Machine Learning in/for Blockchain: Future and Challenges

Fang Chen[*], Hong Wan[†], Hua Cai[‡], and Guang Cheng[§]

September 16, 2019

Abstract

Machine learning (including deep and reinforcement learning) and blockchain are two of the most noticeable technologies in recent years. The first one is the foundation of artificial intelligence and big data, and the second one has significantly disrupted the financial industry. Both technologies are data-driven, and thus there are rapidly growing interests in integrating them for more secure and efficient data sharing and analysis. In this paper, we review the research on combining blockchain and machine learning technologies and demonstrate that they can collaborate efficiently and effectively. In the end, we point out some future directions and expect more researches on deeper integration of the two promising technologies.

Keywords: blockchain, machine learning, deep learning, Bitcoin.

1 Introduction

A blockchain is a shared, distributed public ledger that stores transaction data in a chain of sequential blocks [1]. The data (block) are time-stamped and validated before adding to the chain. Each block contains information from the previous one. The mathematical structure for storing data makes

[*]Ph.D. student, Department of Industrial Engineering, Purdue University.

[†]Associate Professor, Department of Industrial and System Engineering, North Carolina State University.

[‡]Assistant Professor, Department of Industrial Engineering, Purdue University.

[§]Corresponding Author. Professor, Department of Statistics, Purdue University. Guang Cheng gratefully acknowledges NSF DMS-1712907, DMS-1811812, DMS-1821183,and Office of Naval Research, (ONR N00014-18-2759).

it nearly impossible to fake [2]. Thanks to the legacy of cryptocurrency, the term "blockchain" has transformed from a cryptography terminology to a buzz word. Many people believe that cryptocurrency IS blockchain. This is incorrect. While blockchain is the foundation of cryptocurrency, the applications of the blockchain technology are much wider. Scenarios involving data validating, auditing, and sharing can all consider applying blockchains.

In this paper, we review the research on combining blockchain and machine learning technologies and demonstrate that they can collaborate efficiently and effectively. Machine learning (including deep learning [3]) is the core technology for big data analysis [4]. The nature of the blockchain: (1) as a distributed and append-only ledger system; and (2) its incorporation of smart contracts (i.e., a piece of code that will execute automatically in certain conditions), make it a natural tool for sharing and handling big data from various sources. More specifically, blockchain can preserve and encourage data sharing when training and testing machine learning models. Also, it allows us to utilize distributed computing powers (for example, IOT), for developing on-time prediction models with various sources of data. This is especially important for deep learning procedures which requires tremendous amount of computational power. On the other hand, blockchain systems will generate huge amount of data from different sources, and the distributed systems are harder to monitor and control than the centralized ones. Efficient data analysis and forecasting of the system behaviors are critical for optimal blockchain mechanism designs. In addition, machine learning can facilitate the data verification process and identifying malicious attack and dishonest transactions in the blockchain. The interdisciplinary research on combining the two technologies is of great potential.

In this paper, we will review three types of machine learning literature: one for incorporating machine learning into blockchain algorithms or frameworks; another for analyzing attributes or applications of blockchain using machine learning methods; and the third for blockchain-based learning systems, Papers that we include in our review are summarized in the table below, organized by different learning methods used. In the rest of this paper, we first review basic idea and terminology of blockchain in Section 2. The review is by no means exhaustive, but only sufficient for Sections 3, 4, and 5 that introduce how machine learning, deep learning and reinforcement learning can be incorporated into or improve blockchains. Our work is concluded by Section 6 that discusses possible research directions and challenges arising from the ongoing and future fusion of machine learning and blockchain.

Method	Application	Paper
ML	Transaction Entity Classification	Yin et al. (2017), Jourdan et al. (2018)
	Bitcoin Price Prediction	Jourdan et al. (2018), Akcora et al. (2019) Abay et al. (2019), Shah et al. (2014)
DL	Privacy and Security Preserving	Chen et al. (2018), Zhu et al. (2019)
	Computation Power Allocation	Loung et al. (2018)
	Cryptocurrency Price Prediction	Mcnally et al. (2018), Lahmiri et al. (2019) Alessandretti et al. (2018)
RL	IoT, Cryptocurrency Portfolio Management	Liu et al. (2018), Jiang et al. (2017)

Table 1: Summary of Reviewed Papers

2 Review on 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. New block needs to be approved by all or some of the network members to become a valid block. Each approved block includes information of the previous block in the chain, therefore if the block is changed, all blocks *before* this block will be invalid as well. The strategies to reach agreement of the new block (consensus) vary in different types of blockchain. The mathematical structure of the blockchain implies two essential properties: (i) the data (in block) is immutable; (ii) the distributed network with 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 outrages and attacks.

Figure 1 explains how the blockchain works in a financial scenario where a user wants to send assets to another user through the blockchain network. A transaction between two users is first created and then broadcast to every user in the network. Transactions are then validated and combined into blocks. Once a block is validated and approved, it is allowed to be added to the chain. The transaction is then executed and money moves from one user to another.

Depending on who can access to the blockchain and who can validate data, blockchains can be categorized into public chains, private chains, and consortium chains. Most of the cryptocurrencies are based on public chain. Although a fully distributed public blockchain that allows everyone to participate in the network is nearly impossible to forge, shortcomings including high power consumption on transaction validation and low efficiency for processing a transaction occur at the same time. To use in enterprise level,

private and consortium chains are developed for higher efficiency. A private chain is controlled and operated by one organization or a founder who take responsibilities for validating and processing transactions. New users need to apply for permissions from the organization before they can participate in the network. Besides transactions are visible to the organization, an user is able to determine who can access to its transaction rather than every user on the network. An example of a private blockchain is the IBM Hyperledger Fabric [1]. It is a blockchain platform to provide decentralized data storage solutions using smart contracts (Chaincode) for enterprises who enroll in the network through a trusted Membership Service Provider. Consortium chain is similar to the private chain except that it is managed by multiple users or organizations instead of one. Transactions are usually validated and processed by all or a subset of users. An example of a consortium blockchain is Quorum [2]. It is an open-source blockchain platform for companies to collaborate. A selected group of users are assigned voting rights. Private and consortium blockchains process transaction much faster than public blockchain, but they are less secure compared to the public chains. .

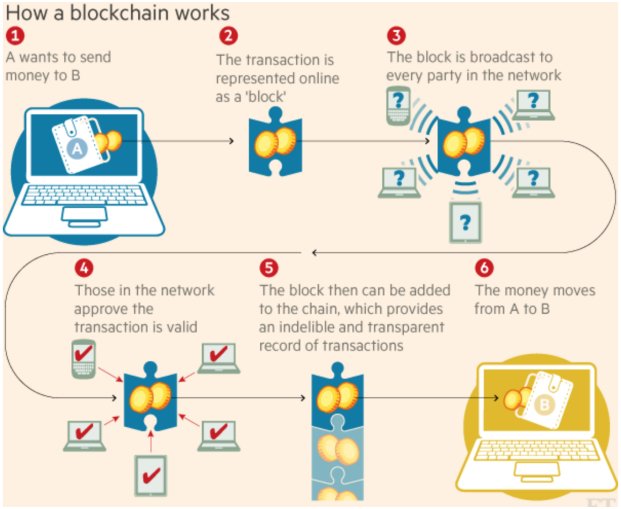

Figure 1: How Blockchain Works in the Financial Scenario [3]

In the remaining of the session, we use bitcoin system, which is the most well-know blockchain application, as an example to demonstrate how

[1] https://www.ibm.com/blockchain/hyperledger
[2] https://www.goquorum.com/
[3] Source: https://www.weforum.org/agenda/2016/06/blockchain-explained-simply

blockchain works in detail [5, 6, 7, 8]. At the end we also briefly discuss how Ethereum[9, 10], another popular public chain system, is different from bitcoin and introduce smart contract concept.

The bitcoin working mechanism is demonstrated in Figure 2. Transactions in Bitcoin are defined as transferring the cryptocurrency from one node (input address) to the other node (output address) without a third party being involved. Here nodes are devices connected to the blockchain network, being responsible for storing, verifying and broadcasting blocks of transactions constantly in order to keep all data up to date. In traditional centralized banking system, the transaction will be handled by a person or machine of the bank. In bitcoin system, the transaction will be broadcast to all users in the network for validation and bookkeeping. Every transaction has a unique hash served as transaction identifier. Hash is generated by the hash function which converts any strings or number to a unique fixed length output. A small change in the input will cause a big change in output. Bitcoin uses SHA-256 Hashing algorithm. Under the SHA-256 algorithm, any lengths of the input is transferred to a fixed 256-bits output.

The transaction between two nodes is completed in two stages. The first stage is in the level of nodes. Suppose a transaction T_{AB} is initiated by Node A who wants to send digital coins to Node B. Both A and B will obtain a unique pair of keys (strings of characters), a private key and a public key. The transaction data contains messages of the input transaction address (Node A address) generated by public key of Node A, the output transaction address (Node B address) generated by public key of Node B, and the original transaction data with corresponding transaction hash. Then Node A sends messages along with its digital signature generated by the private key of A to Node B. After receiving the transaction T_{AB} from Node A, Node B verifies the transaction T_{AB} by comparing two hash values generated by the digital signature and the original transaction data, respectively. If two hash values are equal, the transaction T_{AB} is verified and uploaded to a transaction pool. The process of transaction creation is illustrated in Figure 2. The transaction pool is a set of transactions, which are waiting to be added to blocks in the blockchain network.

In the second stage, the active participants of the network will aggregate transactions to form blocks, and they will compete to append their own block to the blockchain network. The process is also known as mining. The mechanism to determine whether a block can be added to the chain is called consensus. Bitcoin blockchain applies proof-of-work (PoW) as its consensus protocol. Intuitively, the mining nodes will solve a hard math problems to find a specific hash value that fulfill the requirement, usually a specific

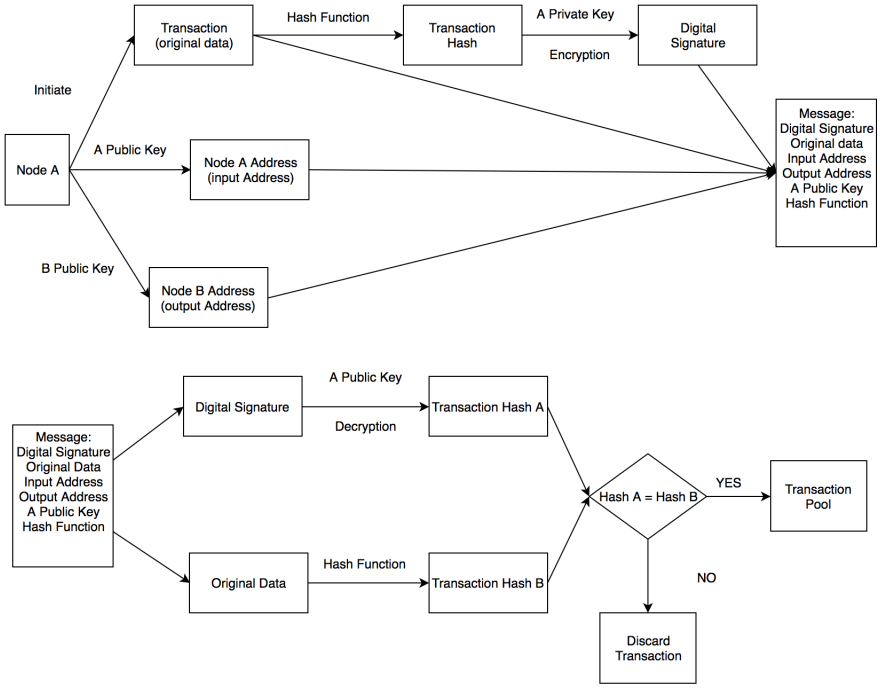

Figure 2: How Bitcoin Blockchain Works

number of zeros at the start of the hash value. The node will broadcast its block to the whole network when it finds the value. Once information on the new block is validated by a majority of nodes, the new block is appended to the blockchain and the node that first creates the block is awarded a certain number of Bitcoin. More specifically, a node receives Bitcoins for its validation work if a new block is created and attached to the blockchain successfully.

The input and output of the transaction is called the spent transaction and the unspent transaction, respectively. The Bitcoin blockchain utilizes values of unspent transaction outputs (UTXOs) to record the balance of each node. Only UTXO from the previous transactions can be used as an input in a new transaction. For instance, suppose a Node A has two UTXOs that record 10 Bitcoins and 20 Bitcoins respectively and plans to send 25 Bitcoins to Node B. Node A need to spend both UTXOs as its input transaction. At the end of the transaction, two input UTXOs are spent and removed from the UTXO set of Node A. A new UTXO recording 5 Bitcoins is returned to

Node A while another new UTXO recording 25 Bitcoins are sent to Node B. The process is shown in Figure (3). Node A can have a new address to receive UTXOs.

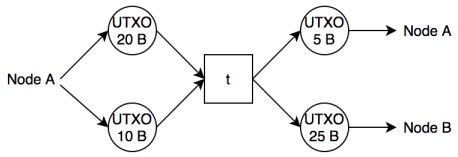

Figure 3: UTXOs Involved in Transaction

Compare to the bitcoin network, under the Ethereum platform, nodes not only can transfer digital coins (Ether), but also can send *smart contracts*. A smart contract is a piece of automatically executed code given certain conditions. More specifically, it allows the execution of a transaction without third parties. Ethereum adopts proof-of-stake (PoS) for generating and adding blocks. Miners are replaced by validators, and they vote on which block will be added next to the chain. The more stakes (usually the cryptocurrency) a node have, the more voting power it will have. Therefore, in PoW, the probability of generating a new block relies on how much computing power every node spends. In PoS, the probability of creating a new block depends on how many coins each node has. The node obtaining a larger number of coins has a larger probability of creating a new block.

3 Machine Learning for Blockchain

In this section, we review several applications of machine learning in the blockchain. Specifically, Section 3.1 reviews two studies regarding transaction entities classification [11, 12] with different purposes. One focuses on the recognition of cybercriminal entities [11], while another on the recognition of common categories of entities that most transactions belong to [12]. Section 3.2 reviews Bitcoin price prediction from different perspectives [12, 13, 14, 15] varying from probabilistic graphic models, Bayesian regression.

3.1 Transaction Entity Classification

In Bitcoin transactions, it is crucial to recognize entities behind those potentially illegal ones. Yin et al. (2017) apply supervised learning to classify entities of transactions that may involve in cybercriminal activities. The

classification algorithm is trained based on 854 observations with categorical identifiers and then applied to study 10000 observations without categorical identifiers, which evaluates 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 accuracy scores of all classifiers, it is found that Random Forests(77.38%), Extremely Randomised Forests(76.47%), Bagging(78.46%) and Gradient Boosting(80.76%) stand out as the best four classifiers. After further comparing precision, recall, and f1 score of these classifiers, bagging and gradient boosting stand out, which are then applied to analyze the 10000 observations. The classification outcome shows that 5.79% (3.16%) addresses and 10.02% (1.45%) coins are from cybercriminal entities according to the bagging method (gradient boosting method).

Jourdan et al. (2018) are interested in classifying entities of transactions into four most common categories: Exchange, Service, Gambling, Mining Pool, based on data collected from 97 sources [16]. The goal of classification is to assist in selecting an appropriate prediction model that is built according to categories of transactions [12]. The applied classification method is a gradient boosted decision tree algorithm along with a Gaussian Process-based optimization procedure that determines optimal hyperparameters. Figure 4 concludes that accuracy in Exchange, Gambling, and Service categories are high. However, the accuracy in the Mining Pool category is poor. This may indicate that mining activity may not be used as an independent label.

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

Figure 4: Classification Performance [12]

3.2 Bitcoin Price Prediction

UTXOs record the number of Bitcoins in transactions, which enable us to track buying and selling information so as to predict the Bitcoin price. Another contribution of Jourdan et al. (2018) is to forecast the value of UTXOs

by creating probabilistic graphical models. The first model is called Block-transaction address model (BT-A) that is a stationary graphic model of a Bitcoin block with conditional dependency structures. As an extension of BT-A, a Block-transaction entity-address model (BT-EA) is further developed by adding a categorical entity to each address. In terms of MSE, RMSE, MAE, RMAE[4], simulation result in Figure 5 shows that this extension significantly outperforms BT-A in all categories except for Exchange.

Metric	BT-EA				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 5: BT-A and BT-EA Performance [12]

The dependent structure of BT-A model to obtain the output UTXOs values, denoted as $V_{o,u}$, is illustrated in Fig 6. Here is some explanation. The BT-A model starts with computing the number of available UTXOs for i^{th} input address A_i, denoted as $k_{A_i}^{UTXO}$. For each input address, the number of UTXOs used in a transaction is uniformly drawn from 1 to $k_{A_i}^{UTXO}$ with the corresponding UTXO value, denoted as $V_{i,u}$. The total input value of a transaction is calculated by summing the input UTXOs value of each input address, denoted as $V_t = \sum V_{i,u}$, and the value of an output UTXO is uniformly drawn from 1 to total transaction value minus validation fee.

Another more direct way to predict Bitcoin price is to use Bayesian regression for "latent source model" [17] as firstly done by Shah et al. (2014). Specifically, the Bitcoin price, denoted as y, is predicted given features x and the latent source model refers to equation 3.1.

$$P(y|x) = \sum_{k=1}^{T} P_k(y) \exp\left(-\frac{1}{2}||x - s_k||_2^2\right) \mu_k \tag{3.1}$$

where $s_1, ..., s_k$ are K distinct unknown latent sources[5] (time series) that are never estimated, P_k is a latent distribution associated with probability μ_k,

[4]MSE is mean squared error; RMSE is the root mean squared error; MAE is mean absolute error; RMAE is the root mean absolute error

[5]An example of latent sources is Twitter activity of a news topic that becomes a trend following one of a finite number of patterns [17]

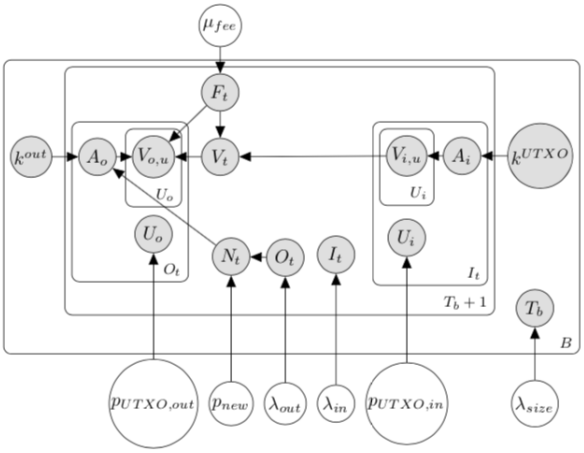

Figure 6: Block-transaction Address Model [12]

denoted as $P_k(T = k) = \mu_k$, where $T \in \{1, ..., K\}$ is a sample index. The expectation of $P(y|x)$ can be estimated as follows:

$$E[y|x] = \frac{\sum_{i=1}^{n} y_i \exp(-\frac{1}{4}||x - x_i||_2^2)}{\sum_{i=1}^{n} \exp(-\frac{1}{4}||x - x_i||_2^2)} \tag{3.2}$$

The future average price change is determined by price changes over three length of historical data, denoted as $\Delta p^j, j = 1, 2, 3$, previous 30 minutes sample, 60 minutes sample and 120 minutes sample. Each Δp^j is calculated by (3.2). Then Δp over a 10-second period is formulated as

$$\Delta p = w_0 + \sum_{j=1}^{3} w_j \Delta p^j + w_4 r \tag{3.3}$$

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

The trading strategy for each user is designed as "buy one bitcoin when $\Delta p > t$; sell one bitcoin when $\Delta p < -t$; otherwise holding the current number of bitcoin when $-t \leq \Delta p \leq t$, where t is a threshold. The designed prediction model is trained by data gathered from Okcoin before May, 2014 and is tested by data after May 2014. It is found that increasing t leads to an increases of average profit per trade.

To better characterize input features, Akcora et al. (2019) introduce a concept of graphic chainlet, which describes the local topological features of Bitcoin blockchain, to explore impacts of the Bitcoin blockchain structure on Bitcoin price formation and dynamics. A transaction-address graph representation of a single transaction of Bitcoin blockchain is shown in Figure 7. Circle vertices represent input and output address. A square vertex indicates the transactions and edges stand for UTXOs (a transfer of Bitcoins). A chainlet model represents x input UTXOs and y output UTXOs involving in a transaction, denoted as $C_{x\to y}$. From the Granger causality test, it is found that the split chainlet cluster defined as when $y < x < 20$, individual chainlet (e.g., $C_{1\to 7}$, $C_{6\to 1}$, $C_{3\to 3}$), extreme chainlets (e.g., $C_{20\to 2,3,12,17}$), certain clusters using Cosine Similarity (e.g., $C_{9\to 11}$, $C_{3\to 17}$, $C_{8\to 14}$, $C_{1\to 1}$) are significant to Bitcoin price dynamics, which can be further used as inputs to develop a more precise prediction model.

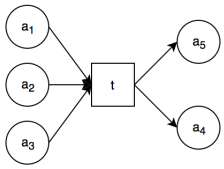

Figure 7: A Transaction-Address Graph

Chainlet model studies topological features from a single transaction aspect and only takes the number of input and output UTXOs into account. Abay et al. (2019) extend chainlet model to a new graphic model "ChainNet" that further assess topological features from all or multiple transactions and take the amount of transferred Bitcoin into consideration. More specifically, from the aspect of all transactions, an occurrence matrix is created to count the number of distinct chainlet among all transactions. An amount matrix records the sum of Bitcoins transferred for distinct chainlet. Combining occurrence and amount matrix, an occurrence matrix with a threshold[6], denoted as O^{ϵ}, is created to count the number of distinct $C_{i\to j}$ larger than ϵ. Different thresholds result in different O^{ϵ}, which are considered as Filtration Features (FL) input in the prediction model.

Although there are other studies related to Bitcoin price prediction using machine learning, i.e., [18, 19], it is hard to include all papers in the review. As a result, we will move on to review more articles in prediction

[6]threshold, denoted as $\epsilon, \epsilon \in \{0, 10, 20, 30, 40, 50\}$

of cryptocurrency price using deep learning and reinforcement learning in Section 4 and Section 5.

4 Deep Learning

In this section, we turn to the application of deep learning. In Section 4.1, two learning frameworks [20, 21], which train the predictive model and improve system security in a privacy-preserving manner, are reviewed. In Section 4.2, we review a deep learning work [22] that allocates computation resource to assist mobile blockchain mining. In Section 4.3, we focus on cryptocurrency price prediction [23, 24] and digital portfolio management [25] using Recurrent Neural Network (RNN) and Long-Short Term Memory (LSTM) models.

4.1 Data Privacy and System Security Preserving

Chen et al. (2018) propose a framework called "Learning Chain" to preserve user's privacy by applying a decentralized version of Stochastic Gradient Descent (SGD) algorithm and a differential privacy mechanism. The proposed framework contains three phases: blockchain initialization; local gradient computation; 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 as w_t, and computing its own local gradient. A differential privacy mechanism is then applied to generate a hidden local gradient, denoted as $\nabla g_k(w_t)^*$, by adding a noise factor to the local gradient. The message broadcasts a pseudo identity of P_k, normalized hidden local gradient, denoted as $\nabla \widehat{g_k}(w_t)^*$, together with the norm of its un-normalized version to computing nodes on the network. In the final phase, after solving Proof-of-Work (PoW), the winner node selects top l-nearest local normalized gradients according to the cosine distance between each normalized local gradient and the sum vector of $\nabla g_k(w_t)^*$ to update the global gradient. The predictive model is updated by $w_{t+1} = w_t + \eta \nabla J(w_t)$, where $\nabla J(w_t)$ is the updated global gradient.

"Learning Chain" is trained and tested in three different data sets: synthetic data set; Wisconsin breast cancer data set; MNIST data set; using 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 of test errors on all data sets. This proposed model is further

compared with the "Learning ChainEX", which is implemented with higher differential privacy and has similar test error.

Zhu et al. (2019) develop a blockchain-based privacy-preserving framework to secure the share of updates in federated learning. Federated learning algorithm is developed by [26], which allows each mobile device to compute and upload updates to the global predictive model based on their local data sets. A security issue arises when there exist Byzantine devices in the network. In this case, the blockchain transaction mechanism is adopted to ensure the security of sharing and updating changes. Specifically, model updates are written in a blockchain transaction by nodes. Along with the digital signature of a node, a transaction broadcasts to other nodes information including changes of hyperparameters and weights, public keys (participants' addresses). Other nodes 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 the existing model under their local data sets, the updates are implemented into the existing model.

4.2 Computing Power Allocation

Luong et al. (2018) develop a deep learning based auction algorithm for edge computing resources 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 as v_i, is drawn from a distribution that assigns a higher value v_i when its block size divided by 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, denoted as $\overline{v}_i := \phi_i(v_i)$, to assignment probabilities using a Softmax function. The winner miner i will pay the price $p_i := \phi_i^{-1}(\text{ReLU}(\max_{i \neq j} \overline{v}_j))$. In the end, the loss function of ECSP is defined as

$$\widehat{R}(\mathbf{w}, \beta) = -\sum_{i=1}^{N} g_i^{(\mathbf{w},\beta)}(\mathbf{v}^s) p_i^{(\mathbf{w},\beta)}(\mathbf{v}^s), \tag{4.1}$$

where stochastic gradient descent (SGD) is applied. Here, g_i is the assignment probability and N is the number of miners. The above designed deep learning (DL) based auction mechanism is empirically compared to regular auction mechanism. It is found that DL-based auction generates a higher revenue and converges to the optimal value faster than competitors.

4.3 Cryptocurrency Price Prediction

For forecasting Bitcoin price, Mcnally et al. (2018) compare performances of two deep learning algorithms, i.e., Recurrent Neural Network (RNN) and Long-Short Term Memory (LSTM). It is interesting to note that two hidden layers with 20 nodes per layer are sufficient in both models. Specifically, RNN model adopts tanh as its activation function while LSTM applies tanh and sigmoid functions for different gates, which result in longer training time. The data set used to train and test LSTM and RNN models is the bitcoin price from Aug 19th, 2013 to July 19th, 2016. The traditional time series model, AutoRegression Integrated Moving Average (ARIMA), is empirically compared with these deep learning models. The simulation results show that LSTM, RNN, and ARIMA have similar accuracy, which are 52.78%, 50.25%, and 50.05%. However, deep learning models have much lower RMSE values. In addition, LSTM model is capable of recognizing long-term dependencies in contrast to RNN model.

In contrast with other studies mainly for predictive models, Lahmiri et al. (2019) instead conduct a chaotic time series analysis before building deep learning models. Hence, their first step is to calculate largest Lyapunov exponent (LLE) and apply detrended fluctuation analysis (DFA) to data sets for chaos assessment. Then a deep neural network (DLNN) model with LSTM implementation [27] and a generalized regression neural network (GRNN) model [28] are built to predict three types of cryptocurrency: Bitcoin, Digital Cash, and Ripple price. According to Figure 8, positive Hurst exponent (HE) value indicates long-memory features of data and negative LLE value indicates training data is chaos. As a result, a short-term prediction model would be suitable for data. The simulation results claim that LSTM model outperforms GRNN model in all three cryptocurrencies' price prediction. Although RMSE of LSTM model is still high, the model demonstrates a similar trend to real price changes for all three cryptocurrencies.

Besides cryptocurrency price prediction, Alessandertti et al. (2018) explore a portfolio analysis by forecasting daily prices of 1681 types of cryptocurrencies. Three models are developed to predict prices of each type of cryptocurrency. For each type c, the target is the return of investment (ROI) at each time $t_i \in \{0, ..., 895\}$, which is expressed as:

$$\text{ROI}(c, t_i) = \frac{\text{price}(c, t_i) - \text{price}(c, t_i - 1)}{\text{price}(c, t_i - 1)}. \tag{4.2}$$

The data features considered are price, market capitalization, market share, rank, and volume. The first two models are decision tree models using dif-

	LLE		HE	
	Training sub-sample	Testing sub-sample	Training sub-sample	Testing 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

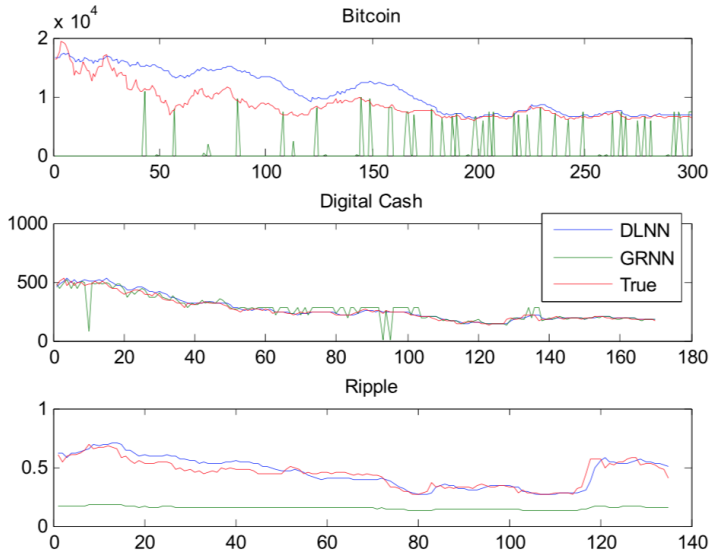

Figure 8: Chaotic Analysis and Prediction Result [24]

ferent target and feature pairing strategies, while the third model utilizes RNN combined with LSTM. A portfolio is constructed based on the predicted prices, and model hyperparameters are optimized by maximizing either sharp ratio or geometric mean of the total return. The result concludes that all three models generate profits and the optimization of parameters using the sharp ratio metric achieve a higher return. Another conclusion is that the first two models implementing gradient boosting decision tree have higher accuracy in short-term (5-10 days) while the third model adopting LSTM has a better prediction accuracy in the long term (around 50 days).

5 Reinforcement Learning

In this section, we review one work [29] that incorporates reinforcement learning into blockchain in order to ensure the security of data collection, storage and processing in the IoT. Another study [30] regarding cryptocurrency portfolio management using reinforcement learning will also be re-

viewed.

Liu et al. (2018) propose a framework to secure data collection and sharing among mobile terminals (MTs) on the IoT network. The framework consists of two phases: data collection and data sharing. In the data collection phase, each MT, denoted as m, adopts multi-agent deep reinforcement learning (DRL) to maximize efficacy of data collection. The state space is defined as $S = \{S_1, S_2, S_3\}$, where S_1, S_2 and S_3 represent locations of Point-of-Interest (PoIs), MTs' locations and sensing time $h_t(k) \in [0, t]$ for the i-th POI. Action space consists of moving direction, denoted as θ_t^m, and moving distance, denoted as l_t^m. Thus, it is written as $A = \{(\theta_t^m, l_t^m) \mid \theta_t^m \in [0, 2\pi), l_t^m \in [0, l_{max})\}$. The reward r_t^m is given as

$$r_t^m = \frac{w_t b_t^m}{\alpha b_t^m + \kappa l_t^m} \tag{5.1}$$

where b_t^m is the amount of collected data, α, κ are the energy consumption per collected data and per travelled distance; w_t is the achieved geographical 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 actor-critic algorithm is applied to maximize the reward.

After MTs finish the data collection, they share data through an Ethereum blockchain network. However, the first step would be to send data to the certificate authority (CA) for verification. Once CA verifies the ownership of MTs' data and checks the consistence of received data and original data stored in the terminal, a digital signature is generated and sent back to the MT. As a result, the MT is able to broadcast its transaction request consisting of digital signature of CA, original data and its public key to other nodes on blockchain network to be further validated. By comparing to randomly moving MTs, MTs implemented DRL collect much more data but consume more energy. The blockchain-based data sharing framework can still store all data sent by MTs even under Dos attack.

Jiang et al. (2017) conduct a study for cryptocurrency portfolio management using deep reinforcement learning. In contrast to other studies whose prediction models output cryptocurrency prices, their CNN model produces a portfolio weight vector instead. The state space is a history price matrix that records all asset prices in each time period. The corresponding action would be to change portfolio weights, denoted as $\vec{w_t}$, at each time

period t. Instead of estimating Q function, deterministic gradient policy is implemented using a direct reward function. The reward function through n periods is to calculate the average logarithmic return as follows.

$$R = \frac{1}{n} \sum_{t=0}^{n} \ln(\vec{w_t} \cdot \vec{y_t}) \tag{5.2}$$

where each $\vec{y_t}$ is the price change vector of t^{th} trading period. A Softmax function is applied to the output layer to ensure $\sum_i w_{t,i} = 1$. The model is trained by gradient based methods. Adam Optimization is used to find the optimal hyperparameters and l_2 regularization is applied to avoid overfitting. Simulation results show that the CNN model with two hidden layers has the best performance. Three benchmarks strategies and three algorithms summarized by [31] are compared to the best designed model. It is found that CNN model is only inferior to Passive Aggressive Mean Reversion model, in term of accumulative return. However, CNN model achieves a significant lower risk by calculating Shape Ratio.

6 Conclusion and Future Challenges

The research we review either applies blockchain in a database to improve users' privacy in learning process; or uses machine learning to optimize computer resource allocation or cryptocurrency investment decisions. The majority can be categorized as applying one technique to another; few is the actual integration of the two technologies. Hence, it is fair to say the current research is still very preliminary from an interdisciplinary perspective.

However, we expect many research emerging in the following areas:

- Design "smart agents" with learning abilities to regulate the blockchain and detect abnormally behaviors. The former is especially important for consortium chain and private chain that requires coordination among users, while the latter is critical for public chain;

- The learning-based analysis of blockchain-based system is rare. From financial system to supply chain, there are enormous amount of data available to evaluate the performance of the decentralized structure of blockchain compared with the traditional centralized one. Learning-based analysis can shed insights on the mechanism design of the blockchain structures and provide on-time forecasting models;

- Blockchain to allow anonymously data sharing. With the development of IOT and wearable device, the privacy issue catches more and more attention of users. Combining with data fusion, we can design multiple-layer blockchain structures that allow sophisticated authorization of data for different users.

References

[1] Thang N Dinh and My T Thai. "AI and Blockchain: A Disruptive Integration". eng. In: *Computer* 51.9 (2018), pp. 48–53. ISSN: 0018-9162.

[2] MIT Technology Review Editor. "Explainer: What is a Blockchain?" In: (2018). URL: `https://www.technologyreview.com/s/610833/explainer-what-is-a-blockchain/`.

[3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep Learning". In: *Nature* 521 (2015), pp. 436–444.

[4] Peter Buhlmann et al. "Handbook of Big Data". In: *CRC Press* (2019).

[5] Zibin Zheng et al. "Blockchain challenges and opportunities: a survey". eng. In: *Int. J. of Web and Grid Services* 14.4 (2018). ISSN: 1741-1106. URL: `http://www.inderscience.com/link.php?id=95647`.

[6] I.-C. Lin and T.-C. Liao. "A survey of blockchain security issues and challenges". In: *International Journal of Network Security* 19.5 (2017), pp. 653–659. ISSN: 1816353X.

[7] Zibin Zheng et al. "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends". eng. In: *2017 IEEE International Congress on Big Data (BigData Congress)*. IEEE, 2017, pp. 557–564. ISBN: 9781538619964.

[8] Satoshi Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System". In: (2018). URL: `https://bitcoin.org/bitcoin.pdf`.

[9] G. Wood. "Ethereum: A secure decentralised generalised transaction ledger". In: (2014).

[10] V. Buterin. "A next-generation smart contract and decentralized application platform". In: (2014).

[11] Haohua Sun Yin and Ravi Vatrapu. "A first estimation of the proportion of cybercriminal entities in the bitcoin ecosystem using supervised machine learning". In: *2017 IEEE International Conference on Big Data (Big Data)*. Vol. 2018-. IEEE, 2017, pp. 3690–3699. ISBN: 9781538627150.

[12] Marc Jourdan et al. "A Probabilistic Model of the Bitcoin Blockchain". In: *CoRR* abs/1812.05451 (2018).

[13] Devavrat Shah and Kang Zhang. "Bayesian regression and Bitcoin". In: (2014).

[14] Cuneyt G. Akcora et al. "Forecasting Bitcoin Price with Graph Chainlets". In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Dinh Phung et al. Cham: Springer International Publishing, 2018, pp. 765–776. ISBN: 978-3-319-93040-4.

[15] Nazmiye C. Abay et al. "ChainNet: Learning on Blockchain Graphs with Topological Features". In: (2019).

[16] D. Ermilov, M. Panov, and Y. Yanovich. "Automatic Bitcoin Address Clustering". In: *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. Dec. 2017, pp. 461–466. DOI: 10.1109/ICMLA.2017.0-118.

[17] George H. Chen, Stanislav Nikolov, and Devavrat Shah. "A Latent Source Model for Nonparametric Time Series Classification". In: (2013).

[18] Alex S. Greaves and Benjamin Au. "Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin". In: 2015.

[19] Isaac Madan. "Automated Bitcoin Trading via Machine Learning Algorithms". In: 2014.

[20] Xuhui Chen et al. "When Machine Learning Meets Blockchain: A Decentralized, Privacy-preserving and Secure Design". In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 1178–1187. ISBN: 9781538650356.

[21] Xudong Zhu, Hui Li, and Yang Yu. "Blockchain-Based Privacy Preserving Deep Learning". In: *Information Security and Cryptology*. Ed. by Fuchun Guo, Xinyi Huang, and Moti Yung. Cham: Springer International Publishing, 2019, pp. 370–383. ISBN: 978-3-030-14234-6.

[22] N. C. Luong et al. "Optimal Auction for Edge Computing Resource Management in Mobile Blockchain Networks: A Deep Learning Approach". In: *2018 IEEE International Conference on Communications (ICC)*. May 2018, pp. 1–6. DOI: 10.1109/ICC.2018.8422743.

[23] Sean Mcnally, Jason Roche, and Simon Caton. "Predicting the Price of Bitcoin Using Machine Learning". In: *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*. IEEE, 2018, pp. 339–343. ISBN: 9781538649756.

[24] Salim Lahmiri and Stelios Bekiros. "Cryptocurrency forecasting with deep learning chaotic neural networks". eng. In: 118 (2019), pp. 35–40. ISSN: 0960-0779.

[25] Laura Alessandretti et al. "Anticipating Cryptocurrency Prices Using Machine Learning". In: *Complexity* 2018 (2018). ISSN: 1076-2787.

[26] H. Brendan McMahan et al. "Communication-Efficient Learning of Deep Networks from Decentralized Data". In: (2016).

[27] Sepp Hochreiter and Jüürgen Schmidhuber. "Long Short-Term Memory". eng. In: *Neural Computation* 9.8 (1997), pp. 1735–1780. ISSN: 0899-7667.

[28] Specht DF. "A general regression neural network". In: (1991).

[29] Chi Harold Liu, Qiuxia Lin, and Shilin Wen. "Blockchain-enabled Data Collection and Sharing for Industrial IoT with Deep Reinforcement Learning". eng. In: *IEEE Transactions on Industrial Informatics* PP.99 (2018), pp. 1–1. ISSN: 1551-3203.

[30] Z. Jiang and J. Liang. "Cryptocurrency portfolio management with deep reinforcement learning". In: *2017 Intelligent Systems Conference (IntelliSys)*. Sept. 2017, pp. 905–913. DOI: 10.1109/IntelliSys.2017.8324237.

[31] Bin Li et al. "PAMR: Passive aggressive mean reversion strategy for portfolio selection". In: *Machine Learning* 87.2 (May 2012), pp. 221–258. ISSN: 1573-0565. DOI: 10.1007/s10994-012-5281-z. URL: https://doi.org/10.1007/s10994-012-5281-z.