International Centre for Innovation
Technology and Education

Solvay Brussels School
Economics & Management

ULB

FORECASTING SHORT-TERM TRANSACTION FEES

ON A SMART CONTRACTS PLATFORM

Authors

Charles Hoffreumon
Solvay Brussels School of Economics and Management, Université libre de Bruxelles, iCite - email: Charles.Hoffreumon@ulb.ac.be

Nicolas van Zeebroeck
Solvay Brussels School of Economics and Management, Université libre de Bruxelles, iCite - email: Nicolas.van.Zeebroeck@ulb.ac.be

iCite Working Paper 2018 - 028

Forecasting short-term transaction fees on a smart contracts platform

Charles Hoffreumon, Nicolas van Zeebroeck
iCite, Solvay Brussels School of Economics and Management, ULB

September 6, 2018

Abstract

In the present article, we analyze the transaction fees market on smart contracts-enabling blockchains. On such systems, as opposed to traditional on-premise and cloud computing solutions, users are effectively competing for computational resources through an auction for priority. This paper proposes a way to estimate the bid one has to offer to have a transaction included in the next block. This method outperforms naive bidding (bidding the optimal value of the last block) if the user is realistically "impatient" to have a transaction processed. It also shows that users collectively spend several million of dollar every years for transaction fees that could be avoided without degrading the service received. This is this "waste" we seek to reduce through our forecasting method.

Keywords: *Blockchain, Auctions, Forecasting, Transaction Fees, Smart Contracts*

JEL: *C22, C53, D44, M15, 039*

1 Introduction

This paper tackles a problem that arises from the emergence of blockchain-based smart contracts. Such contracts can be conceived as "programs" that live on distributed computer systems and regulate some aspects of the interactions between people (the most simplistic being monetary transactions subject to uncertainty).
While such systems are technically viable today and instantiated by, for example, the Ethereum network, there are still few inquiries into the costs of transaction arising from them. It is however, as we will argue in the present article, an important question that may have an impact on a widespread adoption of smart contracts systems in companies. IT practitioners are indeed well acquainted with the fact that costs in IT systems must be managed with care in order not to become major costs centers with negative returns.
The cost of using blockchain comprises two components: the first one is the erosion of the value of the tokens due to "mining". Indeed, at constant demand, the increase of the supply of tokens will decrease the value of an individual token. This cost does not depend on the use or not of the chain to perform transactiona or execute contracts. On the other hand, the transaction fee, expressed in the native token of the chain is the main marginal cost of carrying a transaction. It is proposed by the user and will determine how fast and for how much the transaction of contract execution will be written on the chain. As such, it is the main component of the cost of executing a contract of registering a transaction. Forecasting it in the short term is therefore of paramount importance to optimize costs of decentralized applications. Indeed, if an error of

forecast on a single, non-recurring simple transaction will usually cause a loss (or regret, see below) of a few cents, it will accrue fast for large application that require the execution of complex contracts several times per hours. Any non-trivial decentralized application (be it a token based on smart-contract, a decentralized exchange or a full-fledged set of contracts proposed as a solution by a startup) will therefore benefit from a deeper understanding of how transaction fees are forms but also, and maybe more importantly, from the ability to recommend the bids to be done in the future. Investing in such systems entails therefore taking into account the transaction costs that will be incurred as well as their potential variability and uncertainty. This paper will aim at helping practicioners to forecast the minimum fee to be offered to have their transaction executed in the short run.

With smart contracts, the price of computing is determined by market conditions and the current demand for the "shared resources" of the blockchain. It is based, as will be explained in Section 2, on a system of auction, which, as will be explicited in Section 4.1.1 makes it frequent to pay more than one should or would want at any point of time. From the data collected, we can show that, over a period of 2 months, users paid around $114,507.16$ ETH or around $47M\$$[1] in fees for no additional service. This "waste" arises from the fact that the behavior of bidders on this market is still mostly based on heuristics and relatively unsophisticated recommendation systems that usually takes into account broad timespan to estimate an order of transaction fee (usually with little explanation about what it is trying to optimize).

This is the problem that the present article proposes to describe. We then present a step in the direction of its "solution". We do so by considering the optimal bid in this repeated auction as a time series that can be forecasted through standard econometric models. While not an entirely new idea (such models have been used for predicting the auction-based electricity price in some market as is shown in 3 it is, to the best of our knowledge, the first time it is used in the context of transaction fees in blockchain markets. We show that this method allows for better predictions in the short run than the naive forecast (produced by the assumption that the optimal price in the next auction is the same as the one in the present auction) in terms of expected regret for the bidder.

This article starts by describing the market of transaction fees in bids for execution of smart contracts in Section 2. It then briefly reviews the literature in Section 3 before clarifying the concepts that are then used in the empirical part of the paper in Section 4. We then decribe the collection method and highlight interesting features of the data in Sections 5.1 and 5.2. It is worth mentioning that the data was collected by the author of the article and are made available to the scientific community, along with the code needed to retrieve an updated version of it. It is, to the best of our knowledge, the first time this data receives such a detailed examination in a scientific article. We then proceed with the analysis of the properties of the econometric model in the case studied here in Section 6 and proceed to assess the performance of the forecast produced by applying this method in a rolling fashion.

We show that this method outperforms naive forecast if the user is "impatient enough" (these concepts are defined in Section 4.1.1). Finally, we conclude in Section 7 by highlighting the limitations of such an early work on a topic, providing leads for future research and recontextualizing the findings in the broader framework of an endeavor to turn smart contracts a viable tool for businesses.

2 The Transaction Fees Market for Smart Contracts

In this section, we briefly describe how transaction fees are formed on the largest smart contract platform in activity today, the Ethereum blockchain. Like most blockchains today, it can be conceived as an auction for priority as detailed in (Huberman et al., 2017). Subsection 2.1

[1] At the exchange rate of April 22, 2018

describes how this happens and subsection 2.2 provides the details about the tools available to people wanting to process code on the blockchain in order to estimate the bid they have to submit in function of their preferences.

2.1 Transactions Fees and Gas Price on Ethereum

2.1.1 A short note on blockchains

We start by defining the concepts about blockchains that will be relevant for the sequence of this analysis. At its core, a blockchain is a decentralized ledger, a way to keep a record of informations such that everybody involved in the system (both "providers" and "consumers") know them to be true and that is extremely hard to tamper with (meaning that it is nearly impossible to modify information that was previously recorded). The system is maintained by "miners", the providers of the system that are in charge of collecting, and including the information sent by the "users" in the chain. They do so through a costly process of decentralized lottery and verification of the whole chain of information. A general reference about the workings of these systems can be found in the foundational (Nakamoto, 2008).

2.1.2 Transaction Fees

When a user wants to record an information in the system, or execute a contract, she will send a notification to one or several miners that are going to broadcast it to the network. Along with this notification, she will propose a certain amount of tokens to the miner who will include the "transaction" in the chain. In ethereum, the value of these fees are often expressed in GWei (giga-Wei, 10^9 wei where $10^{18} wei = 1 ETH = 635.20\$$[2]). The miners then participate to the (costly) lottery for the right to include a collection of transactions into the chain. The winner receives the transaction fees of all the transactions comprised on the block he formed.
This lottery happens at a regular rythm and depends on the system considered. For reference, on the currently largest blockchain, bitcoin, a new block is "mined" (or created) approximately every 10 minutes while on the ethereum blockchain, studied here, a new block is formed every 15 seconds.
Conceptually, the transaction fees offered by users can be conceived as "bids" in an auction for priority. Offering a higher fee is supposed to increase the speed of inclusion of the transaction into the chain (The model in Huberman et al. (2017) makes it explicit in section 3).
As we will show in Section 5.2.1, the ethereum blockchain is very often saturated. This mechanism allows therefore the miners to allocate processing power to the most "motivated" users. Such, at least partial, saturation is a necessary feature in the incentive scheme constituting a blockchain and this auction mechanism, be it for simple transaction recording or more elaborated execution of smart contracts on the blockchain makes it complicated to have absolute a-priori near-certainty about either their cost or the execution time.

2.1.3 Smart Contracts

One of the most important uses of blockchain systems is their enabling of so-called smart contracts. The concept of smart contracts, originally popularized by (Szabo, 1997) and implemented by the Ethereum project (Buterin, 2014), implies executing code on a decentralized network of computers. Indeed, while the first blockchains (such as bitcoin) where mostly designed to record pure unconditional transactions, the addition of a semantic allowing for checking conditions before enacting a transaction makes it possible to record *contracts* or sets of conditional transactions that are executed on demand and, if a set of criteria is met, writes a transaction

[2]Valuation on April 22, 2018

on a chain. This idea in economics, was discussed in (Varian, 2010) although there was no mention of the fact that such contracts could be decentralized. Decentralization is interesting in a number of use cases, such as for critical middleware systems between companies or for public applications needing the utmost transparency.
The concrete examples of use cases for smart contracts are legion. The ones given here are by no mean an extensive survey of this vast scope.
First and foremost, and probably owning to the original purpose of blockchain systems, quite a number of smarts contracts today are designed to enable cryptocurrencies. These contracts set the rules about the creation and conditions of spending of such currencies. This is such a popular application that a standard interface, the ERC20, has been created to standardize them.
Other popular contracts are actually "repositories" or wallets of tokens for organizations requiring multiple signature for the use of funds. The security is enforced by conditions placed on the identity and circumstances allowing one transfer of fund to be enacted.
A last example are sets of contracts that, together, constitute "decentralized applications". These are contracts calling and executing one another to perform complex tasks. These are heralded as an enabler for possible future "decentralized autonomous organizations", which would be organizations ruled solely by a set of rules by which the "participants" choose to abide. Their viability is, however, not entirely and convincingly proven today and the early attempts at creating such organizations have had underwhelming results so far (such organizations are briefly discussed in, among other, (Davidson et al., 2016)).
Smart contracts may be compared to cloud computing solutions in their enabling of some applications out of the premises of a company. While both seek to fulfill similar goals, smart contracts are fundamentally different from classical cloud computing as the code is simultaneously ran on a large array of servers (belonging to the miners that we describe below) simultaneously. Moreover, since users compete in an auction (as explained in section 2.1.1), either the cost or the execution time (or both) depend on the valuation of all the other users for the service as well as the backlog of transactions waiting to be processed. Indeed, during periods of heavy load, it is expected *by design* to see price surges and the other way around. This is a significant departure from classical cloud computing offers as, even if surges in price exist in the offers of most major cloud infrastructure providers, they are usually not based on auctions but rather on different price tiers. This is also what causes the problem (the necessary trade-off between higher cost and higher speed of execution) that we endeavour to analyze in the present article.
Another difference between the use of smart contracts and cloud computing is the fact that operation costs in smart contracts systems are, and must be *by design*, tied to a token (sometimes, as in the ethereum case, also a cryptocurrency). Relying heavily on smart contracts to perform run-of-the-mill computation in a company would therefore increase the volatility of an organization's IT costs as the exchange rate of cryptocurrencies is currently extremely volatile and misunderstood (an attempt to find the driver of the exchange rate vis-à-vis the US dollar is presented in (Kristoufek, 2015)).
In this paper, we are solely concerned with the first kind of volatility (the one arising from the auction for priority). The study of exchange rate (or cryptocurrency prices) is, however, a very important topic in its own right for anyone interested in managing costs of smart contracts-based solutions.

2.2 Information and Heuristics on the Current Market Price

Using (or "executing" as it is often called) a smart contract thus involves bidding a certain amount of tokens as a reward for the miners. This task, as it was already mentioned, essentially amounts to trading-off the desired speed of execution for higher transaction fee.
In order to assess the bid she is willing to offer, the user have access to different information

sources. First and foremost, every transaction included on the chain contains the transaction fee that was offered. This information is the one we will use in the remainder of this paper.
Moreover, the set of transactions waiting to be executed, along with their proposed transaction fees is available to the user. While users can set up a so-called "node" (that is, a computer part of the ethereum network that downloads a copy of the chain), the procedure is rather complex for the average user and requires an important amount of disk space. Most users may want to use freely available websites providing this information, such as Etherscan. It may be worth noting that, with both methods, there is the risk of not having the exhaustive list of pending transactions. Indeed, as explained in section 2.1.1, transactions are broadcasted through the miners network on a peer-to-peer basis and there is some latency before a transaction reach any node in the network (the websites usually rely on a dedicated node, but it is just one of the many nodes in a vast network).
Most users, however, will rely on heuristics or third parties providers in order to set up their bid. We will mention two sources of informations on the current transaction fees: the first are the client software for ethereum, such as *geth*, that usually implement a method for estimating the current transaction prices. The second are dedicated websites. The most famous is the ETH Gas Station, which "fit*[s]* a Poisson regression model that estimates the expected number of blocks it will take for a transaction to confirm based on the gas price and the amount of gas used by a transaction based on data from the last 10,000 blocks"[3] every 100 blocks.

3 Literature Review

The present paper is, to the best of our knowledge, the first attempt at predicting the prices on blockchain transaction fees markets in the academic literature. This is probably due to the recency of the topic. There are, however, papers analyzing blockchains under the scope of game theory and market design, as well as attempts to forecast other prices through the same techniques as the ones applied here.
The economic literature on the design of blockchain is a fast-growing corpus. While the first attempts (such as (Babaioff et al., 2012)) were dedicated to modeling problems arising in the coordination between the miners, there has recently been several attempts at modelling the markets for transaction fees or the equilibrium of the system as a whole.
The most notable paper to date is (Huberman et al., 2017) where the authors identify an equilibrium by trading-off the waiting time for higher bids in a bayesian game. While the paper aims at making a point about the design of the system, the model described there is useful in its own right to help understand the dynamics at play in the problem at hand in the present paper. It does not, however, provide a tractable way to forecast the equilibrium price at any time as it remains too generic on the shape of the waiting function arising from blockchain systems from the point of view of the user.
The second paper worth of mention is (Ma et al., 2018) in which the authors, beside looking for an equilibrium in the transaction fee market, model specifically the arms race between miners to increase their hash-generating capabilities. It is, however, not possible to derive a closed form expression of the next equilibrium price from that paper either.
While there are already a number of papers analyzing the impact of cryptocurrencies as an investment assets (e.g. (Brière et al., 2015)), there has been fewer attempts to determine structurally what constitutes the price of cryptocurrencies (with the notable exception of (Kristoufek, 2015), previously mentioned). The present paper aims at demonstrating a technique to estimate the transaction fee at a very short time horizon (between 15 and 30 seconds). While the papers cited above mainly attempt at quantifying the variation of the cryptocurrency itself, the present one aims in fact at preventing wasting too much of that currency on operations on the

[3] Quote from the ETH Gas Station FAQ

blockchain.
The model we will use throughout this paper and detailed in section 4 is the one of a repeated auction for multiple goods. Such models have been studied in, among other, (Goldberg et al., 2001) and then (Goldberg et al., 2006) although never in a repeated setting. Another significant difference is that this paper attempts at defining optimal auction mechanisms in order to maximize (or optimize) the revenue of the auctioneer while, in the present paper, we will take the point of view of a bidder in the auction.
On the methodology side, the ARMA-GARCH model is a staple of financial forecasting. The GARCH-based models have been amply studied since their inception in (Bollerslev, 1986). Two ARMA-GARCH applications on empirical data were presented in (Liu Shi, 2013) and (Solibakke, 2001). It has, however, never been applied to predict prices in the repeated auctions on blockchains transaction fees markets.

4 Model

4.1 A description of repeated auctions with several goods

The market for transaction fees is supposed to be a repeated auction. Each time a user wants to run the code contained in a contract, he broadcasts a proposed transaction along with a transaction fee he is willing to pay for the transaction to be written on the chain. The miners observe all the pending transactions and select the highest bidding to be included in the next block.
It can be shown that, if information was perfect and the users knew who was playing as well as their type, the optimal bid, for a player for whom the value of executing a transaction is high enough, would be v_{N-K+1} where N is the number of players, K the number of transactions included in each block and v_i the valuation of player's i transaction (where players are organized by ascending order of valuation). To see that, one can just consider it as the iteration of the identification of the equilibrium with perfect information of a first price sealed bid auction with several goods.
However, this is not what we observe in the data (see Section 5.2). Indeed, the bids contained in different blocks are sometimes order of magnitudes apart.
A possible explanation is that users ignore the valuation distribution or do not have the possibility to compute the expected lower price and rely on heuristics to form their bid. This would certainly explain the "levels" we observe in the data (bids tend to be concentrated around some values, as will be seen in Section 5.2.2).
As we will see in the remainder of this paper though, the minimum bid to be included in a block may be partially estimated based on the lowest bids for transactions in the blocks preceding it. This is due to the fact that, as a block wipes clean the top of the distribution of pending transactions, the top of the remaining transaction pool becomes a reference point to infere the minimum of the next block.

4.1.1 Empirical Total Regret

In this article, we consider the problem of a user wanting to execute a smart contract by writing a transaction on the chain. We assume that this user derives a "high enough" utility from executing the contract and is only concerned about not incurring too much delay or overpaying for something he could get for less (we will defined those terms later). The likelihood of these assumptions come from the peculiar nature of the auction at hand. It differs from the most common types of auctions studied in economics, such as the traditional Second Price, First Bid on two major aspects.
The first one is the fact that, in the present setting, several bidders end up with the "prize".

Indeed, as the data shows, a block routinely contains more than 100 transactions where each of the users having broadcasted these transactions will actually pay their bid, akin to a first price sealed bid auction.
The second one is the fact that, if a user looses the auction at one point in time, her bid is carried over to the next rounds where she stands a chance to have her transaction executed. This is what allows us to adapt the notion of Missed Opportunity to Win from (Engelbrecht-Wiggans Katok, 2007) in order to measure it and trade it off for a higher chance to get included in a block sooner at the cost of a higher Money Left on the Table (or $MLoT$, as it is called in the cited paper).

Definition 1 *Let agent i be characterized by $\lambda_i \in R^+$, her cost of waiting 1 block to have her transaction or contract executed. We define the* Empirical Total Regret *of agent i as:*

$$ETR_i(b_i) = \text{MLoT}_i(b_i) + \lambda_i D(b_I) \tag{1}$$

Where b_i is the bid that agent i submits, and D is the delay of transaction due to underbidding in the block. $MLoT$ is the difference between bid b_i and the minimum value a user could have offered and still be included in the block in which he was included.

There are several remarks to be made about this "loss function". First and foremost, it can be remarked that this "loss function" does not take into account the utility one derives from having one transaction executed in a block. Indeed, as previously mentioned, the regret is a notion that is observed by the user *ex post*. In this case, we assume that the utility derived from processing a transaction or executing a contract is high enough to justify the bid and that the user is solely looking at her *a posteriori* optimal bid.
The second point that can be made is regarding the optima of the $ETR(b_i)$. One can easily observe that the maxima are located at $b_{it} = \infty$ and $b_{it} \in R^-$ where either the Regret or the waiting time tend to $+\infty$. Moreover, this function only admits one global minimum, which is characterized by $b_{it} = \underline{p_t}$ meaning that, if the agent makes an offer during the auction for a block at time t that corresponds to the lowest accepted offer in that block, then her regret is minimized. This is what leads us to look for a way to forecast this minimum price of the next block.
The optimal bid, according to this measure, is unique for each block (bidding the minimum fee on the next block). On the other hand, the parameter λ is crucial: indeed, it will determine which regret is prefered. A λ close to 0 will mean that a user only cares about not overpaying and does not care about waiting any time for the transaction or contract to be processed. On the contrary, a high λ implies a high cost of waiting. The user will then be willing to accept paying a high value above the minimum transaction fee of the block for the certainty of being included in the next block or one shortly afterwards.
The dimension of λ is therefore in unit of money per unit of time. In the assessment of our forecast and its comparison with naive forecast, we will actually look for the value of λ around which one method of forecasting becomes more interesting than the other.

4.2 Econometric Estimation

As mentioned a the onset, the goal of this article is to propose a simple technique to help a player estimate the optimal bid for the next block. We start from the idea that the bidder has an aversion for both waiting and overpaying to have his transaction fulfilled and that his utility for realizing the transaction is positive at all possible bid. A challenge facing this user is the difficulty to assess the bid he should submit to have a chance to get into the next block. We argue that, because most blocks are full or nearly full, the minimum transaction fee of each block is not completely random and depends on past values of the same indicator. We therefore

consider the time series consisting of the minimum transaction fee for each block to estimate the next point in the series. We use classical tools from the well-researched field of Time Series studies to perform such forecast.
The model selected here is a classic ARMA and GARCH model. We make the assumption that a good part of the information about the next block minimum transaction fee is actually contained in the evolution of the prices in the previous blocs. We will see in Section 5.2.3 that, instead of working directly with the minimum fee, it is more convenient to apply the monotone transformation of taking the difference of the logarithms and to analyze the resulting series. As such, $\underline{p_b}$ denoting the minimum fee in block b, we will have:

$$y_b = log\left(\underline{p_{b+1}}\right) - log\left(\underline{p_b}\right) \tag{2}$$

To forecast the mean (and hence our point estimate), we use an $ARMA(p,q)$ process. In such a process, the mean of our forecast for block b can be modeled as:

$$Y_b = c + \sum_{i}^{p} \varphi_i Y_{b-i} + \varepsilon_b + \sum_{i}^{q} \theta_i \varepsilon_{b-i} \tag{3}$$

(for the details, see e.g. (Hamilton, 1994)). As for the $GARCH(r,s)$ model, that we will use to forecast the variance, it is described in (Bollerslev, 1986) as: $\varepsilon_b|\Psi_{b-1} \sim N(0, h_b)$
$h_b = \alpha_0 + \sum_{i=1}^{s} \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^{r} \beta_i h_{t-i}$ We will therefore reused the conditional variance estimated by the GARCH to estimate the next mean (and hence our best estimate for the forecast). For such models applied specifically to economic forecasting problem, we advise the reader to refer to e.g. (Andersen et al., 2006)).

5 Forecasting the minimum transaction fee in the next block

5.1 Collection Method

The data to carry this analysis was extracted through scripts querying an Ethereum node. The scripts are available upon request. The data was extracted between March 18 and April 12 2018 and we targeted transaction fees in blocks mined between January 15 and March 15 of the same year. In total we have information on 360.000 blocks.
On a technical note, the scripts make repeated calls to the JSON interface of the *geth* blockchain client. The client was previously allowed to fully synchronize with the network (with the "fast sync" option). In order to replicate the extraction, one has to configure an ethereum node, sync it and then run the Python scripts provided in complement to this article. A simple parameter system allows to extract other blocks if need be.
We mostly extracted the "gas price" (which is the level of price at which the whole of the code executed will be priced). We are confident this is the most relevant indicator of a transaction price as it is independent of the quantity of computation requested by the user and reveal the "motivation" of the user to see her transactions and corresponding contracts executed. In our simple model, miners observing a large transaction (one that is going to consume a large share of the "gas" available in the block) offering a high gas price and a smaller transaction offering a smaller gas price will choose the first one.
Moreover, we do not consider the transaction fees for internal transactions (transactions that are not recorded on the chain but happen as a result of contract calling one another) since those "transactions" will be priced at the same price level as the transaction at the origin of the call.

Finally, we also do not consider transactions in the so-called "uncle" blocks[4].
The extraction is based on the information available to a node located in Belgium on March 18, 2018. It is possible that it does not reflect all the information on the chain. However, this is mitigated by the fact that the data was extracted for dates relatively far in the past in terms of blocks. We are therefore confident that the transactions considered in the following study constitute the largest part of the data available for these blocks.

5.2 Data

5.2.1 Blocks

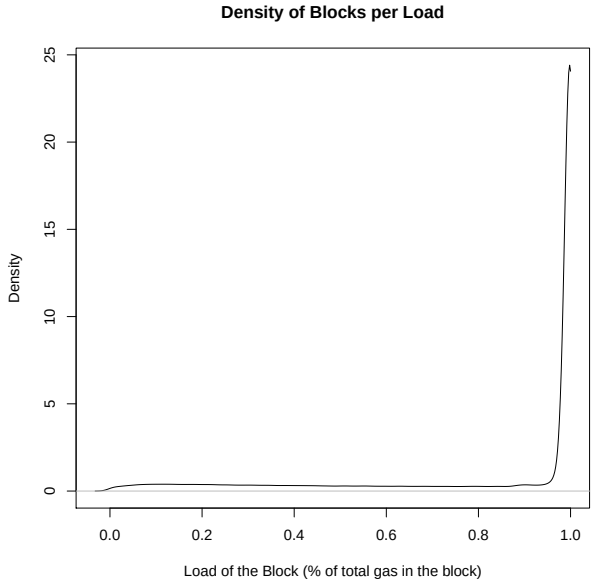
It could be tempting to believe that, while they could in fact accept more transactions per blocks, the miners knowingly make their reserve price vary (meaning that they systematically exclude transactions offering the smallest bids) to lead the users to make higher bid to increase the probability to have their transaction processed fast. We can, however, rule out this hypothesis by looking at the data.
We first extracted data about the load of the blocks. In the ethereum system, each block mined has a maximum "gas"[5] amount.
In order to justify a dependence of the minimum transaction fee on a block on the one of previous blocks, one has to assume that the blocks are mostly "full" (meaning that most of the processing power allocated by the miners to the execution of instructions is indeed used by the instructions executed by the users through their transactions). Otherwise, the use of a model of auction with an infinite amount of goods such as the one presented in (Goldberg et al., 2006) should be used. In such a model, since the auctioneer randomizes the cut price to incentivize "honest" bidding, there is no dependence of a cut-price on the other and the next price can be modeled as a random pick in a fixed distribution.
As can be seen in Figure 1a, however, it seems that, for most of the blocks, the system is at or close to full capacity and there will be dependence in the lowest accepted bids in each block.
In total, we collected information about $360,000$ blocks. The distribution and evolution of the number of transactions in each of these blocks is given in Figure 1b. The summary statistics of the information regarding blocks are available in Table 1.

Table 1: Summary Statistics for Blocks Informations

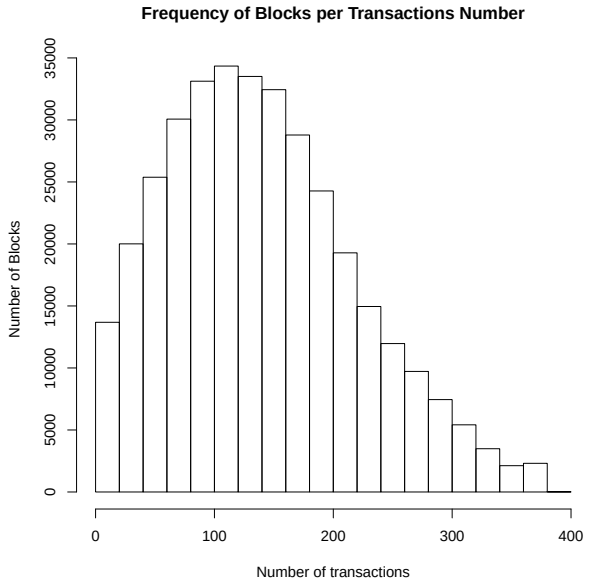
Statistic	N	Mean	St. Dev.	Min	Max
Transactions Count	352,325	140.409	79.421	1	381
Load	352,325	0.832	0.293	0.003	1.000

[4]Blocks that were mined shortly after valid blocks started to be broadcasted and could therefore not be part of the chain. The execution of the transactions in these blocks are executed, the miner rewarded at a lesser rate and the block is kept in the system, however.

[5]In ethereum, Gas is an abstraction of the processing power necessary to process code. Each elementary operation (such as checking a condition or executing a transfer of ethers from one account to another) consumes a certain quantity of gas. Once the maximum amount of gas in a block is reached, the miner does not include any more transaction in the block he is trying to mine.



(a) The density of blocks per load (load being defined as the used gas over total available gas per block). We see that, in most blocks, whole the available gas is consumed by transactions



(b) Number of Transaction per blocks. The median is at 132 transactions per block and the distribution is skewed to the right.

Figure 1: Load and Number of Transactions per Block

5.2.2 Transaction Fees

We also extracted the totality of the transactions fees in blocks in the chain during the period under study (see section 5.1 for more information about what data was collected exactly). In total 49,469,721 transactions were processed during the whole period studied here. The descriptive statistics regarding the transaction fees (in GWei) can be found in Table 2. As seen

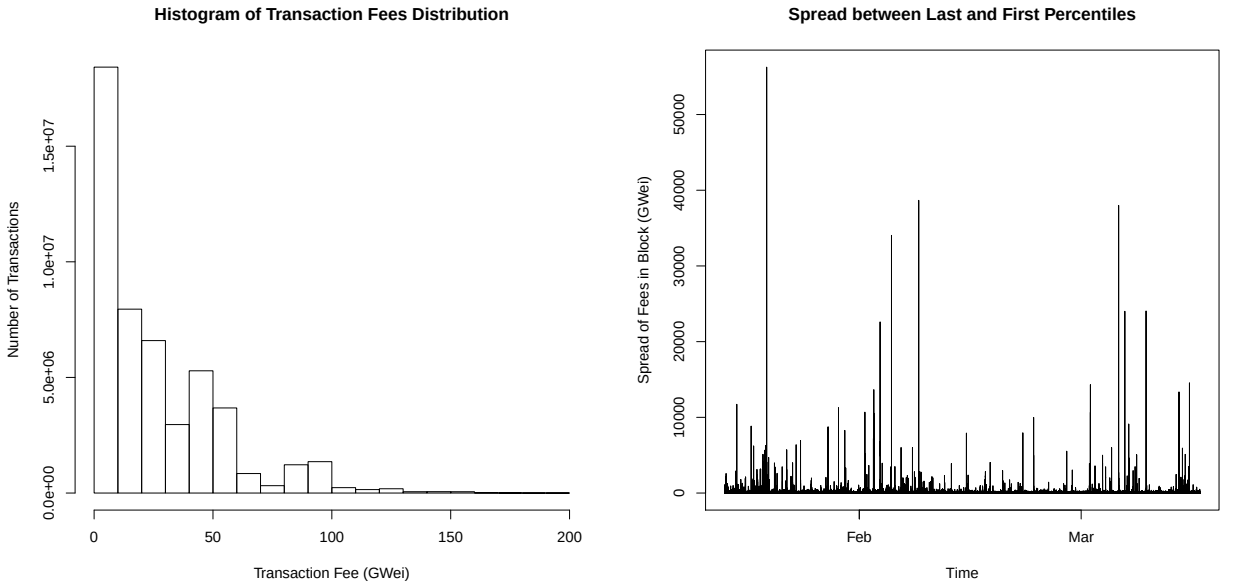
Table 2: Summary Statistics for Transactions Informations

Statistic	N	Mean	St. Dev.	Min	Max
Transaction Fees	49,469,721	28.333	142.940	0.000	750,000.000

in the previous section, the number of transaction inside each block varies a lot. This holds true also for the proposed transaction fees of the transactions contained in each of these blocks and the values of transactions inside blocks over time. This can be observed at Figures 2 and **??**.

From this information, we also compute the aggregated Money Left on the Table as defined in (Engelbrecht-Wiggans Katok, 2007).
Over the two month period observed, we can compute that around $114,507.16 ETH$ (ethers, 1 ether is worth 10^9 GWei - or $635.20 [6]) were spent above the minimum included in the

[6] At the April 22, 2018 exchange rate



(a) Histogram of Transaction Fees

(b) Spread between Percentiles 1 and 99 through Time

Figure 2: Transactions Fees and Spread over Time

block. This represents around $\$57,253,578.58$ over two months and approximatively \$1.16 per transaction written on the chain (this figure would decrease would we include the internal transactions as well). Taken cumulatively, it is a significant amount and it translates directly into more miners producing more pollution in order to participate in the race to mine the next block (one can see that in either (Huberman et al., 2017) or in (Ma et al., 2018)). This is one of the main motives for trying to estimate more accurately the minimum of each block.

5.2.3 Minimum Transaction Fees

As mentioned in our econometric model, we will strive to forecast the next minimum value of transaction fees accepted in a block. This is a complex task as the series oscillate significantly through time. Figure 3 illustrates it. As can be seen, this value varies a lot through time (beware the logarithmic axis for the ordinates) and there are periods of increased volatility followed by calmer periods.

From the data, one can observe that there seems to be values for which we have a large number of observations. Those seem to be "psychological thresholds" in the sense that these are "round" values (mostly in GWei) that are usually recommended by the standard recommendation systems previously described. Different power of 10 will, for example usually be considered more frequently than multiple of those values. This can be interpreted as a sign that users rely mostly on heuristics and relatively simple rules to form their bids and do not seem to think strategically about finding optimal values. As the system gets more stable and professional applications start to become more frequent, though, it will become more critical to avoid overpayment as for-profit companies may tend to be more cost-cautious (or have more sophisticated way of managing costs) than individuals and early adopters.

It is to be noted that the frequency of the data is high. Indeed, as a new block is mined approximately every 15 seconds, we observe a new point at the same rate. It is usually believed that high-frequency data exhibits high persistence of the autocorrelation. This is indeed something

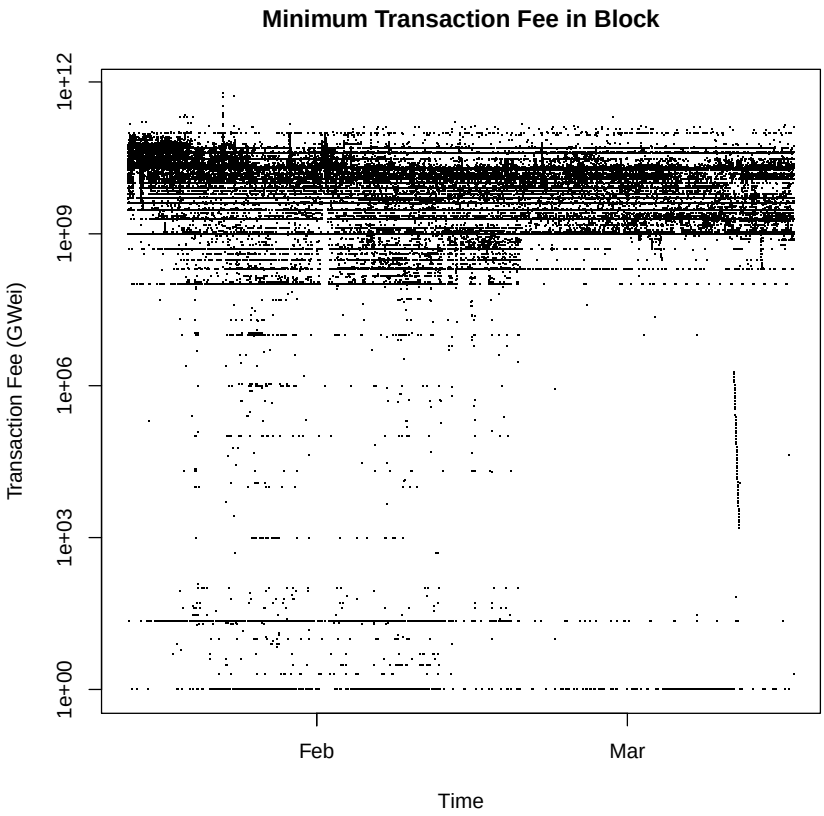


Figure 3: Minimum Transaction Fee per Block

we observe in the data. This seems to indicate that the order of the ARMA part of the forecasting model will need to be relatively high when compared to other, more traditional economic time series. This feature can be seen when we observe the autocorrelogram and the partial autocorrelogram, presented at Figures 4 and **??** (see (Hamilton, 1994)).
We can apply the Augmented Dickey-Fuller test to assess the stationarity of the series. We obtain a test statistic of -21.143 with Lag order = 71, which gives us a p-value = 0.01 on the stationarity hypothesis. While stationarity is indeed a condition for estimating an ARMA (and a GARCH) model with confidence and it would therefore be possible to work directly with this untreated series, we choose to work with the difference of the logarithms of the observations of the series. This allows to decrease the lag order given on the autocorrelogram and, hopefully, to allow us to make predictions with a more parcimonious model.
The plot and autocorrelogram of this modified series is given in Figure 5. Moreover, a plot of the data for approximately an hour is given in Figure 8 in the appendix to help get an idea of how the fees evolve block-by-block rather than in an aggregated fashion.

The examination of these graphs seems to indicate two things. The first one is that the autocorrelogram seems to be "simpler" in the sense that the autocorrelation seems no to be distinguinshable from zero for most lags, as compared with the previous situation where the decay seemed much "slower". The second is that the series seems to pass through periods of high volatility and through periods of lesser volatility. This explains why we chose to couple the GARCH model to the ARMA one as there seems to be heteroskedasticity in the series.

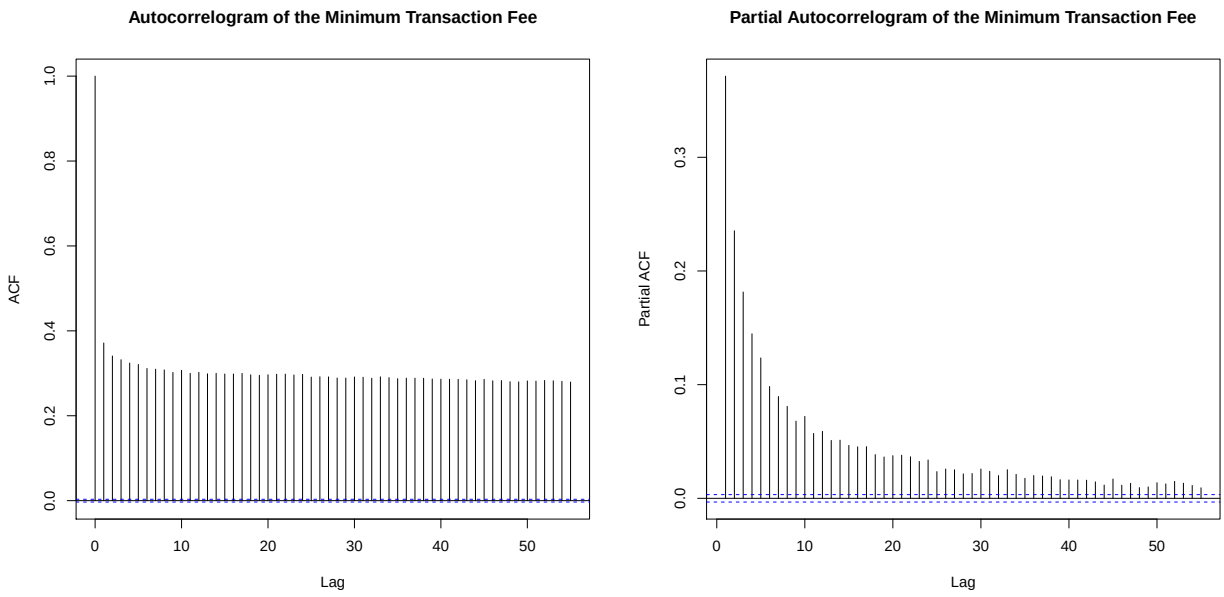


Figure 4: Autocorrelogram and Partial Autocorrelogram of the Minimum Transaction Fee in each Block

Applying the McLeod and Li test to test for the presence of heteroskedasticity leads us not to reject the null hypothesis (the presence of heteroskedasticity) with a p-value of 0.00 up to the 55 degrees of freedom tested. We are therefore conforted in our choice to include a $GARCH$ component to assess the conditional variance of the model.

6 Application and Forecast

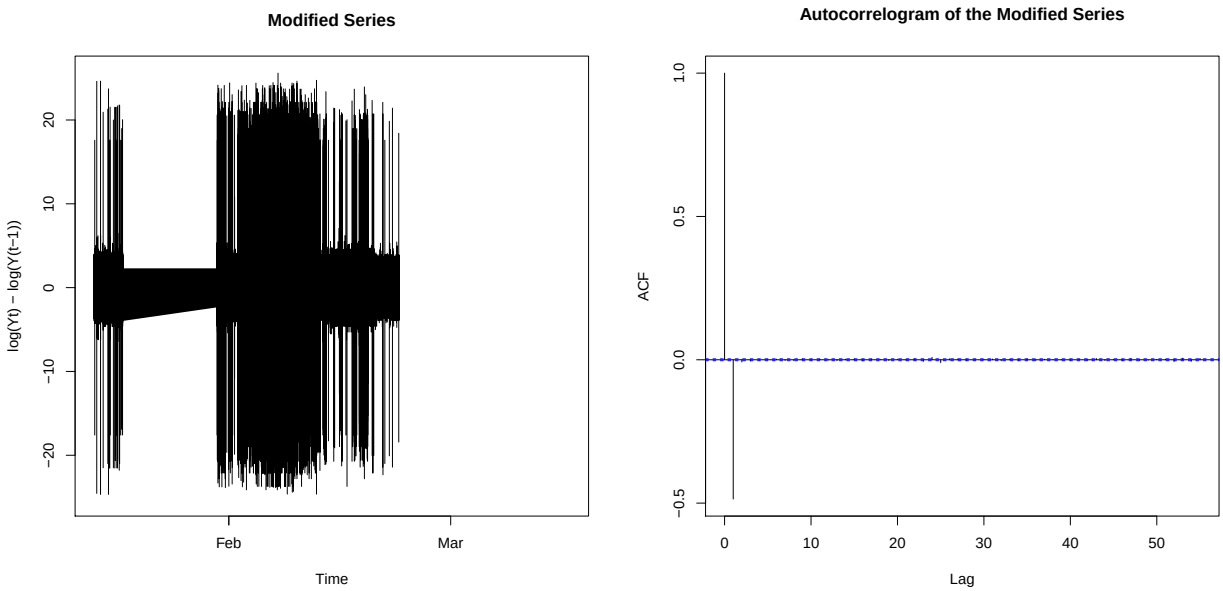
In this section, we present the softwares that were used to perform this study (in the interest of reproducibility) in Subsection 6.1. We detail the model selection procedure in Subsection 6.2, discuss the selected model in Subsection 6.3 and then perform and assess the forecast in Subsection 6.4.

6.1 Softwares

All computations presented here were done using the R statistical software. We used several publicly available (and free/open-source) library. Most of the time series and garch models were estimated and the forecasts produced using the "rugarch" package ((Ghalanos, 2018)). The tables in the present paper were formatted using the Stargazer package ((Hlavac, 2015)). Other packages, such as "dplyr" were used to perform data cleaning and formatting to produce the tables and graphics presented here ((Wickham Francois, 2015)).

6.2 Model Selection Procedure

The ultimate goal of our endeavour is to carry a forecast of the minimum price of gas. However, before assessing the forecasting capacity of ARMA-GARCH models to predict this price, we must first make sure that this class of processes accurately represents the evolution of such a series. We do so by fitting and assessing the model to a small subset of the data.



(a) Modified Time Series

(b) Autocorrelogram of the Modified Series

Figure 5: Time series and Autocorrelogram of the Modified Series

We choose to do so on the prices during the first day of the dataset collected. These are 5000 observations corresponding to the whole day of January 15 2018. We have tested several model specifications by varying the order of both the ARMA and the GARCH processes. The resulting Information Criteria are presented in Table 5 in the appendix.
By analyzing the data, we remark that, although both the Akaike and Schwartz Criteria is minimized for the $ARMA(3,2)-GARCH(0,1)$, this is not the selected model. Indeed, contrarily to the model presented in next section, using this specification results in models that leaves a significant quantity of information in the residual. We choose to select an $ARMA(3,3) - GARCH(1,1)$ model to produce the forecast and assess its properties.
We have also tested the model for several distributions of errors, selecting the one that provides the residuals having the most random structure. Indeed, as we will see, the error have a skew that do not lend itself to the traditional modelling through a normal distribution. We also tested several variant of the $GARCH$ models to make sure that the selected model was indeed the one modelling the process in the best way possible. The results of this selection procedure are presented in Section 6.3.
The last step of this selection procedure is the confrontation of the model to data that was not used to estimate its parameters (the so-called out-of-sample forecasting procedure). This is done during the assessment of the forecasting phase, where we show that, indeed, that class of models is useful to predict prices in the short run on that kind of auctions. As we will show, it performs well against naive forecasting techniques.

6.3 Selected Model

We can now estimate the model. We do so using the standard functions of the "rugarch" package presented in Section 6.1. The selected model is the standard $ARMA(3,3) - GARCH(1,1)$ with normally distributed residuals. We present the coefficients presented in the Table 3. As we can see, there is a short term persistence and most parameters are significative at 5% level

(save for the mean).

We perform the Box-Ljund test on the residuals and the squared residual and find no evidence

Table 3: Maximum Likelihood Coefficients

	Estimate	Std. Error	$t - value$	$Pr(>\|t\|)$
μ	0.000112	0.000210	$5.3154e-01$	0.60
ar_1	0.567191	0.014707	$3.8566e+01$	0.00^{***}
ar_2	0.042226	0.010493	$4.0241e+00$	0.00^{***}
ar_3	0.029448	0.008677	$3.3936e+00$	0.00^{***}
ma_1	−1.537192	0.000001	$-1.3797e+06$	0.00^{***}
ma_2	0.502058	0.000000	$1.0914e+06$	0.00^{***}
ma_3	0.038532	0.000091	$4.2179e+02$	0.00^{***}
ω	0.052537	0.005752	$9.1344e+00$	0.00^{***}
α_1	0.006914	0.002142	$3.2282e+00$	0.00^{***}
β_1	0.972890	0.001210	$8.0402e+02$	0.00^{***}

of remaining $ARMA$ or $GARCH$ structure in the residual. This is what drove the choice for this model against other mentioned in Section 6.2. Indeed, it is the most parcimonious model where the LB tests do not reject the null hypothesis.

We can now analyze the autocorrelation and partial autocorrelation functions of the residuals. These are presented in Figure 6a and 6b. We can see that, although not completely inside the zone where we cannot reject the null hypothesis of no autocorrelation at every lag, we are remarkably close to it for such a highly irregular, complex and high-frequency time series. This is a positive sign that our model manages to explain a good part of the variance of the series.

Considering the frequency of the data, we also tried modelling the conditional mean of the series through a much higher $ARMA$ process (from $ARMA(15, 15)$ to $ARMA(20, 20)$), as well as modelling the conditional variance through a $GARCH(15, 15)$. Beside the considerable estimation time that makes the practical application of the present paper more complicated, the result are not as good as with this much simpler model with less persistence.

This is surprising (although consistent with the conclusions in (Mélard, 2013)), but may actually arise from the fact that, currently, most agents offer their true valuation of having a transaction executed in the next block. This would explain both the weak persistence (as the lowest price of one block is mostly driven by the fact that pricier transactions were wiped out by the preceding block) as well as the fact that the spread of transaction fees in each block varies widely through time (as we have shown previously in Figure 2).

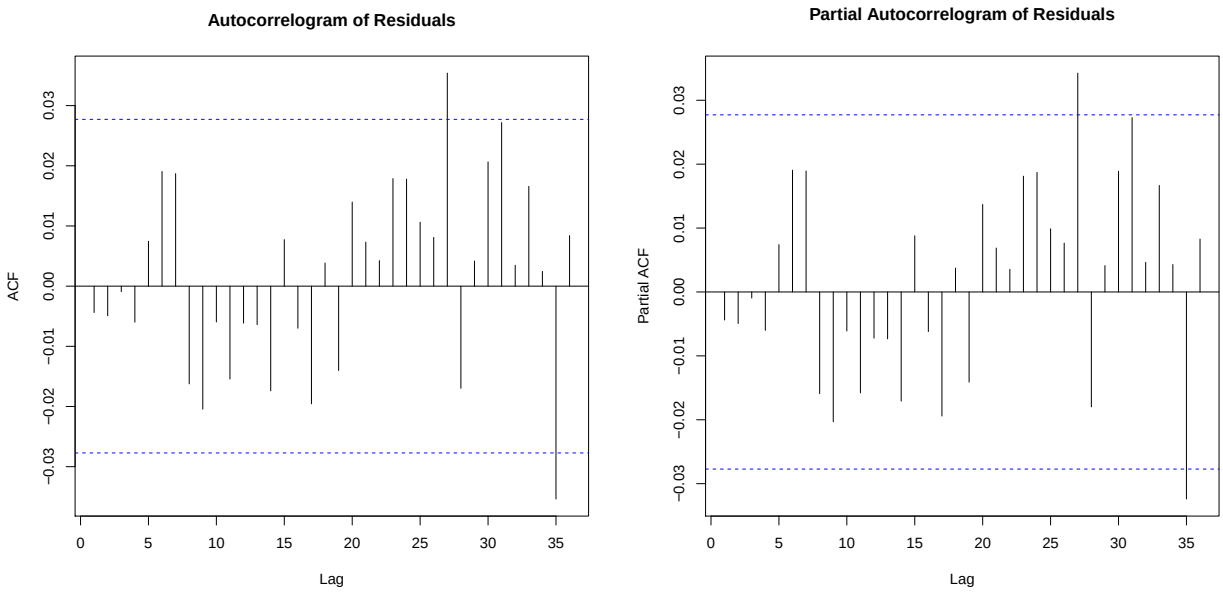
Now that we have shown that the $ARMA(3, 3) - GARCH(1, 1)$ adequately represents the process at hand, we can start applying it to forecast the optimal value for the next block.

6.4 Results

While last section indicated that the model actually represents the process on a small part of the dataset, we now set out to perform the main analysis of the present paper: the forecasting of the minimum value of the next block using the $ARMA(3, 3) - GARCH(1, 1)$ model identified previously. We start by describing the procedure used and then analyze the results obtained.

6.4.1 Forecast Procedure

The forecasting was done using a "rolling forecast" procedure. This means that we performed the estimation of parameters for the model on 5000 blocks at a time (roughly 1 day worth of data) and used the estimated parameters for predicting the next 2500 blocks (roughly, the next 12h). Indeed, it may be suspected that, as market conditions change, so does the influence of past periods and noise. We do so repeatedly to obtain the out-of-sample forecast for most of



(a) Autocorrelation of the Residual of the $ARMA(3,3) - GARCH(1,1)$

(b) Partial Autocorrelation of the Residual of the $ARMA(3,3) - GARCH(1,1)$

Figure 6: Autocorrelation of Residuals for the First Day of Data

the dataset.

In the present paper, we restricted ourselves to 1-step-ahead forecast. This may be conceived as a limitation and will be discussed in the conclusion. Overcoming it is a simple extension of this model (involving estimating the next period on top of the forecast for the period ahead) but will not be presented here.

The forecast, as well as the estimation of its performance were performed on the entire dataset (the full 2 months of the data). The first day of data (the one used to produce the model analysis in Section 6.3) had to be "discarded" as it was not possible to perform out-of-sample prediction and the first 5000 observations were used to estimate the initial parameters. The whole forecast is therefore composed of out-of-sample forecast to be able to assess the performance of the model in a real setting or application.

We can argue that, as the market conditions changed over the period of data collected, the forecast was tested on a fairly diversified dataset. Indeed, the start of year 2018 was characterized by a high turbulence in the cryptocurrencies markets as the bitcoin peak of the end of 2017 came to an end at the onset of the year. The period that followed was far less intense in news and interest for blockchains in general and we could therefore argue that the model was tested on a diverse dataset, lowering the likelihood of having a fortuitous result.

An important consideration here is the speed of applying this model. While current heuristics, such as the ETH Gas Station presented in Section 2.2 include more data, they are refreshed with less frequency (every 100 blocks for the ETH Gas Station), do not include point estimates and do not serve the same purpose.

The forecasting method presented here could be estimated by running an average-grade computer for a couple of minutes every $12h$ and the estimated parameters could then be fed to a very low-grade instance (or on a heavily shared average-grade server) to produce new point estimates every $15seconds$. The solution is thus practical for industrial applications.

This method of rolling forecast for $GARCH$ models is, of course, not new and has been used

in, e.g., (Blair et al., 2010). Moreover, it is also a method used in, mostly, monetary economics studies, such as in e.g. (Swanson, 1998). More recently, there has been tentatives to propose mixed models of rolling and recursive forecasting methods such as in (Clark McCracken, 2009) but there has been so far, to the best of our knowledge, few widespread use of such kind of methods.
The main innovation in the present paper is therefore the object of the forecast as well as the way of assessing forecast results inspired by the peculiar nature of the auctions. As we will see in the next section, the forecast yields better results both in mean squared error and mean absolute deviation, it also performs well again the naive forecast in term of total regret for a wide array of realistic values for waited time (our λ parameter).

6.4.2 Results and Comparison with Naive Forecast

We first compare the results in the difference of logarithms. We compare the outcome (forecasts) using our model to the ones generated by naive forecasting. Naive forecasting is the forecasting method consisting of assuming that the value at $t+1$ is the value at t (with the underlying assumption that the series has martingale properties). In terms of difference of logarithms, it amounts to assuming that the result will always be 0 (i.e., the logarithm at the value next period will be the same as the one this period).
We thus produce forecasts using these two methods. The results, expressed in terms of forecast error can be found in table 4. It seems that the proposed estimation technique performs better than the naive forecast of this transformed series.
If we look at the forecasting result in monetary terms rather as the difference of logarithms, we

Table 4: Comparison of the error of rolling ARMA-GARCH and naive forecast

Metric	ARMA-GARCH	Naive
MSE	7.69	13.11
MAD	1.34	1.66

observe the opposite situation, based on both the mean squared error and the mean absolute deviation. The ARMA-GARCH forecast seems to fare worse than the naive forecast by several orders of magnitudes (in GWei, for the absolute deviation), which can be explained by the fact that an error is very costly if it is done at the level of the exponents and that skew in a forecast brings dire consequences when done in the difference of the logarithms.
This could seem to forebid the demise of the ARMA-GARCH method for forecasting the transaction price in the short run (see table 8 in the appendix). However, we argue that it is not the case. Indeed, the consequences of the error are asymmetrical in nature. The cost for the user to bid $1Gwei$ above the minimum of a block is of approximately $635.20\$ \times 10^{-9}$ which, in monetary terms, is usually negligible. Making an error of $1Gwei$ below the minimum costs the monetary equivalent of at least $15seconds$.
We therefore choose to compare both forecast at the light of the Empirical Total Regret, described in Section 4.1.1. The parameter we seek to identify is λ^*, which is the value of a second of waiting above which one forecasting method is more interesting than the other for a determined user. The result is presented in Figure 7. We can see that, for a user valuing a second of waiting time above roughly $10^{-3}\$cts/s$, forecasting the bid in the next block using the ARMA-GARCH method outperforms the naive forecast.
It is important to remark that the value of time is exogenous to the problem of selecting the optimal bid. Indeed, it depends on the value one assigns to the transaction being registered or

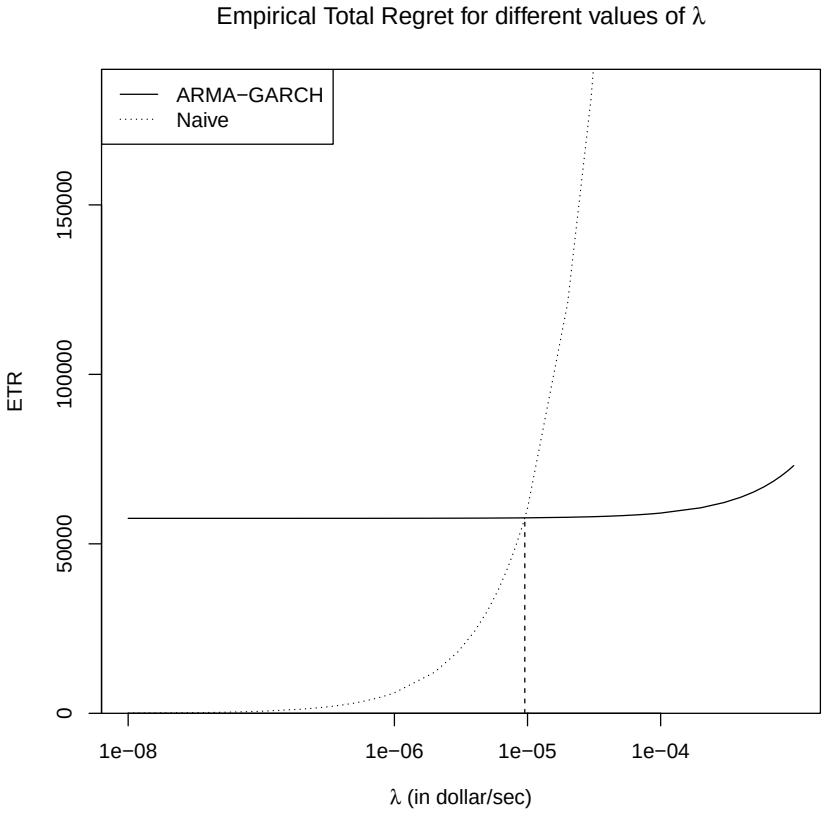


Figure 7: The Empirical Total Regret metric for different values of λ

the contract being executed. Moreover, for a user, this value can fluctuate in time according to the criticality of the operation.
It depends, however, on the complexity of the operation being executed. Indeed, this coefficient has to be multiplied by the total "gas" value to find the indifference λ. For example, a user realizing a transaction (a transfer of tokens from one account to another) will favour the method presented here over the naive forecast if he values a second of waiting above $4.75cts$ as such an operations uses $2,100$ units of gas.
It is conceivable that one will assign a lower value of waiting to a non-time-critical operation (such as the sale of low-value token when the volatility of the price is low) than for high-stake operation (such as the execution of a high-value contract critical to one company operation). The choice of the method will thus depend on the use case.
We consider the value of the λ^* found in the present paper as an encouraging sign that the method analyzed here is useful for non-trivial use of the smart contracts. Indeed, one can expect that, as decentralized applications become more pervasive and gain broader acceptance into mainstream, the criticality of these systems will rise and solving issues of timing will become important for some companies and individuals wanting to enjoy the benefits of decentralized computing. Today, this criticality aspect can already be seen in the sales of tokens, in which, during period of turbulences, the value of a token can evolve by the minute or even the second.

7 Conclusion and Future Research

7.1 Conclusion

We believe that this paper is the first of its kind in the way it approaches the issue of pricing transactions on blockchain. While most previous work emphasized the relatively complex nature of the task of assessing the optimal bid for the user, we show that a relatively simple and parcimonious method fares better than the most sensible heuristics based on the assumption that markets are efficient and that the price evolution, as is often assumed in finance, has martingale properties. Applying the method is fairly straightforward and is practical even with limited computing power at one's disposal. This is a desirable quality if we intend to uphold the original ideals of the blockchain that, along with a complete decentralization and a drift towards a trustless society, often praise the fact that everyone is equal in the system, be it a large or a small player.

We certainly hope to have convinced the reader of two things: The first is the interest of more inquiry in the topic of prices in the complex environments created by blockchain systems. The second is the fact that the market is still in its infancy and, as such, there is a vast amount of simple improvements to be found and implemented that may, in the end, make a difference between a cryptic system for quirky cryptolibertarians and a real societal advance that will bring us forward in the sophistication of the ways we collaborate.

While the decidedly empirical approach adopted here may seem risky in the absence of an underpinning sound theory of such kind of auctions, it is made necessary by the fact that players in the market are improvising their way through the journey (relying mostly on simple heuristics presented above that may distort the way they behave if compared with what theory would predict) and possible by the vast quantity of data freely available to the person who knows where to look for it. As such, the present article was conceived more as a proof of concept and hopefully an inspiration to embark on this research journey than as an endpoint. There is plenty to be done and even more to be discovered.

7.2 Limitations

The present paper is only a small step in a research endeavor that, hopefully will help turn decentralized smart contracts more widespread. As such, there are several limitations to the conclusions presented here and it is of paramount importance to understand that this paper does not, by any mean, have any pretention of being an end to the topic.

First and foremost, we make no claim regarding the optimality of the methodology presented here. Indeed, while we show that ARMA-GARCH forecasts yield better results in terms of regret for a realistic range of time valuation than the naive forecast, there may be other methods that yield similar or even better outcomes on the long run. There is a wealth of statistical forecasting methods and heuristics to be tested and assessed before making any general claim about the source of transaction fees variations in blockchain systems. We mostly stand behind the idea that, by looking at previous transaction fees, one can extract useful information about the likelihood of the next fees to be paid to avoid overspending on computing costs. Understanding the dynamics of price formation on such auctions through computationally realistic methods, considering the high frequency of the process, will be an interesting challenge that we look forward tackling.

The second major limitation is the inclusion in the model of the fact that, despite the best efforts of the designers of most smart-contracts enabling blockchain, broadcasting incoming transactions is not instantaneous. Indeed, more reseach should be carried on the speed of broadcasting of transactions through the miner network before being able to confidently claim to be able to account for this latency. However, knowing that we are able to forecast adequately the price for the next block turns the possibility of doing so at a longer horizon more likely. This would,

obviously entails accounting for both uncertainty on the next forecast and for the uncertainty on the latency of the network and will, hopefully, be tackled in a future article.
It is important to highlight, once more, the important assumptions made to derive the Empirical Total Regret from Section 4.1.1. First and foremost, it assumes that the user of the system derives a high enough utility from using the decentralized app. Indeed, if this is not the case, there is a natural cap on the value she would be willing to offer on the auction. Moreover, we only consider the regret minimization problem, without consideration for other concerns that may arise from using decentralized applications and smart contracts. Finally, we only consider a (admittedly rather simple) linear trade-off between money and time. It is easy to imagine that, in some cases, large deviations in one dimensions or the other may cause the emergence of non-linear effects on the loss or regret one feels. This is, however, a research project of a much more theoretical aspect than what we pretend to have accomplished here.
Finally, it should be rather clear to the reader that the present paper is empirical in nature and that no attempt was done at any point to provide any unambiguous explanation about why prices evolve the way they do. Indeed, the peculiar format of the auction generated by offering processing power repeatedly spaced at regular time interval create trade-offs that are unusual, be it in classical auction theory (which is usually concerned with one-off auctions) or in the IT management literature (which usually deals in deterministic processing costs). This is, in itself, a rather intriguing topic of research that will be evoked in the next section.

7.3 Future Research

It is certain that distributed computing, while being restricted to a narrow field of general computing is destined to play a role in the future. In this context, the understanding of the mechanics of processing costs is important as companies commit resources to the development of solutions involving smart contracts and decentralized applications. In an environment where returns are unclear, volatility in transaction fees induces an undesirable premium on projects that are already very risky for any company that would like to adopt an even slightly conservative stance. We consider therefore any attempts at understanding costs, both in the long and the short run, worthy endeavors.
Blockchain systems are often considered near-perfectly competitive markets, and the fact that several of the early attempts at modelling such systems (e.g. (Huberman et al., 2017) and (Ma et al., 2018)) take such hypothesis as the founding assumption only seem to confirm this belief. Considering this and the high volume of data available to the bidders in the auctions, it seems difficult to justify honest bidding as an optimal strategy. We believe therefore that the present paper is merely one among several attempts at understanding more and making transaction price formation more efficient on the long run.
We will continue this endeavour in two main ways. The first will be the improvement of the predictions, be it through the application of other techniques or through the analysis of forecast at a further time horizon. This will be necessary if we want to turn the costs of computing manageable in the short run and, at least partially, predictable in the long run. The future of decentralized application remains uncertain and some question their ability to scale when compared to more traditional cloud or on-premises application. However, decreasing the volatility of the costs of such systems seems to be an important condition to warrant their adoption by corporations and large-scale projects. It is also an important step towards highlighting the return on investment of such projects with more certainty.
The second endeavour, more theoretical in nature is to design a fundamental understanding on repeated multi-good auctions and the price formation mechanics that arise from them.

References

Andersen TG, Bollerslev T, Christoffersen PF, Diebold FX (2006) Volatility and correlation forecasting. *Handbook of economic forecasting* 1:777–878.

Babaioff M, Dobzinski S, Oren S, Zohar A (2012) On bitcoin and red balloons. *Proceedings of the 13th ACM conference on electronic commerce*, 56–73 (ACM), URL `http://dl.acm.org/citation.cfm?id=2229022`.

Blair BJ, Poon SH, Taylor SJ (2010) Forecasting S&P 100 volatility: the incremental information content of implied volatilities and high-frequency index returns. *Handbook of Quantitative Finance and Risk Management*, 1333–1344 (Springer).

Bollerslev T (1986) Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics* 31(3):307–327.

Brière M, Oosterlinck K, Szafarz A (2015) Virtual currency, tangible return: Portfolio diversification with bitcoin. *Journal of Asset Management* 16(6):365–373.

Buterin V (2014) A next-generation smart contract and decentralized application platform. *white paper* .

Clark TE, McCracken MW (2009) Improving forecast accuracy by combining recursive and rolling forecasts. *International Economic Review* 50(2):363–395.

Davidson S, De Filippi P, Potts J (2016) Economics of blockchain. *Proceedings of Public Choice Conference*, URL `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2744751`.

Engelbrecht-Wiggans R, Katok E (2007) Regret in auctions: Theory and evidence. *Economic Theory* 33(1):81–101.

Ghalanos A (2018) rugarch: Univariate GARCH models., 2012a. *R package version* 1–0.

Goldberg AV, Hartline JD, Karlin AR, Saks M, Wright A (2006) Competitive auctions. *Games and Economic Behavior* 55(2):242–269.

Goldberg AV, Hartline JD, Wright A (2001) Competitive auctions and digital goods. *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, 735–744 (Society for Industrial and Applied Mathematics).

Hamilton JD (1994) *Time series analysis*, volume 2 (Princeton university press Princeton).

Hlavac M (2015) *stargazer: well-formatted regression and summary statistics tables. R package version 5.2*.

Huberman G, Leshno JD, Moallemi CC (2017) Monopoly Without a Monopolist: An Economic Analysis of the Bitcoin Payment System. SSRN Scholarly Paper ID 3025604, Social Science Research Network, Rochester, NY, URL `https://papers.ssrn.com/abstract=3054887`.

Kristoufek L (2015) What Are the Main Drivers of the Bitcoin Price? Evidence from Wavelet Coherence Analysis. *PLOS ONE* 10(4):e0123923, ISSN 1932-6203, URL `http://dx.doi.org/10.1371/journal.pone.0123923`.

Liu H, Shi J (2013) Applying ARMAGARCH approaches to forecasting short-term electricity prices. *Energy Economics* 37:152–166, ISSN 01409883, URL `http://dx.doi.org/10.1016/j.eneco.2013.02.006`.

Ma J, Gans JS, Tourky R (2018) Market Structure in Bitcoin Mining. Working Paper 24242, National Bureau of Economic Research, URL http://dx.doi.org/10.3386/w24242.

Mélard G (2013) Forecasting daily and high frequency data (Paris), URL http://homepages.ulb.ac.be/~gmelard/rech/Forecasting_daily_and_high_frequency_data_v9.pdf.

Nakamoto S (2008) *Bitcoin: A peer-to-peer electronic cash system*. URL http://www.cryptovest.co.uk/resources/Bitcoin%20paper%20Original.pdf.

Solibakke PB (2001) Efficiently ARMA-GARCH estimated trading volume characteristics in thinly traded markets. *Applied Financial Economics* 11(5):539–556, ISSN 0960-3107, URL http://dx.doi.org/10.1080/09603100010029234.

Swanson NR (1998) Money and output viewed through a rolling window. *Journal of monetary Economics* 41(3):455–474.

Szabo N (1997) Formalizing and securing relationships on public networks. *First Monday* 2(9).

Varian HR (2010) Computer Mediated Transactions. *The American Economic Review* 100(2):1–10, ISSN 0002-8282, URL http://www.jstor.org/stable/27804953.

Wickham H, Francois R (2015) dplyr: A grammar of data manipulation. *R package version 0.4* 3.

8 Additional Graphics and Tables

This section aims at providing additional plots in case the reader wants to visualize aspects not covered in the main sections of this document.

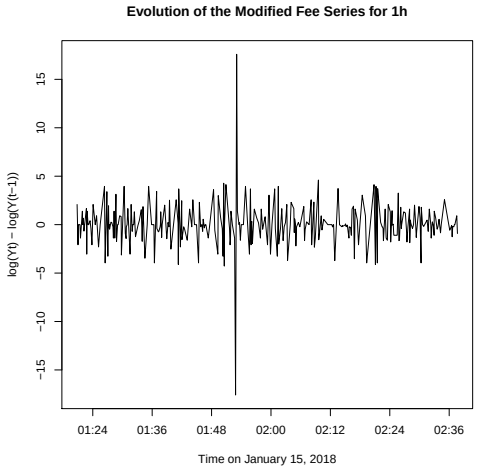


Figure 8: Series for approximately 1h of the difference of the logarithms of the minimum transaction fees per block on January 15, 2018

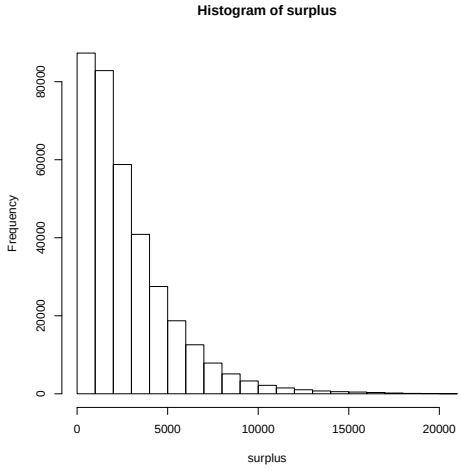


Figure 9: The Distribution of the Regrets (at the Transaction Fees Level)

Table 5: Information Criteria for Several Orders

MA	AR	Alpha	Beta	AIC	BIC
1	1	0	1	3.250	3.259
1	1	0	2	3.199	3.209
1	1	1	1	3.230	3.240
1	1	1	2	3.231	3.243
1	1	2	1	3.231	3.243
1	1	2	2	3.233	3.246
1	2	0	1	3.124	3.134
1	2	0	2	3.127	3.139
1	2	1	1	3.113	3.125
1	2	1	2	3.113	3.126
1	2	2	1	3.114	3.127
1	2	2	2	3.114	3.128
2	**1**	**0**	**1**	**3.153**	**3.164**
2	1	0	2	3.245	3.256
2	1	1	1	3.227	3.238
2	1	1	2	3.232	3.245
2	1	2	1	3.226	3.239
2	1	2	2	3.227	3.241
2	2	0	1	3.033	3.045
2	2	0	2	3.031	3.044
2	**2**	**1**	**1**	**2.996**	**3.009**
2	2	1	2	3.017	3.031
2	2	2	1	3.006	3.021
2	2	2	2	3.004	3.020

Table 6: Comparison of the error of rolling ARMA-GARCH and naive forecast in GWei terms

Metric	ARMA-GARCH	Naive
MSE	4.43×10^{15}	189.16
MAD	2.6×10^{5}	6.91

Table 7: Weighted Ljung-Box Test on Standardized Residuals
d.o.f=6
H0 : No serial correlation

	Statistic	p-value
Lag[1]	0.002149	0.9630
Lag[2*(p+q)+(p+q)-1][8]	5.278279	1.0000
Lag[4*(p+q)+(p+q)-1][14]	10.361039	0.9594

Table 8: Weighted Ljung-Box Test on Standardized Squared Residuals
d.o.f=1
H0 : No serial correlation

	Statistic	p-value
Lag[1]	0.008088	0.9283
Lag[2*(p+q)+(p+q)-1][2]	0.170735	0.9946
Lag[4*(p+q)+(p+q)-1][5]	0.233544	0.9999

WORKING PAPERS 2013

001 - Exploring europe's r&d deficit relative to the us: differences in the rates of return to r&d of young leading r&d firms - Michele Cincera and Reinhilde Veugelers

002 - Governance typology of universities' technology transfer processes - A. Schoen, B. van Pottelsberghe de la Potterie, J. Henkel.

003 - Academic Patenting in Belgium: Methodology and Evidence – M. Mejer.

004 - The impact of knowledge diversity on inventive performance at European universities – M. Mejer

005 - Cross-Functional Knowledge Integration, Patenting and Firm's Performance – M. Ceccagnoli, N. van Zeebroeck and R. Venturini.

006 - Corporate Science, Innovation and Firm Value, M. Simeth and M. Cincera

WORKING PAPERS 2014

007 - Determinants of Research Production at top US Universities – Q. David.

008 - R&D financing constraints of young and old innovation leaders in the EU and the US – M. Cincera, J. Ravet and R. Veugelers

009 - Globalization of Innovation Production; A Patent-Based Industry Analysis – J. Danguy

010 - Who collaborates with whom: the role of technological distance in international innovation – J. Danguy

WORKING PAPERS 2015

011 - Languages, Fees and the International Scope of Patenting – D. Harhoff , K. Hoisl, B. van Pottelsberghe de la Potterie , C. Vandeput

012 - How much does speed matter in the fixed to mobile broadband substitution in Europe? – M. Cincera, L. Dewulf, A. Estache

013 - VC financing and market growth – Interdependencies between technology-push and market-pull investments in the US solar industry – F. Schock, J. Mutl, F. Täube, P. von Flotow

014 - Optimal Openness Level and Economic Performance of Firms: Evidence from Belgian CIS Data – M. Cincera, P. De Clercq, T. Gillet

015 - Circular Causality of R&D and Export in EU countries – D. Çetin, M. Cincera.

016 - Innovation and Access to Finance – A Review of the Literature – M. Cincera, A. Santos.

WORKING PAPERS 2016

017 - Effectiveness of Government intervention in the SME sector: Evidence from the Brussels-Capital Region – G. E. Fombasso, M. Cincera.

018 - A review of corporate R&D intensity decomposition – P. Moncada-Pastemò-Castello.

019 - The laws of action and reaction: on determinants of patent disputes in European chemical and drug industries – R. Kapoor, N. van Zeebroeck.

020 - How do the normativity of headquarters and the knowledge autonomy of subsidiaries co-evolve? – M. Hansmans, G. Liu.

WORKING PAPERS 2017

021 - The case for offensive strategies in response to digital disruption – J. Bughin, N. van Zeebroeck.

022 - Access to finance as a pressing problem: Evidence from innovative European firms – A. Santos, M. Cincera.

023 - Platform play among incumbent firms: the wrong focus? – N. van Zeebroeck, J. Bughin.

024 - Social Movements – M. Hensmans, K. van Bommel.

025 - Decoding Patent Examination Services – L. Gimeno-Fabran, B. van Pottelsberghe de la Potterie.

026 - Countries Attractiveness: an Analysis of EU Firms's Decisions to (De) Localize R&D Activities – M. Cincera, A. Santos.

WORKING PAPERS 2018

027 - The impact of EUREKA projects on the economic performance of R&D SMEs – M. Cincera, G. Fombasso.

028 - Forecasting short-term transaction fees on a smart contracts platform – C. Hoffreumon, N. van Zeebroeck.