

GRID Storage Optimization in Transparent and User-Friendly Way for LHCb datasets

Mikhail Hushchyn, Andrey Ustyuzhanin, Philippe Charpentier, Christophe Haen

22nd International Conference on Computing in High Energy and Nuclear Physics,
San Francisco, October 10-14, 2016

Data Popularity

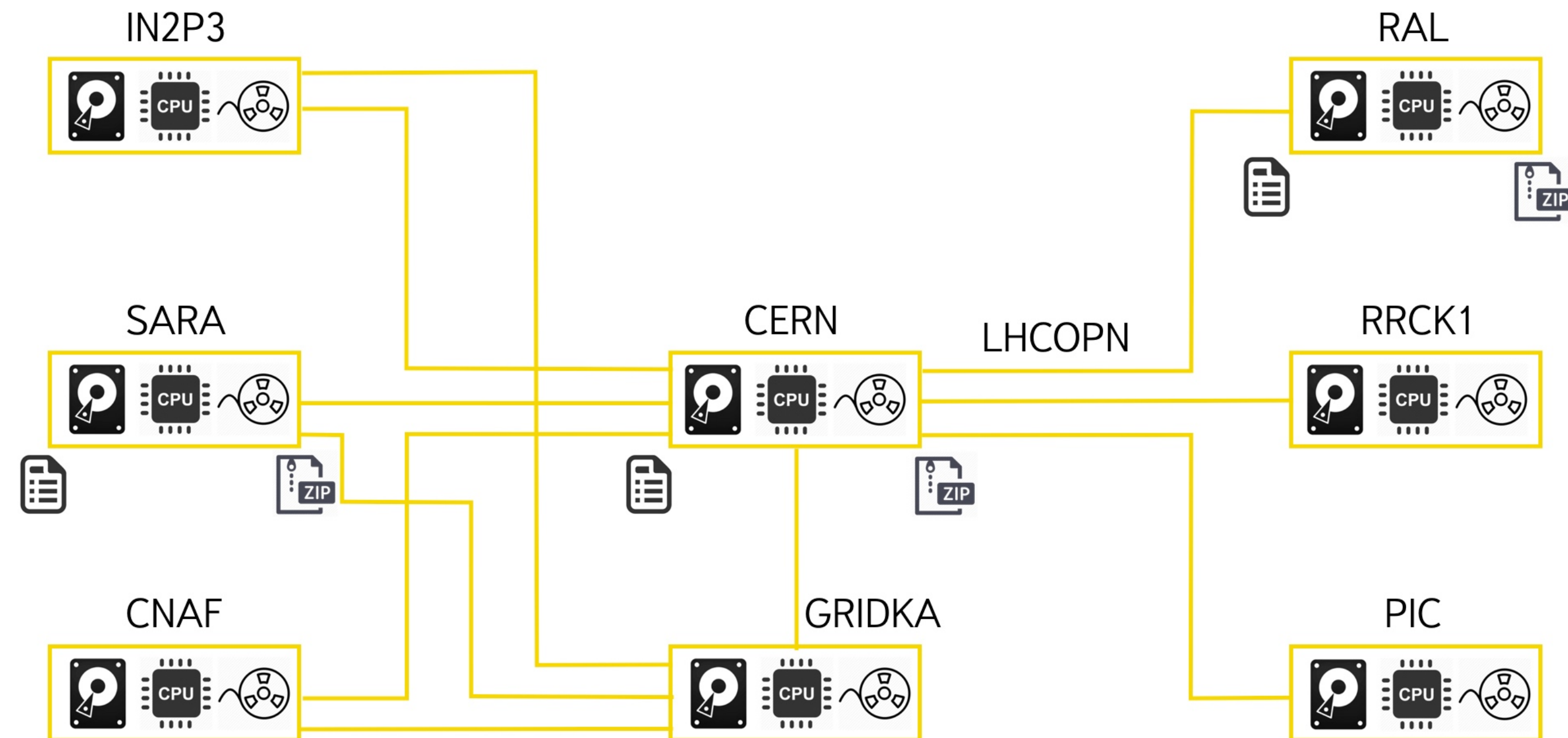
Problem Statement



Problem Statement

The disk storage capacities are limited. Thus, number of replicas for the unpopular files should be reduced or they should be totally removed from the disks.

This study provides a replication strategy for the LHCb based on the data popularity prediction using ML techniques.



Data Popularity

Related Works



Related Works

1. Thomas Beermann, etc. A Popularity-Based Prediction and Data Redistribution Tool for ATLAS Distributed Data Management, 2014

- › Uses Artificial Neural Networks to predict a dataset number of accesses for the next week
- › Removing unpopular replicas
- › New replicas for the popular datasets

2. Valentin Kuznetsov, etc. Predicting CMS Dataset Popularity, ACAT, 2016

- › CMS dataset popularity in up-coming week is predicted using classifiers
- › Several 'popularity' definitions
- › Several ML algorithms are used

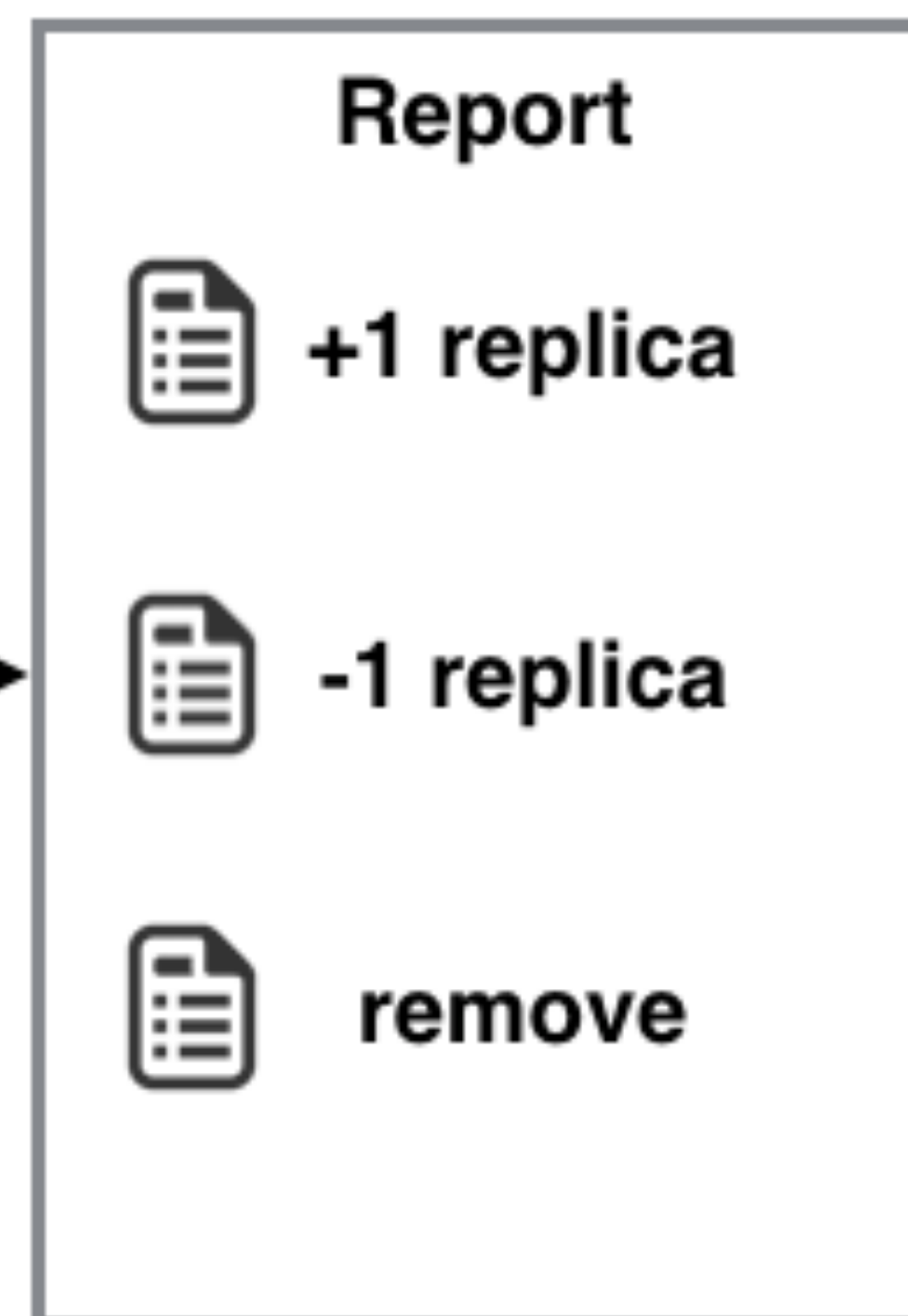
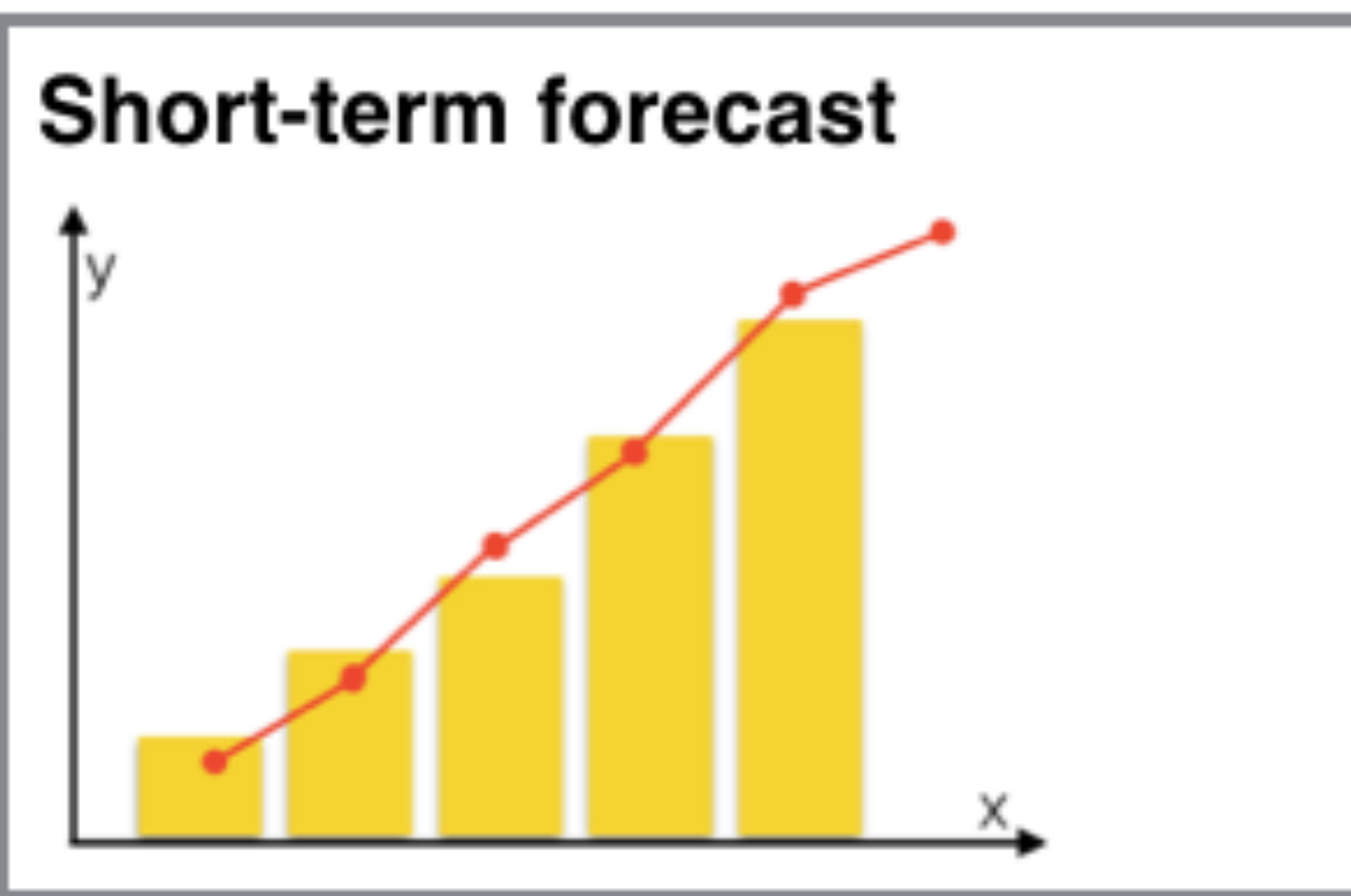
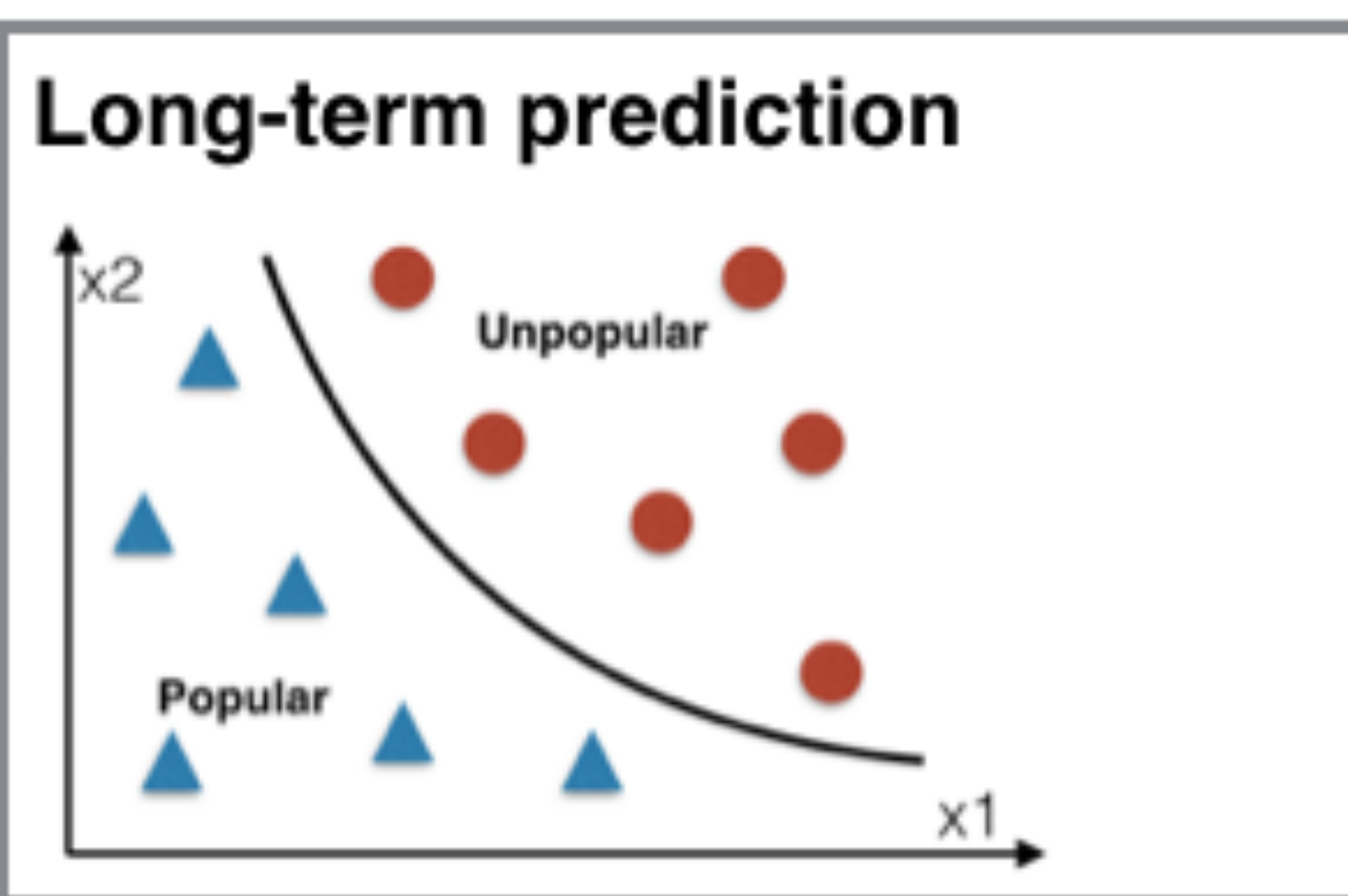
3. Mikhail Hushchyn, etc. Disk Storage Management for LHCb Based on Data Popularity Estimator, CHEP, 2015

- › A dataset long-term popularity is predicted using a classifier and a regressor
- › A lot of parameters, too complicated models, non-user-friendly

Data Popularity

General concept





Data Popularity 3.0

Long-term prediction



Long-term prediction

Random Forest Classifier is used to predict that a data file will be used in future period of time.

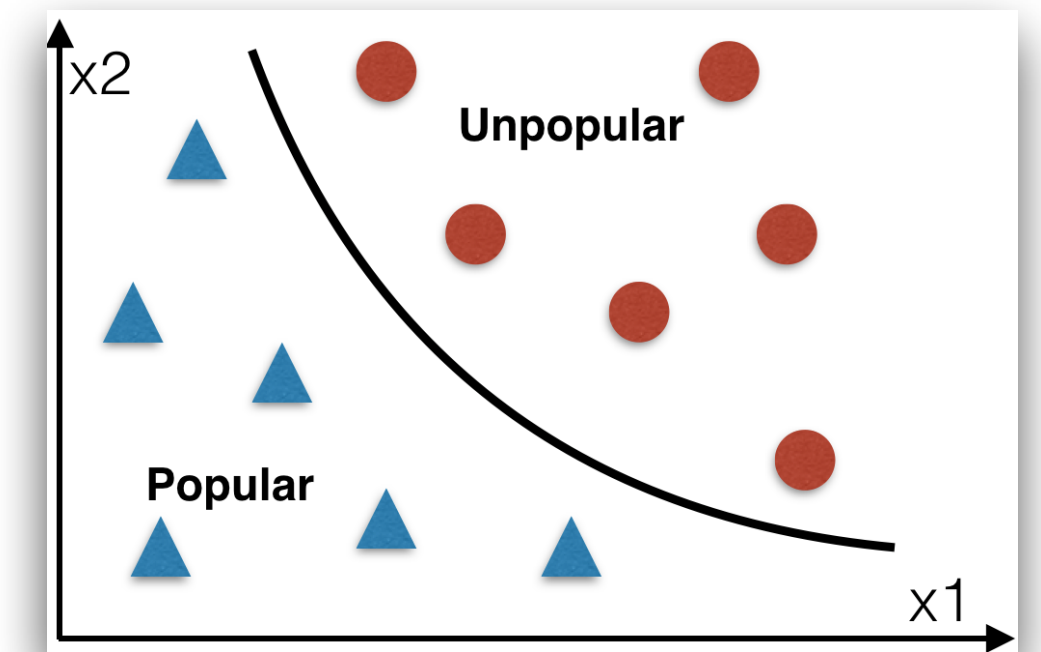
In this study, the data access history and its metadata for the 2.5 years are used.

Labels of the files:

- > 'Popular' if a file will be used during the next 6 months
- > 'Unpopular' otherwise

Features:

- > recency,
- > reuse distance,
- > time of the first access,
- > creation time,
- > access frequency,
- > type,
- > extension and size.



Long-term prediction

Random Forest Classifier is used to predict that a data file will be used in future period of time.

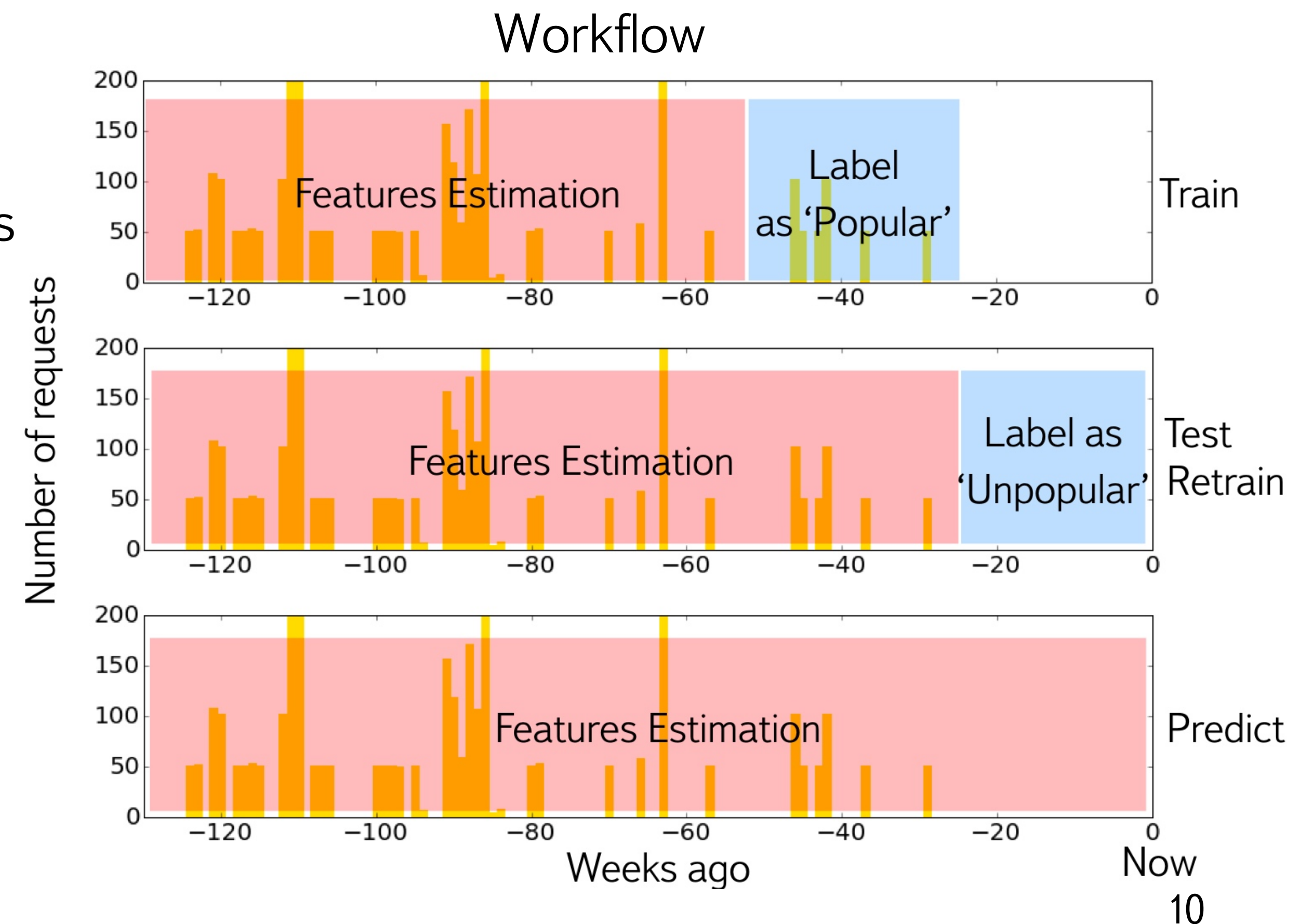
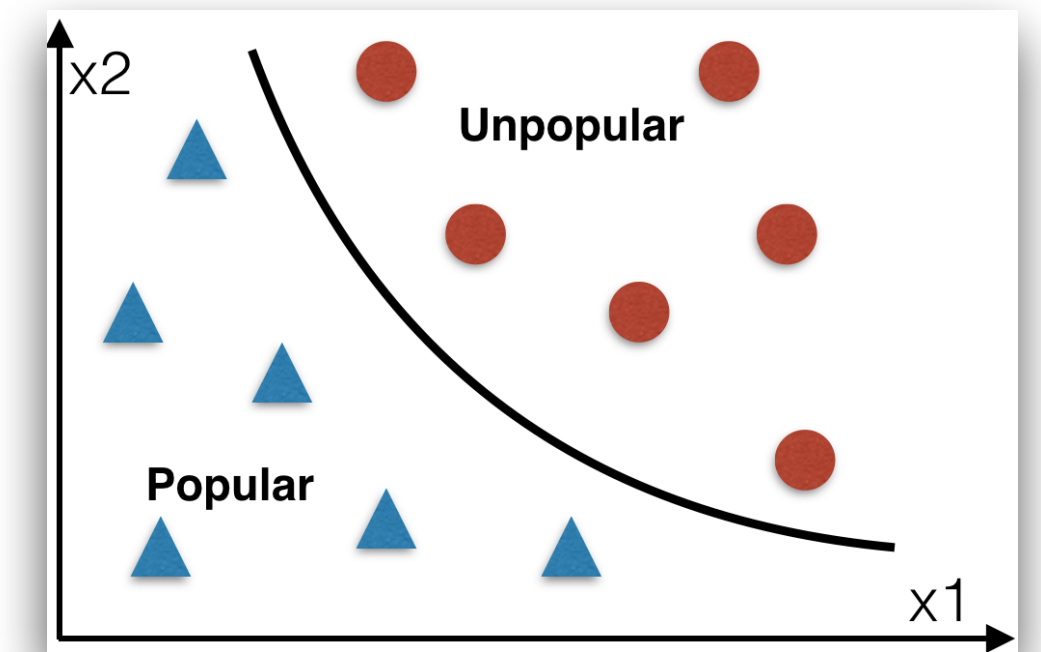
In this study, the data access history and its metadata for the 2.5 years are used.

Labels of the files:

- > 'Popular' if a file will be used during the next 6 months
- > 'Unpopular' otherwise

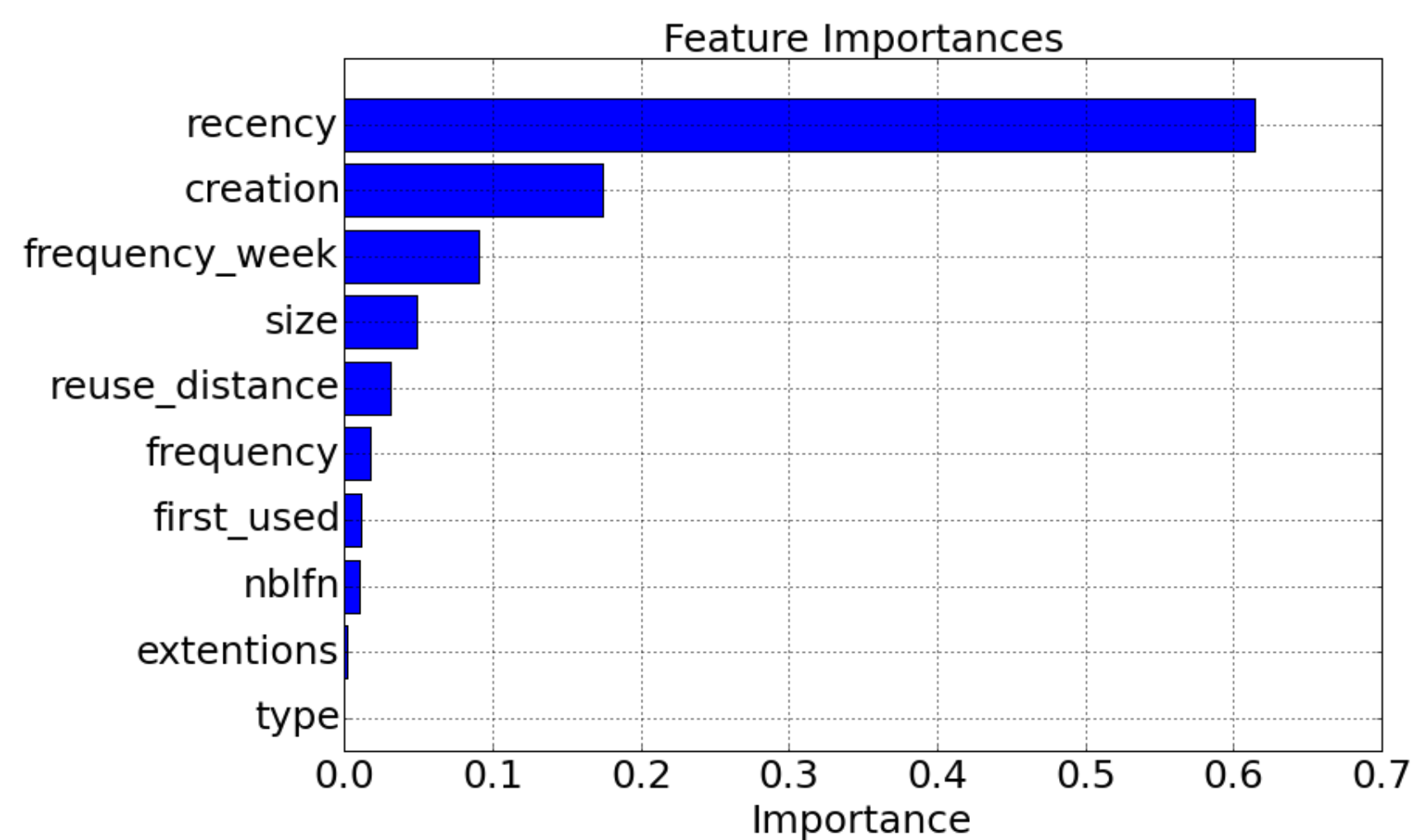
Features:

- > recency,
- > reuse distance,
- > time of the first access,
- > creation time,
- > access frequency,
- > type,
- > extension and size.

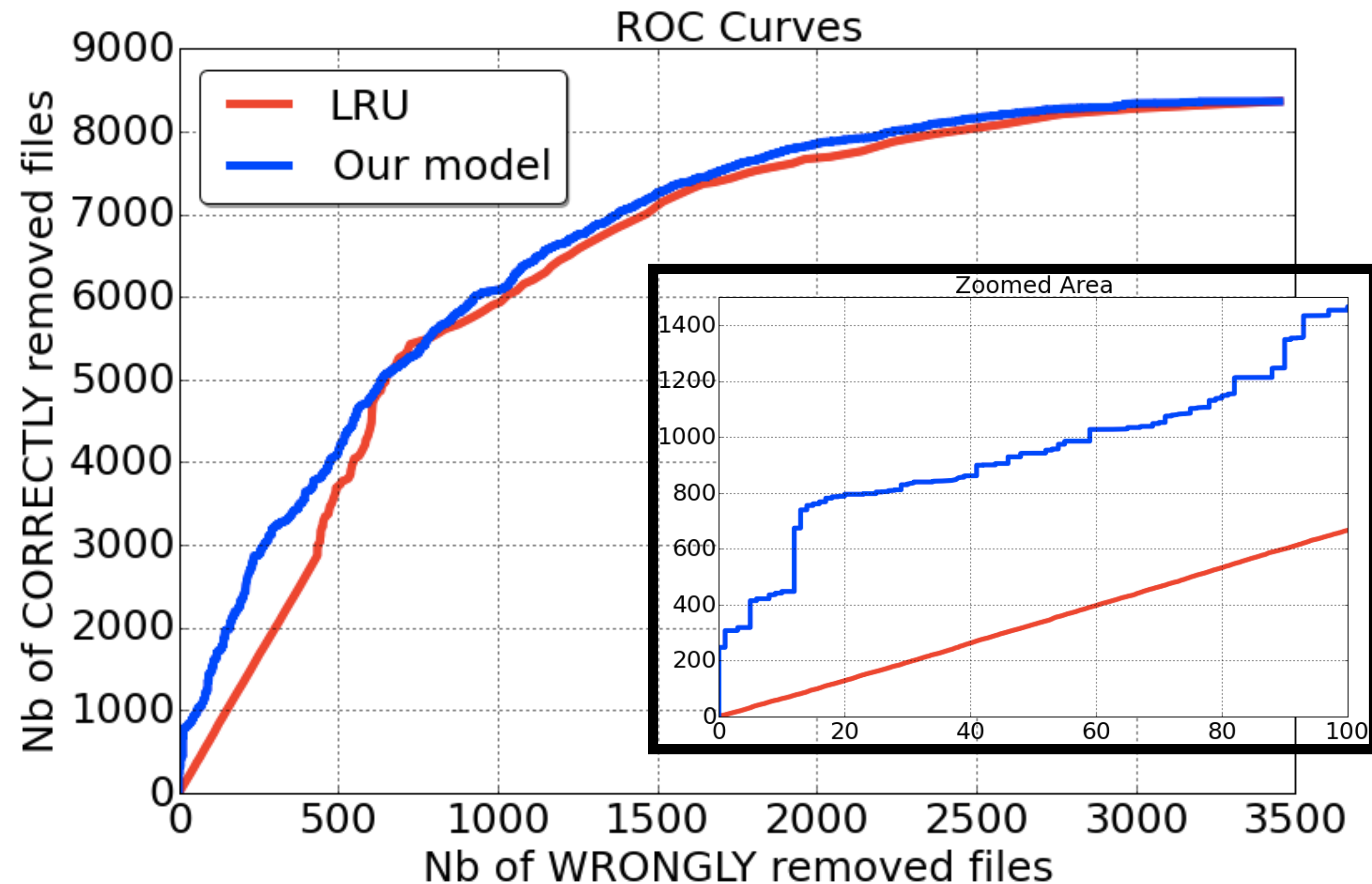


Long-term prediction

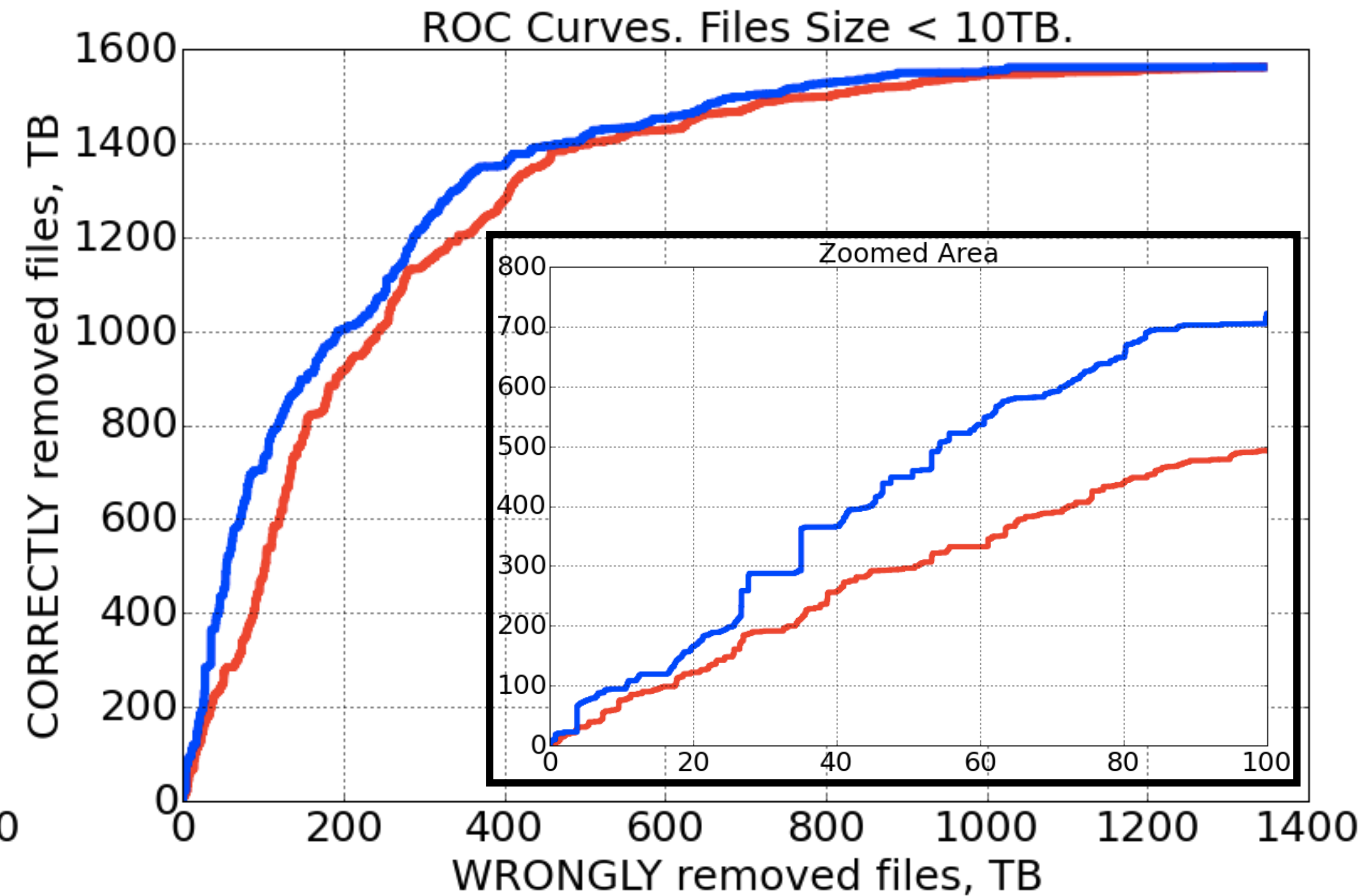
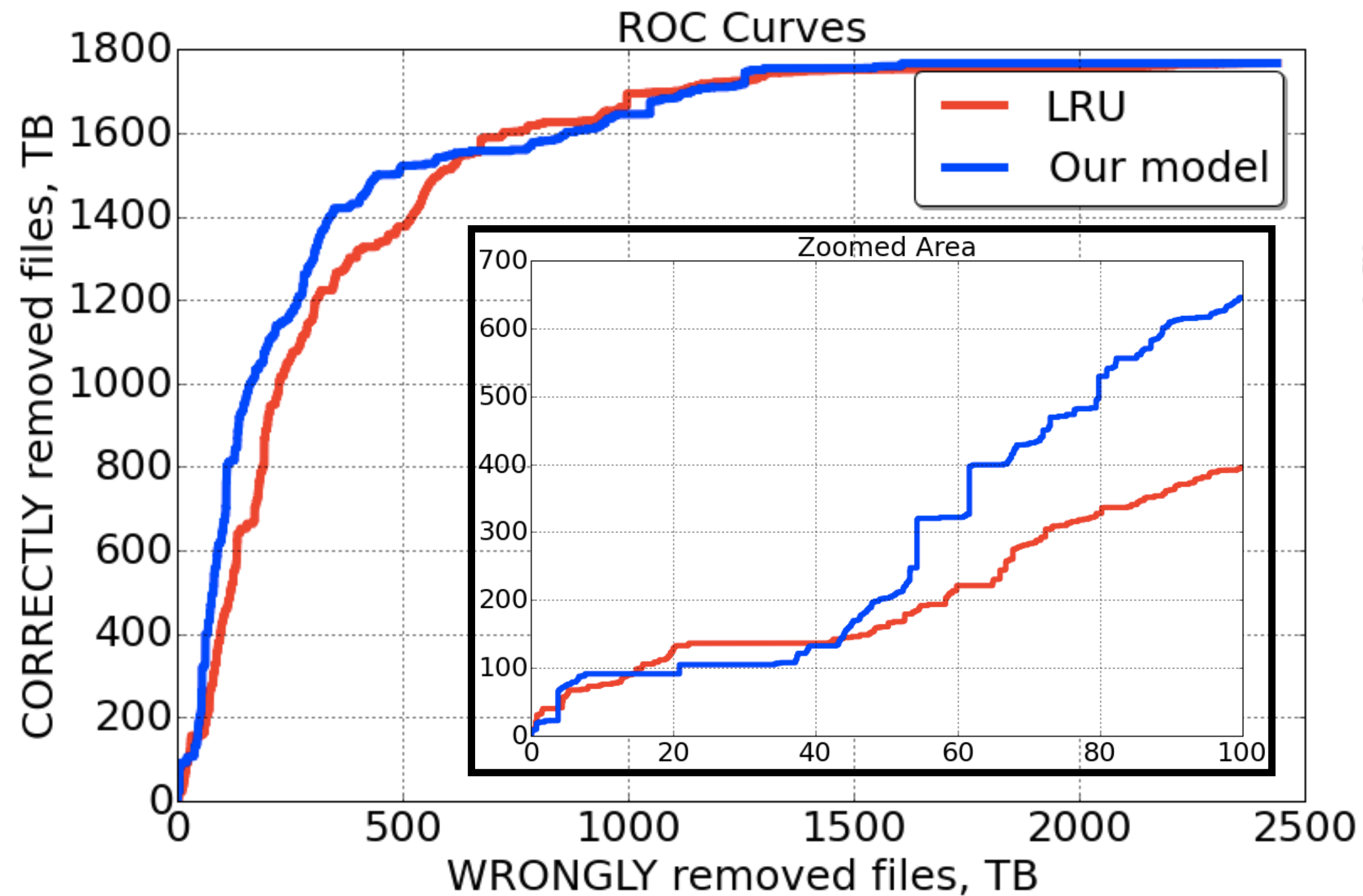
The model is compared with Least Recently Used (LRU) algorithm.



Based on the classifier output, **our model** allows to remove more files from the disks correctly than **LRU** with the same number of mistakes.



Long-term prediction



Number of files with size $\geq 10\text{TB}$ is 0.5%

Data Popularity 3.0

Short-term forecast



Short-term forecast

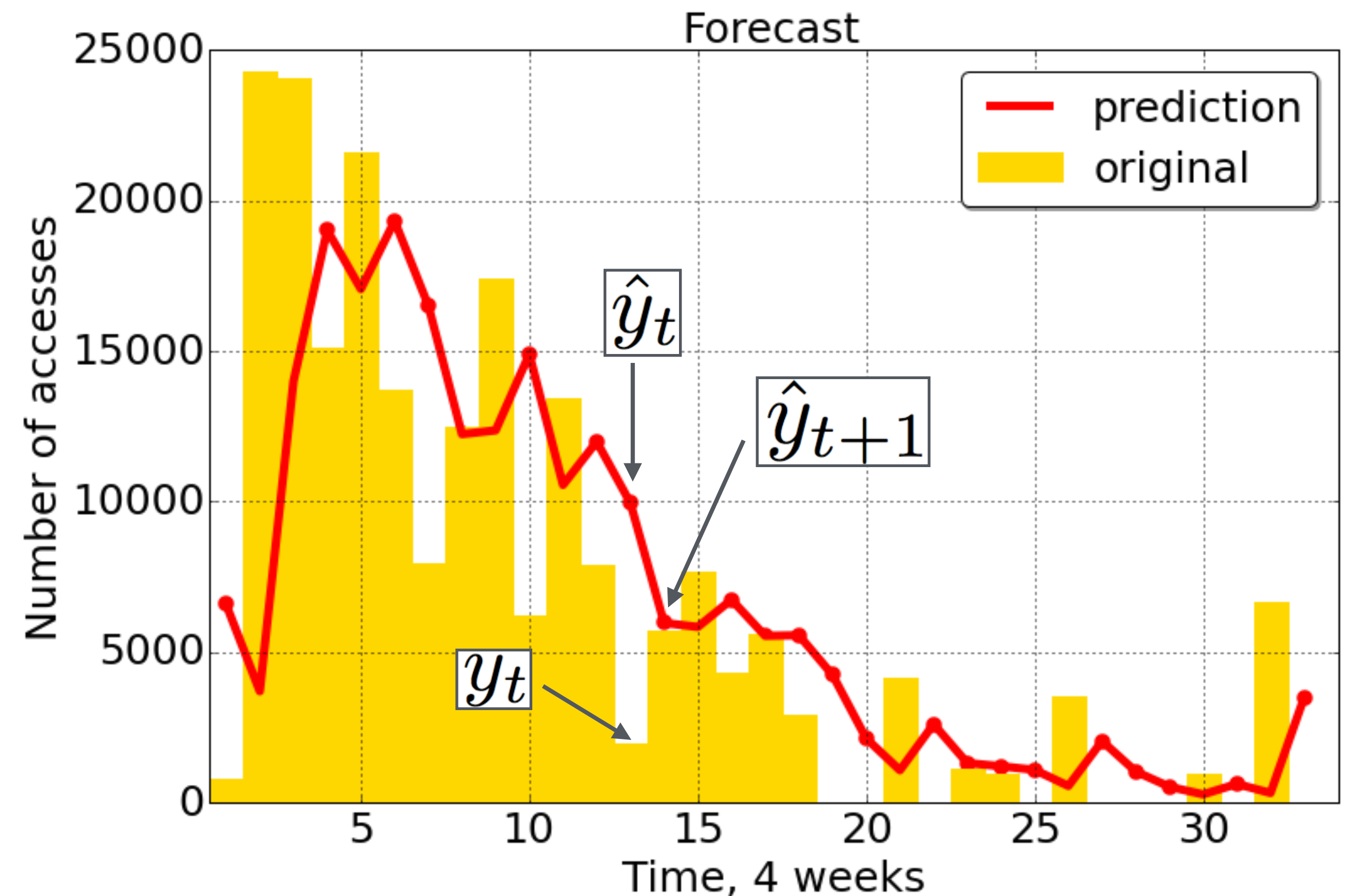
For a data file number of accesses prediction Brown's simple exponential smoothing model was used.

The model is defined as:

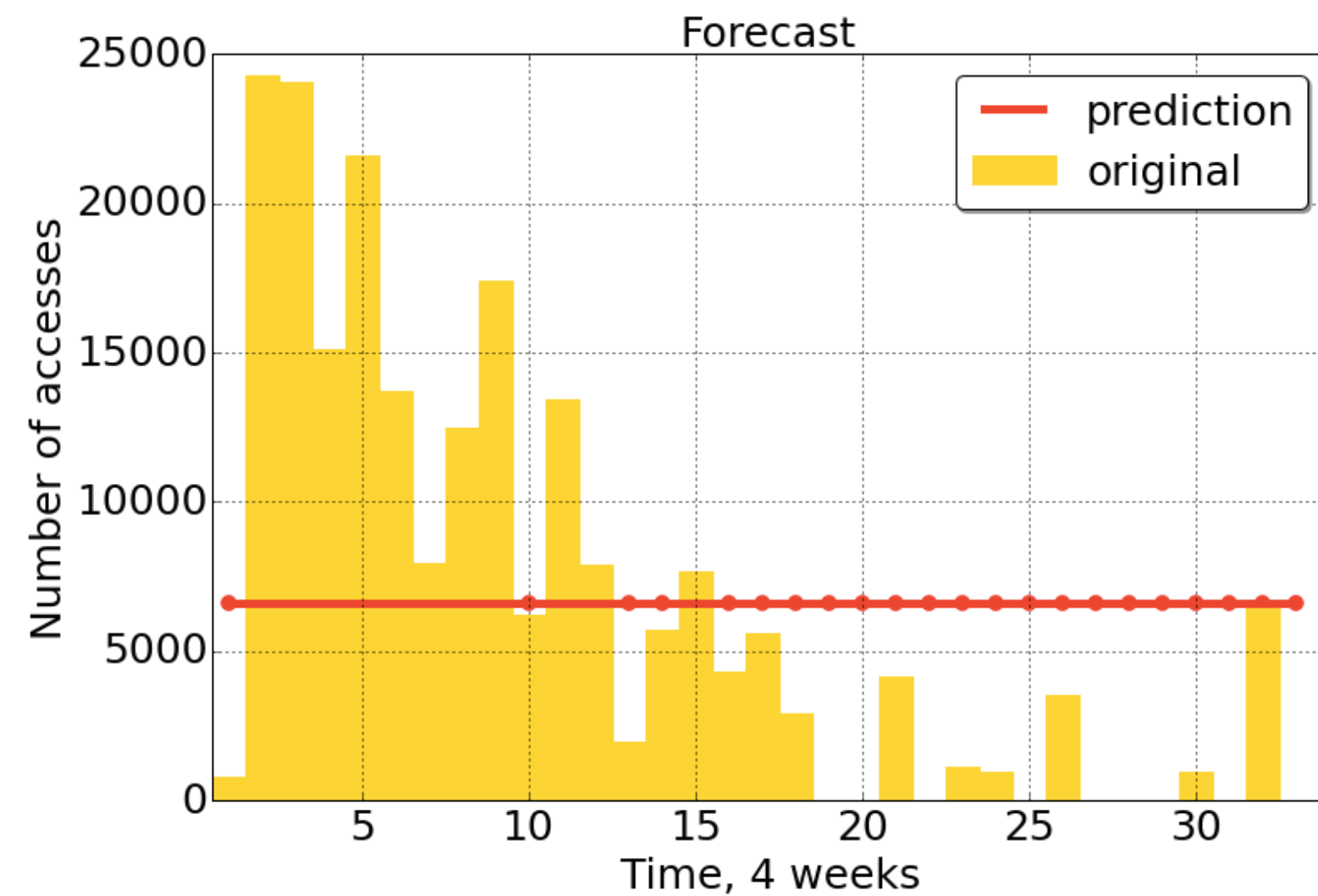
$$\hat{y}_{t+1} = \hat{y}_t + \alpha(y_t - \hat{y}_t)$$

$$\alpha = \operatorname{argmin} \sum_t (\hat{y}_t - y_t)^2$$

$$\hat{y}_0 = \frac{1}{n} \sum_{i=0}^n y_i, \quad \alpha \in (0, 1)$$



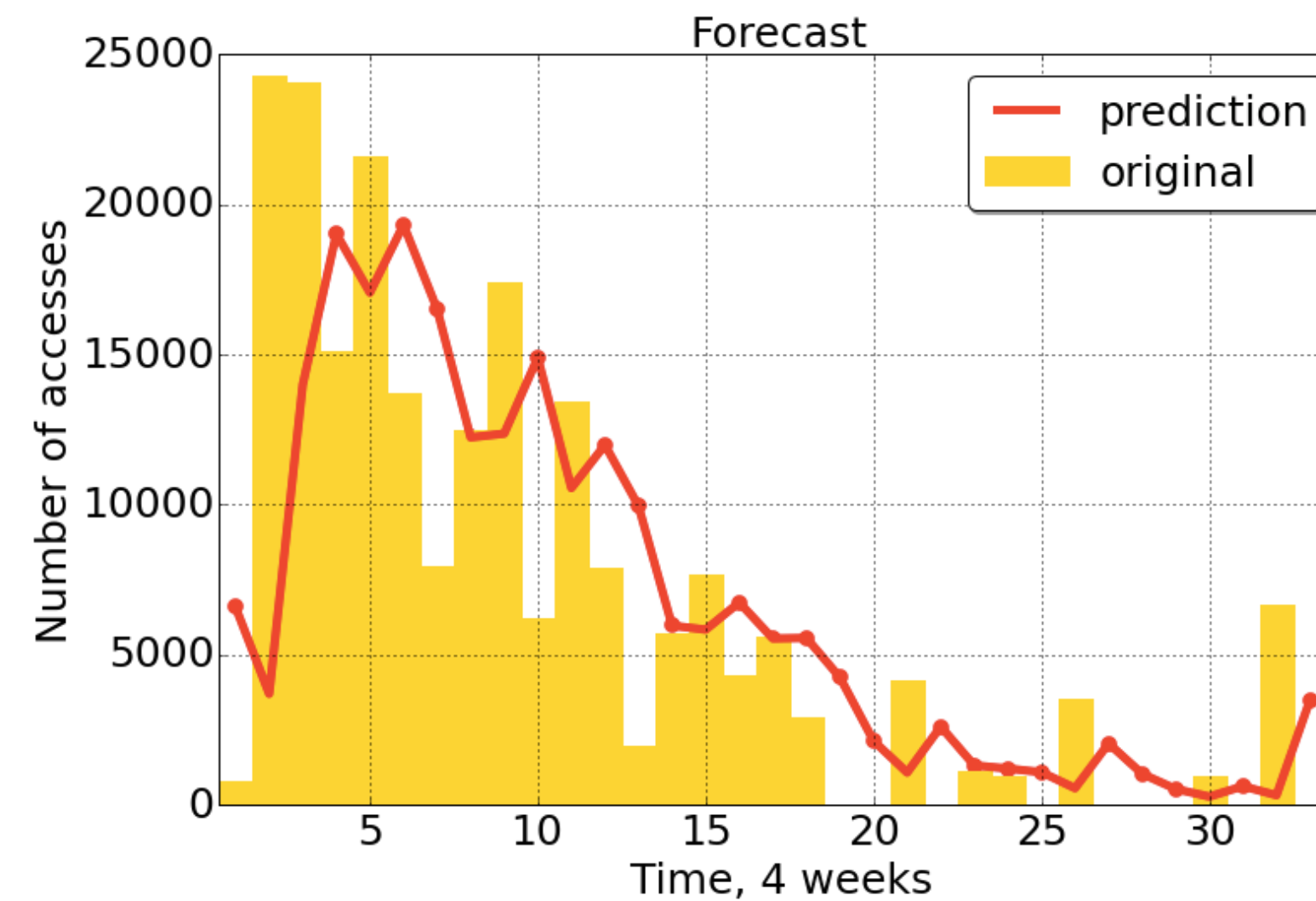
Brown's Model



$\alpha = 0$

$$\hat{y}_{t+1} = \hat{y}_t$$

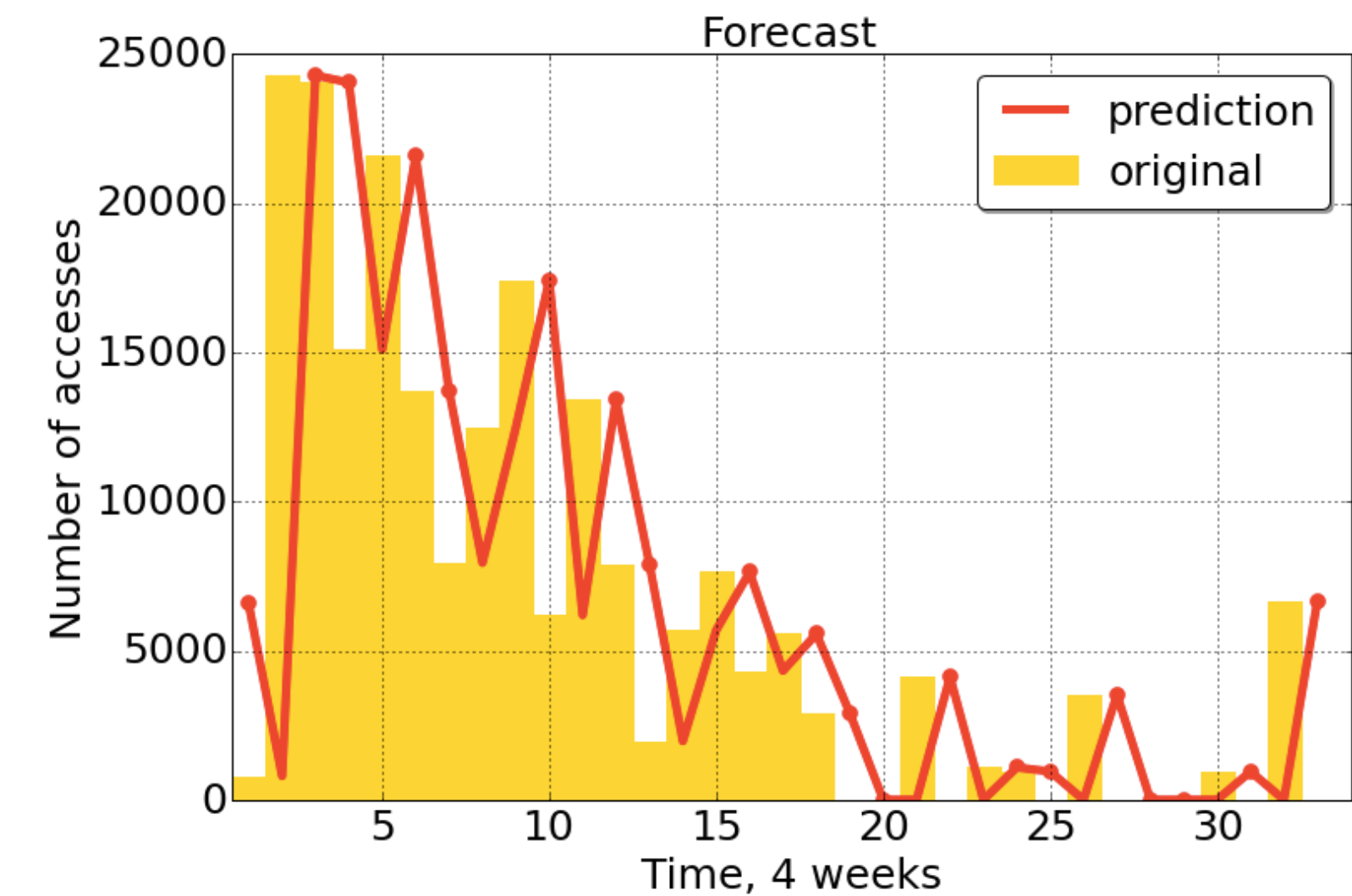
Average model



optimal alpha

$$\hat{y}_{t+1} = \hat{y}_t + \alpha(y_t - \hat{y}_t)$$

Brown's model



$\alpha = 1$

$$\hat{y}_{t+1} = y_t$$

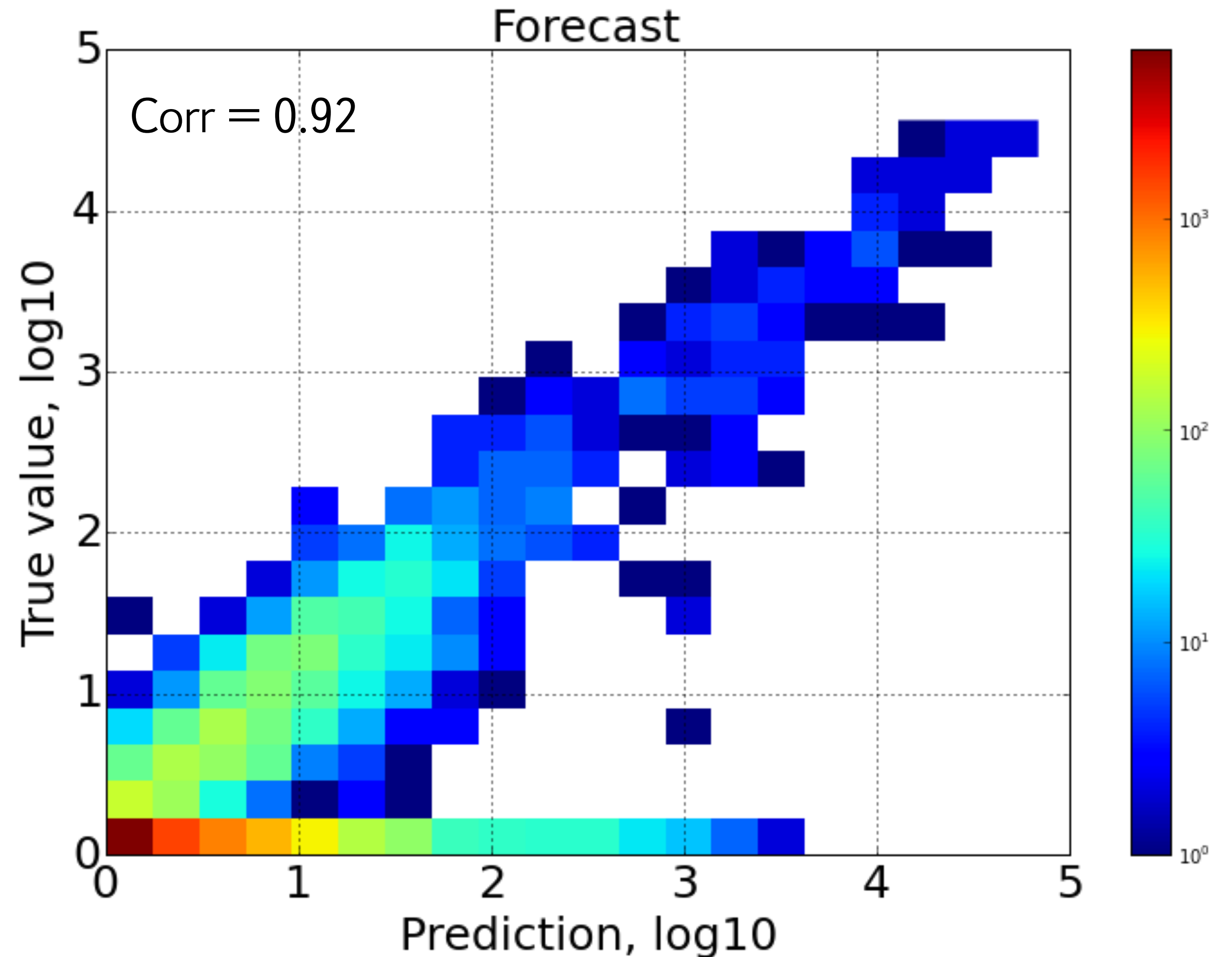
Static model

Brown's model is simple, widely-used and works well with short and sparse time series.

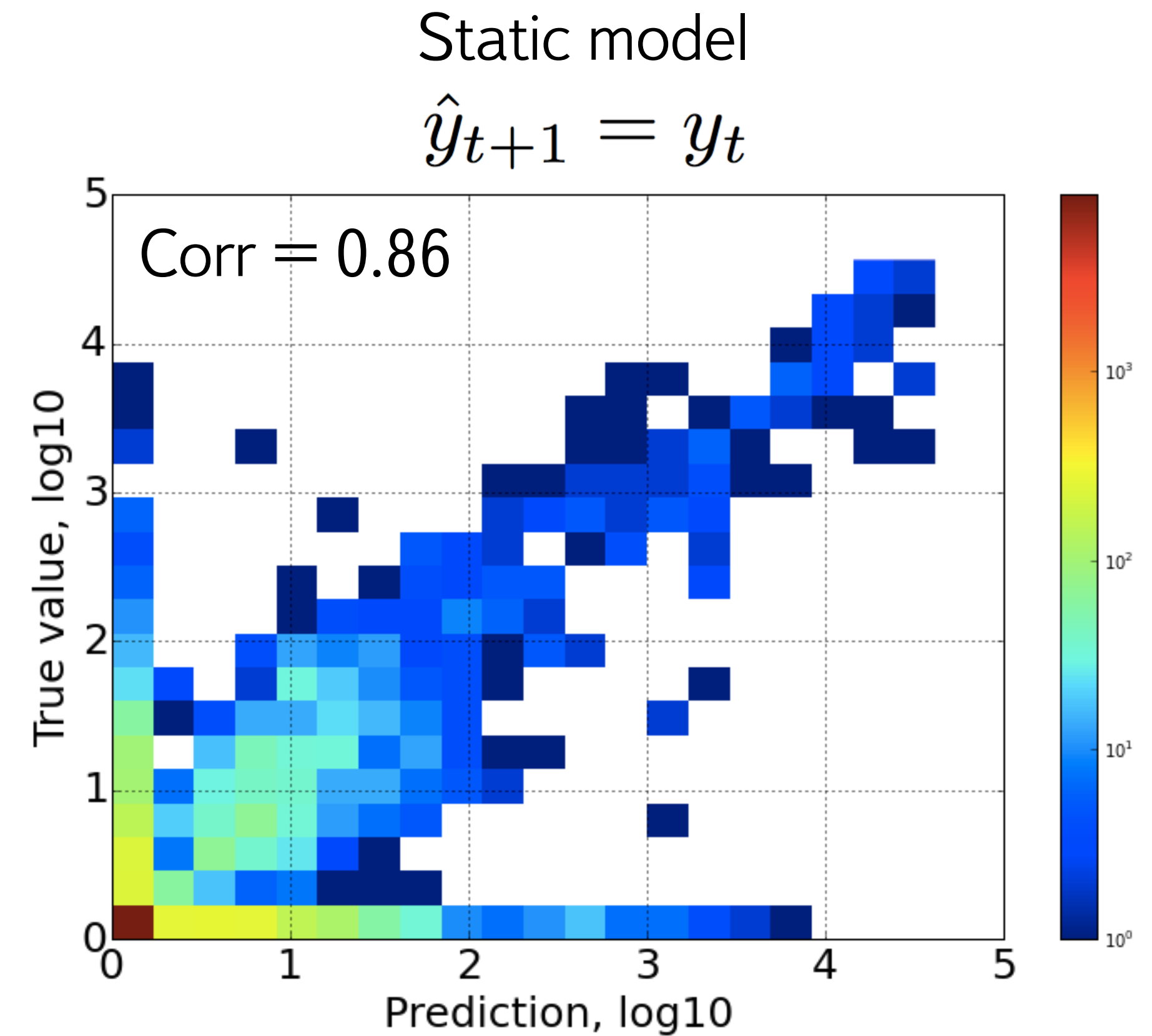
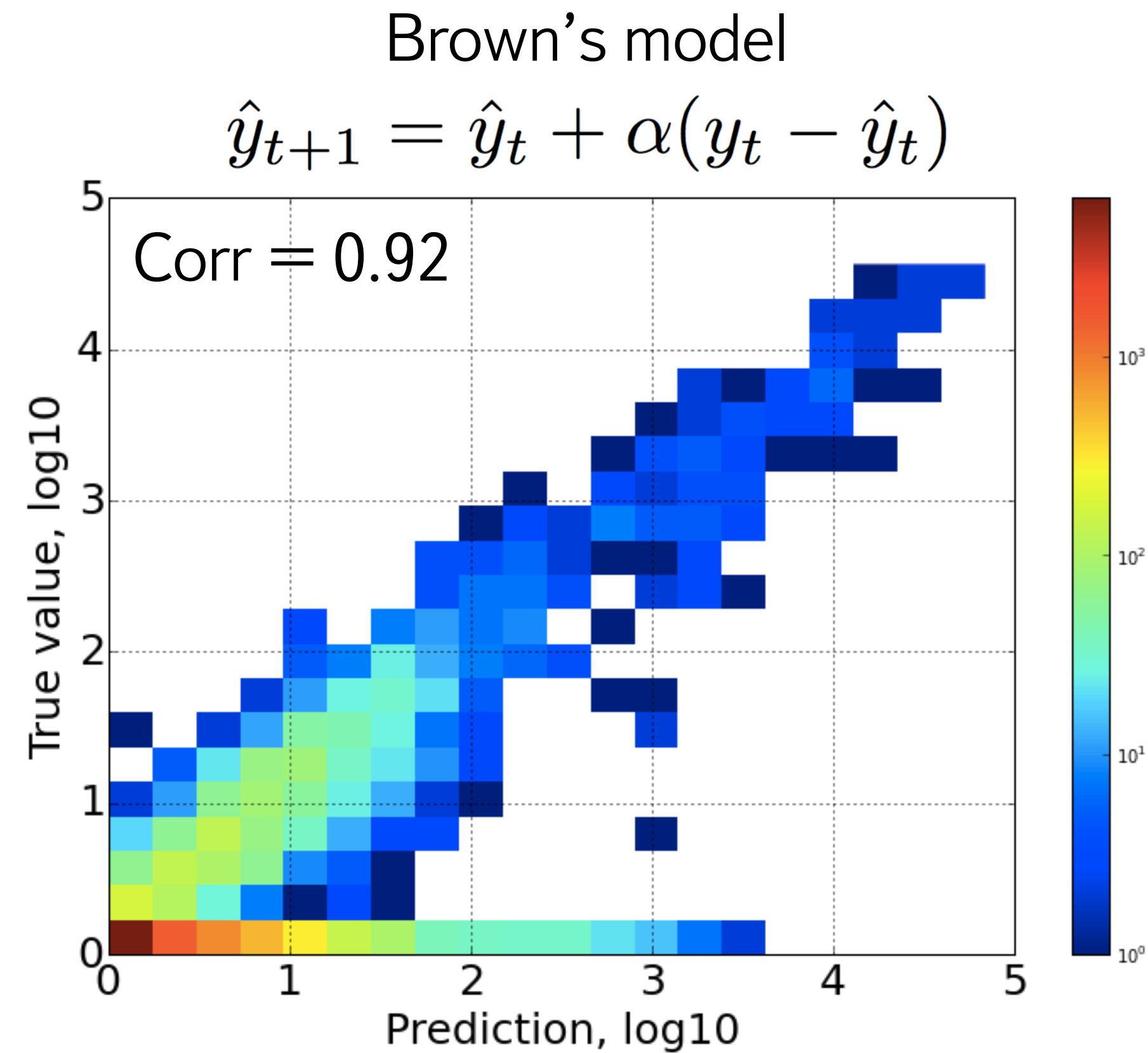
Short-term forecast

The Brown's model demonstrates high correlation between the prediction and true values.

On the other hand, the model is quite conservative for the small true values.



Short-term forecast



Situations when the low predictions correspond to the high true values are undesirable.

Data Popularity 3.0

Replication strategy



Replication strategy

The results of the long-term prediction and short-term forecast are used to calculate one of the following metrics:

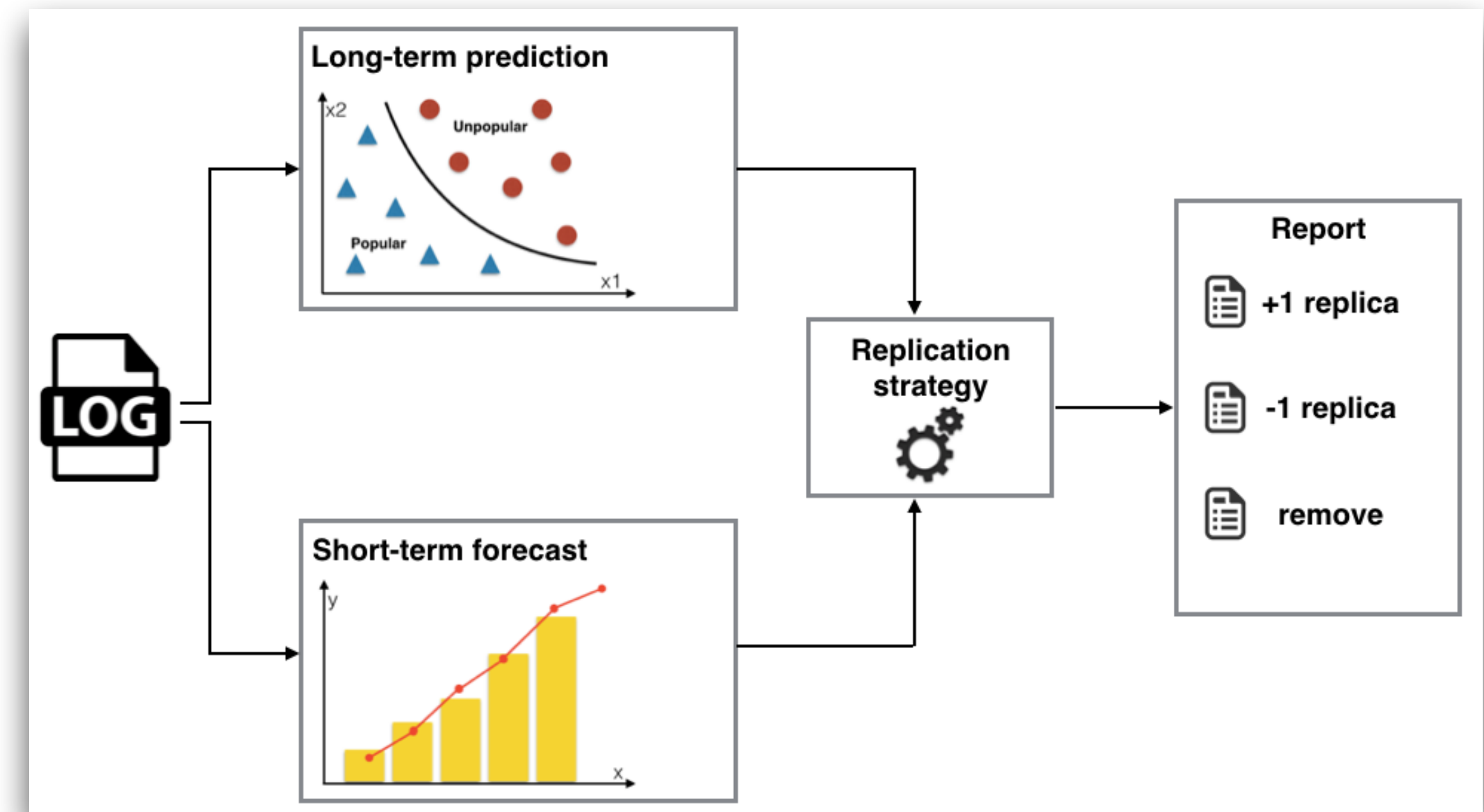
$$M = \frac{\hat{y}_{t_{curr}+1}}{n_{replicas}} \quad \text{default, more conservative}$$

$$M = \frac{\hat{y}_{t_{curr}+1}}{n_{replicas}} (\alpha + classifier_output)$$

To save some disk space:

1. remove 1 replica for a file with the minima \underline{M}
2. recalculate the \underline{M} for that file
3. repeat steps 1-2 until the required amount of space is saved.

This will remove replicas for the files which will be less popular in the future.



Min number of replicas	Space can be saved
remove all	14.8 PB
1	9.5 PB
2	4.2 PB

How much space can be saved leaving the minima number of replicas for the each file

Replication strategy

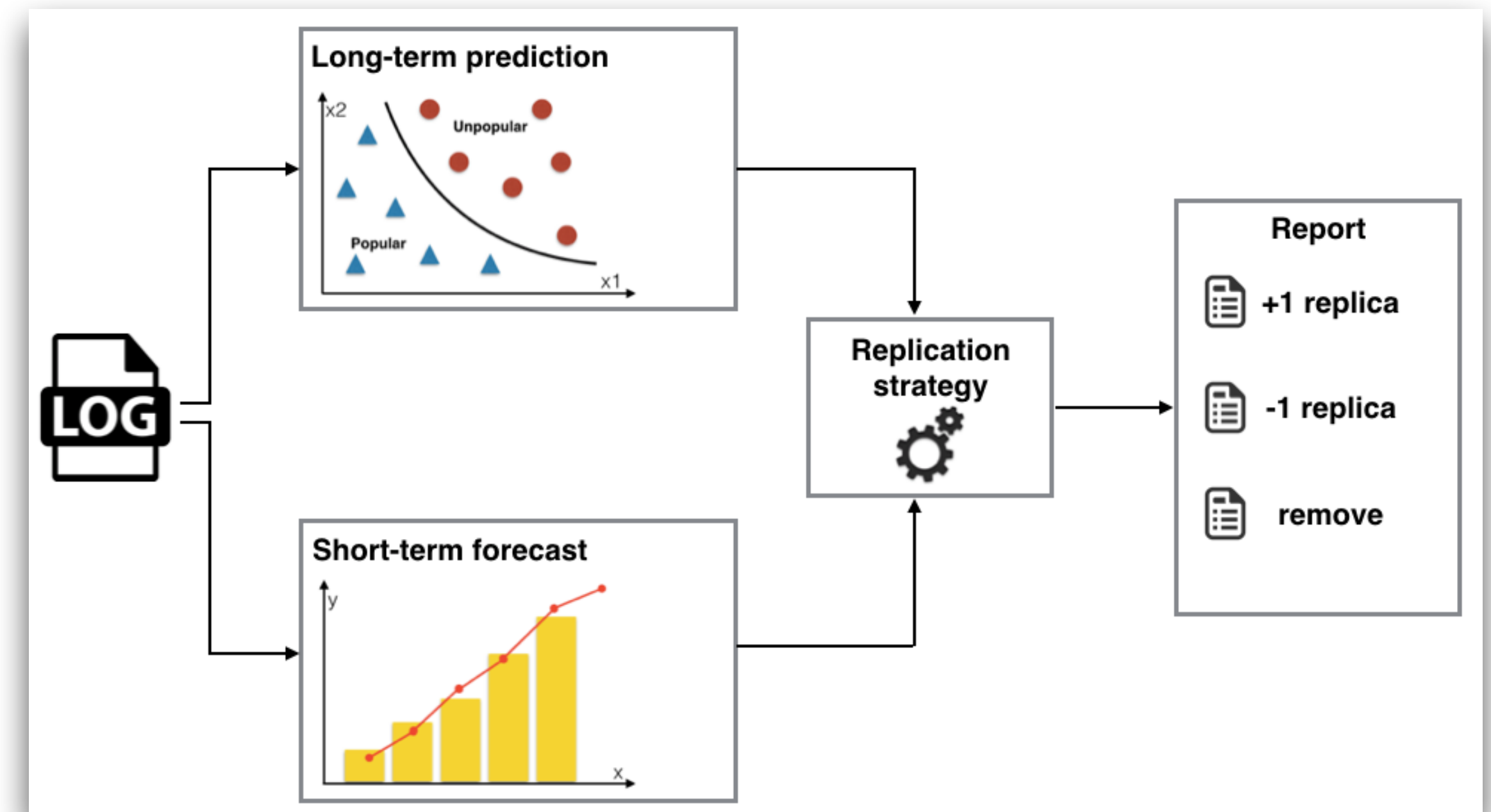
The results of the long-term prediction and short-term forecast are used to calculate one of the following metrics:

$$M = \frac{\hat{y}_{t_{curr}+1}}{n_{replicas}} \quad \text{default, more conservative}$$

$$M = \frac{\hat{y}_{t_{curr}+1}}{n_{replicas}} (\alpha + classifier_output)$$

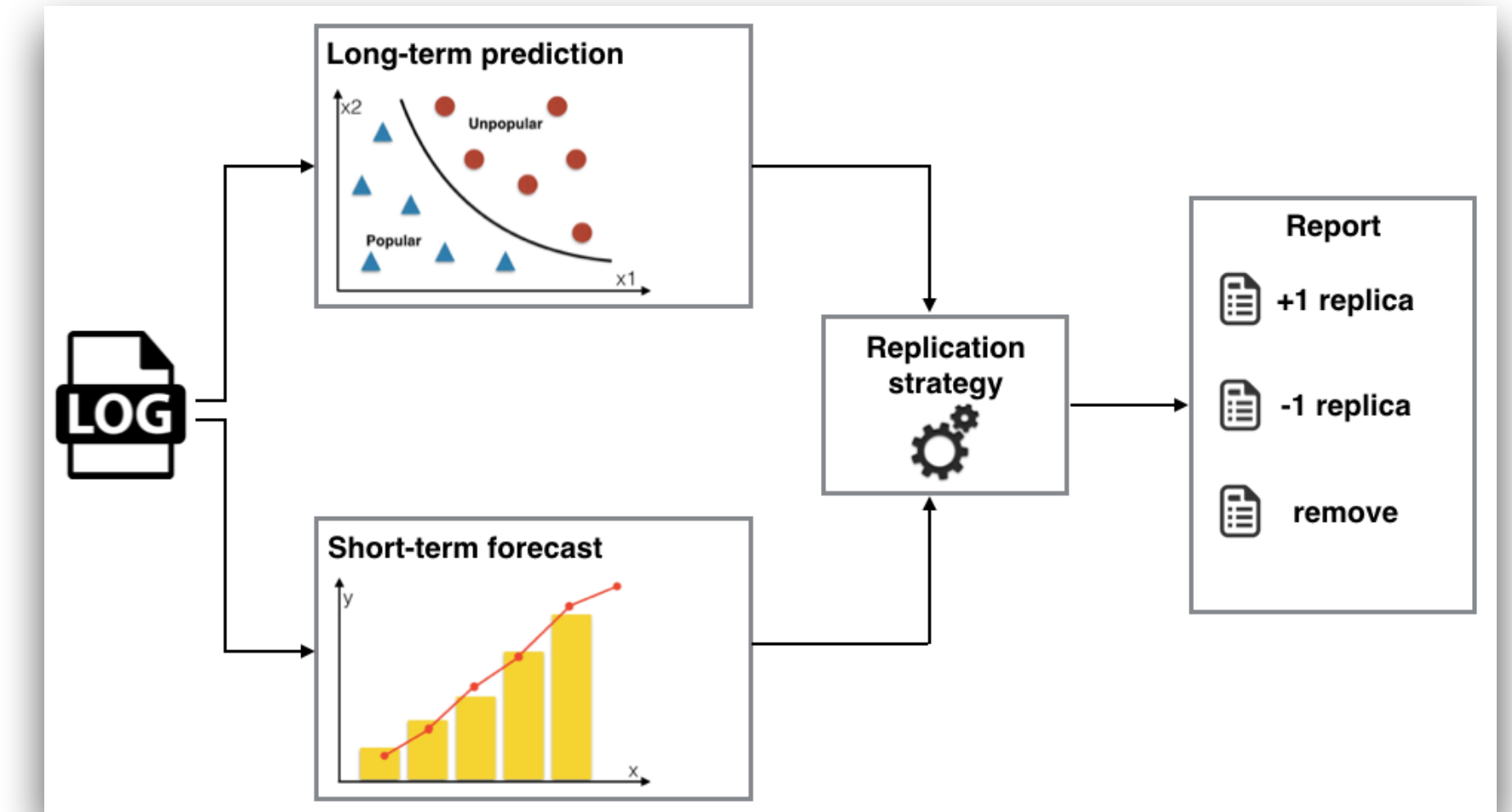
How to use it:

- › Use the metrics above for the decreasing or increasing number of replicas
- › Use long-term prediction for the datasets removing from the disks



Summary

1. ML approaches were developed for the data popularity prediction
2. Short- and long-term popularity predictions are used
3. Decreasing, increasing and removing are separated
4. Least popular files are removed first



Conclusion

The Data Popularity prediction helps to save Pb's of space.

Contacts:

mikhail.hushchyn@cern.ch, andrey.ustyuzhanin@cern.ch