Empirical risk analysis of mining a Proof-of-Work blockchain

Hansjoerg Albrecher, Dina Finger, Pierre-Olivier Goffard

Empirical risk analysis of mining a Proof-of-Work blockchain

Hansjörg Albrecher* Dina Finger† Pierre-O. Goffard‡

May 24, 2024

Dedicated to the memory of Ermanno Pitacco

Abstract

The process of mining blocks on a blockchain utilizing a Proof-of-Work consensus mechanism carries inherent risks, particularly when the operational expenses associated with mining exceed the rewards earned. Building on previous findings on mining in pools, this paper delves into the question of whether the theoretical formulas for the ruin probability and the expected value of future surplus obtained under particular model assumptions are indeed validated empirically. In particular, we include the presence of transactions fees in the block rewards in our analysis. We also provide algorithms to fit the involved generalized hyperexponential distributions to actual data. Moreover, we perform a sensitivity analysis for different factors of interest, and we quantify the relevance of incorporating temporal dependence and transaction fees in the model.

1. Introduction

A blockchain is a data ledger which is maintained by a *Peer-to-Peer* network. The database entries, referred to as transactions, are recorded by batches called *blocks* resulting from the application of a consensus protocol. In the case of the bitcoin, the consensus protocol is the *Proof-of-Work*. The network participants, called *miners*, compete to solve a cryptoproblem via a trial and error approach. Computers are running 24/7 which consumes a lot of electricity. This operational cost is borne by the miners and compensated by a reward expressed in cryptocurrency units whenever a new block is found. The stability of blockchain systems relies heavily on this incentive mechanism. The balance between cost and reward is a stochastic process, denoted by $(R_t)_{t\geq 0}$, which we model as

$$R_t = u - C_t + B_t,\ t \geq 0, \tag{1}$$

where the initial capital u is augmented by the income $(B_t)_{t\geq 0}$ net of the expenses $(C_t)_{t\geq 0}$. Such models were studied in [3, 1, 21] assuming that

$$C_t = c \cdot t,\ \text{for } c, t \geq 0,$$

*The Faculty of Business and Economics, University of Lausanne and Swiss Finance Institute, Quartier UNIL-Chamberonne Bâtiment Extranef, 1015 Lausanne, Switzerland, hansjoerg.albrecher@unil.ch

†The Faculty of Business and Economics, University of Lausanne, Quartier UNIL-Chamberonne Bâtiment Extranef, 1015 Lausanne, Switzerland, dina.finger@unil.ch

‡Université de Strasbourg, IRMA (UMR 7501), Strasbourg, France, goffard@unistra.fr

where c is the intensity of the mining cost, and

$$B_t = \sum_{i=1}^{N_t} U_i, \ t \geq 0,$$

where $(N_t)_{t\geq 0}$ is a Poisson process and the block rewards U_i form a sequence of positive random variables. One goal of this paper is to propose a more accurate model for these rewards.

Block rewards consist of a protocol-specified bounty augmented by transactions fees. When passing a transaction, users will typically attach a transaction fee to it. This fee mostly depends on the transaction volume, since each block has a fixed allotted space inside. The pending transactions are stored in the `memory pool` (often abbreviated as `mempool`), where they await confirmation, hereby forming a queue. The transaction fee level closely relates to the network congestion that may be tracked by looking at the `mempool` size. Algorithms have been developed to inform the users of the appropriate transaction fee levels, see for instance the book of Antonopoulos [5, Chapter 5]. Although common practice suggests that miners prioritize transactions with the highest transaction fee per byte rate, some deviations can be observed due to potential arbitrage opportunities linked to include specific transactions or ordering them in a given manner. Such considerations are beyond the scope of the present study, and we refer the reader to the work of Messias et al. [25]. In the bitcoin blockchain, the impact of the transaction fees is currently still minor when compared to the bounty for finding a new block. However, as the reward gets halved approximately every four years, the need to understand the underlying dynamics of transaction fees will become pivotal in the future. In Carlsten et al. [10], the authors envision the stability of the system when the block reward reduces to the transaction fees. Möser & Böhme [26] analyse the main drivers of the fees and conclude that higher fees lead to faster transaction processing. Easly et al. [16] link the proportion of zero-fee transactions to the bitcoin price and memory pool size through a linear model. Tedeschi et al. [30] build a neural net that outputs the probability for a transaction to be included based on the transaction features. Finally, Rossi et al. [27] consider a queueing model to estimate the confirmation time.

The block reward is expressed in crypto-currency units, but the operational cost in (1) is likely to be expressed in fiat currency. The question of modelling the exchange rate of crypto against fiat currencies naturally arises when studying the profit and losses of blockchain miners via model (1). The evolution price of cryptocurrencies has been extensively studied in the literature. Ciaian et al. [13] study the bitcoin price formation incorporating market information, such as the Dow Jones stock market index or the oil price. Bouoiyour et al. [9] decompose the bitcoin price index using Empirical Mode Decomposition, which is similar to signal-processing techniques, but does not assume periodicity, see e.g. [33]. Many authors have applied neural network techniques to fit and predict bitcoin prices. For instance, Almeida et al. [4] use Artificial Neural Networks (ANN) and find that trading volumes are irrelevant. McNally et al. [24] use Recurrent Neural Networks (RNN) of Long Short Term Memory type to accomododate their three-year long dataset. Time series models like GARCH [20] and ARIMA [6, 32] have also been considered.

In this paper, we aim to compare two modelling strategies. The first one assumes an independence framework, and the second one will adopt a time-dependent point of view. To introduce the ap-

proaches, we assume in the first part that the block rewards are independent and identically distributed (i.i.d.) random variables with a generalized hyperexponential (GH) distribution, studied in Botta et al. [8], also referred to as the combination of exponentials model by Dufresne [15]. The probability density function of a GH distribution is a linear combination of exponentials which does not need to be convex (in contrast to mixture of exponential distributions). The GH class is a proper subset of matrix-exponential (ME) distributions which are probability distributions with rational Laplace transform, see already Cox [14] and Bladt & Nielsen [11] for a recent overview. The GH class leads to tractable calculations and is itself already dense in the set of probability distributions on the positive half-line. For applications in insurance risk theory, see e.g. Lin & Willmot [22, 23]. In that framework, (1) is called the dual model of the standard insurance risk model, as the wealth process performs upward jumps and decreases linearly in time. In our previous work [1], we found closed-form expressions for the ruin probability and the expected profit when the rewards are GH distributed. In this paper we want to go one step further and fit such a distribution to actual bitcoin data. With the help of the obtained results, Bitcoin miners can assess their risk and profitability trade-off, thus adapting their mining strategy by joining a pool if necessary, or, in the worst case, stop their activity.

Since the construction is not probabilistic, a priori one can not guarantee that a particular fitted set of parameters for a GH (and more generally ME) distribution represents a proper probability distribution. Conditions based on the roots and poles of the Laplace transform have been given in the works of Bean et al. [8] and Fackrell [18]. These conditions apply to ME distributions and were integrated in Fackrell [17] within a maximum likelihood estimation procedure. Dufresne [15] used GH distributions to approximate any distribution on the positive half-line via an expansion in terms of Jacobi polynomials. The non-negativity problem is addressed then by expanding the square root of the probability density. In this work, we explore the polynomial approach to fit data to a GH distribution. The non-negativity of the resulting probability density function is checked via a bisection procedure suggested in the work of Hanzon & Holland [19]. To the best of our knowledge, this paper is the first to fit combinations of exponentials to actual data in a non-parametric way instead of approximating a predefined distribution. The results may therefore also be applicable in other modelling contexts where such a distributional assumption is assumed, see e.g. [2] for an example in ruin theory.
The assumption of stationarity, a key premise in our first approach, might be considered too restrictive, as block rewards are influenced by transaction fees and fluctuations in cryptocurrency prices. Consequently, we adopt a second approach that views block reward data as an ARIMA time series. After fitting it to the data, we generate scenarios that enable us to estimate risk and performance indicators through Monte Carlo simulations. We then conduct a numerical comparison between the results obtained under this time series framework and the ones based on the i.i.d. assumption in our first approach, together with a series of further sensitivity results on various assumptions underlying the model for which a tractable formula for the key quantities is available. One of the main goals of this paper is to assess the practical accuracy of formulas provided in [1] which relied on an i.i.d. assumption for the rewards distributions and did not consider transaction fees. The results can be used by the miner for his risk/profitability analysis. In addition, we provide a method to fit combinations of exponentials to empirical data.

The rest of the paper is organized as follows. Section 2 provides a brief description of our data together with reminders about the way that Proof-of-Work blockchains operate. Section 3 describes

the combination of exponentials model and our block reward distribution function estimators. The proposed estimation is first back-tested on synthetic data, before it is applied to the actual block reward data. Section 4 presents the result of our time series analysis. Section 5 compares the two modelling approaches, looking at their impact on the profitability and ruin of blockchain miners. It also contains sensitivity tests with respect to the inclusion of transaction fees in the modelling as well as the electricity price. Section 6 concludes.

2. Transaction fee concepts and descriptive data analysis

In this section, we will analyse empirical data on the transaction fees in the bitcoin cryptocurrency. Let us first give a short reminder on the definition of transaction fees and their importance in the mining process.

In the bitcoin *Proof-of-Work* verification algorithm, each miner solves a cryptoproblem in order to validate a block. Whenever a block is validated or *mined*, the miner receives the corresponding block reward set to 6.25 BTC at the time of writing. In addition, the sum of transaction fees attached to all the transactions included in this block are also given as an additional reward to the miner. As part of the validation system, the miner obtains the authorization from the community to choose which transactions from the *memory pool* ("waiting line") are entering the newly mined block. The user will typically attach a transaction fee to the required payment. This fee mostly depends on the transaction volume, since a block has a fixed allotted space inside. It also depends on the market congestion and the individual user's decisions [5]. There exist algorithms that help the user choose an appropriate fee. For example on https://privacypros.io/tools/bitcoin-fee-estimator/, one can estimate the expected confirmation time as a function of the chosen fee. In the mining process, the miners have to decide which transactions to incorporate in the block, since it has limited memory space. They will consider the priority of the transaction based on the attached fee size per byte, but recent research shows that there may be other (selfish) interests in promoting transactions [25]. It is important to note again that with the scheduled halving of the fixed block rewards every four years, the share of the transactions fees in the total rewards will gain decisive importance over time. Thus, including and modelling this stochastic part will become even more relevant in the future.

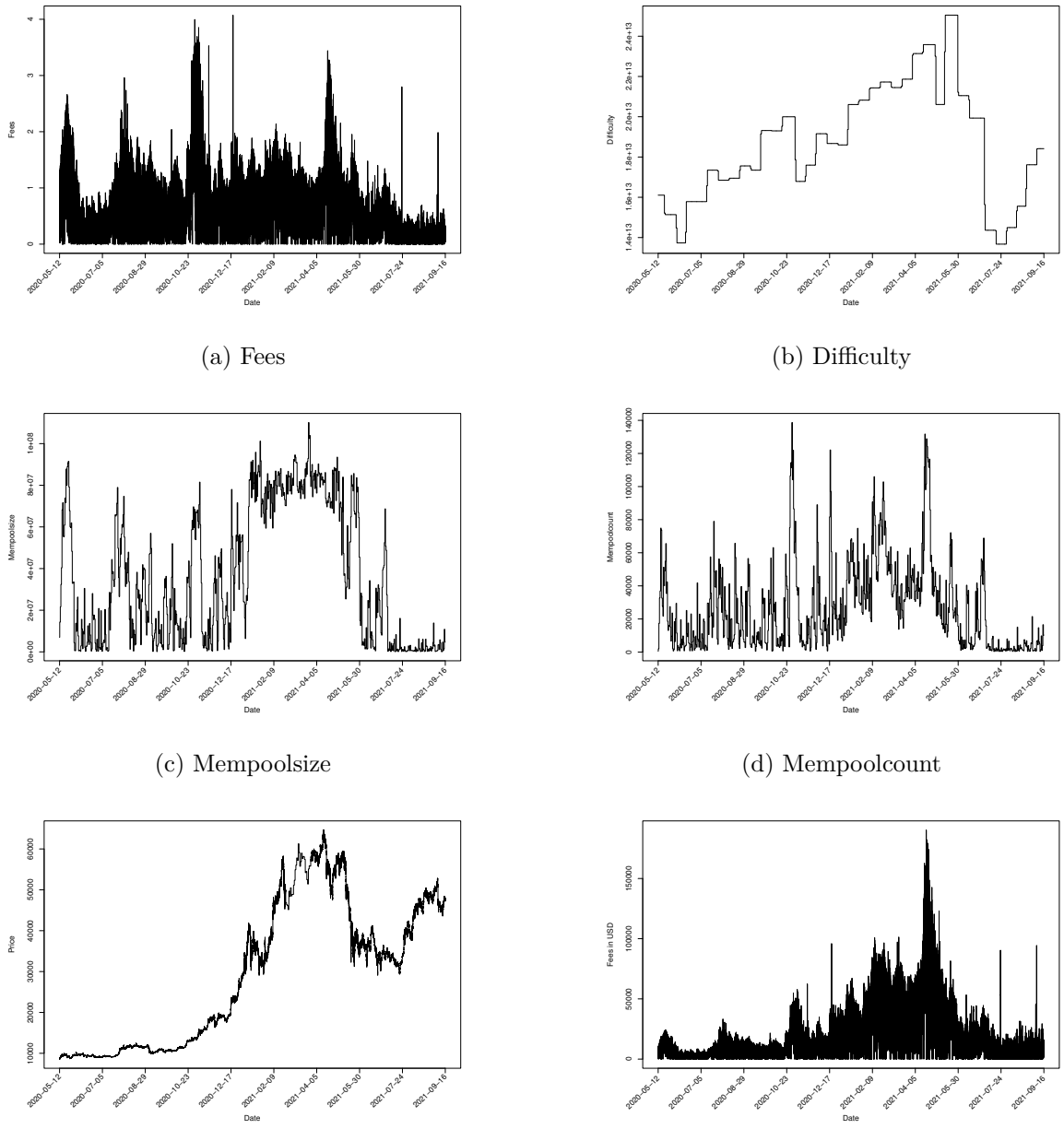
For this analysis, we use publicly available data. Scrapping bitcoin-related data is possible by making calls to some APIs. This was done by looping through necessary blocks. For example, to access block number 650000 information, one can follow the address https://chain.api.btc.com/v3/block/650000[1]. It extracts information in JSON format, which can then be reformatted to our needs. With this method, we gather data for the period spanning from the last halving of the bitcoin reward on May 12, 2020 until September 16, 2021. For this time frame, we collect the following information:

- The transaction fees per block;
- The exchange rate BTC-USD per minute;
- The current difficulty of the cryptoproblem (adjusts bi-weekly);

[1] last accessed on 12/09/2023

- The size of the memory pool of the transactions in bytes (daily);
- The number of transactions in the memory pool (daily).

The difficulty of the cryptoproblem is the ratio between the current target of the Proof-of-work algorithm and the maximal possible target value. For an example of a hash and the difficulty, see [1]. Figure 1 depicts the extracted data and Table 1 contains some statistics of the dataset. One can observe a strong correlation between the transaction fees and the number of transactions in the memory pool, which is confirmed in Table 2.



(a) Fees

(b) Difficulty

(c) Mempoolsize

(d) Mempoolcount

(e) BTC/USD price

(f) Fees in USD

Figure 1: Illustration of the data

For the analysis in this paper, we consider a smaller sub-sample of the data containing 10,000 data points ranging from February 10, 2021 to April 21, 2021. Also, we merge the series of fees and prices by converting fees in USD, since we aim to model the latter. In the sequel of this paper,

Table 1: Main statistics of the dataset.

	Block	Date	Fees in BTC	Price in USD	Mempool size	Mempool count	Difficulty
Mean			0.594	29,535	36,090,588	25,610	1.86E+13
Min	630,014	12.05.2020	0	8,584	120,928	290	1.37E+13
25^{th} pct.			0.1731	11,324	3,889,528	4,757	1.68E+13
Median			0.4652	31,700	27,446,996	18,670	1.86E+13
75^{th} pct.			0.8872	46,331	70,698,937	38,942	2.08E+13
Max	700,851	16.09.2021	4.074	64,617	110,094,398	138,640	2.51E+13

Table 2: Pearson correlation matrix.

	Fees in BTC	Mempool size	Mempool count	Difficulty	Price
Fees in BTC	1	0.489	0.636	0.331	-0.018
Mempool size	0.489	1	0.746	0.672	0.437
Mempool count	0.636	0.746	1	0.534	0.255
Difficulty	0.331	0.672	0.534	1	0.544
Price	-0.018	0.437	0.255	0.544	1

we opt for modelling this latter time series directly, the dynamics of which then aggregate all the potential covariate effects (in future work, one may want to refine that analysis to take into account covariate information more explicitly). As illustrated in Figure 2, the time series of fees exhibits strong autocorrelation, so that a stationarity assumption would not be appropriate. However, by taking the first difference, the one-time differentiated series exhibits again features of stationarity to a large extent, see the right-hand side of Figure 2; only one lag is still significant for the fees. We will therefore consider only one lag when fitting a time series model in Section 4.

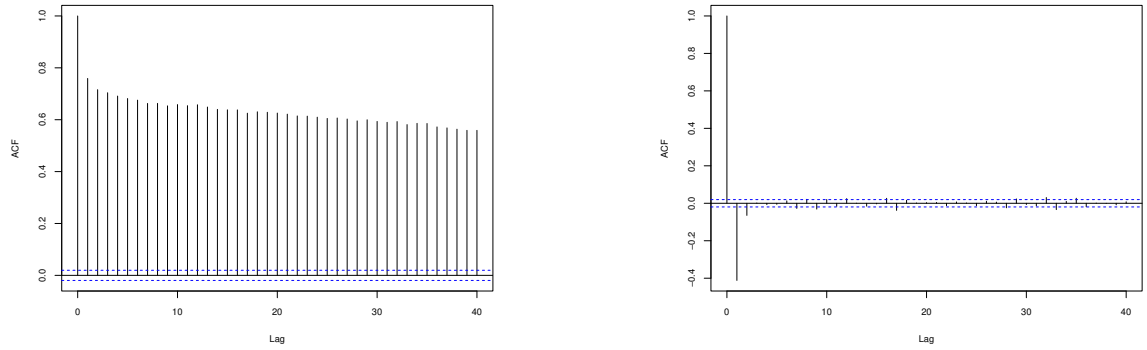


Figure 2: Autocorrelation function of fees in USD (left) and its first differences (right)

3. Block reward as a combination of exponentials

A random variable U has a generalized hyperexponential distribution if its cumulative distribution function (CDF) is given by

$$F_U(x) = 1 - \sum_{i=1}^{d} a_i e^{-\lambda_i x}, \text{ for } x \geq 0, \tag{2}$$

where $\lambda_1, \ldots, \lambda_d > 0$ and $a_1, \ldots, a_d \in \mathbb{R}$ with $\sum_{i=1}^{d} a_i = 1$. In the sequel, we assume that $\lambda_1 < \ldots < \lambda_d$. Define the vectors

$$\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_d)\,, \; \boldsymbol{a} = (a_1, \ldots, a_d)\,,$$

and the diagonal matrix $\Lambda = \mathrm{diag}(\boldsymbol{\lambda})$, so that

$$F_U(x) = 1 - \boldsymbol{a} \cdot e^{-\Lambda x} \cdot \mathbf{1}_d, \text{ for } x \geq 0, \tag{3}$$

where $\mathbf{1}_d = (1, \ldots, 1)$. The probability density function (PDF) is given by

$$f_U(x) = \boldsymbol{b} \cdot e^{-\Lambda x} \cdot \mathbf{1}_d, \text{ for } x \geq 0, \tag{4}$$

with $\boldsymbol{b} = \boldsymbol{a} \cdot \Lambda$.

The Laplace transform of U

$$\mathbb{E}(e^{-\theta U}) = \boldsymbol{b} \cdot (\Lambda + \theta I_d)^{-1} \cdot \mathbf{1}_d, \; \theta \geq 0, \tag{5}$$

where I_d is the identity matrix, is rational, which implies that combinations of exponential distributions are instances of matrix-exponential distributions.

3.1 Non-negativity of GH probability density functions

Given a set of parameters $\boldsymbol{a}$ and $\boldsymbol{\lambda}$, there is no straightforward way to ensure that (3) is a proper CDF or that (4) is a proper PDF. Some characterization, based on the Laplace transform, have been provided by Bean et al. [7] and Fackrell [18] for ME distributions. We take a different road here, following up on the work of Hanzon and Holland [19]. Consider the function

$$f(x) = \boldsymbol{b} \cdot e^{-\Lambda x} \cdot \mathbf{1}_d, \quad x \geq 0. \tag{6}$$

In order for f to be a proper PDF, we need to ensure

$$f(x) \geq 0, \; \forall x > 0, \text{ and } \int_0^\infty f(x)\mathrm{d}x = 1. \tag{7}$$

Necessary conditions for (7) to hold include $b_1 > 0$ and $\boldsymbol{b} \cdot \Lambda^{-1}\mathbf{1}_d = 1$, but the latter are not sufficient. We therefore use a verification method via bisection, suggested in Hanzon and Holland [19]. We define the sequence

$$f_0(x) = \boldsymbol{b} \cdot e^{-\Lambda x} \cdot \mathbf{1}_d = f_U(x) \text{ and } f_k(x) = \boldsymbol{b} \cdot \prod_{i=1}^{k} (\lambda_i \cdot I_d - \Lambda) \cdot e^{-\Lambda x} \cdot \mathbf{1}_d, \text{ for } k = 1, \ldots, d.$$

For a given upper bound S, it is characterized by the following property: for $x \in [0, S]$, the function f_k has at most one sign-changing zero between two sign-changing zeros or boundary points of f_{k+1} for $k = 0, 1, \ldots, d-1$. As $f_d(x) = 0 \; \forall x$, one can recursively, starting from $f_d(x)$, check the presence of sign-changing points on a closed interval through a bisection procedure.

3.2 Fitting GH distributions to data via polynomial expansions

Dufresne [15] presents a method to approximate any PDF of a distribution on the positive half-line via a combination of exponentials. The approximation formula takes the form of an expansion in terms of the shifted Jacobi polynomials defined as

$$R_k^{(\alpha,\beta)}(x) = \sum_{j=0}^{k} \rho_{k,j} x^j, \ x \in \mathbb{R}, \tag{8}$$

where

$$\rho_{k,j} = \frac{(-1)^k(\beta+1)_k(-k)_j(k+\alpha+\beta+1)_j}{(\beta+1)_j k! j!}, \tag{9}$$

where $(z)_k = z \cdot (z+1) \cdot (z+2) \cdot \cdots \cdot (z+k-1)$ denotes the Pochhammer symbol. These polynomials are orthogonal on $[0,1]$ w.r.t. the weight function $\phi(x) = (1-x)^\alpha x^\beta$, where $\alpha > -1$ and $\beta > -1$. Any function $g : (0,1) \mapsto \mathbb{R}$, square integrable w.r.t. $\phi(x)$ can be expanded as a shifted Jacobi polynomial expansion with

$$g(x) = \sum_{k=0}^{\infty} c_k R_k^{(\alpha,\beta)}(x), \ x \in [0,1], \tag{10}$$

where

$$c_k = \frac{1}{h_k} \int_0^1 g(x)\phi(x) R_k^{(\alpha,\beta)}(x) \mathrm{d}x, \tag{11}$$

and

$$h_k = \int_0^1 \phi(x) R_k^{(\alpha,\beta)}(x)^2 dx = \frac{\Gamma(k+\alpha+1)\Gamma(k+\beta+1)}{(2k+\alpha+\beta+1)k!\Gamma(k+\alpha+\beta+1)}. \tag{12}$$

The convergence in (10) takes place in the L^2 sense, see for instance the book of Nagy [12, Ch.7]. Our target is a PDF f on the positive half-line. Following Dufresne [15], we expand the function $f^*(t) = e^{prt} f(t)$ for some $p \in \mathbb{R}$ and $r > 0$ and use the change of variable, that maps the interval $(0,1)$ onto $(0,\infty)$, $y = -\frac{1}{r}\log x$, such that

$$g(x) = f^*\big(-\frac{1}{r}\log x\big).$$

The normalizing constant from Equation (12) is not affected by the change of variable and can be expressed as $h_k = r\int_0^\infty \phi(e^{-ry}) R_k^{\alpha,\beta}(e^{-ry})^2 e^{-ry} \mathrm{d}y$. The expansion of f is then

$$f(t) = e^{-rpt} \sum_{k=0}^{\infty} c_k R_k^{(\alpha,\beta)}(e^{-rpt}), \tag{13}$$

and the square integrability condition on g translates directly to

$$\int_0^\infty e^{-(1-p)rt}\phi(e^{-rt}) f^2(t) \mathrm{d}t < \infty.$$

The coefficients of the polynomial expansion can also be expressed as an integral in terms of f as

$$c_k = \frac{r}{h_k} \int_0^\infty e^{-(1-p)rt}\phi(e^{-rt}) R_k^{(\alpha,\beta)}(e^{-rt}) f(t) dt. \tag{14}$$

A simple truncation of the infinite series in (13), followed by a normalization so that it integrates to 1, yields an aproximation

$$f(t) \approx e^{-rpt} \sum_{k=0}^{d-1} c_k R_k^{(\alpha,\beta)}(e^{-rt}), \tag{15}$$

which is consistent with what is referred to as Method A in Dufresne's work [15]. Replacing the polynomials $R_k^{(\alpha,\beta)}(x)$ by $\sum_{j=0}^{k} \rho_{k,j} x^j$ in (15) yields

$$f(t) \approx \sum_{j=1}^{d} \sum_{k=j-1}^{d-1} c_k \rho_{k,j-1} e^{-(j-1+p)rt},$$

which is a combination of exponentials as in (4) where

$$\mathbf{b} := \left(\sum_{k=0}^{d-1} c_k \rho_{k,0}, \quad \ldots, \quad c_{d-1}\rho_{d-1,d-1} \right) \text{ and } \boldsymbol{\lambda} := \left(pr, \quad (1+p)r, \quad \ldots, \quad (n+p)r \right).$$

Remark 3.1. The approximation method involves selecting parameters α, β, p, r, and determining the truncation order d. In the absence of established selection guidelines, we draw upon the parameter values utilized in Dufresne's work [15]. The choice of the truncation order should strike a balance between accuracy, computational efficiency, and numerical stability, favoring larger values wherever possible. When considering f^* instead of f, it ensures that $f(t)$ approaches zero as t tends towards infinity for any truncation order, provided that $0 < p < (\beta+1)/2$.

For our application, we do not have a known distribution function to approximate, but a dataset to fit. Assume that $\{x_1, \ldots, x_M\}$ form an i.i.d. sample of size M. We can replace the expansion coefficients defined in (14) by their empirical counterpart with

$$\widehat{c}_k = \frac{r}{h_k M} \sum_{m=1}^{M} e^{-(1-p)rx_m} \phi(e^{-rx_m}) R_k^{(\alpha,\beta)}(e^{-rx_m}), \text{ for } k = 0, \ldots, d-1. \tag{16}$$

An a posteriori control with the help of the bisection method from Section 3.1 can ensure the non-negativity of the estimated PDF

$$\widehat{f}_M(t) = e^{-rpt} \sum_{k=0}^{d-1} \widehat{c}_k R_k^{(\alpha,\beta)}(e^{-rt}). \tag{17}$$

The estimated PDF is a nonparametric density estimator relying on orthogonal functions, a method detailed in [31, Ch.8]. A recognized limitation of this approach is its susceptibility to occasional negative values stemming from sampling errors. In instances where our estimates exhibit negativity, we can employ what Dufresne [15] terms as 'Method B' as a corrective measure. Instead of expanding $e^{prt} f(t)$, consider expanding

$$\tilde{f}(t) = e^{prt} \sqrt{f(t)}.$$

We get an approximation formula of the form

$$\sqrt{f(t)} \approx e^{-prt} \sum_{k=0}^{d-1} c_k R_k^{(\alpha,\beta)}(e^{-rt}) = \sum_{j=1}^{d} b_j e^{-\lambda_j t}, \tag{18}$$

and finally squaring it yields an approximation of f which is a proper PDF after normalization:

$$f(t) \approx \sum_{j=1}^{d}\sum_{k=1}^{d} b_j b_k e^{-(\lambda_j+\lambda_k)t} = \sum_{m=1}^{2d-1} \tilde{b}_m e^{-(m-1+2p)rt},$$

with $\tilde{b}_m = \sum_{j=1}^{m} b_j b_{m+1-j}$ and $b_{j>d} := 0$ (i.e. $b_j,\ j > d$), since $\lambda_j = (j-1+p)r,\ j = 1, 2, \ldots, d$. The coefficients of the polynomial expansion of $\sqrt{f}$ are given by

$$b_k = \frac{r}{h_k}\int_0^\infty e^{-(1-p)rt}\phi(e^{-rt})R_k^{(\alpha,\beta)}(e^{-rt})\sqrt{f(t)}dt. \tag{19}$$

To get a statistical estimation of the coefficients, we replace f in (19) by a kernel density estimator

$$\widehat{f}_h(x) = \frac{1}{M}\sum_{m=1}^{M} K_h(x - x_m) = \frac{1}{Mh}\sum_{m=1}^{M} K\Big(\frac{x - x_m}{h}\Big), \tag{20}$$

where $K(x)$ is the Gaussian kernel.

Remark 3.2. In contrast to Method A, Method B guarantees a valid PDF. However, this advantage comes at the expense of significantly increased computational complexity due to doubling the number of terms (with a numerical integration of the kernel density estimator for each term).

3.3 Simulation study

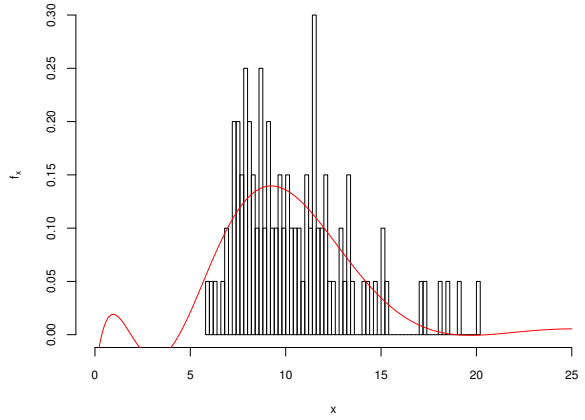
We illustrate our fitting procedure for a combination of exponentials through a brief simulation study. Draw several samples of size n $(x_1, \ldots, x_n)$ from a right-shifted gamma random variable $X = \gamma + Y$, where $\gamma > 0$ and Y has PDF

$$f(x) = \frac{\delta^r x^{r-1} e^{-\delta x}}{\Gamma(r)},\ x > 0,\ r, \delta > 0. \tag{21}$$

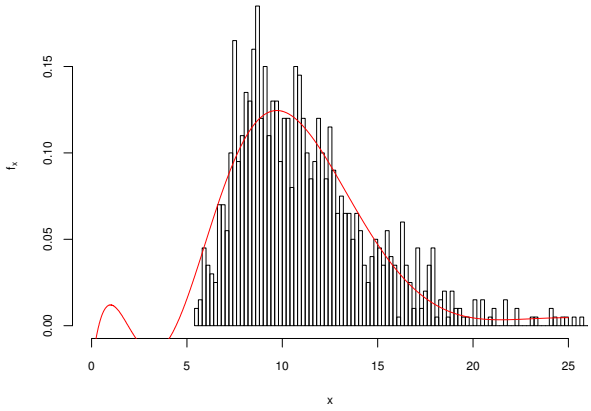
The choice of this shift is motivated by the observed shape of the empirical distribution in our collected block reward data later. We set the parameters to $r = 3, \delta = 0.5$ and $\gamma = 5$. Figure 3 shows the fit (in red) of the combination of exponentials distribution to the data when using Method A with parameters $\alpha = 0, \beta = 0, r = 0.01, p = 0.9, d = 20$ for samples of sizes $n \in \{100, 1000, 10000, 100000\}$. The quality of the fit improves as the sample size increases. However, note that also, for larger sample size, the density estimate occasionally exhibits negative values.
Figure 4 shows the fit (in red) of the combination of exponentials distribution to the data when using Method B with parameters $\alpha = 0, \beta = 0, r = 0.01, p = 0.9, d = 20$, for samples of sizes $n \in \{100, 1000, 10000, 100000\}$. The fit is less good when using Method B; however, it consistently results in a valid probability distribution. Due to this essential property, we have chosen to exclusively employ Method B in our application to the block reward data in the sequel.

3.4 Real data application

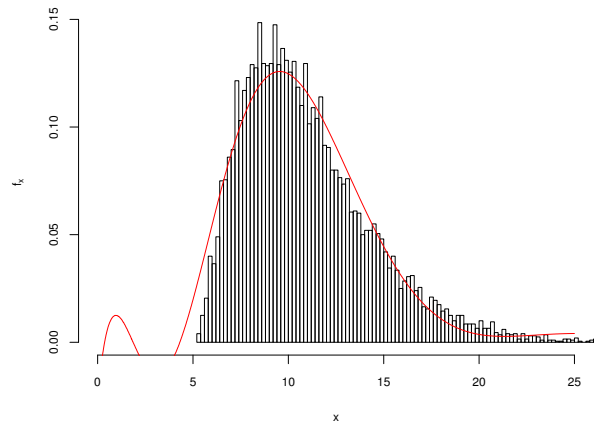
The block reward comprises two components: the reward for discovering a new block, which currently stands at BTC6.25 at the time of writing, and the transaction fees detailed in Section 2. Given that miners typically operate within a fiat currency framework, such as USD, we apply the exchange rate



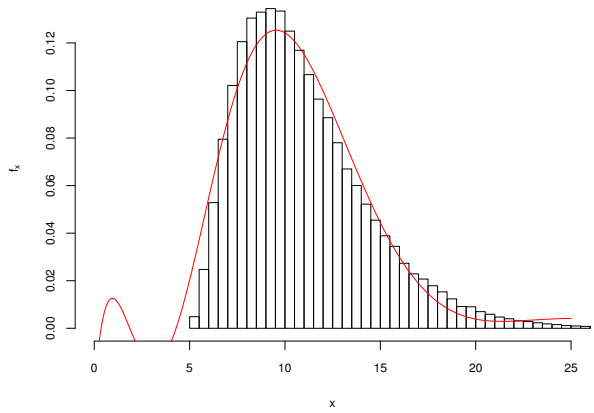
(a) Sample size: 100. $d = 20$, $\alpha = 0$, $\beta = 0$, $r = 0.01$, $p = 0.9$.



(b) Sample size: 1000. $d = 20$, $\alpha = 0$, $\beta = 0$, $r = 0.01$, $p = 0.9$.



(c) Sample size: 10000. $d = 20$, $\alpha = 0$, $\beta = 0$, $r = 0.01$, $p = 0.9$.



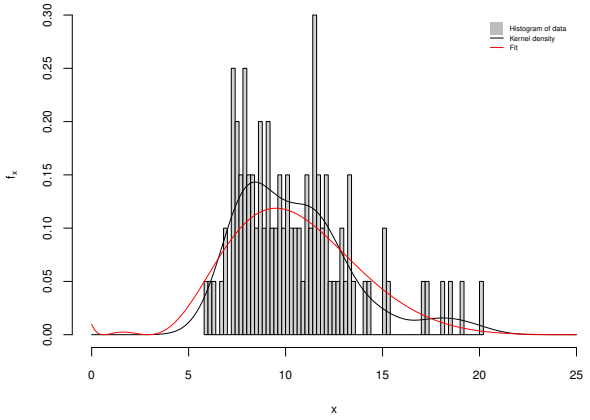
(d) Sample size: 100000. $d = 20$, $\alpha = 0$, $\beta = 0$, $r = 0.01$, $p = 0.9$.

Figure 3: Fitting of the shifted Gamma random sample by modified Method A. The bars depict the empirical density and the red line is the obtained fit.

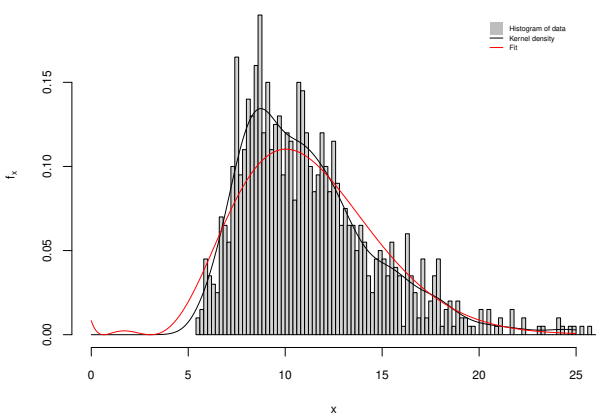
applicable at the moment of block discovery. Figure 5 depicts the histogram of total rewards received by miners, presented both in BTC and USD. The data is indeed shifted away from zero due to the fixed block reward addition, which poses a challenge for parametric fitting methods, as the support of exponential random variables is the entire positive halfline. The non-parametric approach, utilizing a polynomial expansion, provides an advantage by capturing variations in the central mass, even in the absence of data points in the lower tail.

Figure 6 shows the fit of the combination of exponentials using Method B with two parameterizations.

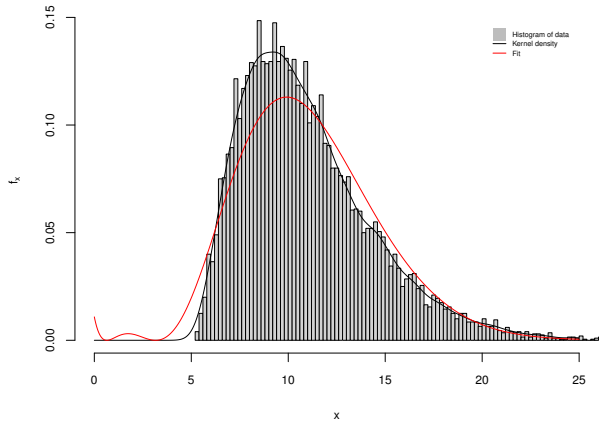
As the number of terms increases, the central part of the density aligns more closely with the actual data, albeit with an increase in instability near the left end. Conversely, reducing the number of terms results in a smoother shape, but the original density's spikes are less pronounced. Renormalization alleviates this effect to some extent (cf. the green line), and we choose to use the parametrization with $d = 10$ for the numerical analysis later.



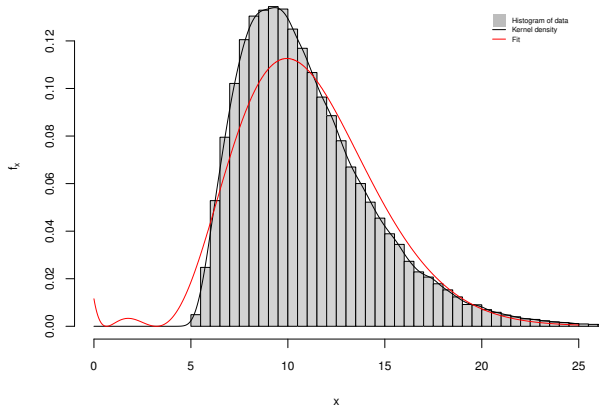
(a) Sample size: 100. $d = 20$, $\alpha = 0$, $\beta = 0$, $r = 0.05$, $p = 0.3$.



(b) Sample size: 1000. $d = 20$, $\alpha = 0$, $\beta = 0$, $r = 0.05$, $p = 0.3$.

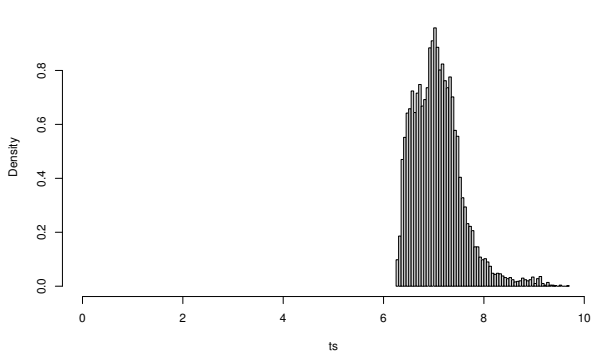


(c) Sample size: 10000. $d = 20$, $\alpha = 0$, $\beta = 0$, $r = 0.05$, $p = 0.3$.

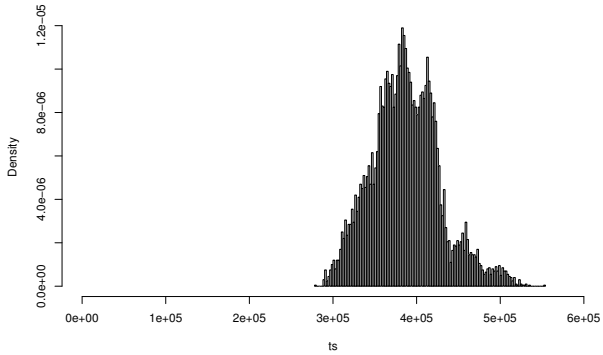


(d) Sample size: 100000. $d = 20$, $\alpha = 0$, $\beta = 0$, $r = 0.05$, $p = 0.3$.

Figure 4: Fitting of the shifted Gamma random sample by modified Method B. The bars depict the empirical density and the red line is the obtained fit.

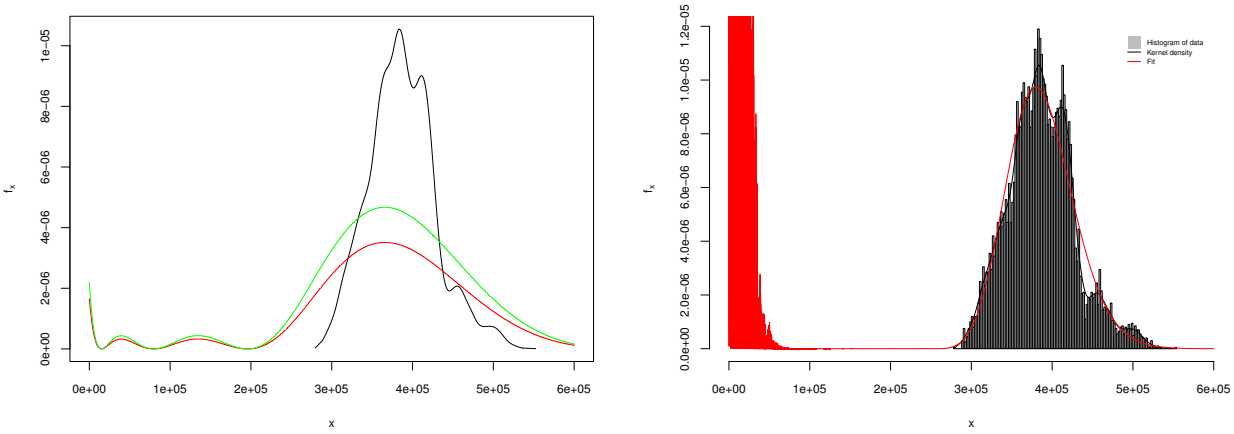


(a) Rewards expressed in BTC.



(b) Rewards expressed in USD.

Figure 5: Histogram of total rewards, period from February 10, 2021 to April 21, 2021.



(a) $d = 10$ ($p = 0.75,\ r = 0.0000008$)

(b) $d = 25$ ($p = 0.3,\ r = 0.000005$)

Figure 6: Approximation of the bitcoin rewards sample (Method B, $\alpha = 0, \beta = 0$)

4. Block rewards as time series

Since we also want to test the model for its sensitivity to non-stationarities, we fit the block reward data to an ARIMA model calibrated using the Box and Jenkins optimization method. Recall that a time series X_t is ARIMA(p, D, q) if $\nabla^D X_t$ is an ARMA(p, q) process, where ∇^D is the D^{th} difference operator. An ARMA(p, q) time series is a stationary process defined as

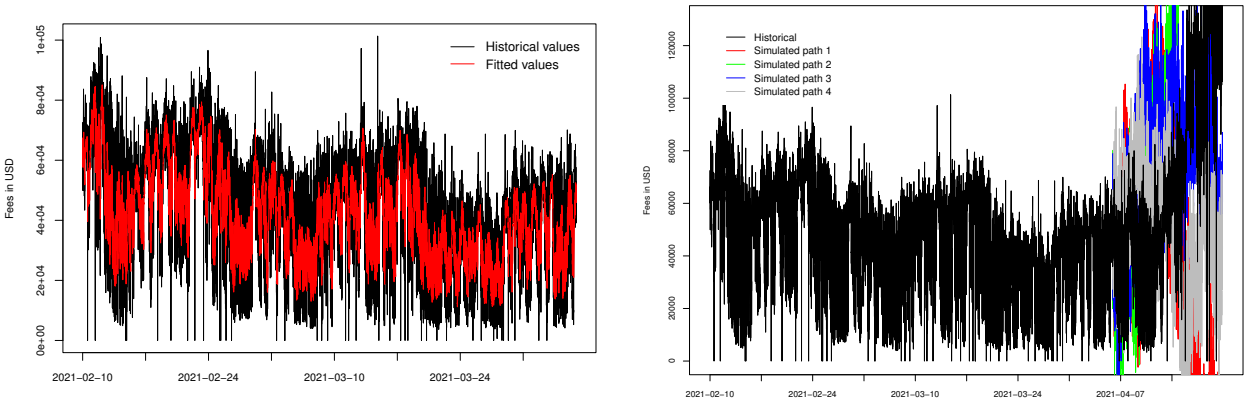
$$X_t = \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + z_t + \theta_1 z_{t-1} + \cdots + \theta_q z_{t-q}, \tag{22}$$

with $\phi_p \neq 0$, $\theta_q \neq 0$ and z_t is white noise with mean 0 and variance σ_z^2, see e.g. [28]. Note that time series modeling is not a focus of this paper, we refer to other papers for this purpose, see e.g. [6, 32]. Our choice of a simple ARIMA model is motivated by its suitability for generating plausible scenarios over a short time horizon while departing from the i.i.d. assumption of Section 3. Also, we opted against more complex GARCH models due to their tendency to overfit the data and their requirement for an extensive number of lags to ensure reliability. Hence, we deliberately select a simple ARIMA models as a pragmatic and reliable choice to conduct our risk analysis over a two weeks time horizon. For fitting our model, we consider a data set from a time frame between February 10, 2021 and April 6, 2021 that we split into a training set and a consecutive test set for checking the data prediction. In Table 3 we summarize the obtained results. The fitted model suggests indeed a once differentiated series. In Figure 7 we show an illustration of the fitted model. On the left-hand side, one can see the

Table 3: Summary of ARIMA(1,1,7) model.

	ar1	ma1	ma2	ma3	ma4	ma5	ma6	ma7
	-0.7799	0.0989	-0.6543	-0.1146	-0.0337	-0.0330	-0.0075	-0.0267
s.e.	0.1122	0.1125	0.0771	0.0194	0.0136	0.0138	0.0123	0.0122
σ^2	219226983							
Log.Lik.	-88159.57							
AIC	176337.1							
BIC	176400							

fitted data points in comparison to the historical values of the training sample. On the right-hand side, we show simulated paths in different colors and the true historical values from our test sample in black. In addition to the statistical fit (cf. Figure 8 for a normal Q-Q plot for the remaining residuals), the fit also seems quite satisfactory visually, which is remarkable for the case of only a few free parameters.



(a) Fitted values.

(b) Simulated rewards per block in USD.

Figure 7: ARIMA(1,1,7) fit to block reward data

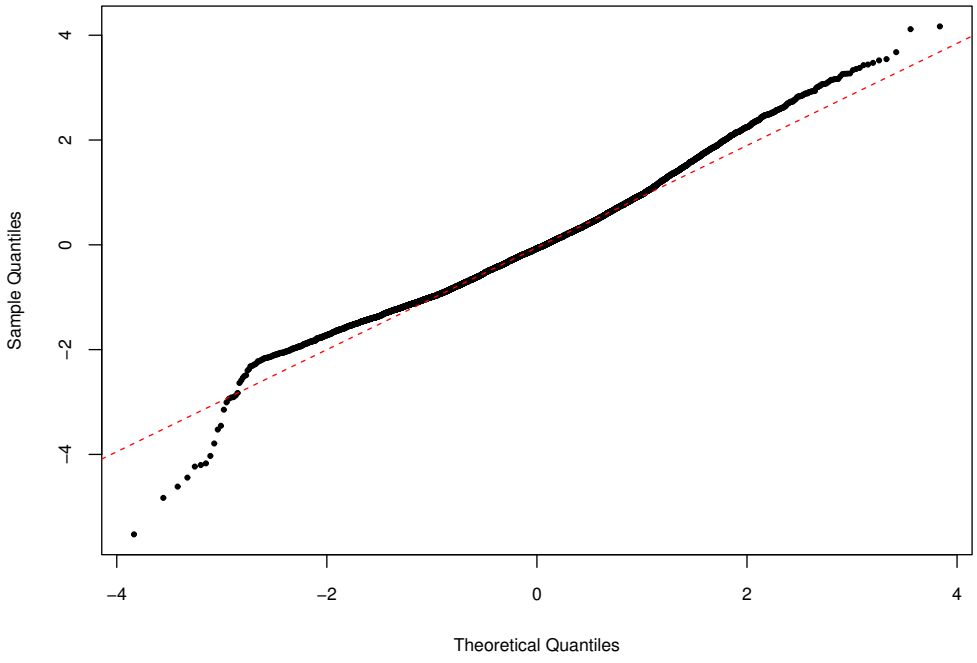


Figure 8: Normal Q-Q plot of ARIMA(1,1,7) residuals.

5. Comparison of the two modelling approaches in terms of risk

In [1, 3], explicit formulas were derived for the ruin probability and the expected surplus of a miner, taking into account the option of participating in a mining pool. Assume the miner's surplus has the form

$$R_t^i = u - c_i \cdot t + \sum_{j=1}^{N_t} U_j, \ t \geq 0,$$

where $N_t \sim Poisson(p_i \mu t)$ and the U_j's are i.i.d. with PDF $f_U(x) = \sum_{j=1}^{d} a_j \lambda_j e^{-\lambda_j x}$, $x > 0$. In [1], it was shown that under the assumption of i.i.d. rewards of GH type, the ruin probability up to an exponential time horizon (with mean t) can be expressed as

$$\hat{\psi}(u,t) = e^{-R \cdot u}, \tag{23}$$

and the miner's expected value of the surplus at that exponential time horizon, given that it did not go negative until then, is given by

$$\hat{V}(u,t) = t\left(c_i - p_i\mu \sum_{j=1}^{d} \frac{a_j}{\lambda_j}\right) e^{-Ru} + u + t\left(p_i\mu \sum_{j=1}^{d} \frac{a_j}{\lambda_j} - c_i\right), \ u > 0, \tag{24}$$

where R is the unique solution with positive real part of the equation

$$c_i R + p_i\mu \sum_{j=1}^{d} \frac{a_j \lambda_j}{R + \lambda_j} - \left(\frac{1}{t} + p_i\mu\right) = 0.$$

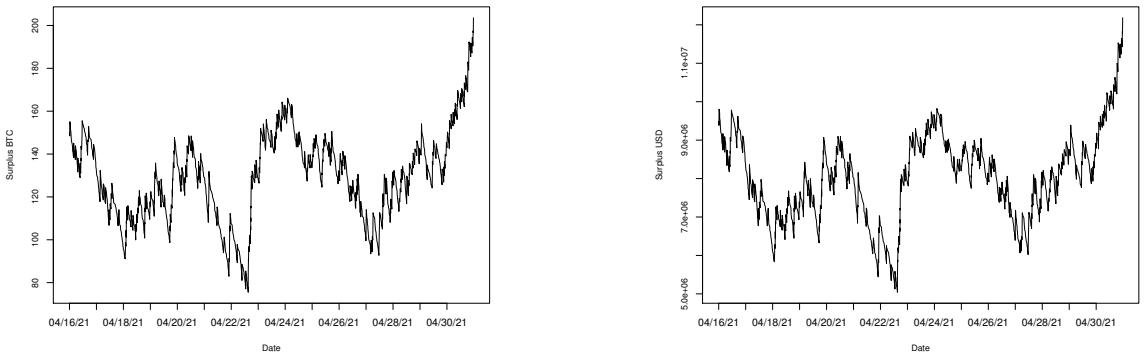
These metrics are of particular significance, given that the process of mining incurs considerable energy costs, and real-world miners may face the risk of financial ruin, which in turn affects their expected earnings.

Our goal in this section is to assess the sensitivity of the risk measures for which we have formulas (23) and (24) w.r.t. some of the model assumptions. First, in Section 5.1 we compare the formulas (using a GH fit to block rewards data from the time period February 10, 2021 to April 21, 2021) to the actual historical realization of the occurrence of blocks and the sizes of the rewards in the period from February 10, 2021 to a random time horizon in the future with a mean of 2 weeks. Note that there are still three random elements to be implemented in the historical path: the actual length of the random time horizon and the probability that a found block was found by a particular miner. Secondly, we consider a 'Full Pay-per-Share' reward system (FPPS) for the pool, in which pool managers instantly reward miners for each share submitted, with the payout determined by the block reward and the entire estimated transaction fees associated with the block. That is, we have to simulate the arrival of shares in addition to the arrival of blocks. As the historical sample path both contains potential time-dependence of transaction fees and the actual block reward values (rather than the GH fit), but only one realization of the two latter effects, we subsequently compare in Section 5.2 the results of (23) and (24) with the simulated counterparts under an i.i.d. assumption, but bootstrapping block reward sizes from the empirical distribution function over the period February 10, 2021 to April 21, 2021. Subsequently, in Section 5.3 we provide a comparison of the formulas (23) and (24) with the simulated counterparts under the ARIMA assumption calibrated in Section 4. Finally, Sections 5.4 and 5.5 consider the sensitivity of the model output concerning the inclusion of transaction fees at all,

as well as concerning the price of electricity.

5.1 Comparison with the historical path

With the collected data on the fees and prices, we can reconstitute the real surplus path of miners or pools in any specific past time period. Indeed, if we position ourselves at some starting date, we can replicate the outflows of mining costs and the inflows of block rewards for the individual miner as well as for the pool. For the following example, we select a two weeks time frame. We select the documented block arrival in the entire system to the particular pool by simulating a Bernoulli random variable with probability equal to the pool's proportion of computational power in the global mining network. For the individual miner, we also simulate the more frequent share payouts by assuming Poisson distributed arrivals of their rewards. In Figures 9 and 10, one can see an illustration of the surplus path for the pool and the miner, respectively. Figures 9a and 9b are almost indistinguishable to the naked eye, but they are not identical. Indeed, for that short particular time horizon, the exchange rate was not very volatile. Here we choose the same parameters as in [1]: $t = 336\text{h} = 2$ weeks, $p_i = 0.001$, $q = 0.1$, $f = 0.02$, $\mu = 6p_i = 0.006$, $u = \$1M$, the cost of electricity c_i is given by $c = p_i \times e_W \times \pi_W$, where e_W is the electricity consumption of the network expressed in kWh, and $\pi_W = 0.04$ is the price of electricity per kWh. For e_W, we choose $\frac{115.541 \times 10^9}{365.25 \times 24}$.[2]



(a) BTC (b) USD

Figure 9: Pool surplus path.

Our analysis distinguishes between scenarios where the miner is engaging in solo mining or participating in a pool. More specifically, we implement the following approach:

1. We choose February 10, 2021 as the starting date.
2. We randomize the time horizon by simulating $nsim$ durations following an exponential distribution with mean equal to 2 weeks.
3. For each simulation run, we simulate $nsim$ share reward payment times.
4. For each block found by an individual miner, mining alone, he will receive the block reward and the transaction fees attached to this block. For a miner in a pool, the miner receives only a fraction of the block reward and of the transaction fees, assuming a FFPS pooling scheme.

[2] https://cbeci.org/

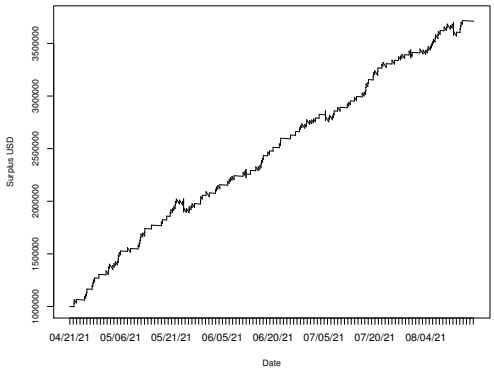


Figure 10: Miner surplus path USD

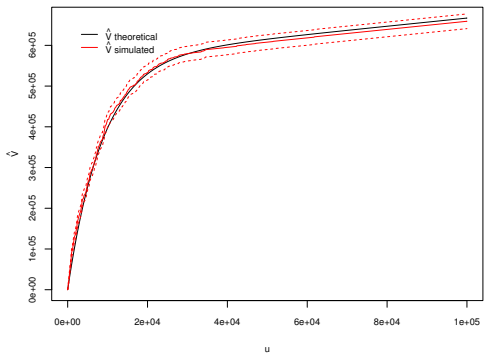
In practice, whereas the bitcoin block reward is known and fixed (at 6.25 BTC at the time of writing), the transaction fees included in a block can only be discovered after the block is appended to the blockchain. It means that the pool has to predict the future fee in order to pay the miners before the block appears. What often happens in practice, is that the fee is computed as the average fee over some short time frame, e.g. the last 24 hours [29]. This is the retained approach in our framework.

5. Combine all data to produce sample paths of the miner's surplus for different values of their initial capital u. Averaging over all iterations yields simulated values for $\hat{V}(u,t)$ and $\hat{\psi}(u,t)$.

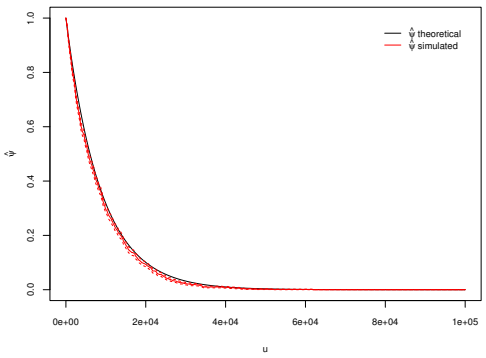
On the other hand, we compute (23) and (24) under the i.i.d. assumption with a GH model for the block rewards as calibrated in Section 3, using the moderate truncation parameter $d = 10$ (a higher number of terms in the combinations of exponentials would significantly complicate the numerical evaluation). Figures 11 and 12 display the ruin probability and expected surplus based on the theoretical formulas and the historical path method, for both a stand-alone miner and one joining a pool. One can see that the results are rather close for the two cases, and the shape as a function of initial capital u is identified rather well. This suggests that the theoretical formulas are quite useful for practical use.

5.2 Sensitivity of the GH fit under the i.i.d. assumption

Under the assumption of i.i.d. block rewards, it is interesting to see how sensitive the results are w.r.t. to the fitted GH model. For that purpose, we compare the formulas (23) and (24) to a situation, where we resample block reward values from the empirical distribution function of rewards from the period February 10, 2021 to April 21, 2021. In a sense, this may also be considered a fairer comparison than the one in Section 5.1, since now the variability of block rewards is similar. Figures 13 and 14 depict the results, where the black lines represent the formulas using the GH fit, and the red lines are simulated values under the resampling together with a 95% confidence interval (also in all the remaining plots of this section, whenever we plot simulated values we do so together with theirs 95% confidence interval). Since this data sample served as the basis for the GH fit calibration, one expects a close similarity, which is indeed the case. Consequently, the GH approximation seems sufficient for purposes of drawing conclusions for our quantities of interest.

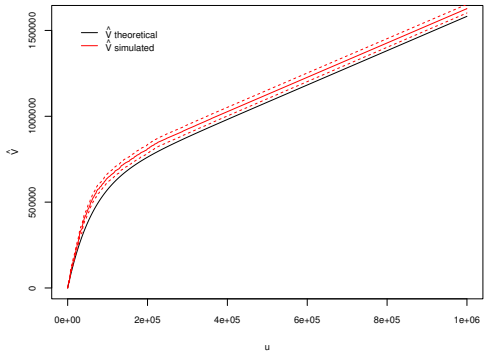


(a) Expected value of the surplus.

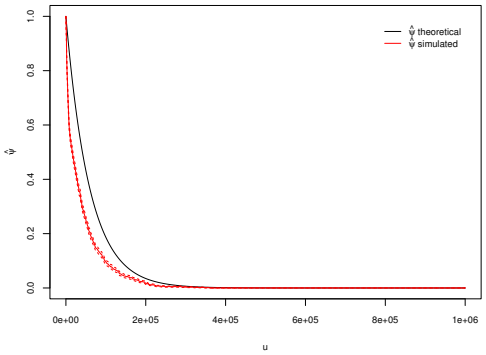


(b) Ruin probability.

Figure 11: Simulation of a historical path vs i.i.d. GH fit, miner in a pool (starting date February 10, 2021).



(a) Expected value of the surplus.

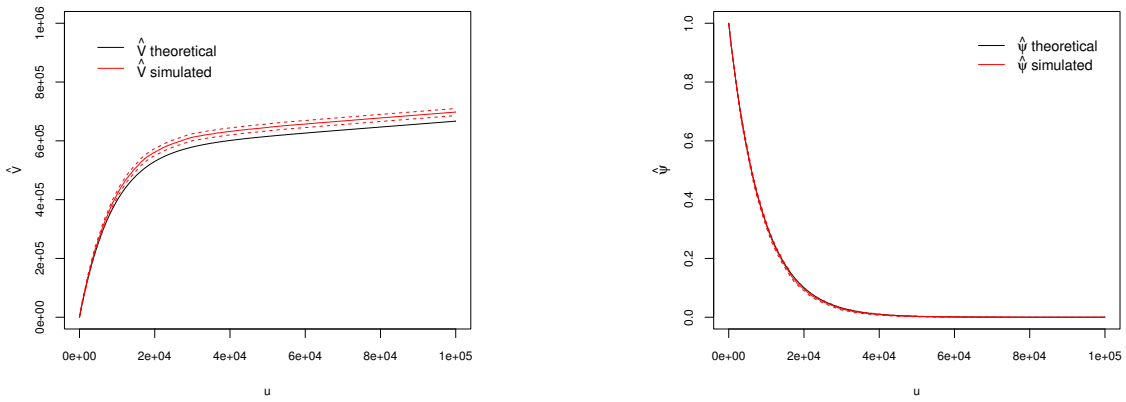


(b) Ruin probability.

Figure 12: Simulation of a historical path vs i.i.d. GH fit, miner alone (starting date February 10, 2021)

5.3 Sensitivity w.r.t. time dependence of rewards

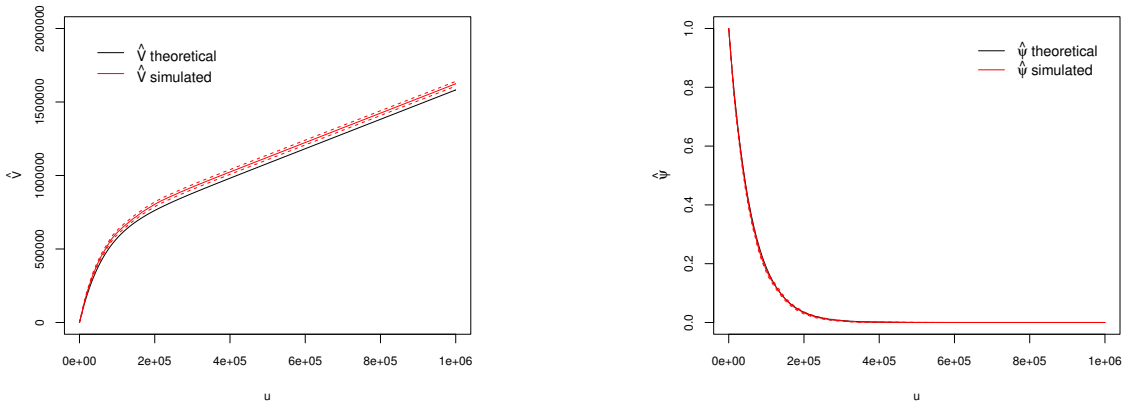
We use the ARIMA model calibrated in Section 4 to simulate trajectories for the transaction fees, as well as another ARIMA model for the bitcoin price in USD relevant for the fixed 6.25 BTC contributions (as was suggested by Azari [6]). For our time window, a calibration of the latter ARIMA model suggests an ARIMA(5,1,1) model, as seen in Section 4. The sum of the two form our block reward process, which feeds a Monte Carlo estimator of the ruin probability and expected surplus under that assumption. As previously, we fit the block rewards expressed in USD. For the miner, the price conversion seems interesting, as rewards enter less often, therefore the timing of the conversion is important and taken in account. From the pool perspective, it virtually operates in BTC for both inflows and outflows and therefore does not suffer conversion risk. Figure 15 and 16 compare the results (in red) with the formulas (23) and (24) (in black), which were derived under an i.i.d. assumption. One observes that for the ruin probability, the deviation from the i.i.d. assumption observed in the data is not relevant for the 2 weeks timeframe, and also for the expected value of the future surplus, the deviations are minor. The difference in the expected profit for large initial capital (where ruin becomes very unlikely) can be explained by the fact that for the calibration of the ARIMA model



(a) Expected value of the surplus.

(b) Ruin probability.

Figure 13: Empirical reward distribution vs i.i.d. GH fit, miner in a pool (starting date February 10, 2021).



(a) Expected value of the surplus.

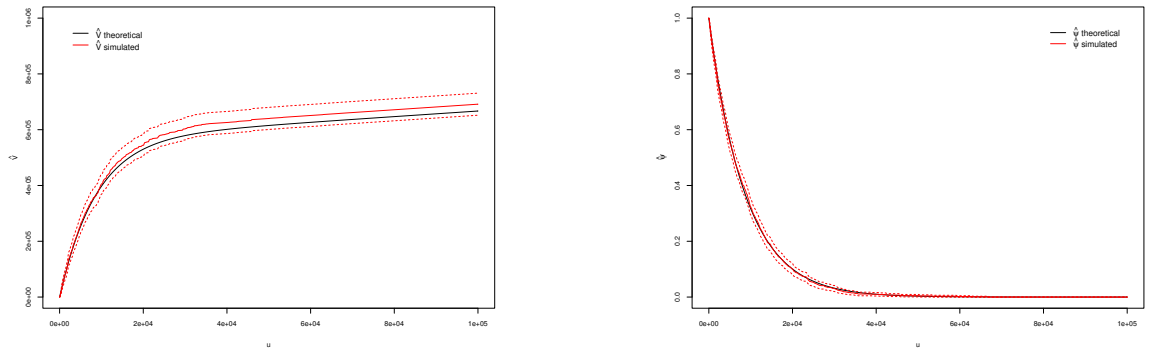
(b) Ruin probability.

Figure 14: Empirical reward distribution vs i.i.d. GH fit, miner alone (starting date February 10, 2021).

the time period is shorter (by 20%) compared to the one of the i.i.d. GH fit due to out-of-sample validation, cf. Section 4.

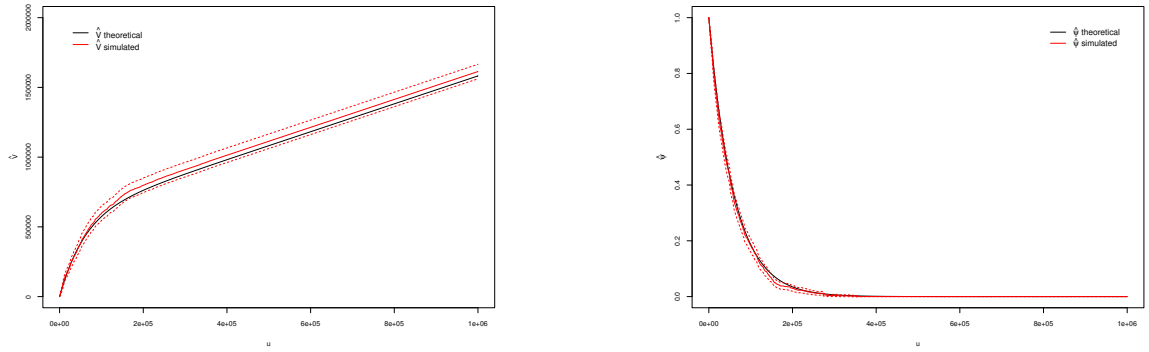
5.4 Sensitivity w.r.t. transaction fees

Transaction fees represent an additional element of randomness in the block rewards, adding to the inherent volatility of the bitcoin (fiat) price. Our objective is to illustrate the effects of including or excluding transaction fees in the reward modelling process. Figures 17 and 18 depict the changes in the ruin probability and expected surplus for miners, both solo and within a pool, when transaction fees are inlcuded or left out (that is, only the ARIMA model for the bitcoin price is simulated in the latter case). One observes that the expected surplus converges to different limits as the initial capital u grows large. In that case, the impact of ruin on the miner's surplus diminishes, and only the positive effect of higher block rewards materializes (statistically, for the period May 12, 2020 to September 16, 2021, the transaction fees mounted to 8.6% of the fixed part of the block rewards 6.25



(a) Expected value of the surplus. (b) Ruin probability.

Figure 15: ARIMA simulation vs i.i.d. GH fit, miner in a pool (starting date February 10, 2021)



(a) Expected value of the surplus. (b) Ruin probability.

Figure 16: ARIMA simulation vs i.i.d. GH fit, miner alone (starting date February 10, 2021).

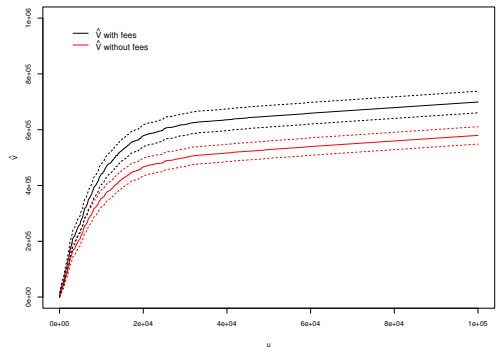
BTC per bounty). In terms of ruin probabilities, while the model including fees prevails over the one without fees, the difference is in fact very small. However, it is important to consider that the fixed block reward undergoes a scheduled halving approximately every four years. According to the current countdown[3], the next halving is expected in April 2024, reducing the fixed reward to 3.125 BTC. In Figures 19 and 20, we therefore conduct the same analysis with 3.125 BTC for the fixed part, all other factors held constant (and assuming that the fee dynamics remain unchanged after that halving). In the latter case, the differences become much more pronounced.

5.5 Sensitivity w.r.t. electricity costs

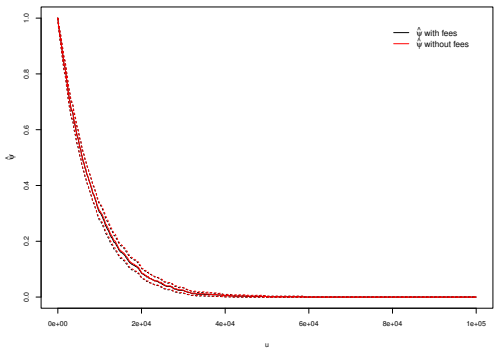
Finally, let us consider the sensitivity of the results to electricity costs under the ARIMA model. Figures 21 and 22 show the ruin probability and expected surplus as a function of the price of electricity for a solo miner and a mining pool participant for a fixed initial capital of $u = 100,000$. Such a graph can help to identify upper bounds for affordable electricity prices needed to ensure specific target levels of ruin probability or expected surplus for a given level of u.

Notably, the results reveal a substantial difference in the ruin probability between individual miners and those participating in a pool, emphasizing the risk mitigation benefits of pooling (already observed in [1]). Additionally, this quantifies the influence of increased electricity costs, underlining the exposure

[3]See, for example, https://www.nicehash.com/countdown/btc-halving-2024-05-10-12-00.

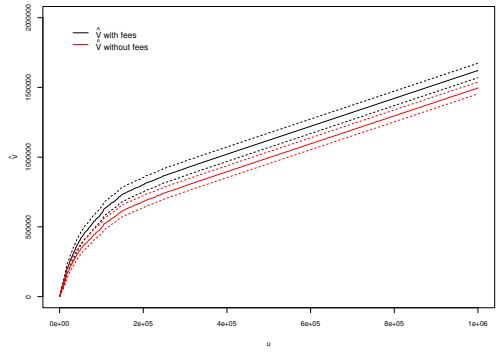


(a) Expected value of the surplus.

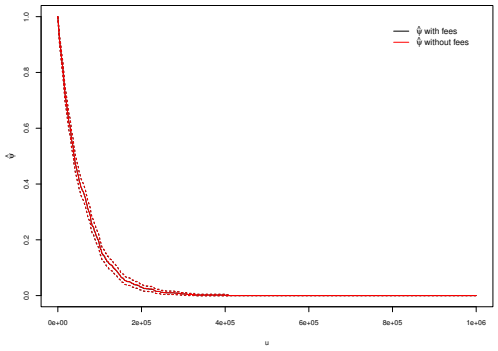


(b) Ruin probability.

Figure 17: Sensitivity to presence/absence of fees, miner in a pool.



(a) Expected value of the surplus.



(b) Ruin probability.

Figure 18: Sensitivity to presence/absence of fees, miner alone.

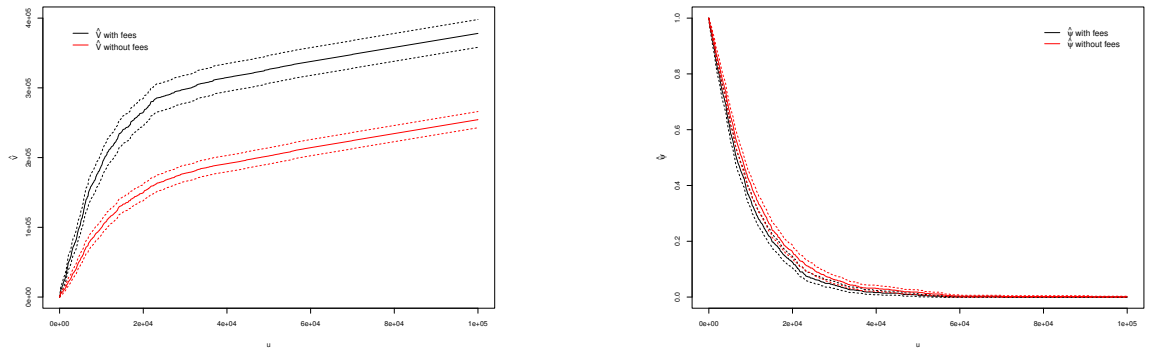
and vulnerability of 'Proof-of-Work' cryptocurrencies to energy-related crises. Such considerations may (even economically) motivate the transition to alternative, less energy-intensive consensus protocols. For instance, Ethereum, the second-largest cryptocurrency, shifted from 'Proof-of-Work' to 'Proof-of-Stake' on September 15, 2022, reducing its energy consumption by a remarkable 99.95%.[4]

6. Conclusion

In this paper, we have delved deeper into the framework for analyzing bitcoin mining from the perspective of risk and profitability. Previously, quantifying the choices available to miners, such as entering a mining pool and selecting the most suitable one, relied on formulas based on assumptions involving combinations of exponential distributions. In this work, we introduce a straightforward and efficient approach for fitting real-world data to these distributions, allowing us to apply theoretical results in practical mining scenarios. The remarkable flexibility of the resulting method enables us to explore shapes that are typically challenging to achieve using other approaches. The code for this work is available upon request.

Furthermore, we have explored the stochastic nature of block rewards, breaking down their variability into two key components: price volatility and the inclusion of transaction fees. Our findings highlight

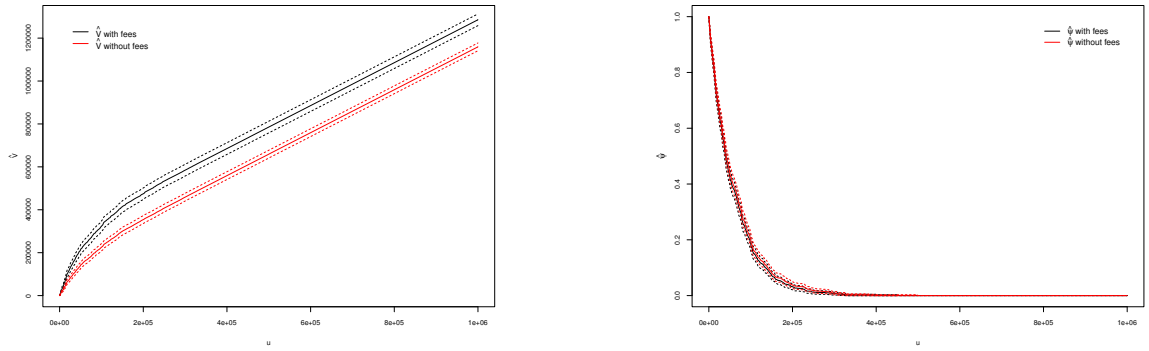
[4]Source: https://ethereum.org/en/upgrades/merge/ (last accessed on 13/09/2023).



(a) Expected value of the surplus.

(b) Ruin probability.

Figure 19: Sensitivity to presence/absence of fees, miner in a pool, block reward 3.125 BTC.



(a) Expected value of the surplus.

(b) Ruin probability.

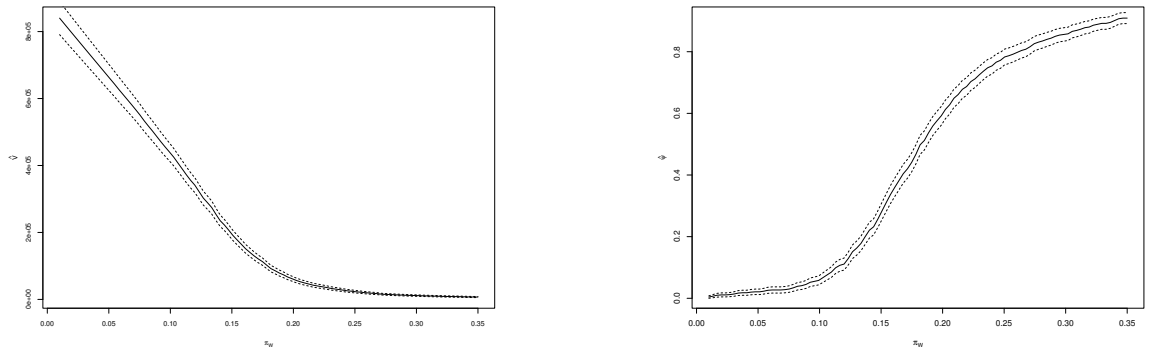
Figure 20: Sensitivity to presence/absence of fees, miner alone, block reward 3.125 BTC.

the growing importance of incorporating fees in modeling, especially with the scheduled halving of the fixed block reward.

By expressing transaction fees as a time series, we can simulate our key metrics of interest — namely, the expected surplus and ruin probability — and compare them to theoretical results. This analysis confirms that the formulas derived in [1, 3], originally under an i.i.d. assumption, stand up well when compared to the outcomes derived from modeling strategies that incorporate time dependency or use empirical data more generally.

Finally, we emphasize the sensitivity of our key metrics to the inclusion of transaction fees. While the differences in magnitude are currently still small, we anticipate significant shifts in results after the next halving, making fees a more substantial factor in modeling. Our analysis also underlines the substantial impact of electricity costs on a miner's profitability and ruin probability. These insights contribute to the ongoing debate surrounding the viability of 'Proof-of-Work' consensus protocols in an increasingly energy-constrained world.

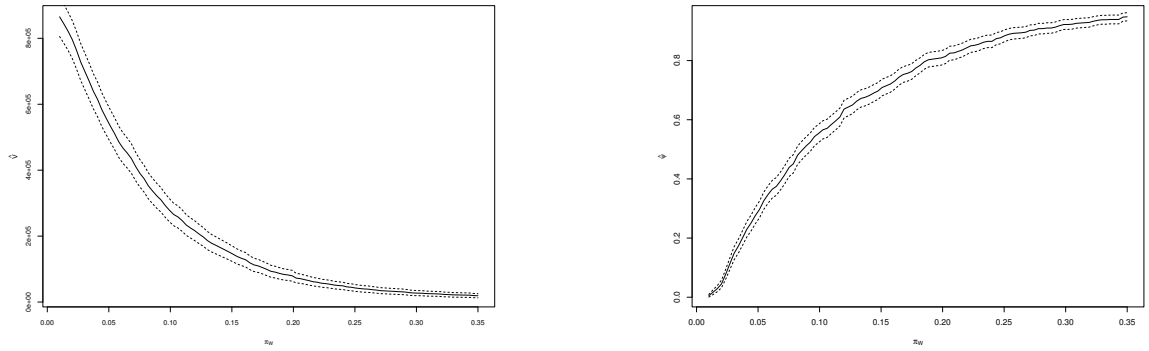
Statement. All authors declare that they have no conflicts of interest.



(a) Expected value of the surplus.

(b) Ruin probability.

Figure 21: Sensitivity to change of electricity price π_W, miner in a pool (starting date February 10, 2021).



(a) Expected value of the surplus.

(b) Ruin probability.

Figure 22: Sensitivity to change of electricity price π_W, miner alone (starting date February 10, 2021).

References

[1] Hansjörg Albrecher, Dina Finger, and Pierre-O Goffard. Blockchain mining in pools: Analyzing the trade-off between profitability and ruin. *Insurance: Mathematics and Economics*, 105:313–335, 2022.

[2] Hansjörg Albrecher, Hans U Gerber, and Hailiang Yang. A direct approach to the discounted penalty function. *North American Actuarial Journal*, 14(4):420–434, 2010.

[3] Hansjörg Albrecher and Pierre-Olivier Goffard. On the profitability of selfish blockchain mining under consideration of ruin. *Operations Research*, 70(1):179–200, 2022.

[4] João Almeida, Shravan Tata, Andreas Moser, and Vikko Smit. Bitcoin prediciton using ANN. *Neural networks*, 7:1–12, 2015.

[5] Andreas M Antonopoulos. *Mastering Bitcoin: unlocking digital cryptocurrencies.* " O'Reilly Media, Inc.", 2014.

[6] Amin Azari. Bitcoin price prediction: An ARIMA approach. *arXiv preprint arXiv:1904.05315*, 2019.

[7] Nigel G. Bean, Mark Fackrell, and Peter Taylor. Characterization of matrix-exponential distributions. *Stochastic Models*, 24(3):339–363, aug 2008.

[8] Robert F. Botta, Carl M. Harris, and William G. Marchal. Characterizations of generalized hyperexponential distribution functions. *Communications in Statistics. Stochastic Models*, 3(1):115–148, jan 1987.

[9] Jamal Bouoiyour, Refk Selmi, Aviral Kumar Tiwari, Olaolu Richard Olayeni, et al. What drives Bitcoin price. *Economics Bulletin*, 36(2):843–850, 2016.

[10] Miles Carlsten, Harry Kalodner, S. Matthew Weinberg, and Arvind Narayanan. On the instability of Bitcoin without the block reward. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 154–167, 2016.

[11] Bladt, Mogens and Nielsen, Bo Friis. *Matrix-Exponential Distributions in Applied Probability*. Springer-Verlag GmbH, May 2017.

[12] Szökefalvi-Nagy, Béla. *Introduction to real functions and orthogonal expansions*. Oxford University Press, 1965.

[13] Pavel Ciaian, Miroslava Rajcaniova, and dâĂŹArtis Kancs. The economics of BitCoin price formation. *Applied Economics*, 48(19):1799–1815, 2016.

[14] D. R. Cox. A use of complex probabilities in the theory of stochastic processes. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(2):313–319, apr 1955.

[15] Daniel Dufresne. Fitting combinations of exponentials to probability distributions. *Applied Stochastic Models in Business and Industry*, 23(1):23–48, 2007.

[16] David Easley, Maureen O'Hara, and Soumya Basu. From mining to markets: The evolution of bitcoin transaction fees. *Journal of Financial Economics*, 134(1):91–109, 2019.

[17] Mark Fackrell. Fitting with matrix-exponential distributions. *Stochastic Models*, 21(2-3):377–400, jan 2005.

[18] Mark Fackrell. An alternative characterization for matrix exponential distributions. *Advances in Applied Probability*, 41(4):1005–1022, dec 2009.

[19] Bernard Hanzon and Finbarr Holland. Non-negativity analysis for exponential-polynomial-trigonometric functions on $[0,\infty)$. In *Spectral Theory, Mathematical System Theory, Evolution Equations, Differential and Difference Equations*, pages 399–412. Springer Basel, 2012.

[20] Paraskevi Katsiampa. Volatility estimation for bitcoin: A comparison of GARCH models. *Economics Letters*, 158:3–6, 2017.

[21] Zongxi Li, A. Max Reppen, and Ronnie Sircar. A mean field games model for cryptocurrency mining. *Management Science*, 2023.

[22] X. Sheldon Lin and Gordon E. Willmot. Analysis of a defective renewal equation arising in ruin theory. *Insurance: Mathematics and Economics*, 25(1):63–84, 1999.

[23] X. Sheldon Lin and Gordon E. Willmot. The moments of the time of ruin, the surplus before ruin, and the deficit at ruin. *Insurance: Mathematics and Economics*, 27(1):19–44, 2000.

[24] 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)*, pages 339–343. IEEE, 2018.

[25] Johnnatan Messias, Mohamed Alzayat, Balakrishnan Chandrasekaran, Krishna P Gummadi, Patrick Loiseau, and Alan Mislove. Selfish & opaque transaction ordering in the Bitcoin blockchain: the case for chain neutrality. In *Proceedings of the 21st ACM Internet Measurement Conference*, pages 320–335, 2021.

[26] Malte Möser and Rainer Böhme. Trends, tips, tolls: A longitudinal study of Bitcoin transaction fees. In *International Conference on Financial Cryptography and Data Security*, pages 19–33. Springer, 2015.

[27] Sabina Rossi, Ivan Malakhov, and Andrea Marin. Analysis of the confirmation time in proof-of-work blockchains. *Available at SSRN 4031244*.

[28] Robert H Shumway, David S Stoffer, and David S Stoffer. *Time series analysis and its applications*, volume 3. Springer, 2000.

[29] Luxor Tech. Different Bitcoin mining pool payment methods PPS vs FPPS vs PPLNS vs PPS+), 2018. Accessed: 2022-06-17.

[30] Enrico Tedeschi, Tor-Arne S Nordmo, Dag Johansen, and Håvard D Johansen. On optimizing transaction fees in Bitcoin using AI: Investigation on miners inclusion pattern. *ACM Transactions on Internet Technology (TOIT)*, 2022.

[31] Larry Wasserman. *All of nonparametric statistics*. Springer Science & Business Media, 2006.

[32] I Made Wirawan, Triyanna Widiyaningtyas, and Muchammad Maulana Hasan. Short term prediction on Bitcoin price using ARIMA method. In *2019 International Seminar on Application for Technology of Information and Communication (iSemantic)*. IEEE, sep 2019.

[33] Angela Zeiler, Rupert Faltermeier, Ingo R Keck, Ana Maria Tomé, Carlos García Puntonet, and Elmar Wolfgang Lang. Empirical mode decomposition-an introduction. In *The 2010 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2010.