant)$	$(1 - \gamma)(1 - \alpha)$	$(0, 0)$

Since Q-learning can only optimize a linear reward function, the authors propose a new multi-dimensional RL algorithm based on off-policy Q-learning. The new algorithm considers two Q-functions, i.e., a pair  $(Q^{(a)}(s, a), Q^{(h)}(s, a))$ . At each time step, the adversarial agent observes  $(s_{t+1}, r_{t+1}^a, r_{t+1}^h)$  from the environment. The two Q-functions are then updated

$$q^{(a)}(s_t, a_t) \leftarrow (1 - \beta)q^{(a)}(s_t, a_t) + \beta[(r_{t+1}^a + \lambda q^{(a)}(s_{t+1}, a')] ]$$

and

$$q^{(h)}(s_t, a_t) \leftarrow (1 - \beta)q^{(h)}(s_t, a_t) + \beta[(r_{t+1}^h + \lambda q^{(h)}(s_{t+1}, a')] ]$$

where  $\beta \in (0, 1)$  is the learning rate,  $\lambda$  is a number close to one, and  $a' = \text{argmax}_a f(s_{t+1}, a)$ . The current best action is chosen by the  $\epsilon$ -greedy strategy to maximize the objective function in Equation (3) with probability  $1 - \epsilon$ . A random action is chosen with probability  $\epsilon$ . Random selection is used to avoid trapping at local maximums. The parameter  $\epsilon$  is determined by  $\epsilon(s_t) = \exp(-\frac{V(s_t)}{T_\epsilon})$ , where  $V(s_t)$  is the number of times that the state was visited and  $T_\epsilon$  controls the rate at which  $\epsilon$  is reduced.

Sensitivity analysis is applied to evaluate the estimated optimal strategy. The simulation result is shown in Figure 4. After setting the discount factor to one, the article suggests that their optimal mining strategy outperforms the mining strategies presented in Eyal & Sirer (2014) and Sapirshtein, Sompolinsky & Zohar (2017).

### 6. CONCLUSION AND FUTURE CHALLENGES

In this article, we reviewed applications of blockchain to databases to improve user privacy in the learning process and uses of machine learning to optimize computer resource allocation or cryptocurrency investment decisions. The majority of these applications can be categorized as applying one technique to another, but few actually integrate two technologies.



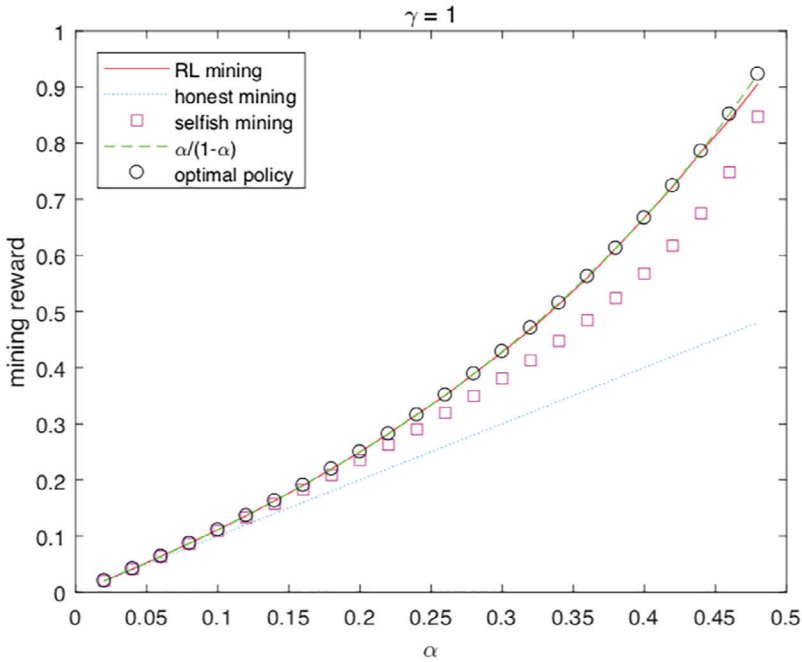


FIGURE 4: Simulation result for different mining strategies (Wang, Liew & Zhang, 2019).

Hence, it is fair to say that current research is still very preliminary from an interdisciplinary perspective.

However, we expect new lines of research to emerge in the following areas.

- “Smart agents” can be designed to regulate the blockchain and detect abnormal behaviours. The former is especially important for consortium and private chains that require coordination among users, while the latter is critical for public chains.
- Learning-based analyses of blockchain-based systems are rare. From financial systems to supply chains, there is an enormous amount of data available to evaluate the performance of the decentralized structure of the blockchain relative to traditional, centralized structures. Learning-based analysis can shed light on mechanism design for blockchain structures and provide on-time forecasting models.
- Blockchain allows anonymous data sharing. With the development of IoT and wearable devices, privacy issues are catching the attention of more and more users. Through combinations with data fusion, multiple-layer blockchain structures that allow the sophisticated authorization of data for different users can be designed.
- Blockchain mining activity can be modelled as an MDP. Although a few works exist related to finding optimal mining strategies using single-agent reinforcement learning in reality, individual mining is not as popular as pool mining. Specifically, miners collaborate and compete with each other to mine blocks. Multi-agent reinforcement learning setting mixing collaborative and competitive agents is more suitable for modelling complex pool mining activity and can help miners find optimal mining strategies in the future.
- Cryptocurrency plays an important role, especially in public chains. Different chains have their own unique cryptocurrencies. Cryptocurrency and cryptocurrency portfolios are now an investment option similar to other financial products. Some works have studied cryptocurrency

price prediction using supervised learning techniques, but only a few explore the potentials of RL or deep RL. In many cases, RL and deep RL perform better in financial forecasting, e.g., for stock market prediction, due to the reason that historical data cannot reflect the dynamic market. We expect that more works adopting RL, deep RL, or inverse RL to study the investment return of cryptocurrencies will emerge soon.

## ACKNOWLEDGEMENTS

Guang Cheng gratefully acknowledges the support of National Science Foundation (NSF DMS-1712907, DMS-1811812, DMS-1821183), and the Office of Naval Research (ONR N00014-18-2759).

## BIBLIOGRAPHY

- Abay, N. C., Akcora, C. G., Gel, Y. R., Kantarcioglu, M., Islambekov, U. D., Tian, Y., & Thuraisingham, B. (2019). ChainNet: Learning on blockchain graphs with topological features. *2019 IEEE International Conference on Data Mining (ICDM)*, IEEE, New York, 946–951.
- Abernethy, J. D. & Frongillo, R. M. (2011). A collaborative mechanism for crowdsourcing prediction problems. In K. Q. Weinberger (Ed.) *Advances in Neural Information Processing Systems 24*, Curran Associates Inc., 2600–2608.
- Agbo, C. C., Mahmoud, Q. H., & Eklund, J. M. (2019). Blockchain technology in healthcare: A systematic review. *Healthcare*, 7, 56–56.
- Akcora, C. G., Dey, A. K., Gel, Y. R., & Kantarcioglu, M. (2019). Forecasting Bitcoin price with graph chainlets. In D. Phung, V. S. Tseng, G. I. Webb, B. Ho, M. Ganji, & L. Rashidi (Eds.) *Advances in Knowledge Discovery and Data Mining*, Springer International Publishing, Berlin, 765–776. (Cham, 2018).
- Akcora, C. G., Li, Y., Gel, Y. R., & Kantarcioglu, M. (2020). BitcoinHeist: Topological data analysis for ransomware prediction on the Bitcoin blockchain. *Twenty-Ninth International Joint Conference on Artificial Intelligence Special Track on AI in FinTech*, 4439–4445.
- Alessandretti, L., ElBahrawy, A., Aiello, L. M., & Baronchelli, A. (2018). Anticipating cryptocurrency prices using machine learning. *Complexity*, 2018, 1–16.
- Bhaskar, N. D. & Lee, K. C. (2015). Bitcoin mining technology. In D. L. K. Chuen (Ed.) *Handbook of Digital Currency: Bitcoin, Innovation, Financial Instruments, and Big Data*, Academic Press, San Diego, IL, 45–65.
- Bhattacharya, P., Tanwar, S., Bodke, U., Tyagi, S., & Kumar, N. (2019). BinDaaS: Blockchain-based deep-learning as-a-service in healthcare 4.0 applications. *IEEE Transactions on Network Science and Engineering*, IEEE, New York, <https://doi.org/10.1109/TNSE.2019.2961932>.
- Bresler, G., Chen, G. H., & Shah, D. (2014). A latent source model for online collaborative filtering. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.) *Advances in Neural Information Processing Systems 27*, Curran Associates Inc., 3347–3355.
- Chen, G. H., Nikolov, S., & Shah, D. (2013). A latent source model for nonparametric time series classification. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.) *Advances in Neural Information Processing Systems 26*, Curran Associates Inc., 1088–1096.
- Chen, X., Ji, J., Luo, C., Liao, W., & Li, P. (2018). When machine learning meets blockchain: A decentralized, privacy-preserving and secure design. *2018 IEEE International Conference on Big Data (Big Data)*, IEEE, 1178–1187.
- Dey, A. K., Akcora, C. G., Gel, Y. R., & Kantarcioglu, M. (2020). On the role of local blockchain network features in cryptocurrency price formation. *Canadian Journal of Statistics*, 48, 561–581.
- Dinh, T. N. & Thai, M. T. (2018). AI and blockchain: A disruptive integration. *Computer*, 51, 48–53.
- Ermilov, D., Panov, M., & Yanovich, Y. (2017). Automatic Bitcoin address clustering. *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, New York, 461–466.
- Eyal, I. & Sirer, E. G. (2014). Majority is not enough: Bitcoin mining is vulnerable. *Communications of the ACM*, 61, 95–102.
- Ferrag, M. A. & Maglaras, L. (2020). Deepcoin: A novel deep learning and blockchain-based energy exchange framework for smart grids. *IEEE Transactions on Engineering Management*, 67, 1285–1297.

- Firdaus, A., Anuar, N., Faizal, M., Hashem, I., Bachok, S., & Kumar, A. (2018). Root exploit detection and features optimization: Mobile device and blockchain based medical data management. *Journal of Medical Systems*, 42, 112–112.
- Ghimire, S. & Selvaraj, H. (2018). A survey on Bitcoin cryptocurrency and its mining. *2018 26th International Conference on Systems Engineering (ICSEng)*, 1–6.
- Gökalp, E., Gökalp, M. O., Çoban, S., & Eren, P. E. (2018). Analysing opportunities and challenges of integrated blockchain technologies in healthcare. In S. Wrycza & J. Maślankowski (Eds.) *Information Systems: Research, Development, Applications, Education (Cham, 2018)*, Springer International Publishing, Cham, 174–183.
- Granger, C. W. J. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37, 424–438.
- Greaves, A. S. & Au, B. (2015). *Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin*. Stanford Publishing, Stanford.
- Harrigan, M. & Fretter, C. (2016). The unreasonable effectiveness of address clustering. *2016 Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCCom/IoP/SmartWorld)*, IEEE, New York, 368–373.
- Harris, J. D. & Waggoner, B. (2019). Decentralized and collaborative AI on blockchain. *2019 IEEE International Conference on Blockchain (Blockchain)*, IEEE, New York, 368–375.
- He, X., Liu, J., Jin, R., & Dai, H. (2017). Privacy-aware offloading in mobile-edge computing. *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, IEEE, New York, 1–6.
- Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9, 1735–1780.
- Jang, H. & Lee, J. (2018). An empirical study on modeling and prediction of Bitcoin prices with Bayesian neural networks based on blockchain information. *Institute of Electrical and Electronics Engineers Access*, 6, 5427–5437.
- Jiang, Z. & Liang, J. (2017). Cryptocurrency portfolio management with deep reinforcement learning. *2017 Intelligent Systems Conference (IntelliSys)*, 905–913.
- Jourdan, M., Blandin, S., Wynter, L., & Deshpande, P. (2018). A probabilistic model of the Bitcoin blockchain. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, IEEE, New York, 2784–2792.
- Juneja, A. & Marefat, M. (2018). Leveraging blockchain for retraining deep learning architecture in patient-specific arrhythmia classification. *2018 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*, IEEE, New York, 393–397.
- Khezr, S., Moniruzzaman, M., Yassine, A., & Benlamri, R. (2019). Blockchain technology in healthcare: A comprehensive review and directions for future research. *Applied Sciences*, 9, 1736.
- Lahmiri, S. & Bekiros, S. (2019). Cryptocurrency forecasting with deep learning chaotic neural networks. *Chaos, Solitons and Fractals*, 118, 35–40.
- Li, Z., Guo, H., Wang, W. M., Guan, Y., Barenji, A. V., Huang, G. Q., McFall, K. S., & Chen, X. (2019). A blockchain and AutoML approach for open and automated customer service. *IEEE Transactions on Industrial Informatics*, 15, 3642–3651.
- Liu, C. H., Lin, Q., & Wen, S. (2018). Blockchain-enabled data collection and sharing for industrial IoT with deep reinforcement learning. *IEEE Transactions on Industrial Informatics*, 15, 3516–3526.
- Luong, N. C., Xiong, Z., Wang, P., & Niyato, D. (2018). Optimal auction for edge computing resource management in mobile blockchain networks: A deep learning approach. *2018 IEEE International Conference on Communications (ICC)*, IEEE, New York, 1–6.
- Mamoshina, P., Ojomoko, L., Yanovich, Y., Ostrovski, A., Botezatu, A., Prihodko, P., Izumchenko, E., et al. (2017). Converging blockchain and next-generation artificial intelligence technologies to decentralize and accelerate biomedical research and healthcare. *Oncotarget*, 9, 5665–5690.
- McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2016). Communication-efficient learning of deep networks from decentralized data. In A. Singh & J. Zhu (Eds.) *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, PMLR*, Vol. 54, 1273–1282.
- McNally, S., Roche, J., & Caton, S. (2018). Predicting the price of Bitcoin using machine learning. *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, IEEE, New York, 339–343.

- MIT Technology Review Editors. (2018). *Explainer: What is a blockchain?*. <https://www.technologyreview.com/2018/04/23/143477/explainer-what-is-a-blockchain/>.
- Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system*. <https://bitcoin.org/bitcoin.pdf>.
- Nguyen, D. C., Pathirana, P., Ding, M., & Seneviratne, A. (2020). Privacy-preserved task offloading in mobile blockchain with deep reinforcement learning. *IEEE Transactions on Network and Service Management*, IEEE, New York, 1–1.
- Özyilmaz, K. R., Doğan, M., & Yurdakul, A. (2018). IDMOB: IoT data marketplace on blockchain. *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, 11–19.
- Qin, R., Yuan, Y., & Wang, F. -Y. (2019). A novel hybrid share reporting strategy for blockchain miners in PPLNS pools. *Decision Support Systems*, 118, 91–101.
- Rathore, S., Pan, Y., & Park, J. H. (2019). BlockDeepNet: A Blockchain-based secure deep learning for IoT network. *Sustainability*, 11, 3974.
- Saleh, F. (2021). Blockchain without waste: Proof-of-stake. *The Review of Financial Studies*, 34, 1156–1190.
- Sapirshtein, A., Sompolinsky, Y., & Zohar, A. (2017). Optimal selfish mining strategies in Bitcoin. In J. Grossklags & B. Preneel (Eds.) *Financial Cryptography and Data Security (Berlin, Heidelberg, 2017)*, Springer, Berlin Heidelberg, 515–532.
- Shah, D. & Zhang, K. (2014). Bayesian regression and Bitcoin. *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 409–414.
- Shen, M., Tang, X., Zhu, L., Du, X., & Guizani, M. (2019). Privacy-preserving support vector machine training over blockchain-based encrypted IoT data in smart cities. *IEEE Internet of Things Journal*, 6, 7702–7712.
- Singla, K., Bose, J., & Katariya, S. (2018). Machine learning for secure device personalization using blockchain. *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 67–73.
- Specht, D. F. (1991). A general regression neural network. *IEEE Transactions on Neural Network*, 2, 568–576.
- Sun, X., Liu, M., & Sima, Z. (2020). A novel cryptocurrency price trend forecasting model based on lightGBM. *Finance Research Letters*, 32, 101084.
- Vyas, S., Gupta, M., & Yadav, R. (2019). Converging blockchain and machine learning for healthcare. *2019 Amity International Conference on Artificial Intelligence (AICAI)*, 709–711.
- Waggoner, B., Frongillo, R., & Abernethy, J. D. (2015). A market framework for eliciting private data. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, & R. Garnett (Eds.) *Advances in Neural Information Processing Systems*, 28, Curran Associates, Inc., 3510–3518.
- Wang, T., Liew, S. C., & Zhang, S. (2019). *When blockchain meets AI: Optimal mining strategy achieved by machine learning*, arXiv preprint arXiv:1911.12942.
- Wang, S., Wang, J., Wang, X., Qiu, T., Yuan, Y., Ouyang, L., Guo, Y., & Wang, F. (2018). Blockchain-powered parallel healthcare systems based on the ACP approach. *IEEE Transactions on Computational Social Systems*, 5, 942–950.
- Wohrer, M. & Zdun, U. (2018). Smart contracts: Security patterns in the Ethereum ecosystem and solidity. *2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, 2–8.
- Wood, G. (2014). *Ethereum: A secure decentralised generalised transaction ledger*. <https://ethereum.github.io/yellowpaper/paper.pdf>.
- Xiong, W. & Xiong, L. (2019). Smart contract based data trading mode using blockchain and machine learning. *IEEE Access*, 7, 102331–102344.
- Yin, H. S. & Vatrupu, R. (2017). A first estimation of the proportion of cybercriminal entities in the Bitcoin ecosystem using supervised machine learning. *2017 IEEE International Conference on Big Data (Big Data)*, IEEE, New York, 3690–3699.
- Zheng, X., Mukkamala, R. R., Vatrupu, R., & Ordieres-Mere, J. (2018a). Blockchain-based personal health data sharing system using cloud storage. *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)*, IEEE, New York, 1–6.
- Zheng, Z., Xie, S., Dai, H. -N., Chen, X., & Wang, H. (2018b). Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services*, 14, 352.
- Zhu, S., Li, W., Li, H., Hu, C., & Cai, Z. (2019a). A survey: Reward distribution mechanisms and withholding attacks in Bitcoin pool mining. *Mathematical Foundations of Computing*, 1, 393.

Zhu, X., Li, H., & Yu, Y. (2019b). Blockchain-based privacy preserving deep learning. In F. Guo, X. Huang, & M. Yung (Eds.) *Information Security and Cryptology (Cham, 2019)*, Springer International Publishing, Berlin, 370–383.

---

*Received 28 October 2019*

*Accepted 24 November 2020*