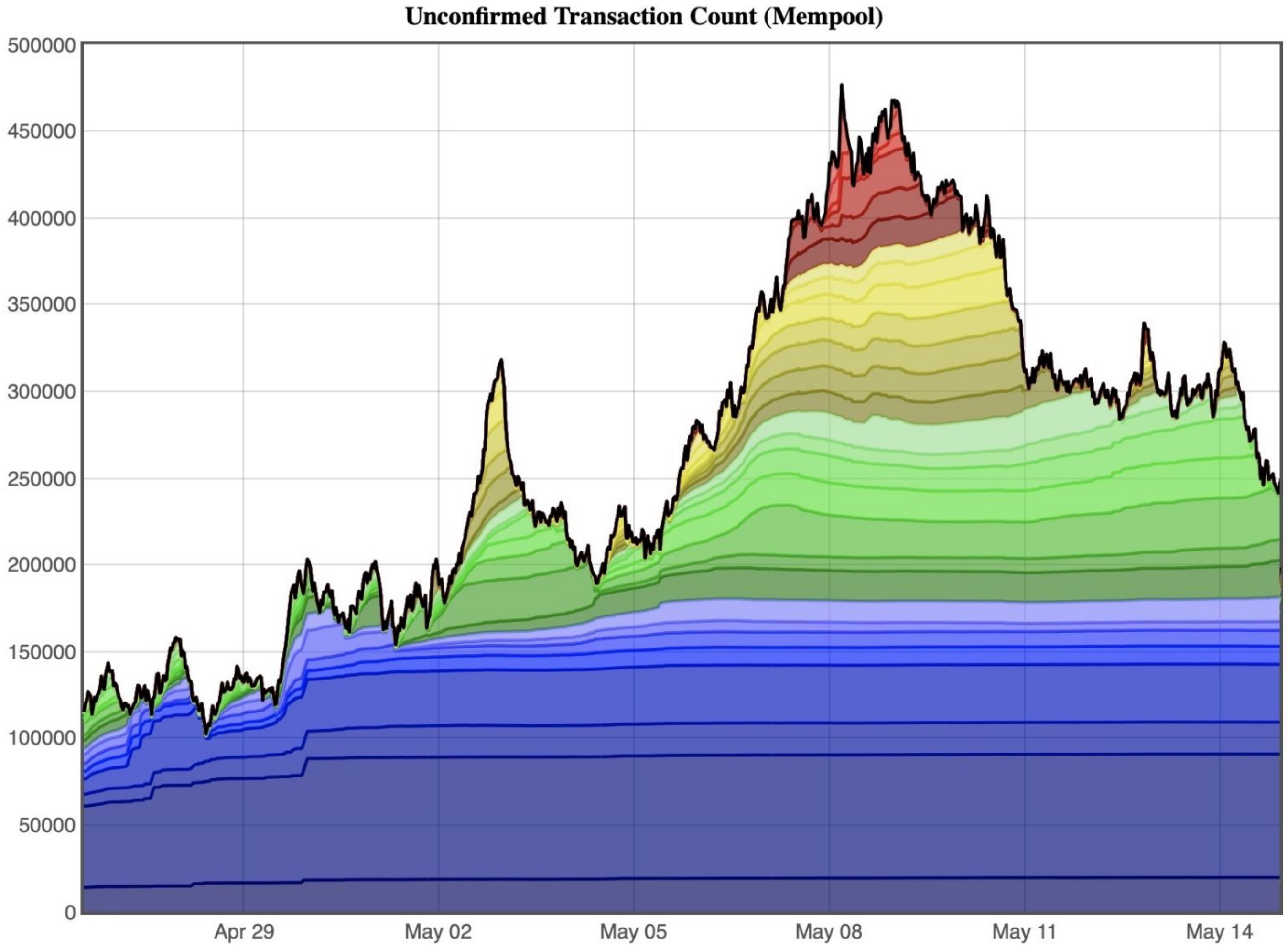
Confirmed or Dropped: Reliability Analysis of Transactions in PoW Blockchains

Andrea Marin, Ivan Malakhov, Sabina Rossi, Daniel Sadoc

Published in IEEE Trans. on Network Science

The behavior of the Mempool (ideal with infinite size)



The problem of transaction dropping

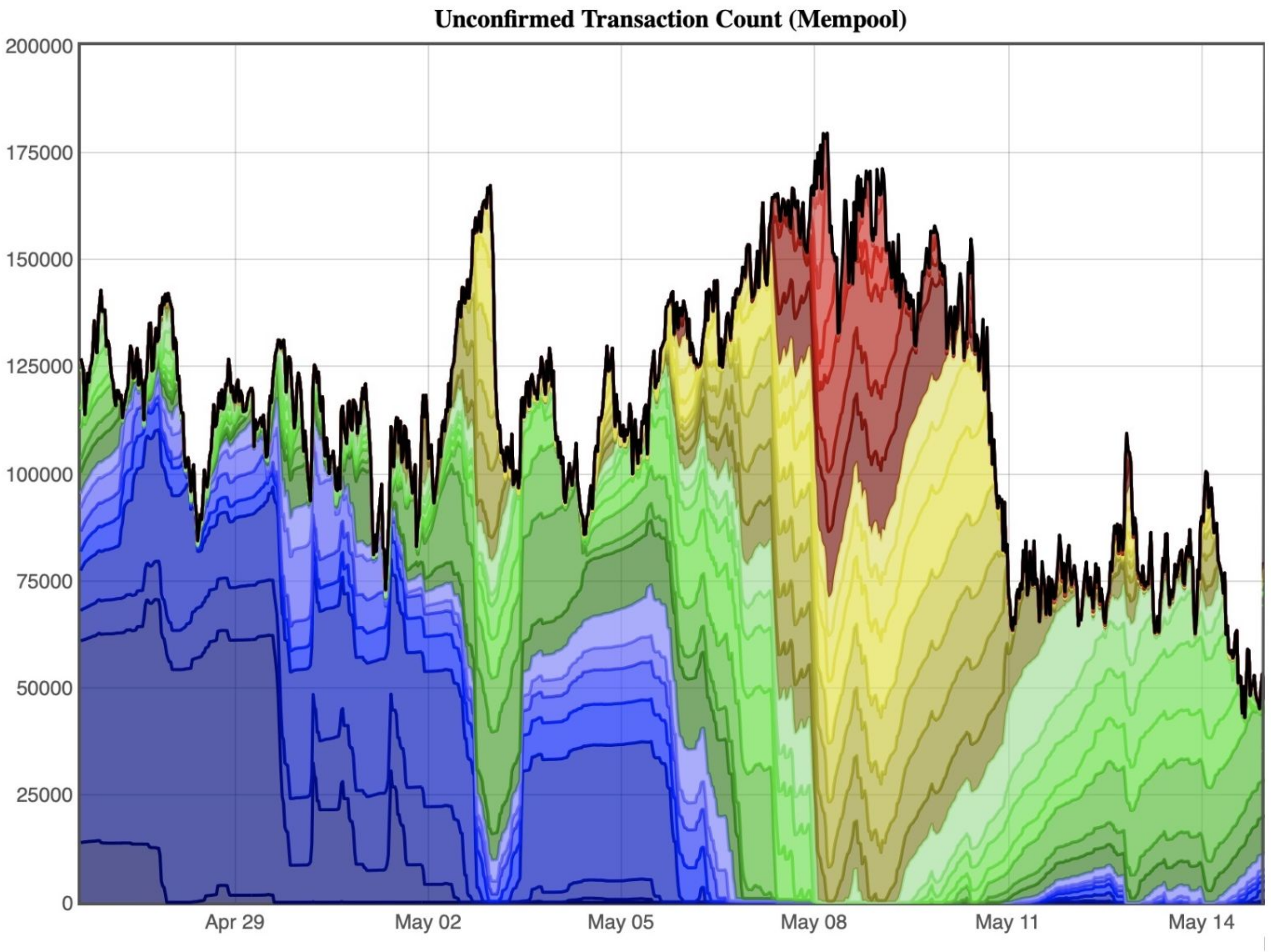
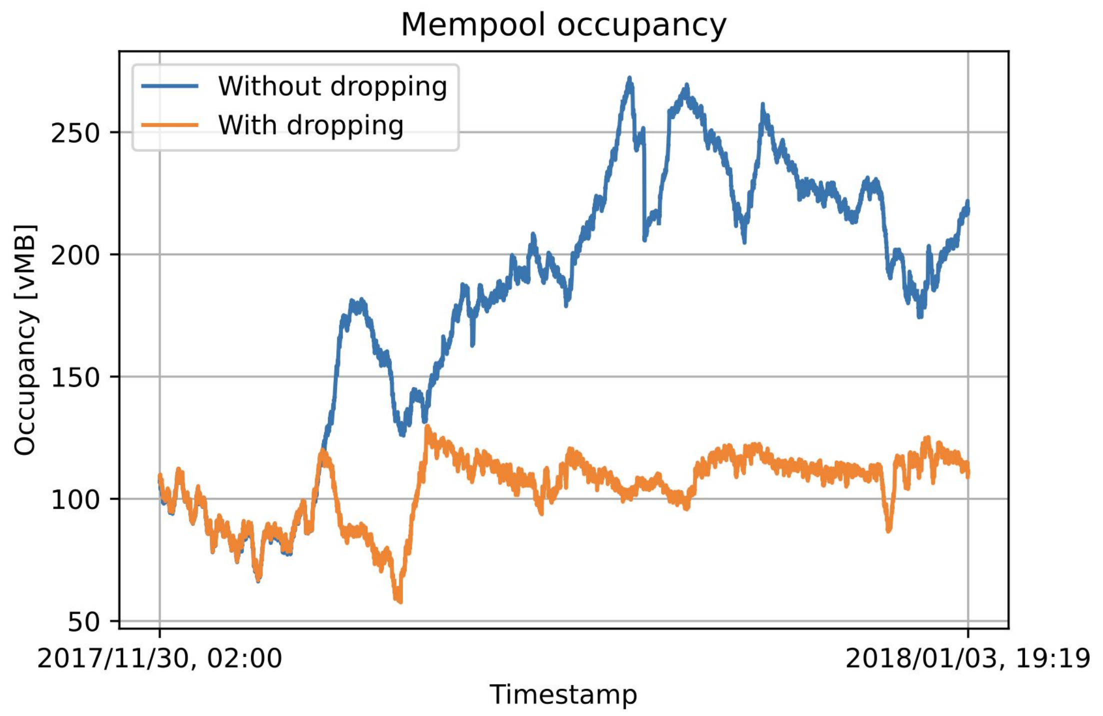
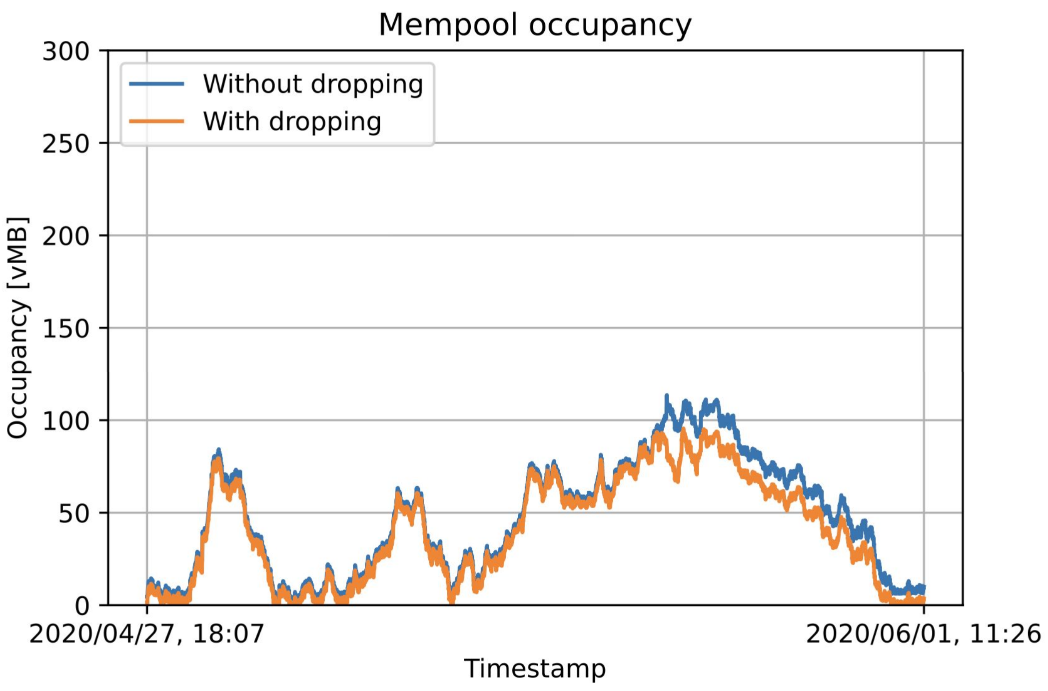


Fig. 1. Mempool occupancy per fee level of the default Mempool.





A deeper view

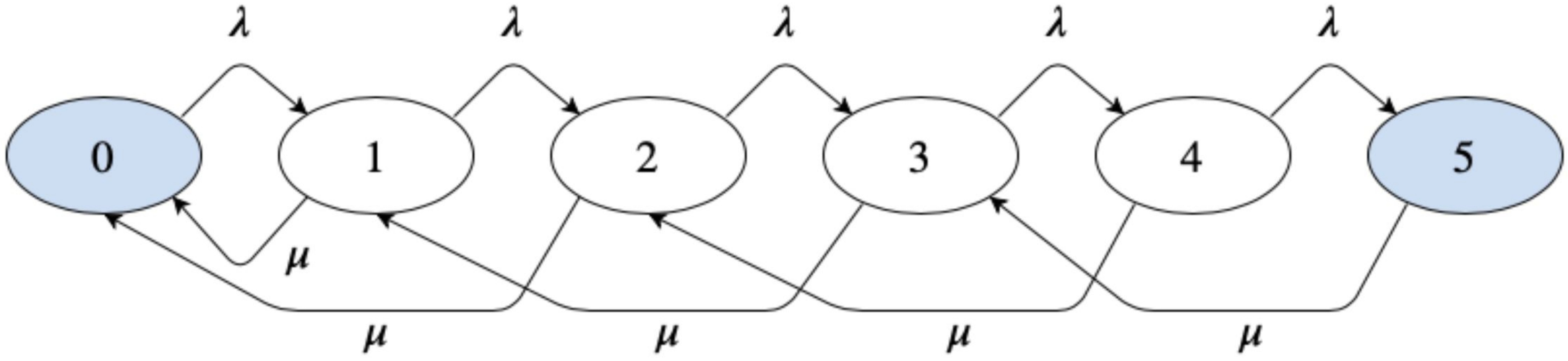
Problem statement

Suppose we know:

- The Mempool state
- Description of the arrival process

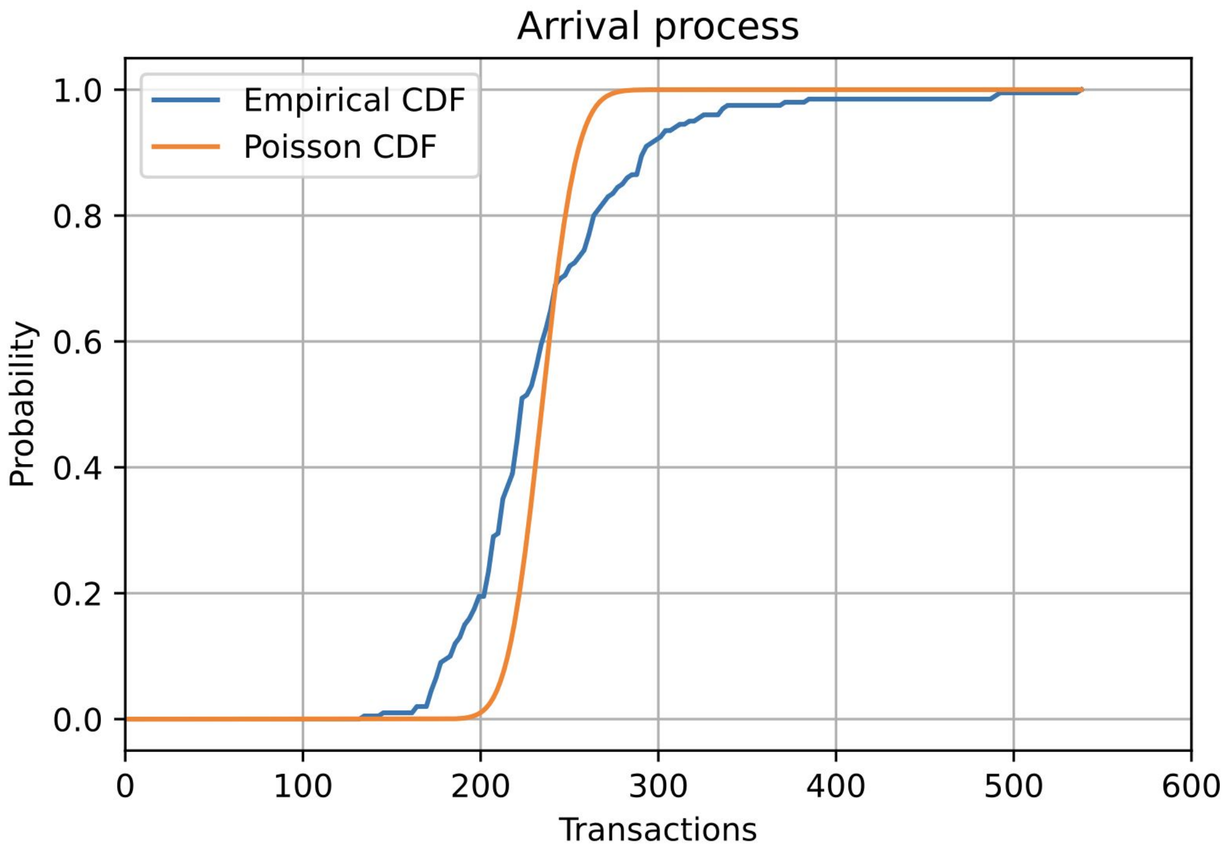
We want to determine:

- If a transaction offers a certain fee per byte, what is the probability that this transaction will eventually be confirmed or dropped?
- Problem particularly relevant for delay tollerant transactions



A variation of the gambler’s ruin problem

Assumptions



- The transactions' arrival process is Poisson
- The inter-block generation time is independent and exponentially distributed
- Miners are fair

Variable	Description
$K-1$	Mempool capacity in number of transactions
B	maximum number of transactions per block
λ	arrival rate of transactions
μ	block generation rate
α	probability of transaction arrival before next block mining
β	probability that B arrivals occur before block mining
τ	arriving transaction
t	arrival time
i	# of pending transactions found by the arriving transaction
p_i	probability that the arriving transaction is eventually dropped

Notation

Theorem 1. *For $0 \leq i \leq K$, the solution of the system of equations (1) is:*

$$p_i = \frac{T_i}{T_K}, \tag{2}$$

where

$$T_i = \frac{1}{\alpha^{i-1}} \sum_{l=0}^{m_i} \beta^l \binom{l(B+1)-i}{l} \tag{3}$$

and

$$m_i = \left\lfloor \frac{i-1}{B+1} \right\rfloor.$$

Main theoretical achievement

A numerical approach: difference equations

- The probability of absorption can be described by the following system of equations:

$$\begin{cases} p_0 = 0 & \\ p_i = \dfrac{\alpha^{1-i}}{T} & 1 \leq i \leq B \\ p_i = (1-\alpha)p_{i-B} + \alpha p_{i+1} & B < i < K \\ p_K = 1\,. & \end{cases}$$

- This requires to find the roots of the following polynomial

$$P(x) = \alpha x^{B+1} - x^B + (1-\alpha)$$

- We prove that the polynomial has different real or complex roots

How to get the solutions

- The probability of absorption has the following form:

$$p_i = \sum_{j=1}^{B+1} C_j^* x_j^i$$

- The coefficients can be determined by solving the linear system:

$$\begin{cases} C_1^* + C_2^* + \ldots + C_{B+1}^* = 0 & i = 0 \\ C_1^* x_1^i + C_2^* x_2^i + \ldots + C_{B+1}^* x_{B+1}^i = \frac{1}{T}\alpha^{1-i} & 1 \leq i \leq B \end{cases}$$

Toy example

- Let's study a system with the following characteristics:
- Blocks contain 3 transactions
- 1.4 transactions per second is the arrival rate
- Block are generated with a rate of 0.6 blocks per second
- Mempool capacity is 50

Solutions

- Find the roots of the polynomial

$$x_1 = 1\,, \qquad x_2 \simeq -0.354 - 0.501j\,,$$
$$x_3 \simeq -0.354 + 0.501j\,, \qquad x_4 \simeq 1.137\,.$$

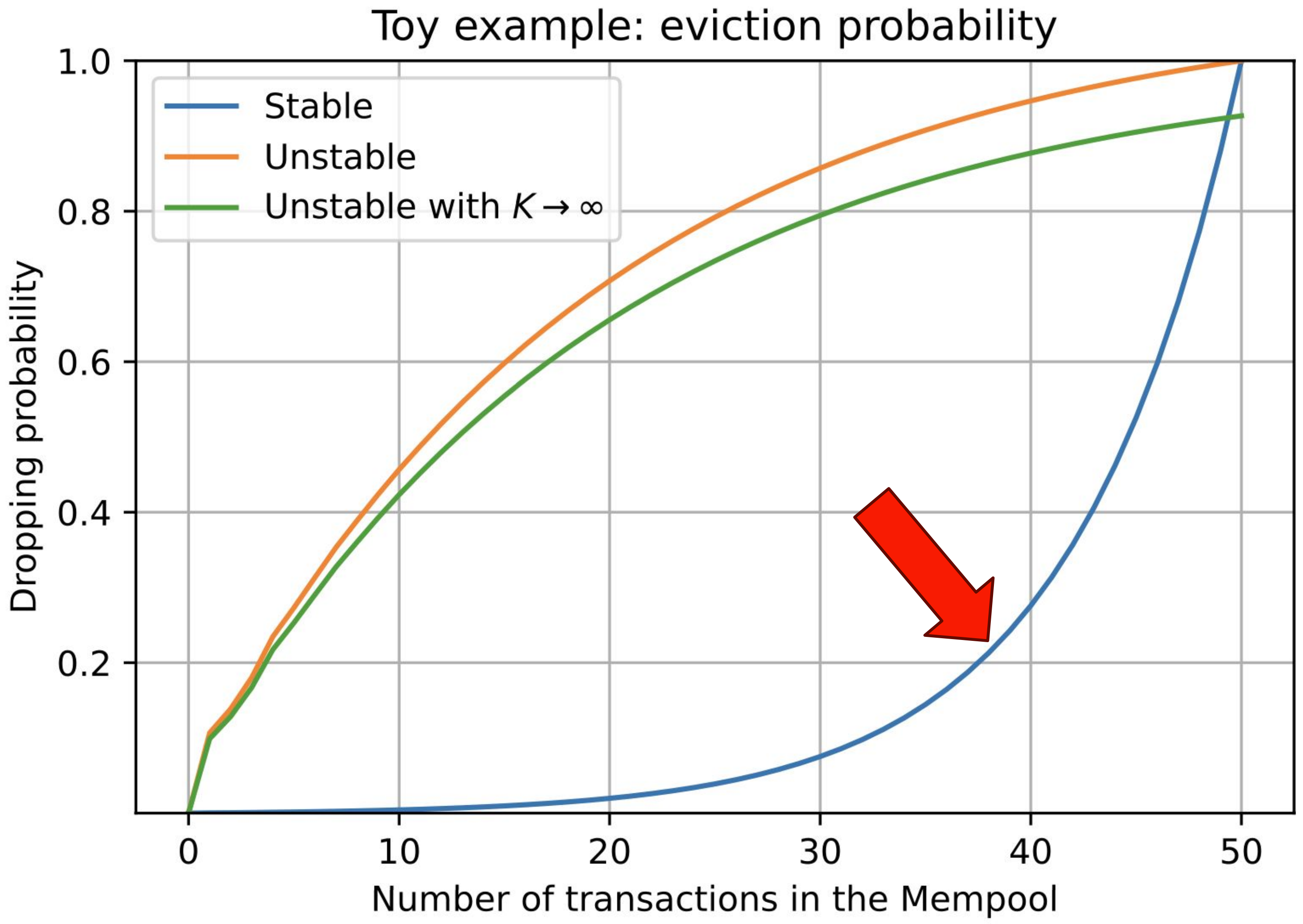
- Find the coefficients by solving the linear system

$$C_1 \simeq -3.500\,, \qquad C_2 \simeq -0.1561 - 0.054889j\,,$$
$$C_3 \simeq -0.156 + 0.05489j\,, \qquad C_4 \simeq 3.812\,.$$

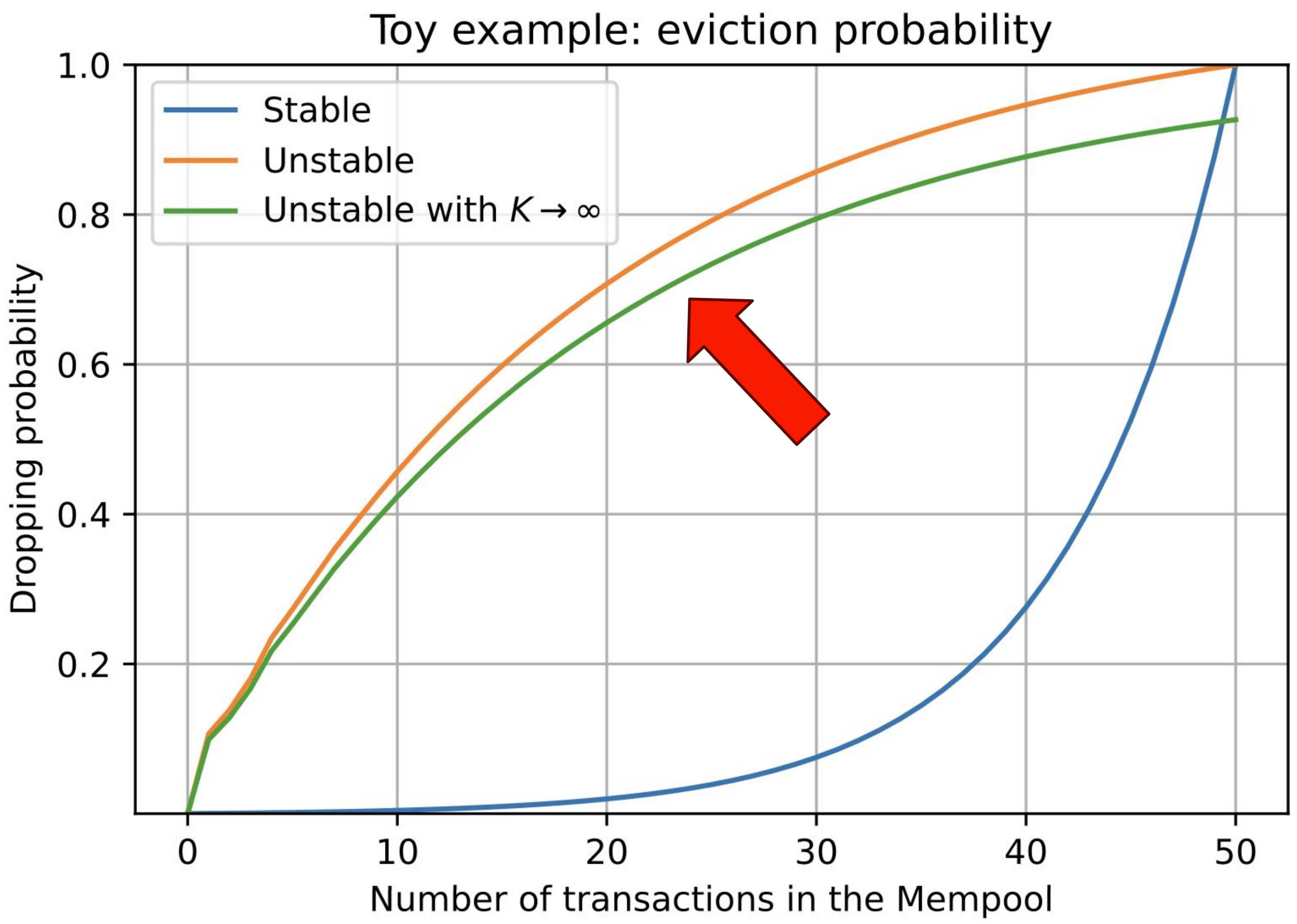
- Get the normalizing factor

$$T \simeq 2337.29155$$

Probability of dropping as function of the mempool state



The case of instability



Real world data

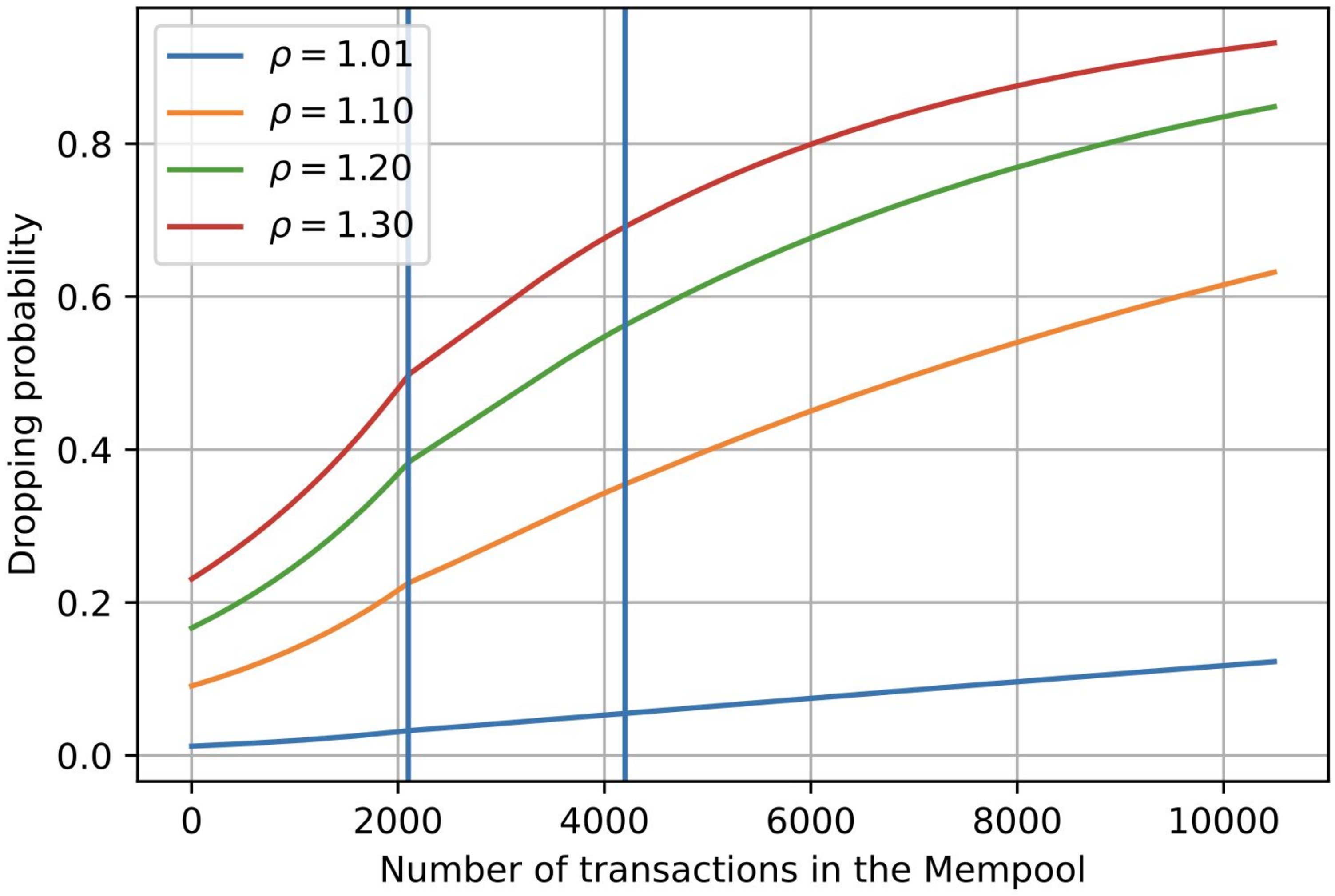
- We consider a dataset collected from Bitcoin
- We use our model to classify the transactions which are dropped and check its accuracy
- We compare the result with a random classifier: the first is totally unaware of the system state or dynamic, the second knows how many transactions will be dropped and makes a random guess

Comparison using Brier's score

TABLE II
BRIER SCORES FOR HEAVY AND MODERATE LOADS.

	Heavy load	Moderate load
Transaction class	$[1, 12]$ satoshis/B	$[1, 5]$ satoshis/B
Fraction confirmed	0.39	0.64
Fraction dropped	0.61	0.36
BS_{model}	0.134	0.161
BS_{rand}	0.465	0.431
BS_{oracle}	0.242	0.232
BSS	**0.447**	**0.306**

Dropping probabilities for different workload intensities



Conclusions

- New results for the Gambler's ruin model
- Use of numerical packages to find the roots of polynomial helps to tackle the numerical problems of the explicit solutions
- Application for delay tolerant transactions
- Bitcoin state at the moment is critical with the Mempool saturated fo cheap transactions