An Analysis of Transaction Handling in Bitcoin

Befekadu G. Gebraselase, Bjarne E. Helvik, Yuming Jiang
Department of Information Security and Communication Technology
NTNU, Norwegian University of Science and Technology, Trondheim, Norway
{befekadu.gebraselase, bjarne, yuming.jiang }@ntnu.no

Abstract—Bitcoin has become the leading cryptocurrency system, but the limit on its transaction processing capacity has resulted in increased transaction fee and delayed transaction confirmation. As such, it is pertinent to understand and probably predict how transactions are handled by Bitcoin such that a user may adapt the transaction requests and a miner may adjust the block generation strategy and/or the mining pool to join. To this aim, the present paper introduces results from an analysis of transaction handling in Bitcoin.

Specifically, the analysis consists of two parts. The first part is an exploratory data analysis revealing key characteristics in Bitcoin transaction handling. The second part is a predictability analysis intended to provide insights on transaction handling such as (i) transaction confirmation time, (ii) block attributes, and (iii) who has created the block. The result shows that some models do reasonably well for (ii), but surprisingly not for (i) or (iii).

Index Terms—Bitcoin, Transaction handling, Linear and non-linear prediction models, Classification, Machine Learning, Artificial Intelligence

I. Introduction

Blockchain has been considered as an essential technique to resolve privacy and security issues of Big Data, such as privacy protection in IoT [40] and Big Data analytics [17, 20]. To this aim, understanding transaction handling in Bitcoin, e.g., explanatory data analysis or predictability analysis, such as transaction confirmation time, block attributes, is necessary. This naturally demands a thorough study of the transaction characteristics of the blockchain [28], which gives helpful insight into designing and developing new blockchains for smart data or managing big data.

Bitcoin is the first and the largest decentralized electronic cryptocurrency system that uses blockchain technology [27]. It adapts a cryptographic proof of work (PoW) mechanism that allows anonymous peers to create and validate transactions through the underlying peer-to-peer (P2P) network. The peers that maintain and update the chain of blocks are called miners [38, 39]. In addition to transaction generation by user nodes, transaction handling in Bitcoin is done by the full nodes, among which, the miners play a central role: They find the mathematical puzzle to generate a valid block confirming the related transactions.

Due to the design and structure of proof of work (PoW) in Bitcoin, the difficulty of finding the mathematical puzzle increases exponentially, every 2016 blocks. As a consequence, independent miners struggle to find the puzzle. This has forced miners to collaborate to form a team to find the puzzle through a combined computational effort, a *mining pool* [22], and earn

a reward, depending on their overall mining power share and the reward mechanism and policy of the mining pool [29] [30]. The mining pools' behavior significantly affects the Bitcoin end users since the mining pools process most of the users' transactions: The throughput of Bitcoin depends partially on those major miners [39]. Additionally, as the number of users increases, the system's internal traffic of transaction handling escalates faster than expected, and at the same time, the throughput requirement increases proportionally with the number of users.

This paper investigates how transactions are handled by the Bitcoin system. The aim is to, through analyzing transaction handling, provide valuable insights to both users and miners:

- A user may expect when his/her transaction will be confirmed and hence choose an appropriate time to request a transaction to reduce the waiting time.
- A miner may define block generation strategies that utilize the current state of the system.
- A miner may also explore which mining pools are more recognizable in the block generation and use this knowledge to join or dis-join a mining pool.

Specifically, through an exploratory data analysis, we reveal key transaction handling characteristics and provide answers to several fundamental transaction handling questions, such as, what is the current throughput, how frequently blocks are generated, how long it takes for a transaction to be approved, and who has created a block. Besides, through a predictability analysis on throughput related features and classification of mining pools, we provide additional insights on these fundamental questions.

The investigation is based on a dataset collected at a Bitcoin full node which contains transaction handling information over a period of 543 days from 7th March, 2019 to 31st August 2020. As a highlight, the dataset includes locally available information that cannot be found on the public ledger blockchain. The results indicate that with a proper prediction model taking into account both internal and external factors, the prediction performance can be appealing for block size and number of transactions in a block, as well as for block generation intensity. However, in terms of predicting when a next block will be generated and a transaction be approved, the effort does not lead to conclusive observation. In addition, also surprisingly, in predicting / classifying the mining pool, clear distinguishing is only found for one specific mining

pool, the F2Pool. Discussion is provided for these findings, including the surprising ones, with the help of findings from the exploratory analysis.

The rest of the paper is organized as follows. Section II illustrates the workflow of transaction handling in Bitcoin, and introduces the dataset used in the analysis. Then, Section III introduces our analysis approach, highlighting the adopted statistical and artificial intelligence techniques. Following that, an exploratory analysis on the dataset is conducted and results are reported and discussed in Section IV. Next, Section V reports results and findings from the predictability study. The current state of the art is covered in Section VI. Finally, Section VII concludes the paper.

II. BITCOIN TRANSACTION HANDLING: WORKFLOW AND DATASET

A. Workflow

Bitcoin is a distributed ledger platform that enables information about transactions to be distributed than centralized, where the ledger is the Bitcoin blockchain that records the transactions. In Bitcoin, all full nodes, also called miners, take part in creating and validating/invalidating transaction blocks and propagating such information, independently [27]. Specifically, the users generate transactions for being processed, and the distributed ledger components, i.e. the full nodes or miners, work together to generate and validate transaction blocks and add them to the blockchain.

Fig.1 illustrates the workflow of transaction handling in Bitcoin, which includes transaction arrival, block formation, propagation and validation. Briefly, after transactions are generated by the users, they are sent to all full nodes for validation. At a full node, upon the arrival of a transaction, the node stores the transaction in its mining pool, called mempool in Bitcoin, waiting for confirmation.
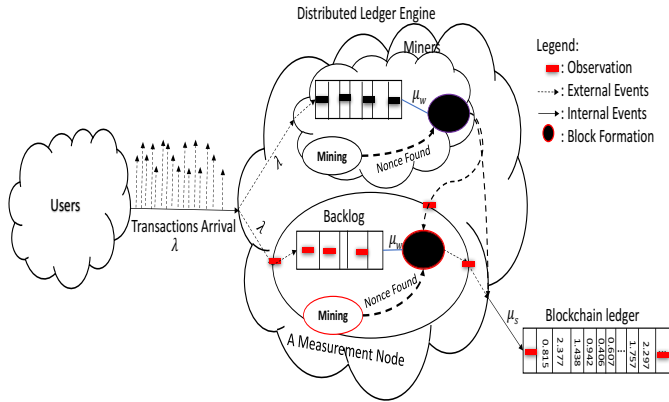

Fig. 1. An illustration of the work flow of Bitcoin

In addition, a full node may choose unconfirmed transactions in the backlog to pack into a new transaction block, and perform mining to find the mathematical puzzle given by the Bitcoin to gain the right to add the block to the ledger. If the puzzle finding is successful, this newly generated block is added to the blockchain, and this information is sent to all the nodes.

At each node, the validity of the newly generated block is checked. If the validity is confirmed with consensus, the updated blockchain is accepted and the transactions in the new block are validated. Such validated transactions are removed from the mempool at each full node that then repeats the above process. Note that, while the above description is brief, the essence of the workflow is kept. For more details about how Bitcoin works, the original introduction [27] is the best source.

B. Dataset

To analyze transaction handling in Bitcoin, we implemented a server installation of a full Bitcoin node to collect related information. The information has two parts. One part records information from the ledger that is globally available, called the *global information part*. Another part records locally available information about the backlog status of the mempool. This part is called the *local information part*.

More specifically, the global information part includes, for each block i on the blockchain, the number of transactions (n_i) in the block, its miner (m_i), the size of the block in bytes (s_i), the timestamp or generation time of the block (T_i), and the average per-transaction fee of the block (f_i). The local information part records the mempool' status (ms_i) in terms of size and fee of backlogged transactions in mempool when each block i is received at our full node.

In total, the dataset consists of information related to 80,408 Bitcoin blocks with more than two hundred million (203432240) transactions for a period of 543 days from 7th March 2019 to 31st August 2020.

III. THE ANALYSIS APPROACH

The dataset is essentially a composition of time series. We hence employ time series analysis on the dataset to provide insights and/or gain findings about transaction handling in Bitcoin. In the rest, the following time series are specifically used: $y = [y_1, y_2, \ldots, y_M]$, $x = [x_1, x_2, \ldots, x_M]$, $c = [c_1, c_2, \ldots, c_M]$, and $D = \{\{y_1, x_1, c_1\}, \{y_2, x_2, c_2\}, \ldots, \{y_M, x_M, c_M\}\}$, defined with:

$$y_i = \{s_i, n_i\}$$
$$x_i = \{Td_i, f_i, ms_i\} \tag{1}$$
$$c_i = \{m_i\}$$

where $Td_i \equiv T_i - T_{i-1}$ denotes the inter-block time, s_i, n_i, f_i, ms_i and m_i are defined in the previous section, and $M = 80408$ representing the total number of blocks in the dataset.

Our analysis consists of two parts. In the first part, i.e., Section IV, the focus is on revealing fundamental characteristics and/or basic statistical properties of transaction handling related time series, using exploratory data analysis techniques such as histogram, scatter plot and curve fitting.

In the second part of the analysis, i.e. Section V, the focus is on investigating if / how Bitcoin transaction handling may be predicted. To this aim, both classical and modern time series

forecasting approaches are considered for prediction of various transaction related attributes. In addition, a decision tree based classification approach is adopted for miner inference. The following subsections give an introduction of these approaches.

A. Autoregressive models for forecasting

For time series forecasting, a large number of approaches are available, including both classical ones and modern artificial intelligence (AI) based approaches [10].

For the former, we tested various autoregressive (AR) models. Due to their generally better performance, this paper focuses on ARIMA (AutoRegressive Integrated Moving Average) and ARIMAX (Autoregressive Integrated Moving Average with Exogenous input). Equations (2) and (3) define these models respectively, where B is the backshift operator and ∇ the difference operator.

$$y_i^+ = \phi_1 y_{t-1} + \cdots + \phi_p y_{i-p} + \theta_1 \varepsilon_{i-1} + \cdots +$$
$$\theta_q \varepsilon_{i-q} + \varepsilon_i, \quad (2)$$
$$\Phi(B)\nabla^d y_i^+ = \Theta(B)\varepsilon_i,$$

$$(y_i^+|T_i = t) = \phi_1\{x_{i-1}, y_{i-1}\} + \cdots + \phi_p\{x_{i-p}, y_{i-p}\}$$
$$+ \theta_1 \varepsilon_{i-1, t_{i-1}} + \cdots + \theta_q \varepsilon_{i-q, t_{i-q}} + \varepsilon_{i, t_i}, \quad (3)$$
$$\Phi(B)\nabla^d(y_i^+|T_i = t) = \beta x_i + \Theta(B)\varepsilon_{i, t_i},$$

where $(y_i^+|T_i = t)$ or y_i^+ is the predicted block, $E(\varepsilon_{i, t_i}) = 0$, $\mathrm{Var}(\varepsilon_{i, t_i}) = \sigma^2$, $\nabla^d = (1\text{-}B)^d$ is difference factor, $\nabla^d(y_i^+|T_i = t)$ is the sequence of y_i by d times differed, $\Phi(B) = 1 - \phi_1 B, \ldots, \phi_p B^p$ is an auto regressive coefficient polynomial, and $\Theta(B) = 1 - \theta_1 B, \ldots, \theta_q B^q$ is a moving smoothing coefficient polynomial of the smooth invertible autoregressive moving average model ARMA (p, q).

To assess the forecasting performance, we use mean average error (MAE) and root mean square error (RMSE), which are respectively defined as: with $e_i = y_i - y_i^+$,

$$MAE = \frac{\sum_{i=1}^{N} |e_i|}{N}$$
$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} e_i^2}{N}} \quad (4)$$

where N denotes the number of predicted data points.

B. AI-based forecasting models

For AI-based models, NAR (nonlinear autoregressive neural network) and NARX (nonlinear Autoregressive Network with Exogenous Inputs) are chosen because they have a feedback connection that encloses several layers of the network, which uses memory to remember the time series's past values to get better performance [18] [36]. Additionally, the models have nonlinear filtering that helps to capture the dynamic fluctuations of the input values.

Equations (5) and (6) describe NAR and NARX network's function to predict a particular value of data series y_i^+ using p previous values of y and x.

$$(y_i^+) = f_{\mathrm{NAR}}(y_{i-1}, y_{i-2}, \ldots, y_{i-p}) \quad (5)$$

$$(y_i^+|T_i = t) = f_{\mathrm{NARX}}(\{x_{i-1}, y_{i-1}\}, \{x_{i-2}, y_{i-2}\},$$
$$\ldots, \{x_{i-p}, y_{i-p}\}) \quad (6)$$

The functions f_{NARX} and f_{NAR} in (5) and (6) are unknown, and the neural network training approximates the function by optimizing the network weights and neuron bias. The NAR and NARX model uses Levenberg-Marquardt, Bayesian regularization, and scaled conjugate gradient training algorithms to train the model [2]. Specifically, Bayesian regularization (BR) is used to conduct the analysis. BR minimizes a combination of squared errors and weights; then determines the correct combination to produce a network that generalizes well. It uses network training function Levenberg-Marquardt to optimize network weights and neuron bias. The Levenberg–Marquardt is a popular numerical solution to find the smallest nonlinear function over parameter space.

The following explains the input and output of the neural network model we use.

- Input: Block values in the form of vector length, which indicate the number of previous values of the block time series. The models without external input take a vector of the input $y_i = \{n_i, s_i\}$ while predicting the next blocks content either n_i or s_i. Similarly, the models with external input additionally take $\{x_i\}$ as an input when the model is used to predict the subsequent blocks.
- Hidden layer: For NAR and NARX, the number of hidden neurons is determined by performing a pre-analysis using the collected dataset. Based on this analysis, the models satisfy the Mean Square Error (MSE) value when the neurons are equal to ten.
- The input delay p and q are approximated by using an autocorrelation (p) and partial-autocorrelation (q) plot.
- Output: The predicted blocks $(y_i^+|T_i = t)$ or y_i^+) contain the predicted values of the blocks $\{n_i, s_i\}$ of the weekend, working, and the combinations.

C. Decision tree based classification

Starting in 2010, there are more than 23 mining pools worldwide, as reported in Fig. 3. It has been illustrated that mining pools compete to find the mathematical puzzle and the mining behavior is a game [16][35].

In this paper, we investigate if the mining pools are detectable using a machine learning, decision tree based approach [1][15][41]. It has a tree structure: Each branch represents the outcome of the test, and each leaf node represents a class label. In some cases, it is essential to combine several decision trees to produce a better classification performance. Such a combination produces an ensemble of different methods. In the present work, we considered two methods: booted and RSUbooted [26].

The accuracy, area under curve (AUC), sensitivity, and miss rate are used to test the classification performance, in addition to false negative rate (FN), true positive rate (TP), and receiver operating characteristic (ROC) curve of TP versus TN, as commonly used for machine learning based classification [37].
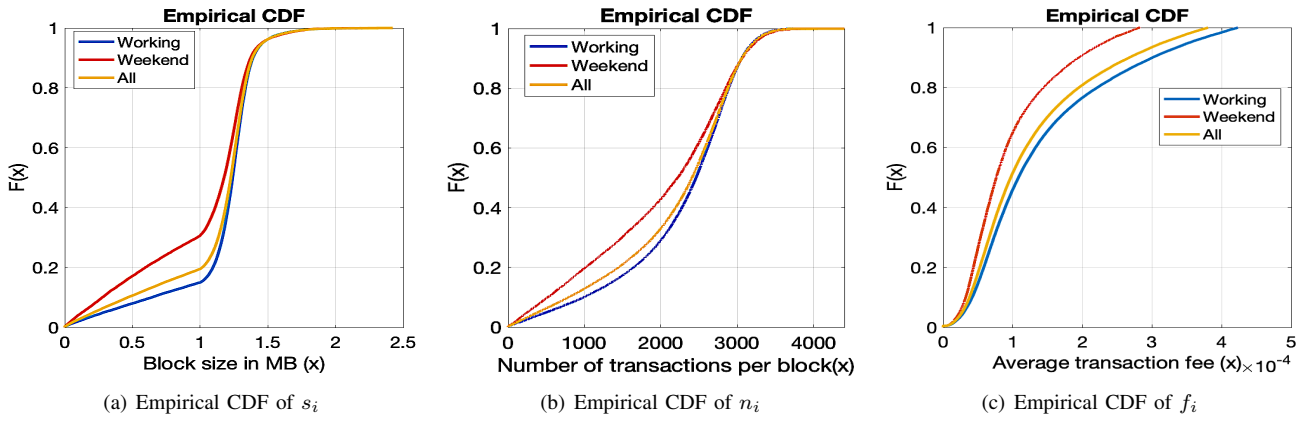
Fig. 2. CDFs of basic block attributes

IV. RESULTS: EXPLORATORY ANALYSIS

This section reports results and observations from an exploratory analysis of the collected data.

A. Basic block attributes

The size s_i, the number of transactions n_i and the fee f_i are fundamental attributes of a block. Since the transaction activities are time-varying process [12], they may have distributions that vary from time to time, e.g., the weekend and working day demands have a different distribution.

Fig. 2(a) reports in most cases (80%), s_i in working days has less than 1.4 MB, whereas it has a 1.2 MB size during weekends. In both cases, the s_i can grow more than 1.5 MB in 1% of the cases. Relatively, 30% of the blocks have a size less than the default legacy size s_i of 1 MB on weekend days; nevertheless, in working days, less than 20% of the blocks have a size less than 1 MB.

Similarly, Fig. 2(b) illustrates that the n_i also varies as s_i: 50% of the blocks have less than 2200 transactions per block in weekend days, while 2500 transactions per block in working days. In working days, only 20% of the blocks have n_i less than 2100 transactions wherein the weekend, 40% of the generated blocks have n_i less than 2200.

In addition, the miner's economic incentives affect which transactions to include in a block and this financial interest may also show some differences over time. Fig. 2(c) reports, 50% of the blocks, during weekdays, the f_i is smaller than $1.3*10^{-4}$ while during working days, the f_i is less than $1.43*10^{-4}$. In both cases, 80% of the f_i is smaller than 0.00033 BTC, and with less than one percent, the f_i can grow more significantly than 0.0004 BTC.

B. Miners

Fig. 3 reports miners' contribution in terms of the number of valid blocks in the main chain. As we can observe from the figure, unknown(?), F2Pool, BTC.com, Poolin, and AntPool contribute a higher number of blocks. Combined, these five major mining pools generate around 50% of the valid blocks.

Driven by the financial interest, a mining pool might use a strategy to increase the financial gain [9]. To explore, we
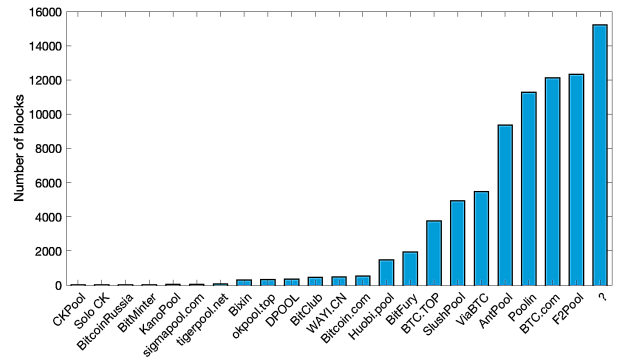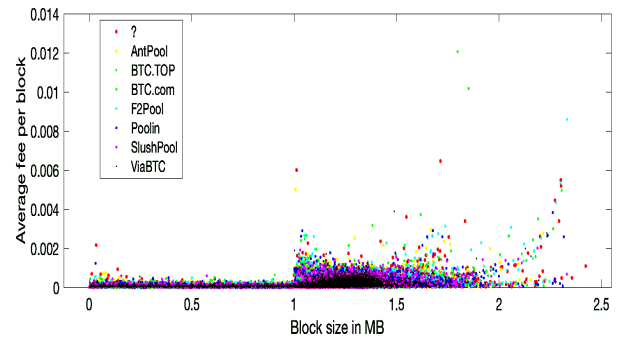

Fig. 3. Miners

Fig. 4. f_i vs s_i

analyze the blocks generated by the major mining pools. Fig. 4 and 5 report that when the size s_i is greater than 1.5 MB, it is visible from the figures that some of the major mining pools become more recognizable. However, when s_i is less than 1 MB, it is challenging to see any difference between the pools. Similarly, When the s_i is between 1- 1.5 MB, we can see a high concentration of the mining pools. The figures also report that as the s_i increases, the n_i and f_i also rise together.

To further investigate the number of block contributions in the working and weekend days, we focus on the five major miners. Fig. 6 illustrates that these miners contribute similar number of blocks in the working days, except for

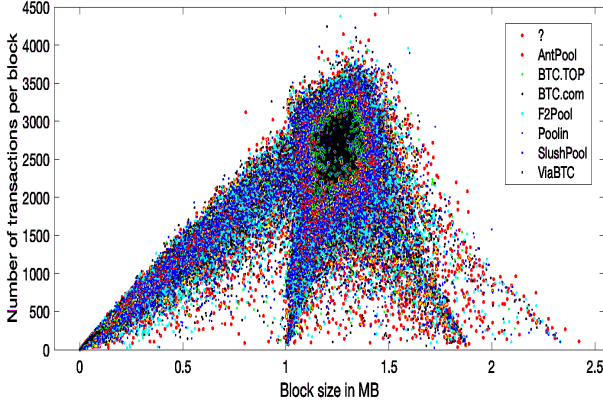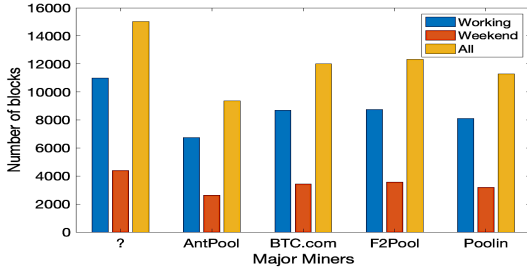| Mining pool	$\mu(s_i, n_i, f_i)$	$\sigma(s_i, n_i, f_i)$	$\min(s_i, n_i, f_i)$	$\max(s_i, n_i, f_i)$
?	$(1.1252, 2.14 * 10^3, 1.83 * 10^{-4})$	$(0.3657, 844.2627, 2.18 * 10^{-4})$	$(2 * 10^{-4}, 1, 0.00)$	$(2.4229, 4402, 0.0065)$
AntPool	$(1.1141, 2.18 * 10^3, 1.8 * 10^{-4})$	$(0.3622, 844.2076, 1.9 * 10^{-4})$	$(3.34 * 10^{-4}, 1, 0.00)$	$(2.2151, 4063, 0.0050)$
BTC.com	$(1.0960, 2.15 * 10^3, 1.86 * 10^{-4})$	$(0.3782, 868.4394, 2.487 * 10^{-4})$	$(2.38 * 10^{-4}, 1, 0.00)$	$(2.3056, 4243, 0.0121)$
F2Pool	$(1.1099, 2.14 * 10^3, 1.76 * 10^{-4})$	$(0.3680, 845.6503, 2.16 * 10^{-4})$	$(2.66 * 10^{-4}, 1, 0.00)$	$(2.3316, 4377, 0.0086)$
Poolin	$(1.1091, 2.17 * 10^3, 1.67 * 10^{-4})$	$(0.3635, 842.1800, 1.87 * 10^{-4})$	$(2.17 * 10^{-4}, 1, 0.00)$	$(2.3165, 3988, 0.0038)$

Fig. 5. n_i vs s_i

Fig. 6. Miners contribution

(a) Inter-block generation (b) Autocorrelation plot

Fig. 7. Fitting of inter-block generation time to n.e.d

C. Block generation

1) Distribution of inter-block generation time: Based on the Bitcoin design [27], it has been expected that the inter-block generation time follows an exponential distribution, and the validity has also been checked [12]. Along the same line, Fig. 7(a) reports the fitting of the inter-block generation time to an exponential distribution. Additionally, to check the independence of block generation time, its autocorrelation plot is illustrated in Fig. 7(b). As can be seen from Fig. 7(a) and Fig. 7(b), **the inter-block generation time fits well with an exponential distribution** with increasing mismatch at the tail, partly due to the limited number of blocks in the dataset, and the autocorrelation is close to zero under all the lags in the figure, with the most significant difference only around 1%, indicating that block generation is little correlated.

(a) 100 minutes; $\lambda = 9.44707$ (b) 1000 minutes; $\lambda = 103.184$

Fig. 8. Block generation histogram fitting to a Poisson distribution with intensity λ under different time slot length

the unknown(?) pool. The same observation is also found for the weekend days. The unknown(?) pool generates a higher number of blocks in all cases.

To gain a deeper insight into the block contents than the number of blocks, Table I is presented, where the mean μ, standard deviation σ, minimum and maximum values of the basic block attributes (s_i, n_i, f_i) are shown. Note that these major mining pools become operational starting 2016, except for Pooling in 2018 [39]. Even though there is a gap in years between Poolin and the rest, Table I shows that Poolin, F2Pool, BTC.com generate blocks with similar average size, standard deviation and max values. However, the unknown (?) and AntPool generates block with size greater than the three. The unknown (?) has a block size mean close to 1.214 MB, and the maximum block size is also found in this mining pool. Additionally, the public mining pool, Poolin, comparing the maximum values of f_i and n_i, has the smallest than the other four mining pools.
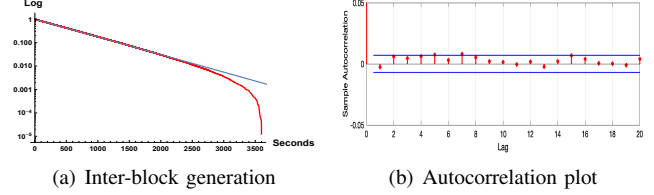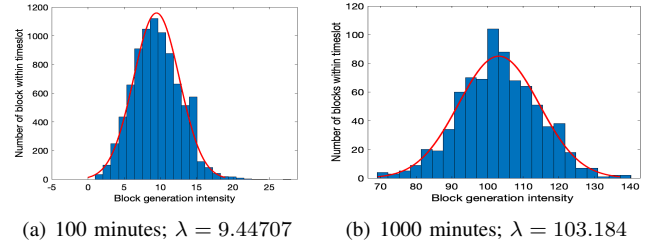
2) Fitting to a Poisson process: Since the block generation process has exponentially distributed inter-generation times, we investigate if it can also be further treated as a Poisson process. For this, we make histograms of the number of blocks generated in different length of time and fit them with Poisson distributions. If the process is Poisson, these Poisson distributions must have the same intensity after being scaled. For this investigation, Fig. 8(a) and Fig. 8(b) are presented, where the best fitting intensity of Poisson distribution is shown under two time lengths, 100 and 1000 minutes. Clearly, the obtained two intensities differ noticeably, after taking into consideration
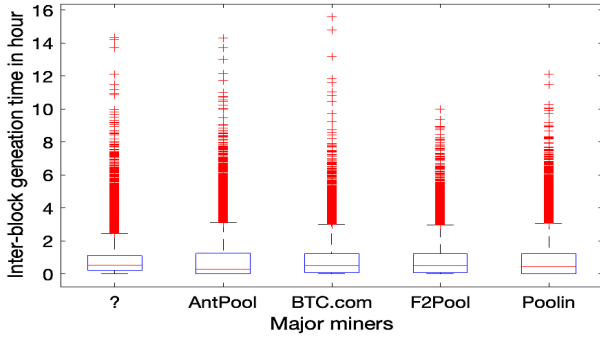
Fig. 9. Major mining pools' inter-block generation time

(a) Inter-block generations vs s_i

(b) Inter-block generations vs n_i

(c) Inter-block generations vs f_i

Fig. 10. Inter-block generations v.s. block size, transaction number and fee

(a) Transaction inter-arrival time

(b) Autocorrelation plot

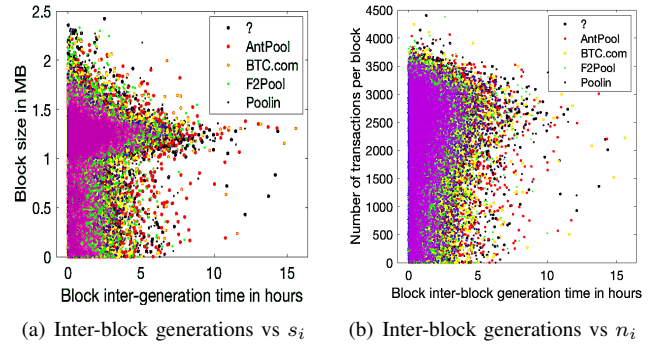Fig. 11. Transaction inter-arrival time fitting n.e.d

that there is 10x scaling difference. This observation, which is surprising, implies that **block generations can at most be approximated but cannot be treated as a Poisson process.**

3) Relation with miners: To have a closer look on block generation, we made further investigation over the five major mining pools. Fig. 9 reports the inter-block generation time of the major mining pools. As the figure shows, **while the average inter-block generation time is almost the same among the major mining pools, there is visible difference for the median**: While for Unknown(?) and F2Pool, the median time is close to 52 minutes, for BTC.com and Poolin, it is near 45 minutes and for AntPool, it is close to half-hour. The minimum inter-block generation time is the same for all major mining pools, close to zero. However, for the maximum inter-block generation time, while AntPool and Unknown(?) need 14 hours and 30 minutes, BTC.com demands 16 hours. In addition, the public mining pool, Poolin, requires 12 hours, and unlike or shorter than the others, F2Pool needs only 10 hours. As a highlight from Fig. 9, F2Pool stands clearly out of the others with shortest tail.
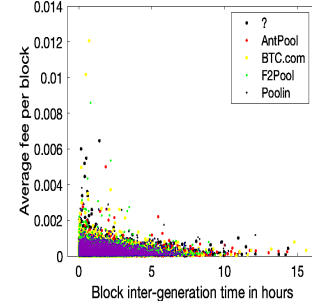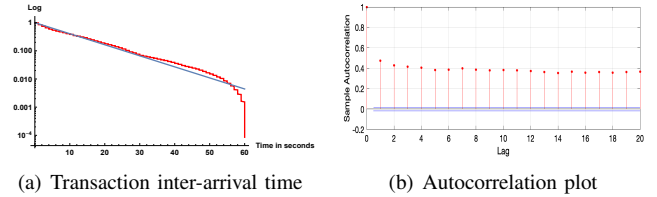
4) Relation with basic block attributes: We further explored the relationship between block generation and the three basic block attributes, shown by Fig. 10(a), 10(b), and 10(c). Specifically, Fig. 10(a) illustrates that when the block size s_i is greater than 1.5 MB, the inter-block generation time seen by the blocks is less than two hours. However, when the block size is concentrated between 1-1.5 MB, Unknown(?), AntPool, and BTC.com block can have the inter-block generation time greater than 13 hours. On the other hand, the blocks from Poolin and F2Pool seem to be generated with shorter interval than the rest three, which is also indicated by Fig. 9.

In addition, Fig. 10(b) demonstrates that the number of transactions n_i in a block of Poolin is on average smaller than the other mining pools. Most of the n_i from F2Pool seem to have a shorter inter-block generation time. However, it is hard to say for the Unknown(?) and AntPool, because the plot shows most of the block with n_i seems to have a larger inter-block generations time. These effects may arise from the state of the mempool, when the mempool contains more transactions then the miners can pick as much number of transaction to include in block.

Furthermore, it is natural the miners prioritize the finical in-

centives, which encourages the miners to pick up transactions with a higher fee. Fig. 10(c) illustrates this fact. Specifically, **when the fee f_i is higher, the inter-block generation time of the block is lower, maybe even shorter than an hour**. The figure also shows that the blocks with the smaller average fee from Unknown(?), AntPool, and BTC.com may experience inter-block generation time greater than 14 hours. On the other hand, the blocks from Poolin seem to have a less average fee and seeingly smaller inter-block generation time.

D. Transaction arrival and confirmation time

Users generate transactions for validation. New arrivals stay at the backlog (memory pool) until the nonce finding is successful and they are picked up by the miner.

1) Transaction inter-arrival time: Fig. 11(a) shows that the fitting of transaction inter-arrival times to a negative exponential distribution is only reasonable well with visible deviation. Additionally, Fig 11(b) reports the inter-arrival between the transactions is correlated. These reflect that *there exists some level of dependence between transaction arrivals.*

2) Transaction confirmation time: Fig. 12(a) reports the transaction confirmation time fitting to a negative exponential
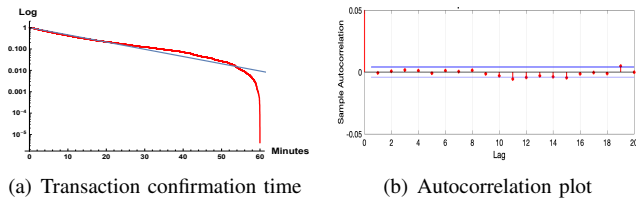
(a) Transaction confirmation time   (b) Autocorrelation plot
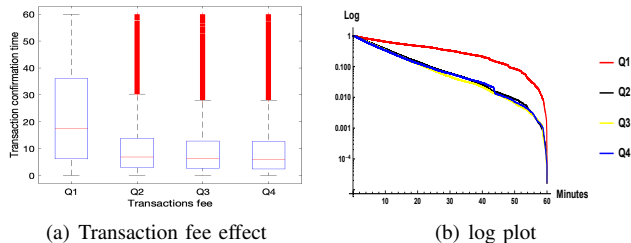
Fig. 12. Transaction confirmation time fitting n.e.d

(a) Transaction fee effect   (b) log plot

Fig. 13. Transaction fee effect on transaction confirmation time

TABLE II
DIVISION OF THE DATASET

Dataset	Training	Test	Validation	#No of blocks
Working_day	40095	8591	8591	57277
Weekend_day	16190	3469	3469	23128
All_db	56286	12061	12061	80408

distribution, with a sharp drop at the tail. Additionally, Fig. 12(b) illustrates that the transaction confirmation time is uncorrelated, reflecting that *the transaction confirmation time is independent.*

Since *a miner tends to choose transactions with a higher fee*, to demonstrate this effect on the confirmation time, Fig .13(a) is presented. Specifically, it demonstrates the relationship of confirmation time and fee for Q1 (25%), Q2 (50%), Q3 (75%), and greater than Q3, i.e., (Q4) for f_b. Their intervals are respectively (0,Q1), (Q1,Q2), (Q2,Q3), and (Q4,∞). As Fig 13(a) shows, low fee transactions exhibit a higher confirmation time. On average, the low fee transactions (Q1) wait 22 minutes for validation. However, for higher fee (Q4) transactions, the average confirmation time is less than half of that of the low fee transactions. For Q2 and Q3, the transactions exhibit close to a ten-minute average confirmation time. Still, transactions from Q2, on average, wait one more minute extra than Q3. Overall, transactions wait on average 13 minutes, and we also observed a few transactions waiting for more than 24 hours at the backlog. At the same, these few transactions also tend to have a fee associated relatively very small.

V. RESULTS: PREDICTABILITY ANALYSIS

Having explored the various characteristics of transaction handling in the previous section, this section is devoted to investigating if and what such characteristics can be predicted. For this predictability analysis, the prediction approaches introduced in Section III are used. The results are reported and discussed in the rest of this section, where the dataset is divided into three parts, i.e, training, test and validation, and the details of this division is reported in Table II.

A. Basic block attributes

Table III compares the performance of the various models in predicting the target block attributes: size s_i and number n_i, where as a benchmark, the basic autoregressive (AR) model is also included. For these models, the symbol p is order

of the autoregressive part, d is the number of nonseasonal differences needed for stationarity, and q is order of the moving average part. In this investigation, the values for $p = 2$ and $q = 2$ are calculated from autocorrelation and partial-autocorrelation plot, and we set $d = 0$. MAE and RMSE are used to compare models' performance. In addition, to give a more direct impression, we illustrate the prediction results by the models for randomly chosen ten consecutive weekend blocks, as an example, in Fig. 14(a) and Fig. 14(b).

Table III, Fig. 14(a) and Fig. 14(b) indicate that, the prediction results by the considered forecasting approaches all follow the actual trend well. However, the models that additionally make use of the locally available information x, which are ARIMAX and NARX, generally produce better results than their counterpart models ARMA and NARX that do not have exogenous input. In addition, the AI-based models perform better than the classical autoregressive models under the same condition. Overall, NARX' performance is best, which is *an encouraging finding for applying AI-based approaches in predicting the basic block attributes' values.*

Remark: The alert reader may have noticed that among the three basic block attributes investigated in the exploratory study, we have left the fee f_i out in the predictability study. This is simply because a large related literature exists, which will be discussed in the related work section, and the results therein show that the price can be excellently predicted.

B. Block generation and transaction confirmation time

Encouraged by the prediction results for the basic block attributes, we used the NARX model to test if block generation and transaction confirmation time can also be predicted. For predicting block generation, we used T_i as the input while x= $\{f_i, n_i, s_i, ms_i\}$ as the external input. Fig. 15(a) reports the model's performance. For predicting transaction confirmation, we used transaction confirmation times as the input, while the size of the transactions and the fee associated are used as an external input. Fig. 15(b) exemplifies the model's performance at a number of random points.

As indicated by Fig. 15(a) and Fig. 15(b), *the predication of block generation and transaction confirmation time does not work.* While this observation seems to be contradictory to the observation in predicting s_i and n_i, a closer look at the characteristics of block generation time and transaction confirmation time enables to explain. Reported in the exploratory analysis in Section IV, both the inter-block generation time and the transaction confirmation time has or can be closely approximated by an exponential distribution. Then, because of the memoryless property of exponential distribution, the likelihood of something happening in the future has little

Models	MAE			RMSE		
	Weekend(s_i,n_i)	Working(s_i,n_i)	All(s_i,n_i)	Weekend(s_i,n_i)	Working(s_i,n_i)	All(s_i,n_i)
AR(p)	0.53, 264	0.6, 117.35	0.5, 127.12	0.5, 122.14	0.5, 141.91	0.3, 264
ARIMA(p,d,q)	0.15, 15.373	0.077, 12.840	0.13, 12.969	0.04, 12.461	0.01, 10.833	0.025, 10.942
ARIMAX(p,d,q)	0.12, 13.364	0.07, 12.092	0.06, 11.735	0.02, 11.052	0.006, 10.408	0.006, 10.408
NAR(p)	0.01, 14.770	0.06, 12.969	0.06, 12.840	0.03, 12.214	0.008, 11.275	0.008, 10.942
NARX(p)	0.011, 10.942	0.06, 10.471	0.013, 10.460	0.01, 10.121	0.006, 10.035	0.0003, 10.030
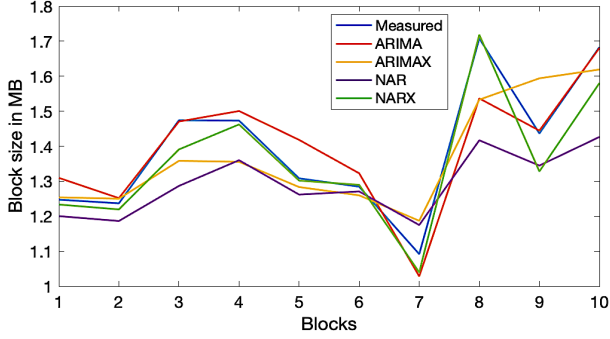
(a) measured vs predicted s_i
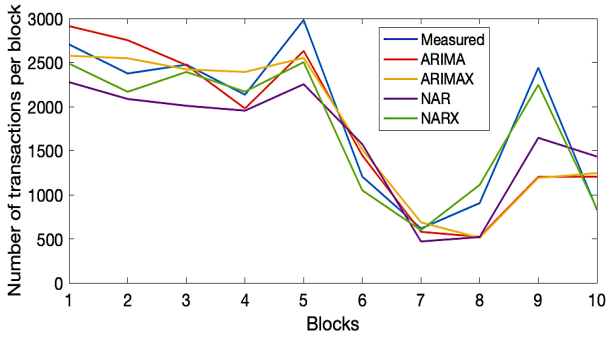
(b) measured vs predicted n_i

Fig. 14. Sample prediction results

relation to whether it has happened in the past. Implied by this and as also confirmed by Fig. 15(a) and Fig. 15(b), **any effort of predicting these two transaction handling aspects may, "surpringly", lead to no solid conclusion**.
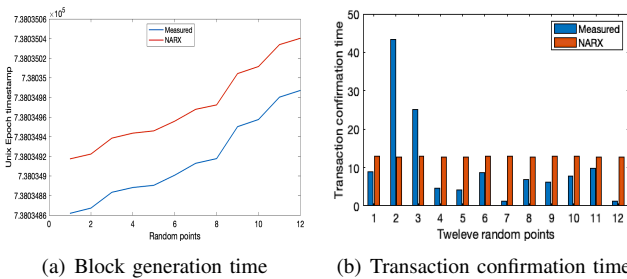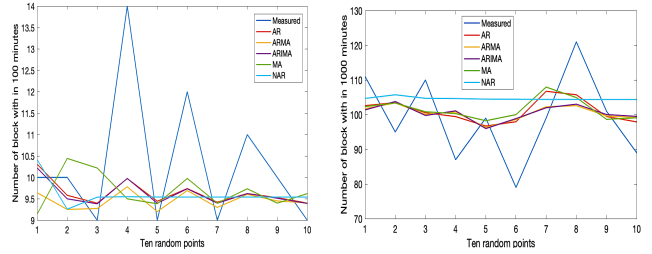

(a) Block generation time

(b) Transaction confirmation time

Fig. 15. Block generation and transaction confirmation time sample prediction

We conduct further investigations on predicting block generation intensity. In this case, for the AI-based models, we only used NAR because we do not have additional input for NARX.

To be in line with the counterpart exploratory investigation, we fixed the slot size of 100 and 1000 minutes and predicted the number of blocks within the slot, respectively. Fig. 16(a) and 16(a) report the performance of both the classical autoregressive models and the AI-based NAR model. In general, the AR models follow the trend better than the NAR model. Nevertheless, all models struggle to perform better than the average. This, we believe, attributes largely from that while not exactly, the number of blocks in a time period can is approximately Poisson-distributed, as reported in Section IV.

(a) Block generation intensity with fixed time slot of 100 minutes

(b) Block generation intensity with fixed time slot of 1000 minutes

Fig. 16. Block generation intensity sample prediction

C. Miner classification

As we saw in the previous sections, the f_i, s_i, Td_i, n_i, and ms_i have a significant effect on the evolution of the Bitcoin ledger. Due to this, we use these feature sets to test if they can help infer a miner's relationship, and if some mining pools use some specified strategies while generating a block. To study these, we take two cases, first working and weekend days, and in the second case, considering all the data together. The feature set, including f_i, s_i, n_i, and Td_i, is used to perform classifications of mining pools (c_i). As a remark, we have also tried other features in the mempool state ms_i but observed that they do not bring significant increase over the accuracy.

1) Case-I (Working and Weekend day): The top-eight mining pools are used to detect the block generation behavior. Fig. 17(a) and Fig. 17(b) report that the major mining pools have a true positive rate (TP) more significant than the rest of the pools. As Fig. 17(a) and Fig. 17(b) report, the better model, the RSUBoosted decision tree with the booted method, shows a promising result classifying the F2Pool in better approximation relative to the other pools. As we can see from Fig. 17(a) and Fig. 17(b), the TP for BTC.com, AntPool, and Poolin is smaller than 25%, but for the SlushPool and

True class	?	AntPool	BTC.top	BTC.com	F2Pool	Poolin	SlushPool	ViaBTC	TPR	FNR
?	19%	8%	16%	5%	27%	3%	20%	3%	19%	81%
AntPool	9%	9%	28%	3%	25%	2%	20%	3%	9%	91%
BTC.top	10%	12%	27%	2%	27%	2%	19%	3%	27%	73%
BTC.com	8%	6%	15%	19%	28%	4%	16%	3%	19%	81%
F2Pool	1%	<1%	1%	2%	90%	3%	1%	2%	90%	10%
Poolin	5%	6%	14%	9%	28%	20%	13%	4%	20%	80%
SlushPool	8%	8%	19%	3%	26%	2%	31%	3%	31%	69%
ViaBTC	10%	7%	22%	7%	28%	7%	14%	5%	5%	95%

(a) Weekend days

True class	?	AntPool	BTC.top	BTC.com	F2Pool	Poolin	SlushPool	ViaBTC	TPR	FNR
?	23%	3%	25%	2%	18%	2%	24%	3%	23%	77%
AntPool	12%	7%	38%	<1%	17%	1%	22%	3%	7%	93%
BTC.top	14%	6%	41%	<1%	17%	1%	20%	2%	41%	59%
BTC.com	8%	2%	23%	22%	18%	4%	20%	3%	22%	78%
F2Pool	<1%	<1%	<1%	<1%	>99%	<1%			>99%	<1%
Poolin	6%	4%	19%	8%	19%	24%	16%	3%	24%	76%
SlushPool	10%	1%	29%	2%	16%	<1%	38%	4%	38%	62%
ViaBTC	13%	3%	34%	4%	19%	7%	14%	6%	6%	94%

(b) Working days

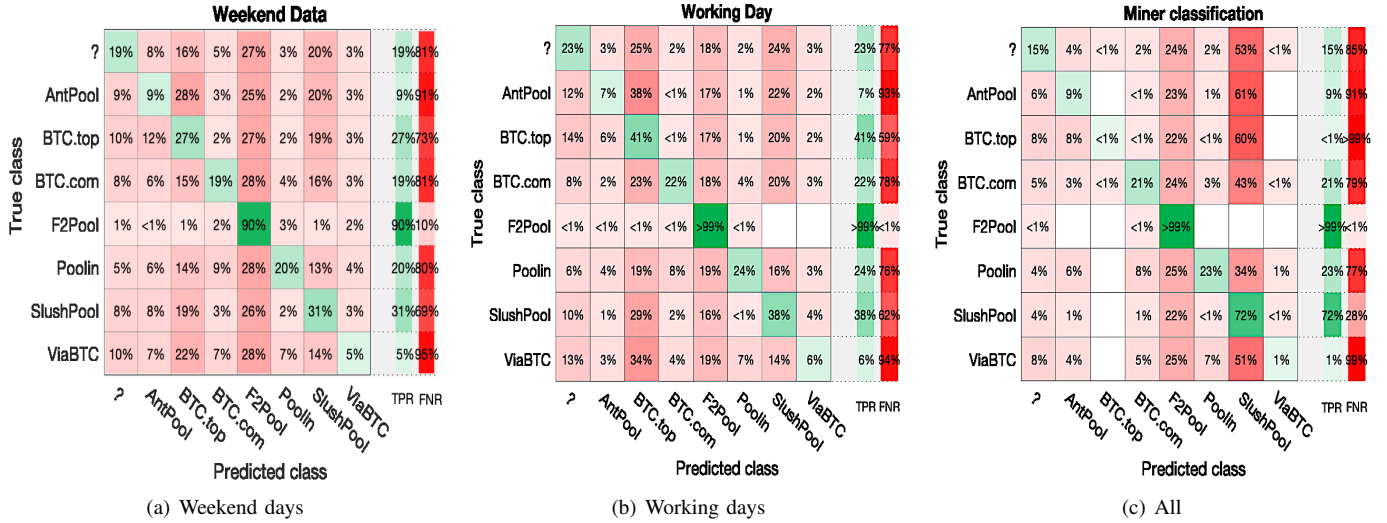| True class	?	AntPool	BTC.top	BTC.com	F2Pool	Poolin	SlushPool	ViaBTC	TPR	FNR
?	15%	4%	<1%	2%	24%	2%	53%	<1%	15%	85%
AntPool	6%	9%		<1%	23%	1%	61%		9%	91%
BTC.top	8%	8%	<1%	<1%	22%	<1%	60%		<1%	99%
BTC.com	5%	3%	<1%	21%	24%	3%	43%	<1%	21%	79%
F2Pool	<1%			<1%	>99%				>99%	<1%
Poolin	4%	6%		8%	25%	23%	34%	1%	23%	77%
SlushPool	4%	1%		1%	22%	<1%	72%	<1%	72%	28%
ViaBTC	8%	4%		5%	25%	7%	51%	1%	1%	99%

(c) All

Fig. 17. Confusion Matrix of major miners (RSUBoosted decision tree)

BTC.TOP, it is more significant than 25%. Especially in the case of the public mining pool, Poolin, the false-negative rate is five times higher than the TP. This indicates *the Poolin has less detectable block generation strategy than the rest. However, for SlushPool, it is has a block generation behaviour more distinguishable than the top five major mining pools.*

2) Case-II (All Data): The previous case showed that F2Pool was approximated very reasonably from the major mining pools. Fig. 17(a) and Fig. 17(b) report a confusion matrix illustrating the F2Pool and SlushPool having a higher positive rate than the rest of the mining pools. Additionally, Fig.17(c) reports that the two major mining pools, SlushPool and F2Pool, the TP are more significant than 70%, which is 40% more accurate than the first case for SlushPool. Similarly, the false-negative rate is less than 20%, especially in F2Pool, which is even less than 3%. To have a better understanding, we performed further investigation on only these two mining pools, F2Pool and SlushPool. The results are reported in Table IV, Fig. 19(a) and 19(b), and Fig. 18. Table IV compares the performance of the two DT methods. Due to better accuracy of the RSUBoosted-tree, it is used in Fig. 19(a) and 19(b), and Fig. 18. Specifically, the true-positive rate (TPR) and the false-negative rate (FNR) are shown in Fig. 18, and Fig. 19(a) and 19(b) further illustrate the model accuracy in terms of AUC and ROC.

TABLE IV
PERFORMANCE OF CLASSIFICATION BETWEEN F2POOL AND SLUSHPOOL

Models	Accuracy	Sensitivity	Miss rate
RSUBoosted-tree	0.90	0.885	0.115
Boosted-tree	0.883	0.881	0.119

3) Discussion: Fig. 17(a), Fig. 17(b), and Fig. 17(c) essentially show that **other than for a few mining pools, particularly F2Pool, mining pools have a minimal positive classification rate, implying they are hard to distinguish**. This is in line with Fig. 9 in the exploratory analysis part,
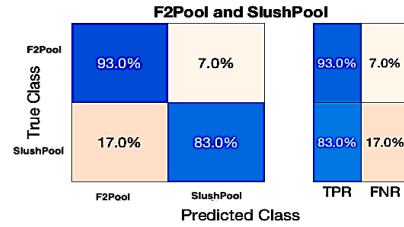

Fig. 18. F2Pool and SlushPool

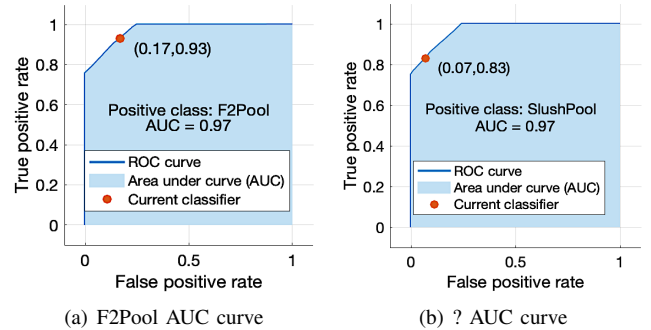(a) F2Pool AUC curve   (b) ? AUC curve

Fig. 19. AUC curves for F2Pool and SlushPool

which shows that while the block generation distributions of other miners are similar, for F2Pool it is visually distinguishable from the others. We believe this characteristic difference has been explored by the decision tree approach in the classification. In addition, a closer investigation as illustrated by Fig. 17(c) and Fig. 18 implies that the two major private mining pools P2Pool and SlushPool use different strategies that have caused their block generations with special properties making the classification with higher accuracy.

VI. RELATED WORK

1) Statistical analysis of transaction handling characteristics: While a lot of such analysis results are available, e.g.,

various Bitcoin statistics [7], block propagation delay [6], block arrival process [3], transaction rate and confirmation time [13] [33], we focus on fundamental aspects underlying transaction handling and particularly their distributions, different from the literature. Through analyzing these distributions, we have been able to reason some seemly surprising observations in the predictability study. In addition, very few results in the literature take into account information that is only locally available. In this sense, the work [12] is most related. However, except for inter-block generation time fitting, which is similar as we already highlighted, the other results are not found in [12], due to different focuses of [12] and the present work.

2) Forecasting transaction handling characteristics: The focus of the literature has been on bitcoin price. For instance, Huisu Jang and Jaewook Lee [19] developed a neural network-based forecast model on the volatility of a Bitcoin price and extended analysis to identify the best feature set that gives more information about the Bitcoin price process. Similarly, Edwin Sin and Lipo Wang [34] implemented an artificial neural network to predict the next Bitcoin price and the amount of profit that could be gained by making such predictions. Shah et al. [32] considered the Bayesian regression method to predict the price of Bitcoin. Pavel Ciaian et al. [4] estimates Bitcoin price formation based on a linear model by introducing several factors such as market forces, attractiveness for investors, and global macro-financial factors. Greaves et al. [14] analyzed the Bitcoin blockchain data to predict the price of Bitcoin using SVM and ANN, which score 55% accuracy. Similarly, models such as Random Forest, SVM, and Binomial Logistic algorithms are used to predict short-term Bitcoin price and achieve a high accuracy result of 97% in [25]. To the best of our knowledge, no previous work combines the feature sets to predict the transaction handling characteristics focused in this paper.

3) Mining pool classification: There have been some research works that studied block withholding and unfair distribution of reward. For instance, Schrijvers et al. [31] analyzed the incentive compatibility of the Bitcoin reward mechanism. In their model, a miner can decide between honest mining and delaying her found blocks' submission. They proved that the proportional mining reward mechanism is not incentive compatible. Eyal [8] computed the pools' optimal strategy in the block withholding attack and their corresponding revenues. It was demonstrated that the no-pool-attack strategy is not a Nash equilibrium in these games because if none of the pools run the attack, one pool can increase its revenue by launching the attack. Luu et al. [24] experimentally demonstrate that block withholding can increase the attacker's revenue. They do not address the question of mutual attacks. Courtois and Bahack [5] have recently noted that a pool can increase its overall revenue with block withholding if honest pools perform all other mining. We consider the general case where not all mining is performed through public pools and analyze situations where pools can attack one another. M. Salimitari et al. [30] used prospect theory to predict a miner's Profit from joining one of the major mining pools. The hash rate power, total number of the pool members, reward distribution policy of the pool, electricity fee in the new miner's region, pool fee, and the current Bitcoin value are used to predict which pools are profitable specific miners.

Most mining pool studies do either emphasis on **(i)** block withholding [16] [21] or **(ii)** unfair distribution of rewards [11] [22] [23] [35], but none or little has been investigated to detect the major mining pools with hidden block generation strategies. Our work tries to further investigate these block formation strategies, by introducing decision tree to distinguish one of the major mining pools following having a detectable block formation strategy.

VII. CONCLUSION

An exploratory analysis on fundamental transaction handling characteristics of Bitcoin is conducted, together with a novel analysis on their predictability. The results from the former have been used to help reason the findings from the latter. Specifically, the focused block attributes include the size, the number of transactions and the fee. In addition, block generation and transaction confirmation, two fundamental processes resulted from transaction handling, are investigated. Furthermore, the contribution of miners to these attributes and processes is particularly taken into consideration.

The results show that while it is possible to use measurement-based collected data in predicting the basic attributes of the next block with reasonable accuracy, care is needed in predicting block generation and transaction confirmation. While the latter seems contradicting the expectation from the former, the explanation is supported and implied by results from the exploratory analysis. Additionally, it shows that combining internal and external factors enables better performance in prediction / classification. Furthermore, although it is difficult to distinguish among mining pools through prediction in general, the investigation shows that F2Pool is well distinguished from the others. A closer investigation in the exploratory analysis shows that block generation of F2Pool has a distribution with visible characteristic difference, implying that it has used a different strategy than the other miners. These results shed new light and may also be considered by users and miners when deciding their transaction strategies.

REFERENCES

[1] A. Abdelhalim and I. Traore. "A New Method for Learning Decision Trees from Rules". In: *2009 International Conference on Machine Learning and Applications*. 2009, pp. 693–698.

[2] Oludare Abiodun et al. "State-of-the-art in artificial neural network applications: A survey". In: *Heliyon* 4 (Nov. 2018), e00938.

[3] R. Bowden et al. "Modeling and analysis of block arrival times in the Bitcoin blockchain". In: *Stochastic Models* 0.0 (2020), pp. 1–36.

[4] Pavel Ciaian, Miroslava Rajcaniova, and d'Artis Kancs. "The economics of BitCoin price formation". In: *Applied Economics* 48.19 (2016), pp. 1799–1815.

[5] Nicolas T Courtois and Lear Bahack. "On subversive miner strategies and block withholding attack in bitcoin digital currency". In: *arXiv preprint arXiv:1402.1718* (2014).

[6] Christian Decker and Roger Wattenhofer. "Information Propagation in the Bitcoin Network". In: *13th IEEE International Conference on Peer-to-Peer Computing*. 2013.

[7] Btc Block Explorere. *Block Explorer*. URL: https://btc.com/. (accessed: 01.07.2020).

[8] Ittay Eyal. *The Miner's Dilemma*. 2014. arXiv: 1411.7099 [cs.CR].

[9] Ittay Eyal and Emin Gun Sirer. *Majority is not Enough: Bitcoin Mining is Vulnerable*. 2013. arXiv: 1311.0243 [cs.CR].

[10] Christos Faloutsos et al. "Classical and Contemporary Approaches to Big Time Series Forecasting". In: *Proceedings of the 2019 International Conference on Management of Data*. ACM. 2019, pp. 2042–2047.

[11] Ben Fisch, Rafael Pass, and Abhi Shelat. "Socially optimal mining pools". In: *International Conference on Web and Internet Economics*. Springer. 2017, pp. 205–218.

[12] Befekadu G. Gebraselase, Bjarne E. Helvik, and Yuming Jiang. "Transaction Characteristics of Bitcoin". In: *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. 2021, pp. 544–550.

[13] S. Geissler et al. "Discrete-Time Analysis of the Blockchain Distributed Ledger Technology". In: *2019 31st International Teletraffic Congress (ITC 31)*. 2019, pp. 130–137.

[14] Alex Greaves and Benjamin Au. "Using the bitcoin transaction graph to predict the price of bitcoin". In: *No Data* (2015).

[15] Y. Gu and W. Guo. "Apply the Decision Tree Model to Enterprise Informatization Indicators Analysis". In: *2010 International Forum on Information Technology and Applications*. Vol. 2. 2010, pp. 6–9.

[16] Alireza Toroghi Haghighat and Mehdi Shajari. "Block withholding game among bitcoin mining pools". In: *Future Generation Computer Systems* 97 (2019), pp. 482–491.

[17] Hossein Hassani, Xu Huang, and Emmanuel Silva. "Big-Crypto: Big Data, Blockchain and Cryptocurrency". In: 2 (Oct. 2018), pp. 34–.

[18] Mona Ibrahim et al. "Nonlinear autoregressive neural network in an energy management strategy for battery/ultra-capacitor hybrid electrical vehicles". In: *Electric Power Systems Research* 136 (2016), pp. 262–269.

[19] H. Jang and J. Lee. "An Empirical Study on Modeling and Prediction of Bitcoin Prices With Bayesian Neural Networks Based on Blockchain Information". In: *IEEE Access* 6 (2018), pp. 5427–5437.

[20] Konstantinos Lampropoulos, Giorgos Georgakakos, and Sotiris Ioannidis. "Using Blockchains to Enable Big Data Analysis of Private Information". In: *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. 2019, pp. 1–6.

[21] W. Li et al. "Mining Pool Game Model and Nash Equilibrium Analysis for PoW-Based Blockchain Networks". In: *IEEE Access* 8 (2020), pp. 101049–101060.

[22] X. Liu et al. "Evolutionary Game for Mining Pool Selection in Blockchain Networks". In: *IEEE Wireless Communications Letters* 7.5 (2018), pp. 760–763.

[23] Y. Liu et al. "An Intelligent Strategy to Gain Profit for Bitcoin Mining Pools". In: *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*. Vol. 2. 2017, pp. 427–430.

[24] Loi Luu et al. "On power splitting games in distributed computation: The case of bitcoin pooled mining". In: *2015 IEEE 28th Computer Security Foundations Symposium*. IEEE. 2015, pp. 397–411.

[25] Isaac Madan, Shaurya Saluja, and Aojia Zhao. "Automated bitcoin trading via machine learning algorithms". In: *URL: http://cs229. stanford. edu/proj2014/Isaac% 20Madan* 20 (2015).

[26] Mathworks. *ensemble algorithms*. URL: https://www.mathworks.com/help/stats/ensemble-algorithms.html. (accessed: 01.07.2020).

[27] Satoshi Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System". In: *Cryptography Mailing list at https://metzdowd.com* (Mar. 2009).

[28] Daniel Perez, Jiahua Xu, and Benjamin Livshits. "Revisiting Transactional Statistics of High-scalability Blockchain". In: *ACM Internet Measurement Conference (IMC)*. 2020.

[29] R. Qin, Y. Yuan, and F. Wang. "Research on the Selection Strategies of Blockchain Mining Pools". In: *IEEE Transactions on Computational Social Systems* 5.3 (2018), pp. 748–757.

[30] M. Salimitari et al. "Profit Maximization for Bitcoin Pool Mining: A Prospect Theoretic Approach". In: *2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC)*. 2017, pp. 267–274.

[31] Okke Schrijvers et al. "Incentive compatibility of bitcoin mining pool reward functions". In: *International Conference on Financial Cryptography and Data Security*. Springer. 2016, pp. 477–498.

[32] Devavrat Shah and Kang Zhang. "Bayesian regression and Bitcoin". In: *2014 52nd annual Allerton conference on communication, control, and computing (Allerton)*. IEEE. 2014, pp. 409–414.

[33] Jun.Kawahara Shoji.Kasahara. "Effect of Bitcoin fee on transaction-confirmation process". In: *Journal of Industrial and Management Optimization* 15.1547 (2019), p. 365.

[34] E. Sin and L. Wang. "Bitcoin price prediction using ensembles of neural networks". In: *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. 2017, pp. 666–671.

[35] Yonatan Sompolinsky et al. "Bitcoin Mining Pools: A Cooperative Game Theoretic Analysis". In: vol. 2. Jan. 2015.

[36] Tsung-Nan Lin et al. "A delay damage model selection algorithm for NARX neural networks". In: *IEEE Transactions on Signal Processing* 45.11 (1997), pp. 2719–2730.

[37] E VALUATIONS. "A REVIEW ON EVALUATION METRICS FOR DATA CLASSIFICATION EVALUATIONS". In: 2015.

[38] C. Wang, X. Chu, and Y. Qin. "Measurement and Analysis of the Bitcoin Networks: A View from Mining Pools". In: *2020 6th International Conference on Big Data Computing and Communications (BIGCOM)*. 2020, pp. 180–188.

[39] C. Wang, X. Chu, and Y. Qin. "Measurement and Analysis of the Bitcoin Networks: A View from Mining Pools". In: *2020 6th International Conference on Big Data Computing and Communications (BIGCOM)*. 2020, pp. 180–188.

[40] Steven A. Wright. "Privacy in IoT Blockchains: with Big Data comes Big Responsibility". In: *2019 IEEE International Conference on Big Data (Big Data)*. 2019, pp. 5282–5291.

[41] Yurong Zhong. "The analysis of cases based on decision tree". In: *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. 2016, pp. 142–147.