# Transaction Fee Optimization

## MAS.S62 Cryptocurrency Engineering & Design

Maxime Amram
Jeremy Toledano

May 14, 2018

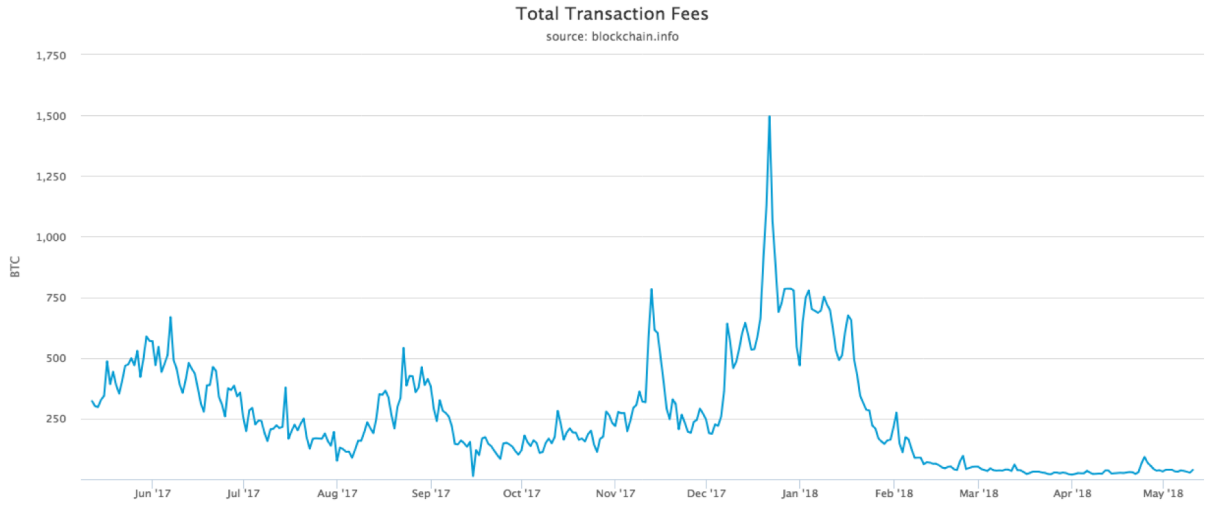

OPERATIONS RESEARCH CENTER

mit media lab

# Why do **fees** matter?

# Why do **fees** matter?

- **Fees** = implicit difference between inputs and outputs
- Decision left to user
- Paid to miners in **auction process**
- Compensation for **transaction space** (expressed in sat/byte)
- As block reward decreases, only **incentive for miners**
- Thus, fees are crucial for system **security** and **viability**



# How are **fees** calculated today?

# How are **fees** calculated today?

- Rule of thumb

# How are **fees** calculated today?

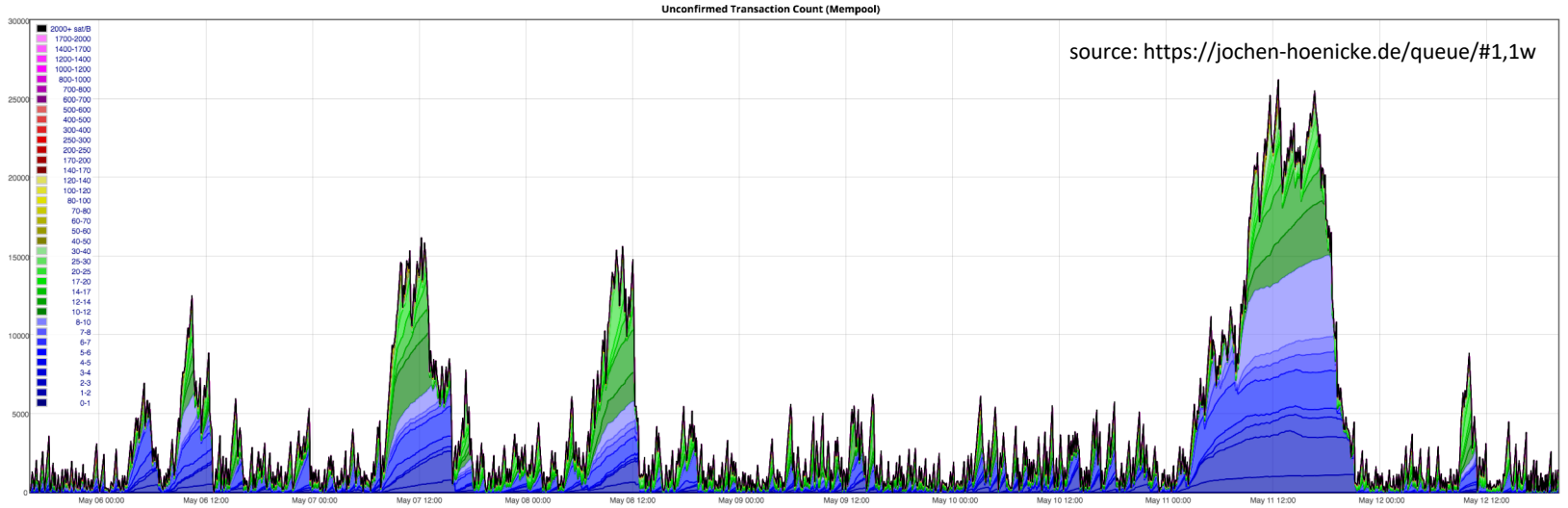
- **Rule of thumb**
- **History of transactions**
  - Built in Bitcoin Core since version 0.15
  - Two modes: **conservative** or **economical**
  - For a given block target, estimate fee rate based on **most recent mined blocks**
  - Reacts to volatility of fee market

# How are **fees** calculated today?

- **Rule of thumb**
- **History of transactions**
- **Trial and error**
  - Child Pays For Parent: create dependent transaction with higher fee
  - Replace By Fee: broadcast the same transaction with higher fee
  - Drawbacks:
    - Only works with **underestimations**
    - CPFP takes **unnecessary space** in a block and creates unnecessary **dependencies**
    - RBF is **double spending** and contributes to flooding the network

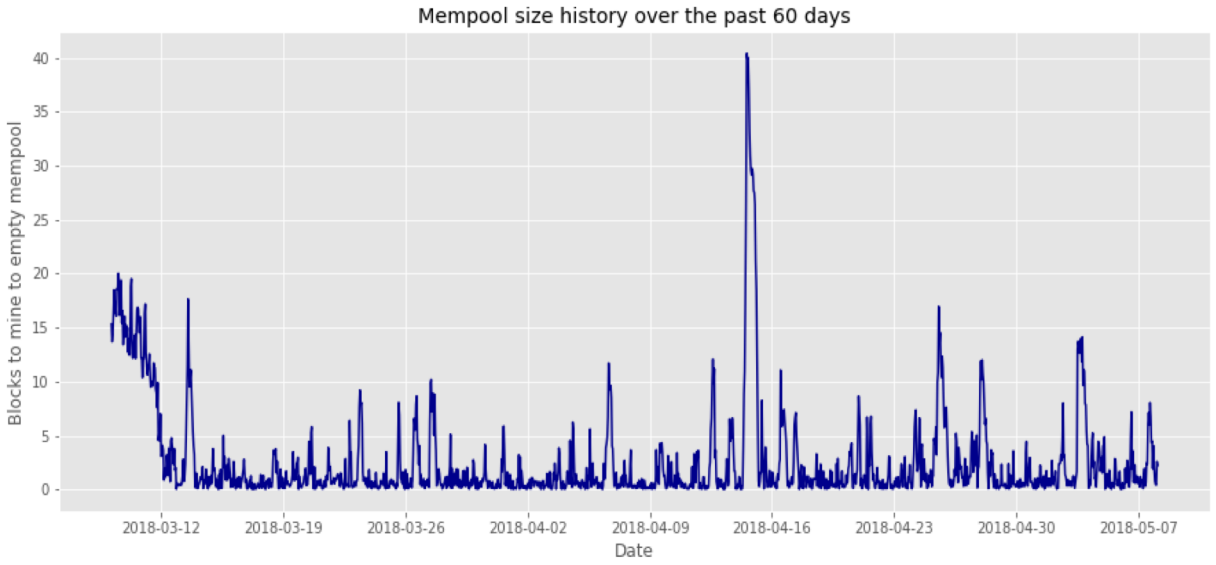
# How are **fees** calculated today?

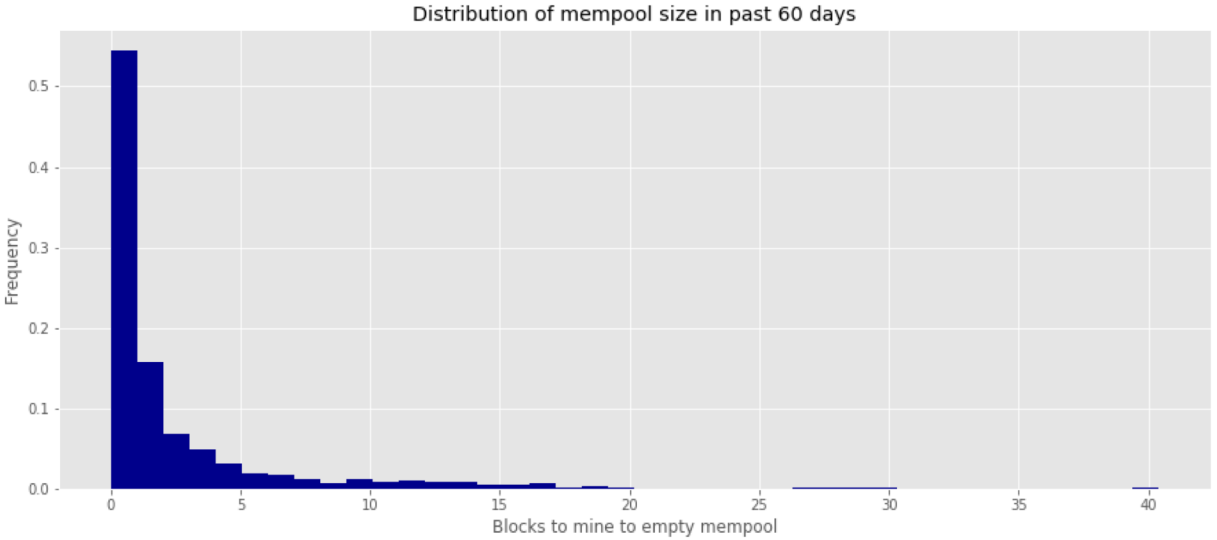
- Rule of thumb
- History of transactions
- Trial and error
- Mempool information
  - Key idea: estimate fees in the same way miners select transactions
  - Even more reactive than history-based methods



# Can we do better?

# Can we do better?

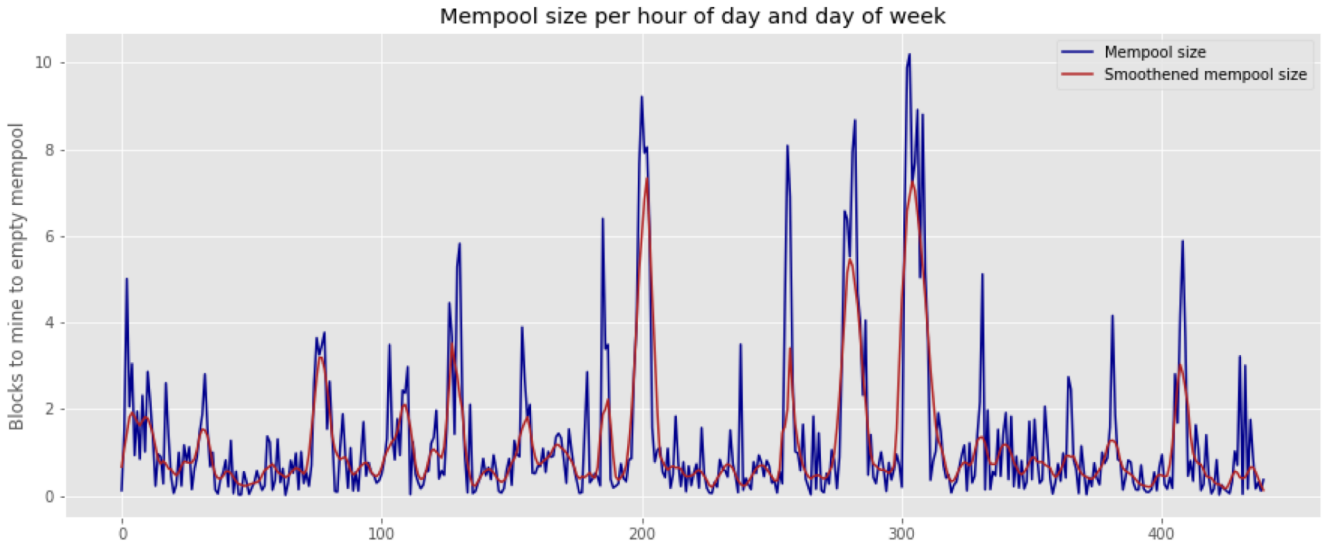


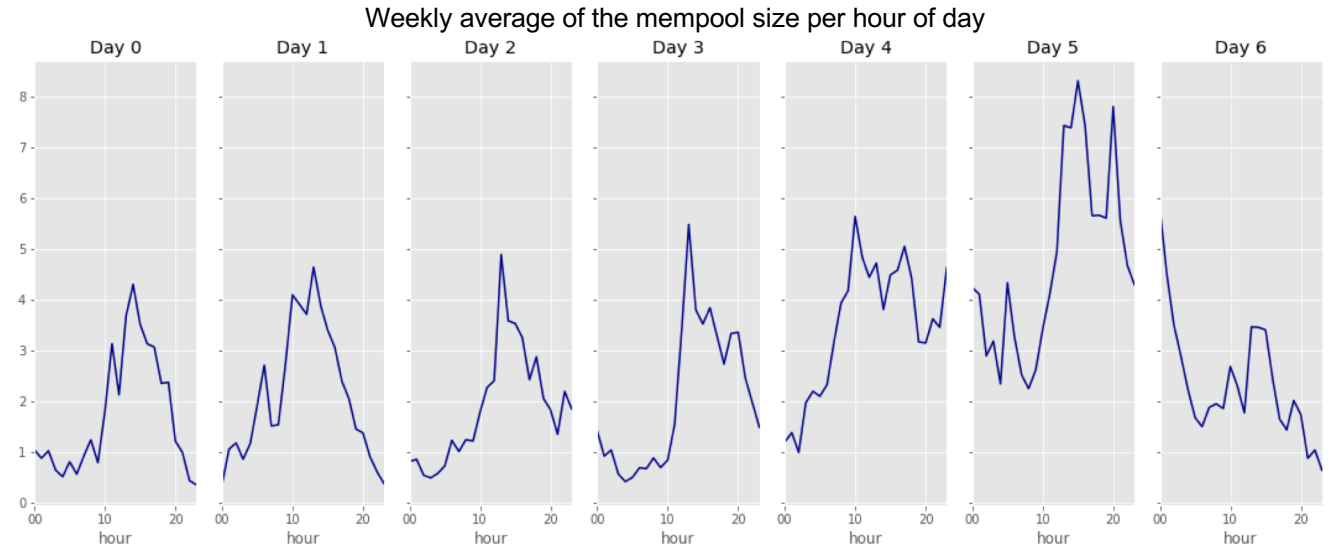




- More than **55%** of the time, there is less than **1 MB** of **transactions** in the **mempool**
- When this happens, **any positive fee transaction** gets in the next block
- What if we could **predict** the **future mempool state** and **anticipate those events**?

# Predicting **Future Mempool State**

# Predicting **Future Mempool State**



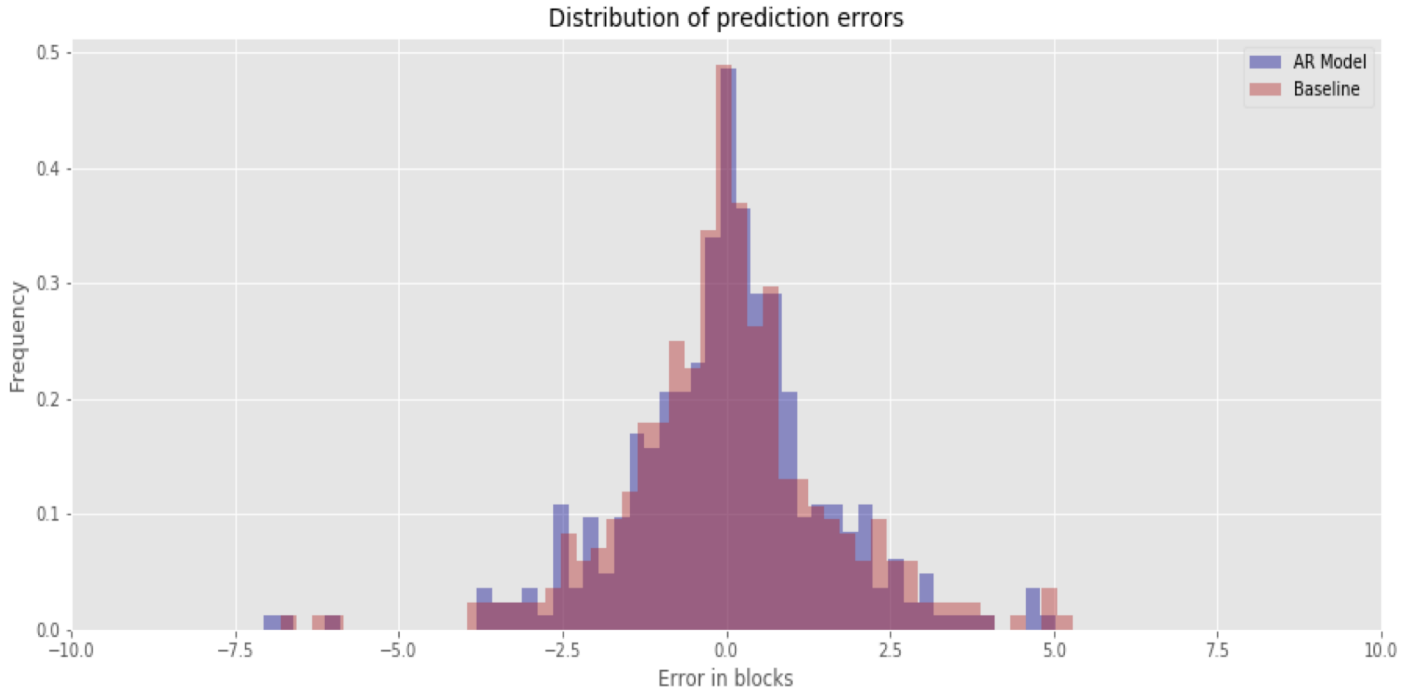

- There are **patterns** in the mempool size data (seasonal changes, weekly and daily trends)
- Goal: use **machine learning** and **deep learning** techniques to learn those patterns and **predict future mempool state**

# Predicting **Future Mempool State**

- **Machine learning models**
  - Target: mempool size in **next hour**
  - **Auto-Regression** (AR) with **Random Forests** and **Ridge Regression**

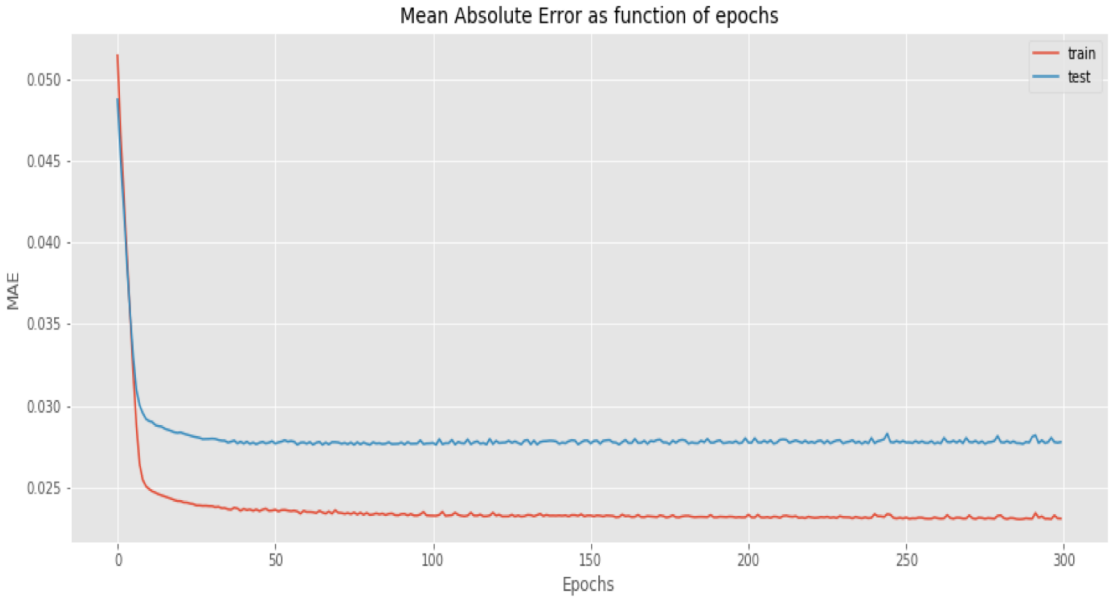
| Model | Out-of-Sample $R^2$ |
|---|---|
| Baseline | 0.7216 |
| Random Forest | 0.7089 |
| Ridge Regression | **0.7340** |

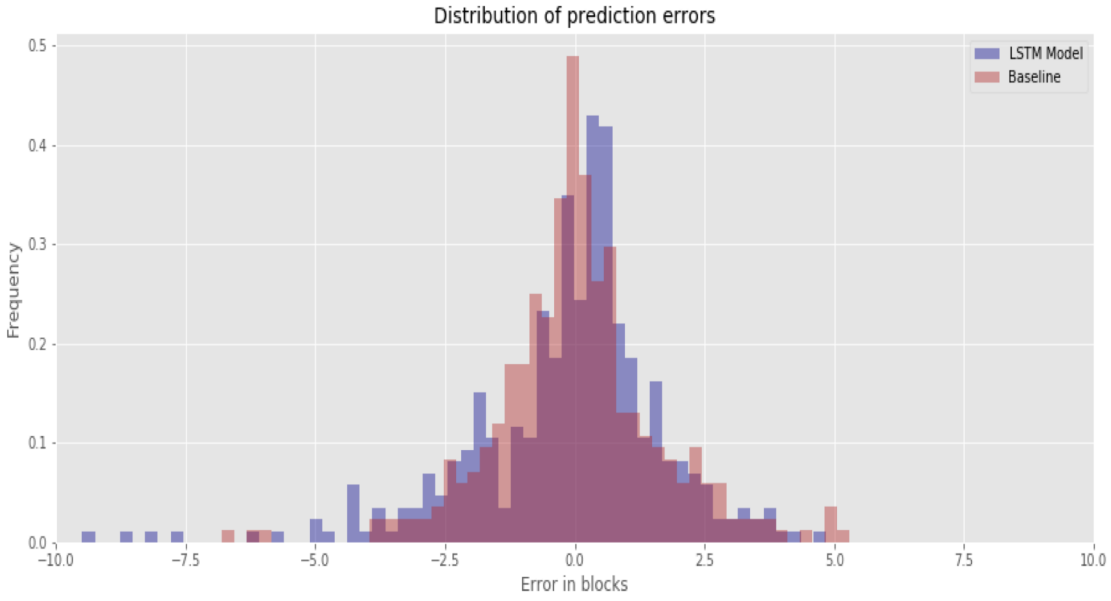
*Baseline model: prediction of future mempool size is its current size*



# Predicting **Future Mempool State**

- Machine learning models
- Deep learning model
  - Long Short-Term Memory (**LSTM**) neural network
  - $R^2_{LSTM} = 0.7396 > R^2_{Ridge\ Regression} > R^2_{Baseline}$



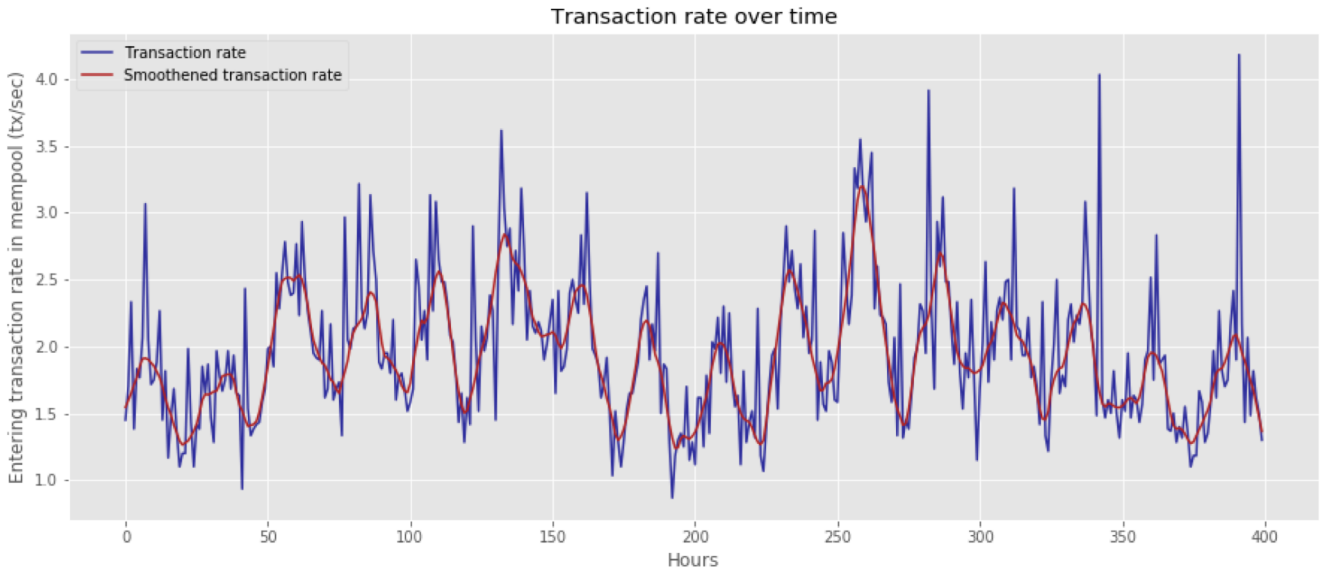



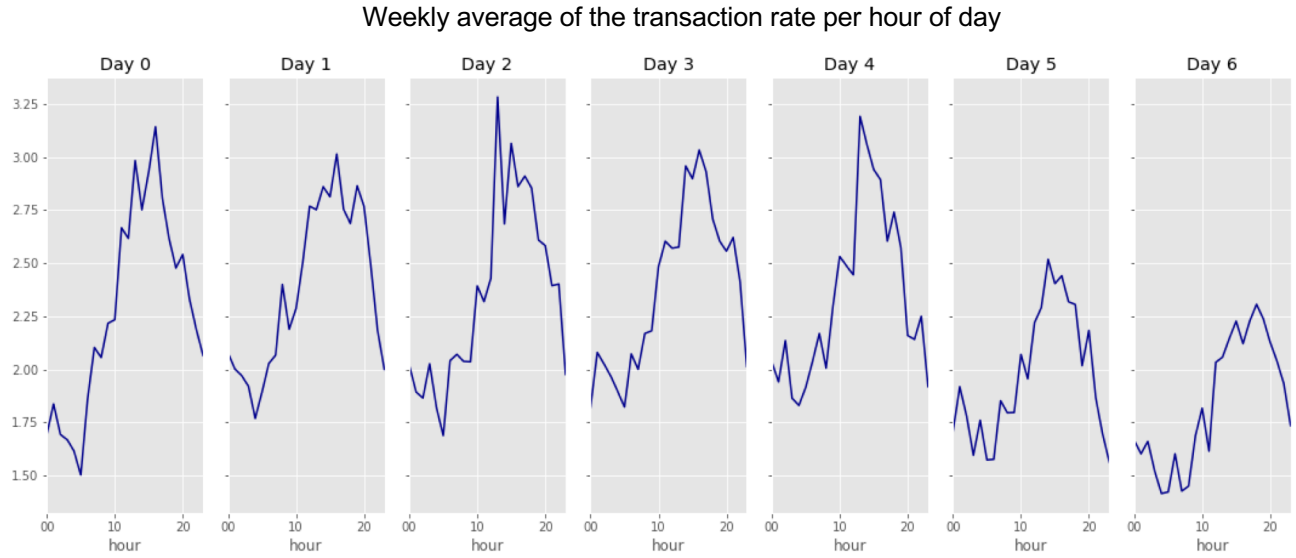
# Predicting **Future Mempool State**

- **Machine learning models**
- **Deep learning model**
- **Results**
  - LSTM yields the best results, not significantly above baseline
  - Explanations:
    - Neural networks need **more data** to learn complex patterns
    - **Randomness** in block mining blurs the signal and makes it hard to learn
  - Precision is **not sufficient** to solve the problem
  - Idea: **predict transaction arrival rate**, which has less randomness

# Predicting Transaction Arrival Rate & Mempool State

# Predicting **Transaction Arrival Rate** & **Mempool State**



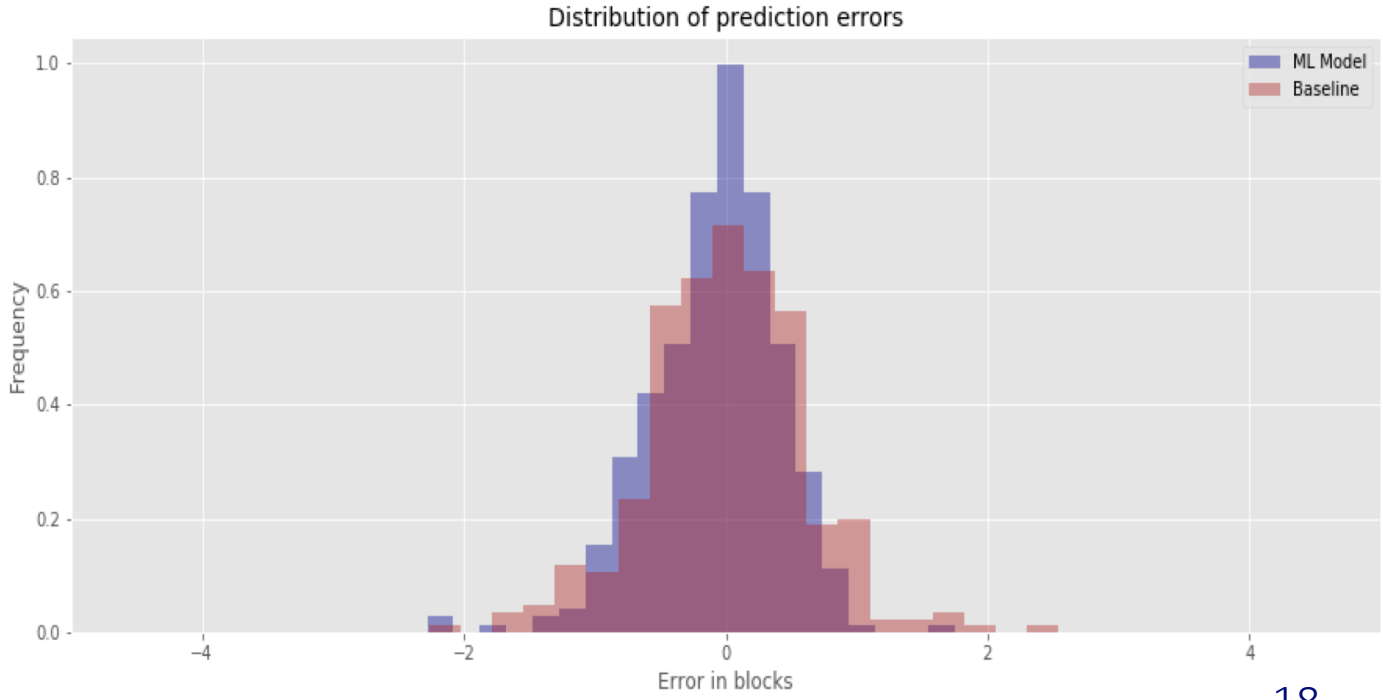

- For **transaction rate**, patterns are clearer than for mempool size data
- Clear **distinction** between **weekday** and **weekend** activity

- New machine learning opportunity
- Let us learn **transaction rate** first, and use it to predict **mempool state**

# Predicting **Transaction Arrival Rate** & **Mempool State**

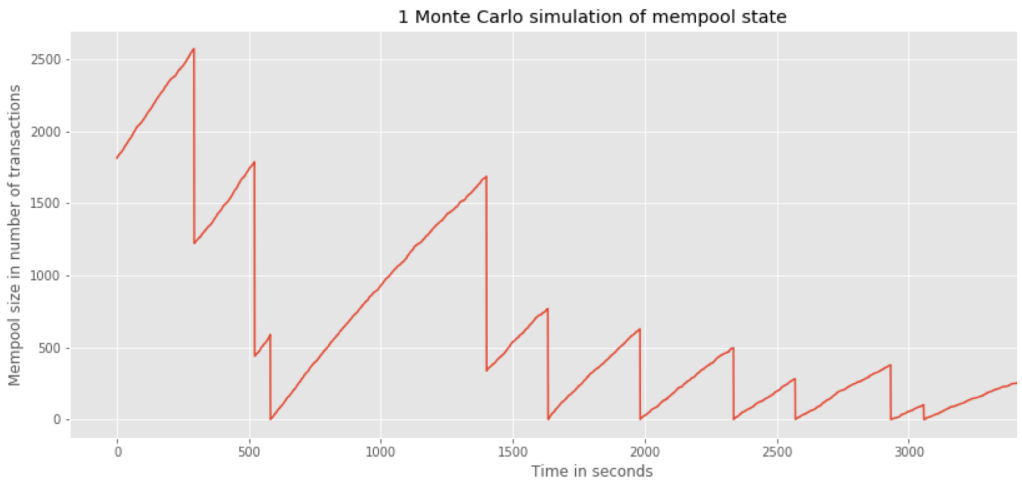
- **Machine/deep learning models**
  - Target: transaction rate in **next hour**
  - **Auto-Regression Ridge** is the best model
  - Much better results than baseline

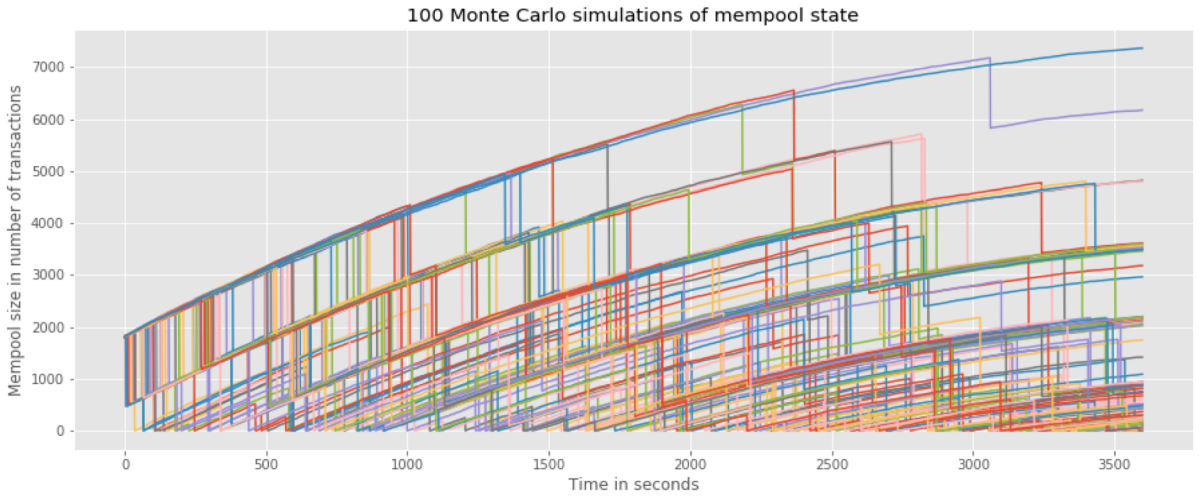
| Model | Out-of-Sample $R^2$ |
|---|---|
| Baseline | 0.0456 |
| Random Forest | 0.3353 |
| Ridge Regression | **0.3801** |
| LSTM | 0.3745 |



# Predicting **Transaction Arrival Rate** & **Mempool State**

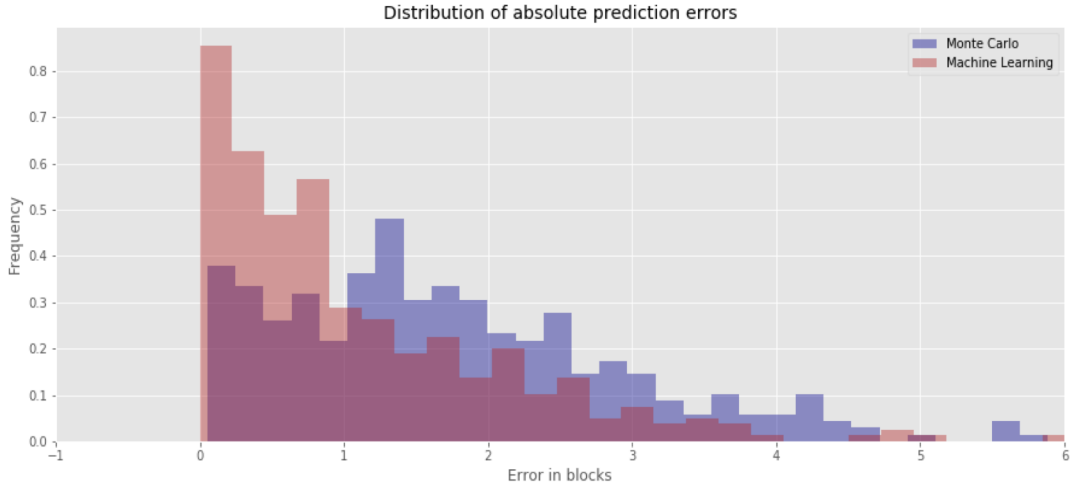
- **Machine/deep learning models**
- **Monte Carlo simulation**
  - Model mining and transactions arriving in mempool as a **slowly time-varying Poisson process**
  - Model mempool state as **Markovian queueing process**

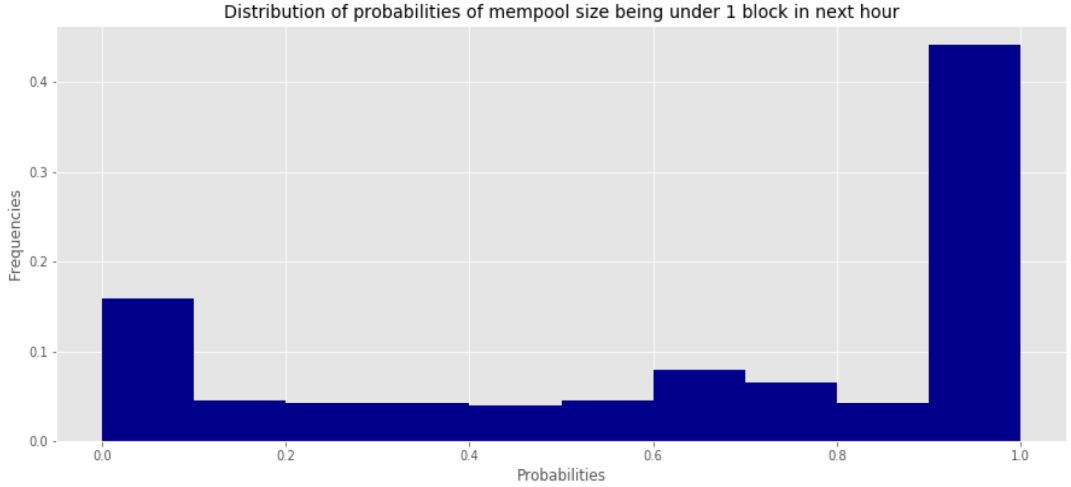


# Predicting **Transaction Arrival Rate** & **Mempool State**

- **Machine/deep learning models**
- **Monte Carlo simulation**
- **Results**
  - Simulation is not very accurate to predict future mempool state
  - However, it gives access to other **interesting metrics**:
    - Probability of mempool being below 1 block in next X minutes
    - Average time before mempool size is below 1 block



# Summary

# Summary

- **Directly predict future mempool state**
  - Not enough precision
- **Predict future rate of transactions entering the mempool, and use Monte Carlo to get insights about future mempool state**
  - Dynamic estimate as Monte Carlo includes future arrival rate
  - New metrics computable
- **Use case**
  - When changes are sudden and other fee estimation methods fail
  - E.g. after a surge in mempool and transaction rate back to normal

# Thank you!
# Questions?


OPERATIONS RESEARCH CENTER

mit media lab