Building a Wide-Area File Transfer Performance Predictor
An Empirical Study

Zhengchun Liu, Rajkumar Kettimuthu, Prasanna Balaprakash, Nageswara S.V. Rao, Ian Foster

Presented by: Rajkumar Kettimuthu

November, 2018 Paris, France

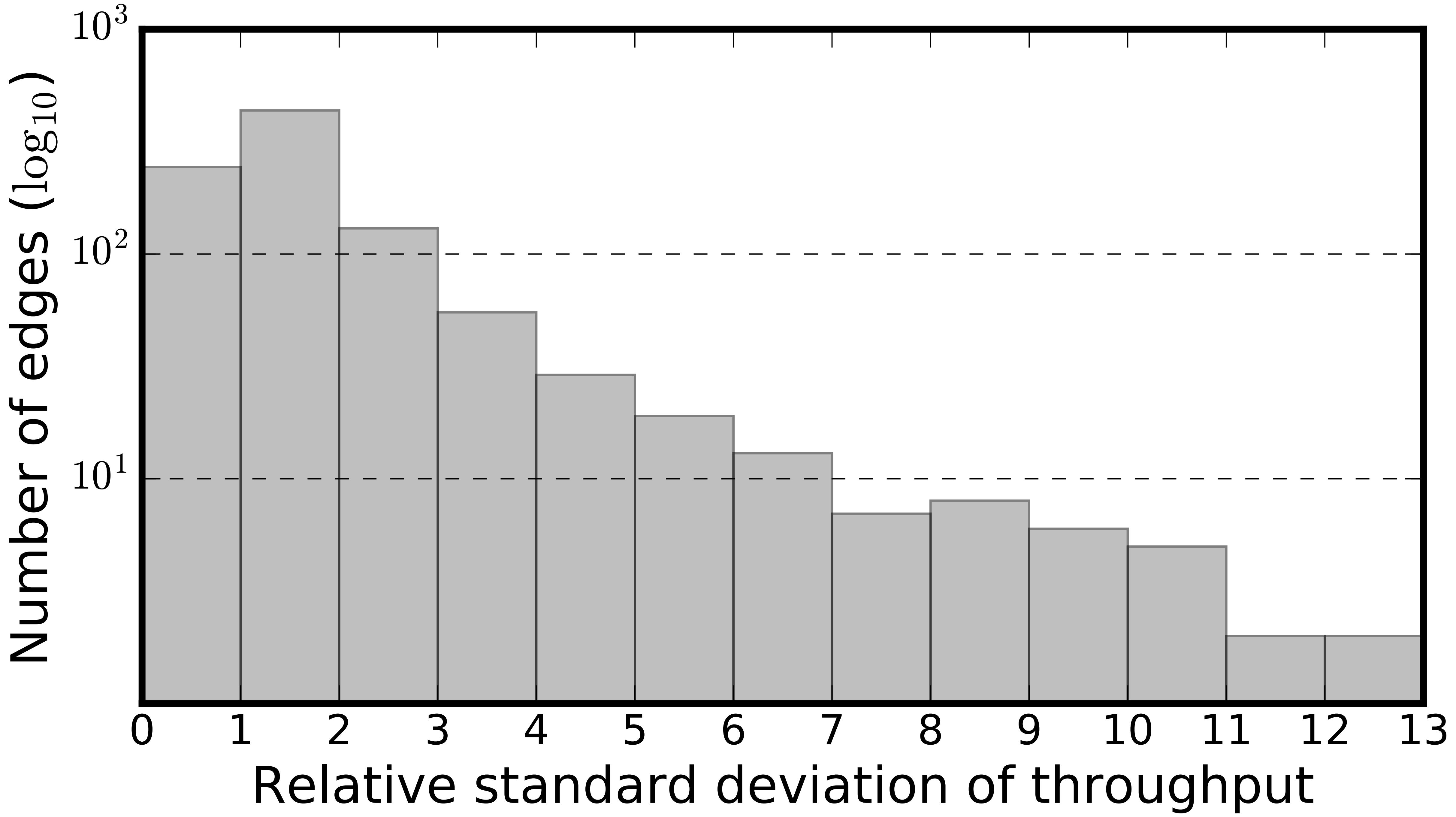
Motivation and summary

- Wide-area data transfer is central to geographically distributed scientific workflows.
- Faster delivery of data is important for these workflows.
- Predictability is equally (or even more) important than transfer rate.
- Providing a reasonably accurate estimate of data transfer time to improve resource allocation & scheduling for workflows.
- Machine learning methods to develop predictive models for data transfer times over a variety of wide area networks.

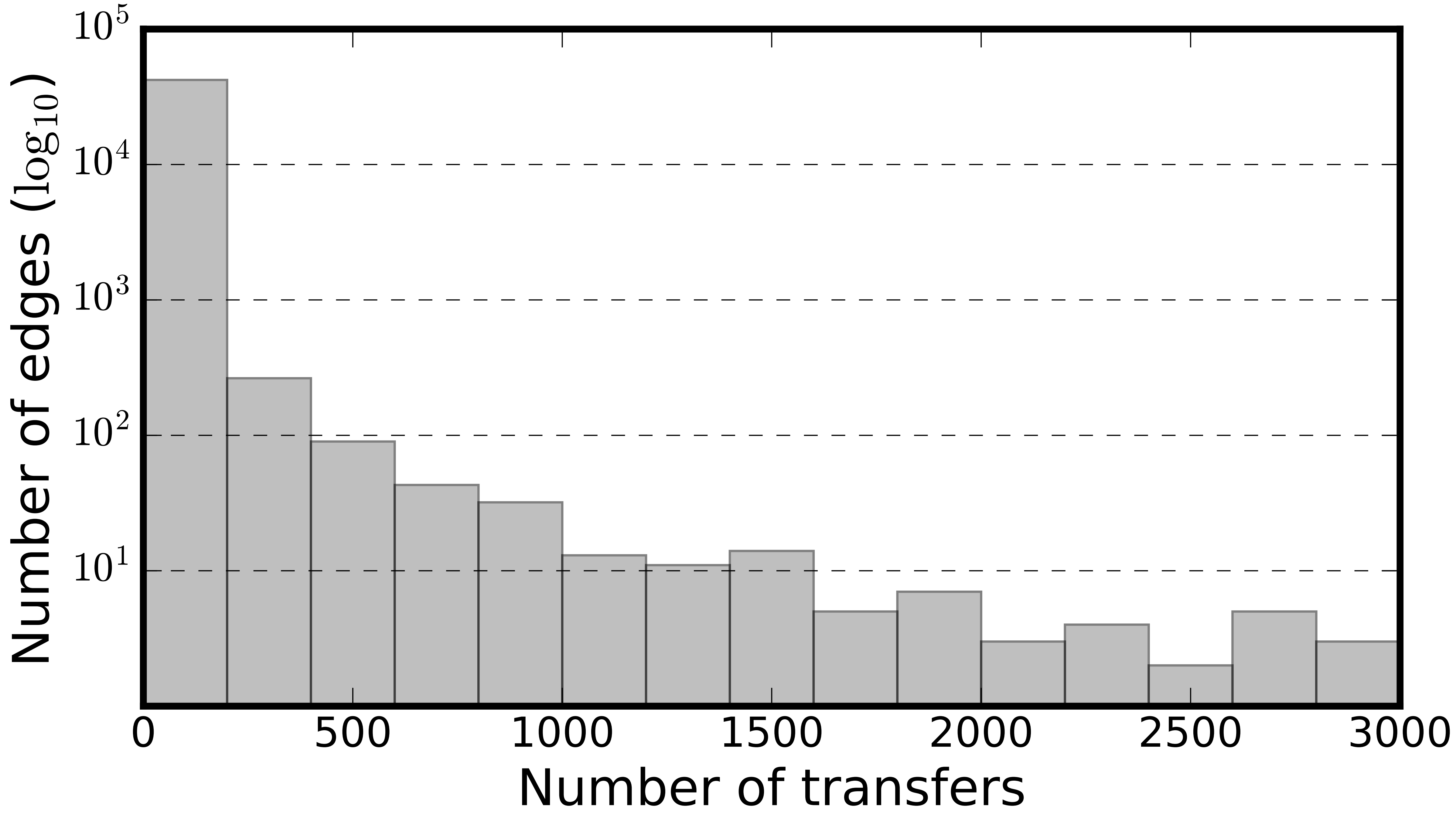
Agenda

- The data
- The way to build the predictor
- Important open questions to build the predictor
- Summaries

Data exploration analysis



Relative standard deviation (standard deviation divided by the mean) for the top 1,000 heavily used edges in Globus

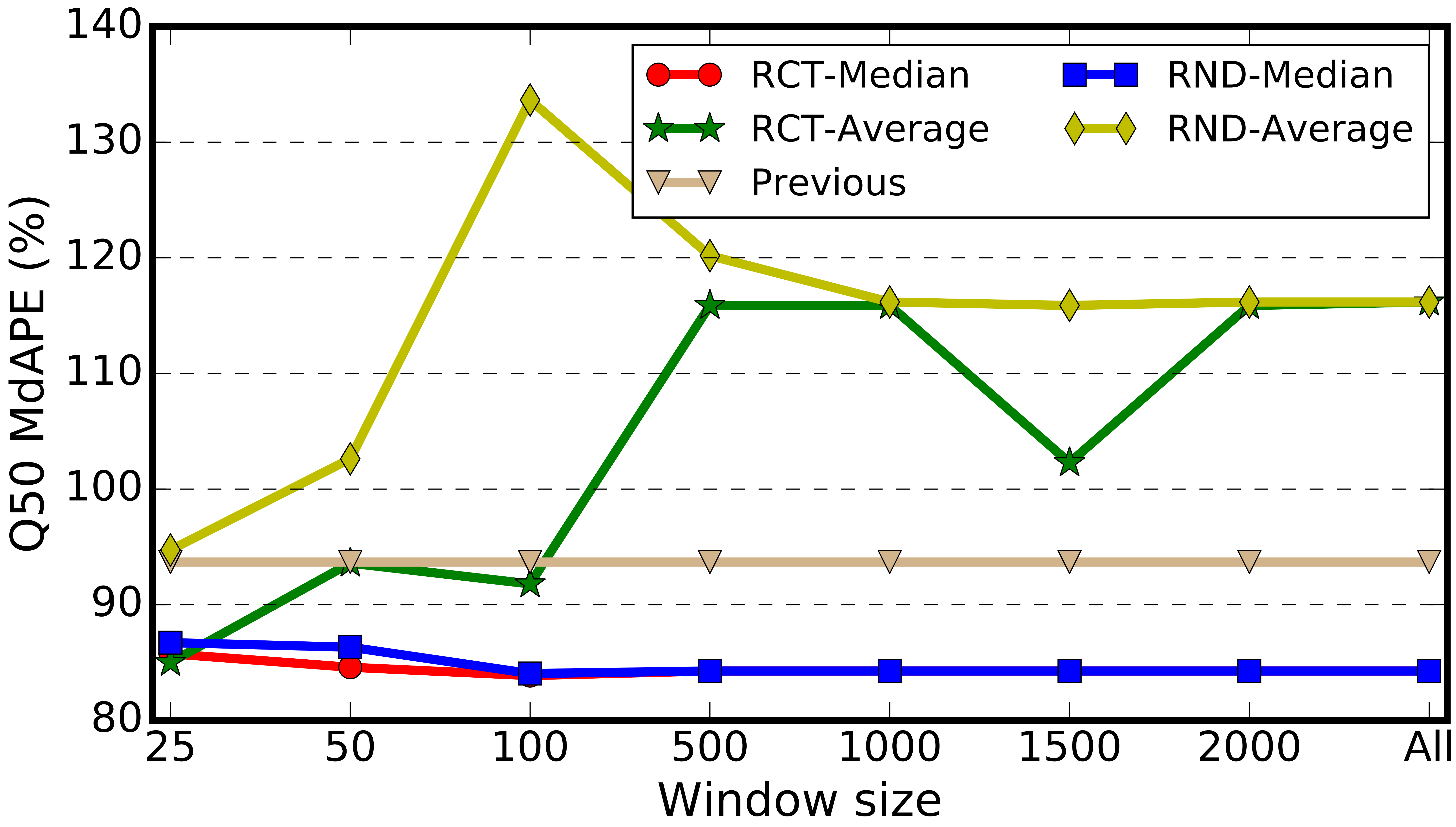


Distribution of the number of transfer over edges.

Throughput on different edges (source endpoint to destination endpoint pair) is quite different.

Transfer load varies largely from edge to edge.

Baseline predictions: Average, Median and Previous

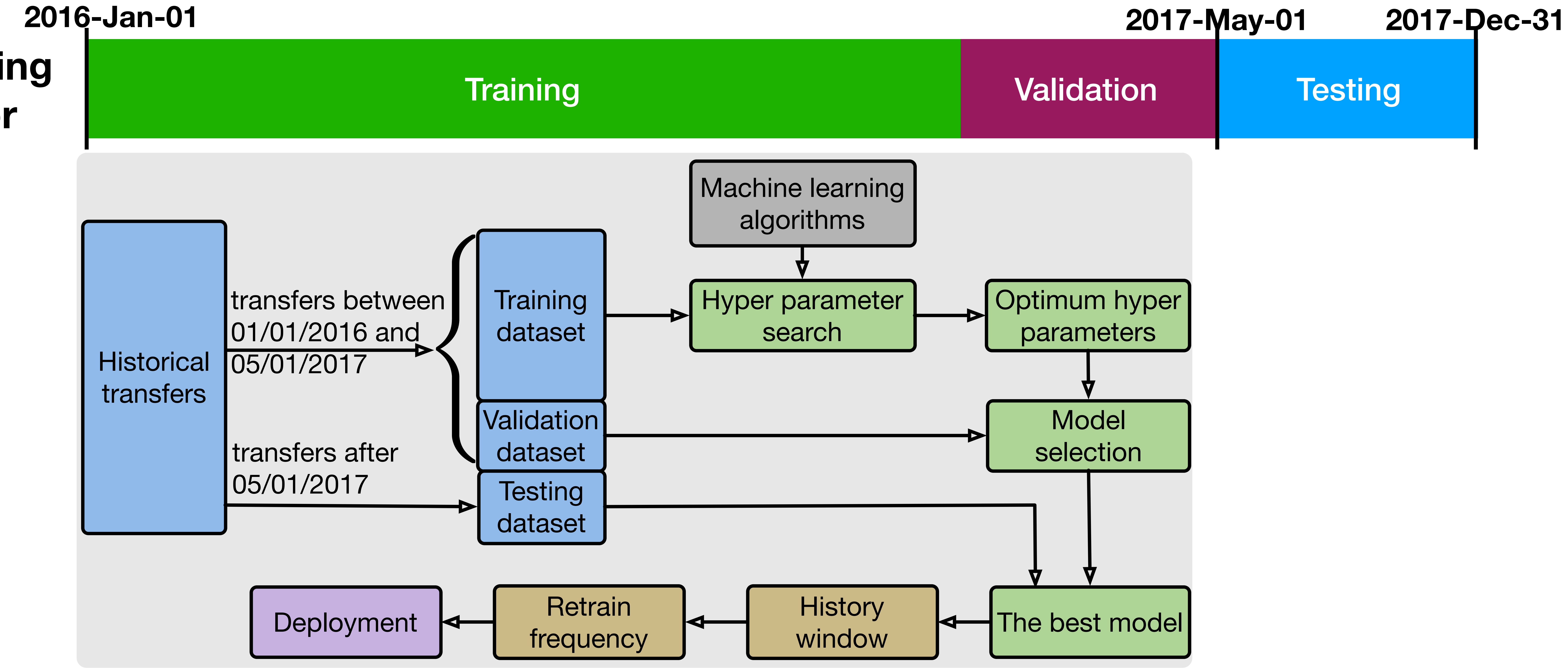


The prediction error (50th quantile MdAPE) when use median, average and previous transfer as performance predictor.

- **Median Absolute Percentage Error (MdAPE) is greater than 80%.**
- **Baseline predictor do not provide any prediction!**
- **Need to give machine learning a try.**

Work flows and open questions

Data for building the predictor



- How to select the most appropriate machine learning algorithms
- What is the appropriate retraining frequency to deal with changes because of software/hardware upgrade?
- How many historical data points needed to train the model? and,
- Randomly choose K transfers versus use the most recent K transfers to train the model (i.e. temporal aspect).

Features constructed & used

Notation used in this article.
The lower 20 terms are used as features in our machine learning algorithms, of which the first 15 are from Liu et al. HPDC'17 **and the remaining five are developed in this paper.**

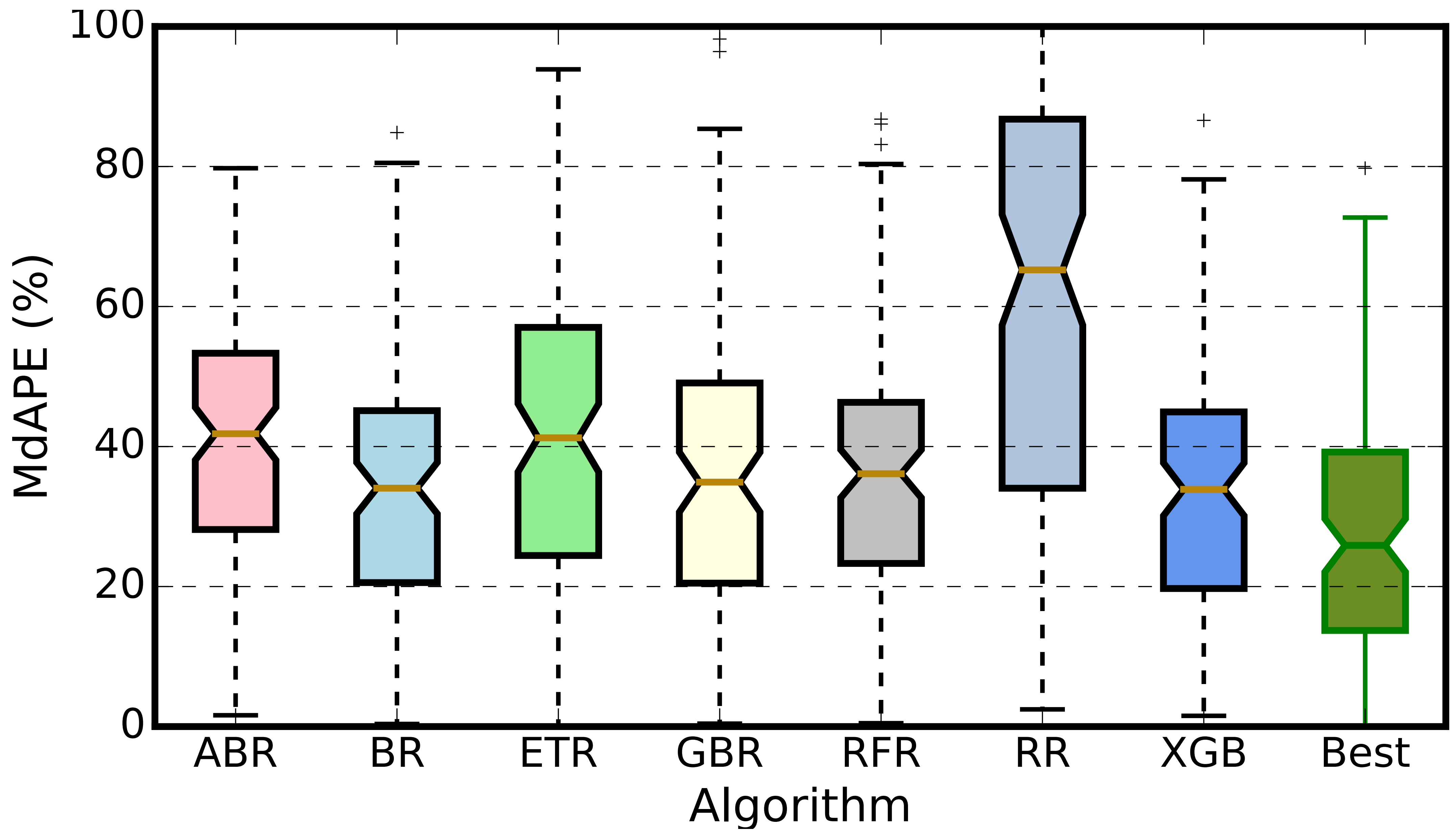
$$Q^{x\in\{sout,sin,dout,din\}}(k) = \sum_{i\in A_x} \frac{\mathcal{O}(i,k)}{Te_k - Ts_k} R_i, \quad (1)$$

where *sout* and *sin* denote outgoing and incoming at institution I_{src}, respectively, and *dout* and *din* represent outgoing and incoming at institution I_{dst}, respectively; A_x is the set of transfers (excluding k) with I_{src} as source, when $x = sout$; I_{src} as destination, when $x = sin$; I_{dst} as source, when $x = dout$; and I_{dst} as destination when $x = din$; and R_i is the throughput of transfer i, and $\mathcal{O}(i,k)$ is the overlap time for the two transfers:

$$\mathcal{O}(i,k) = \max\left(0, \quad \min(Te_i, Te_k) - \max(Ts_i, Ts_k)\right).$$

src_k	Source endpoint of transfer k.
dst_k	Destination endpoint of transfer k.
I_k^{src}	Institution of the source endpoint of transfer k.
I_k^{dst}	Institution of the destination endpoint of transfer k.
Ts_k	Start time of transfer k.
Te_k	End time of transfer k.
R_k	Average transfer rate of transfer k.
K^{sin}	Contending incoming transfer rate on src_k.
K^{sout}	Contending outgoing transfer rate on src_k.
K^{din}	Contending incoming transfer rate on dst_k.
K^{dout}	Contending outgoing transfer rate on dst_k.
C	Concurrency: Number of GridFTP processes.
P	Parallelism: Number of TCP channels per process.
S^{sin}	Number of incoming TCP streams on src_k.
S^{sout}	Number of outgoing TCP streams on src_k.
S^{din}	Number of incoming TCP streams on dst_k.
S^{dout}	Number of outgoing TCP streams on dst_k.
G^{src}	GridFTP instance count on src_k.
G^{dst}	GridFTP instance count on dst_k.
Nf	Number of files transferred.
Nd	Number of directories transferred.
Nb	Total number of bytes transferred.
Q^{sin}	Contending incoming transfer rate on I_k^{src}.
Q^{sout}	Contending outgoing transfer rate on I_k^{src}.
Q^{din}	Contending incoming transfer rate on I_k^{dst}.
Q^{dout}	Contending outgoing transfer rate on I_k^{dst}.
D	Pipeline depth.

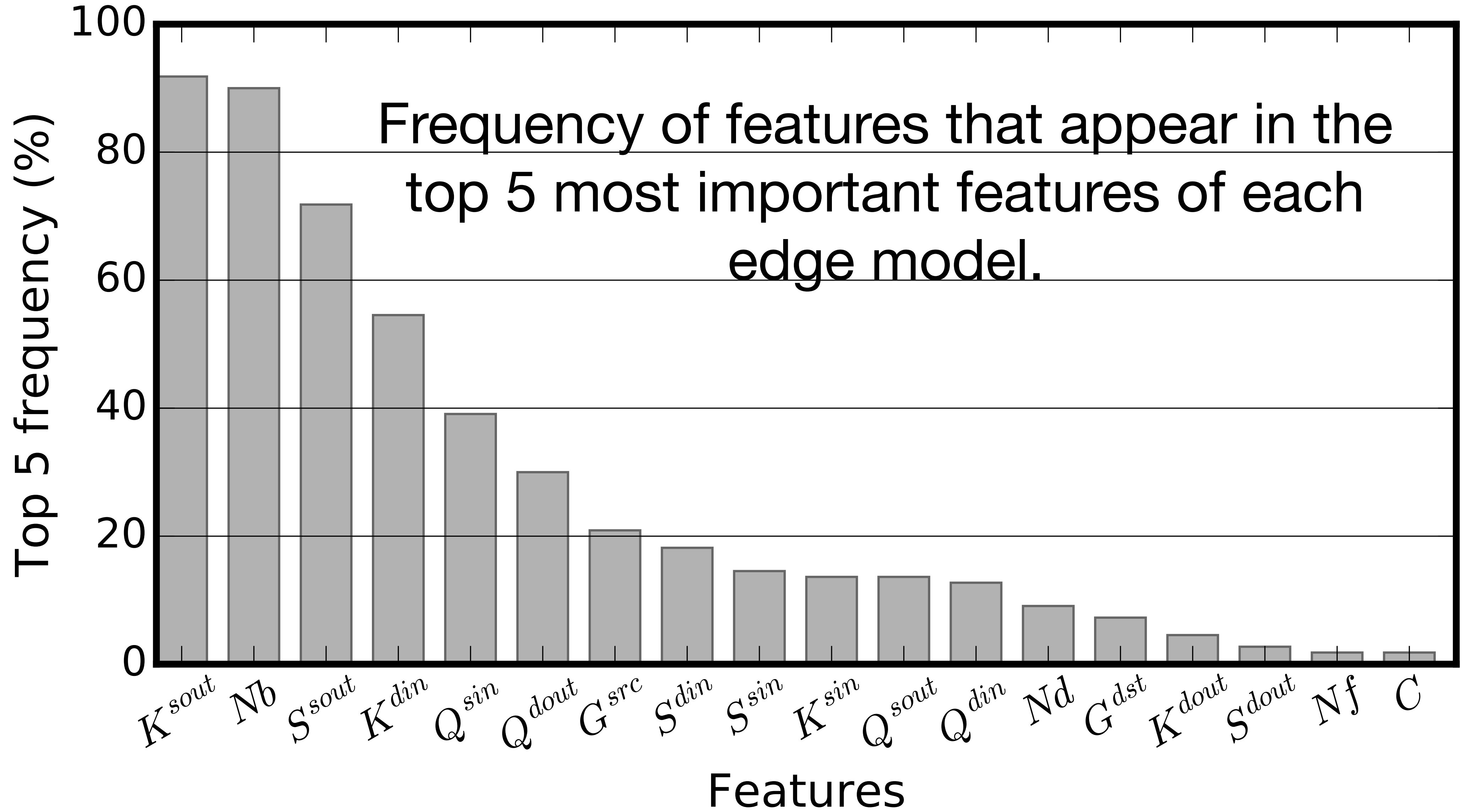
Algorithm selection and Model training and feature importance



Validation results. *Best* represent validation errors when the best algorithm is used for each edges.

Statistics of the best model selection

Algorithm	**Pairs**
GradientBoostingRegressor	23
Ridge	6
XGBRegressor	26
BaggingRegressor	18
AdaBoostRegressor	9
RandomForestRegressor	14
ExtraTreesRegressor	14



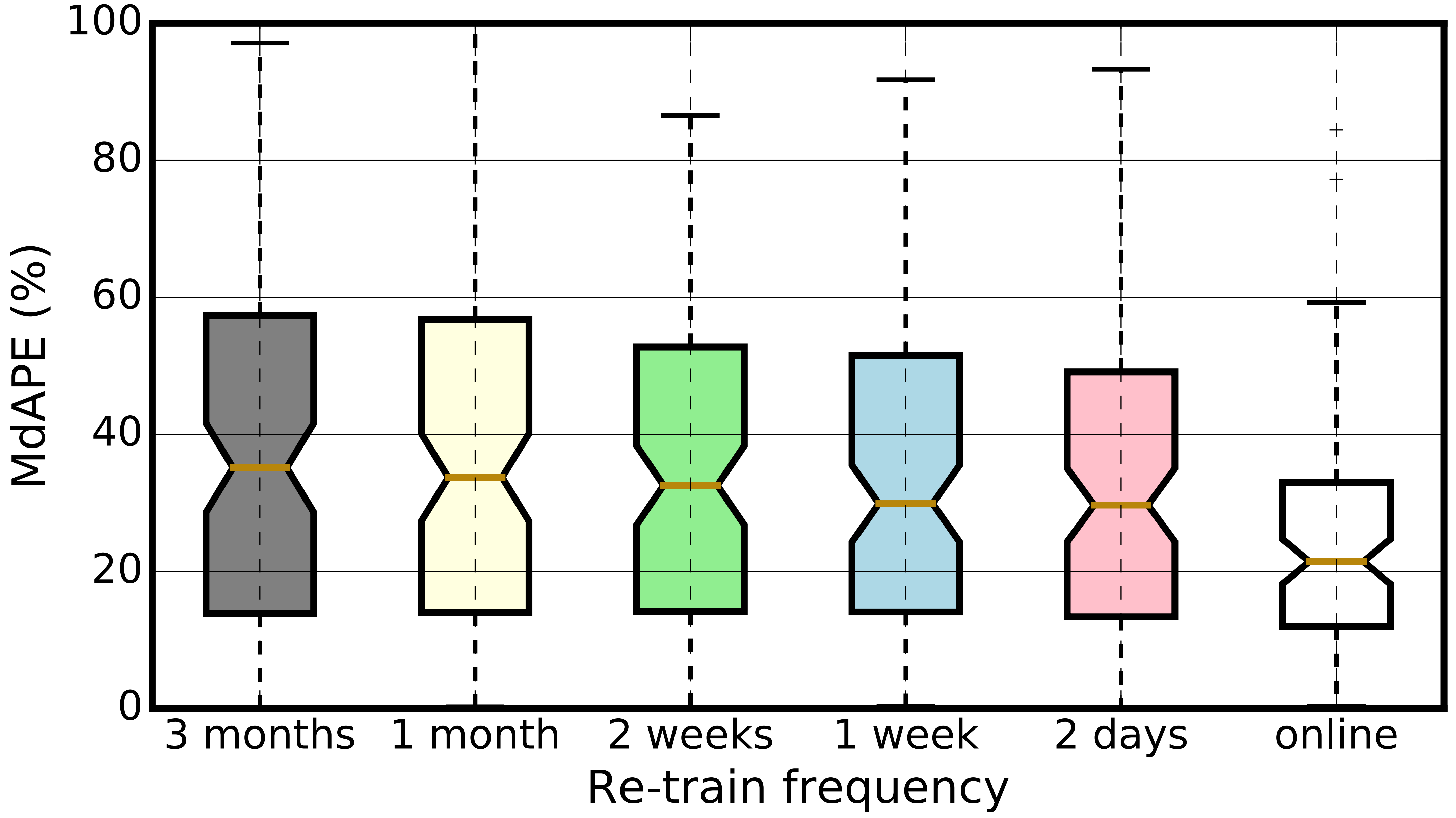
Transfer size Nb and K^{sout}, K^{din} and S^{sout} which have contention in the same direction with the transfer of interest, are important for most of the end-point pairs. The four new introduced features in this paper (Q^{sout}, Q^{din}, Q^{sin} and Q^{dout}), which quantify contention from simultaneous transfers from the same institution (within eight kilometers), are also important.

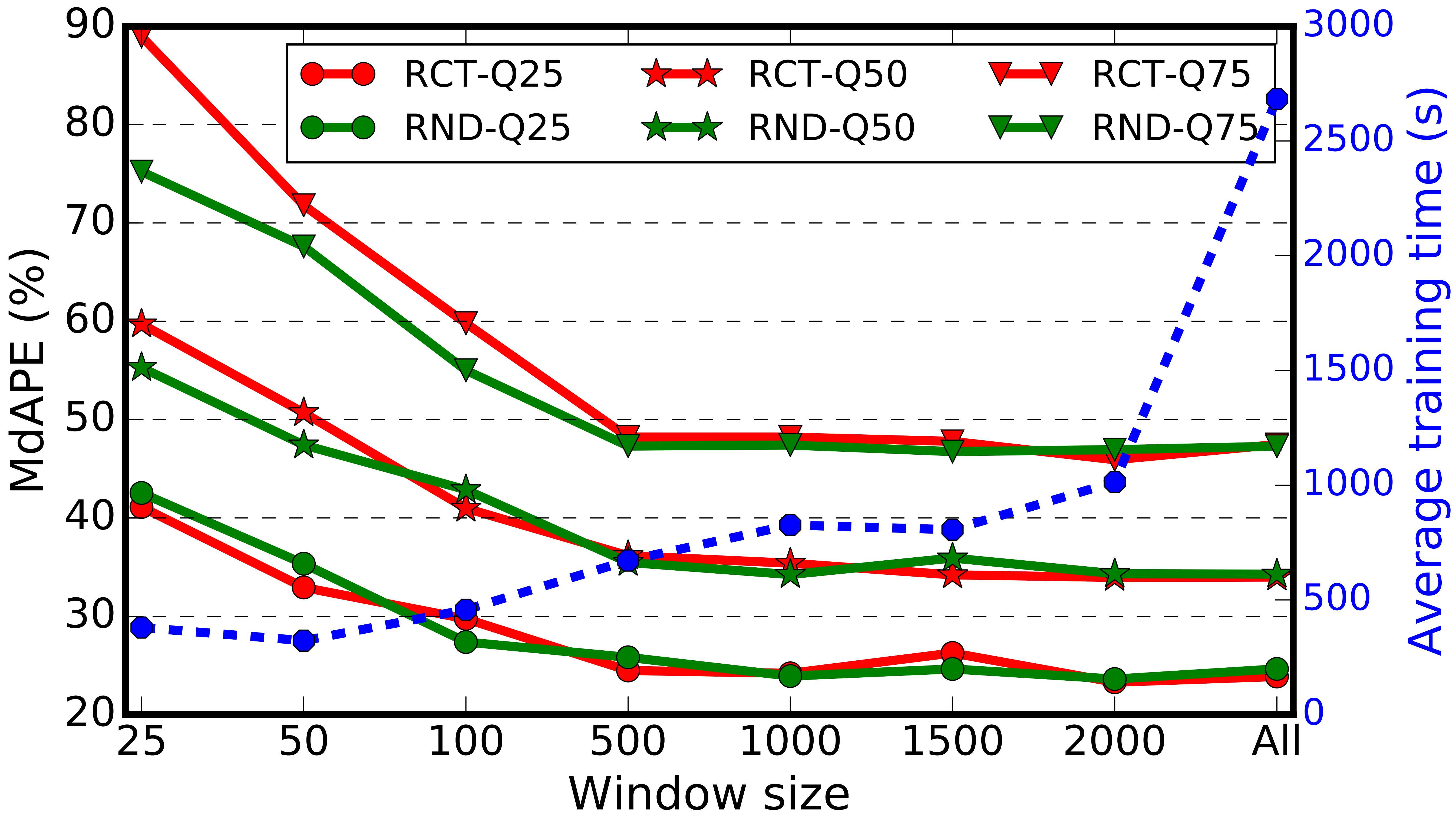
Argonne NATIONAL LABORATORY

Retrain frequency versus prediction error; the selection of samples.

- Accuracy versus training time
- Most recent versus random

Retrain frequency versus prediction error on testing dataset.





Prediction errors (solid lines) and model training time (dotted blue line) as a function of the number of transfers (N_{train}) used to train the model. **Q25**, **Q50** and **Q75** represent 25th, 50th and 75th quantile of MdAPE separately. **RCT** and **RND** denotes most recent N_{train} transfers and randomly chosen N_{train} transfers individually. **All** means that we used all transfers before May 1, 2017, to train the model.

Further insights for the prediction error

For a given endpoint pair we group transfers by:
(these transfers have similar dataset characteristics and application parameters)

Group 1: Transfers with rate greater than 50% of the maximum rate observed over this endpoint pair. These transfers are likely to have less contending load.

Group 2: We sort transfers in descending order by source's aggregate outgoing rate and take the top 5%. Similar for destination. (transfers have lots of external load but mostly are known)

Group 3, We apply the same procedure that created group (2), but extract bottom 5% on source and destination. (known load is less and throughput is low as well)

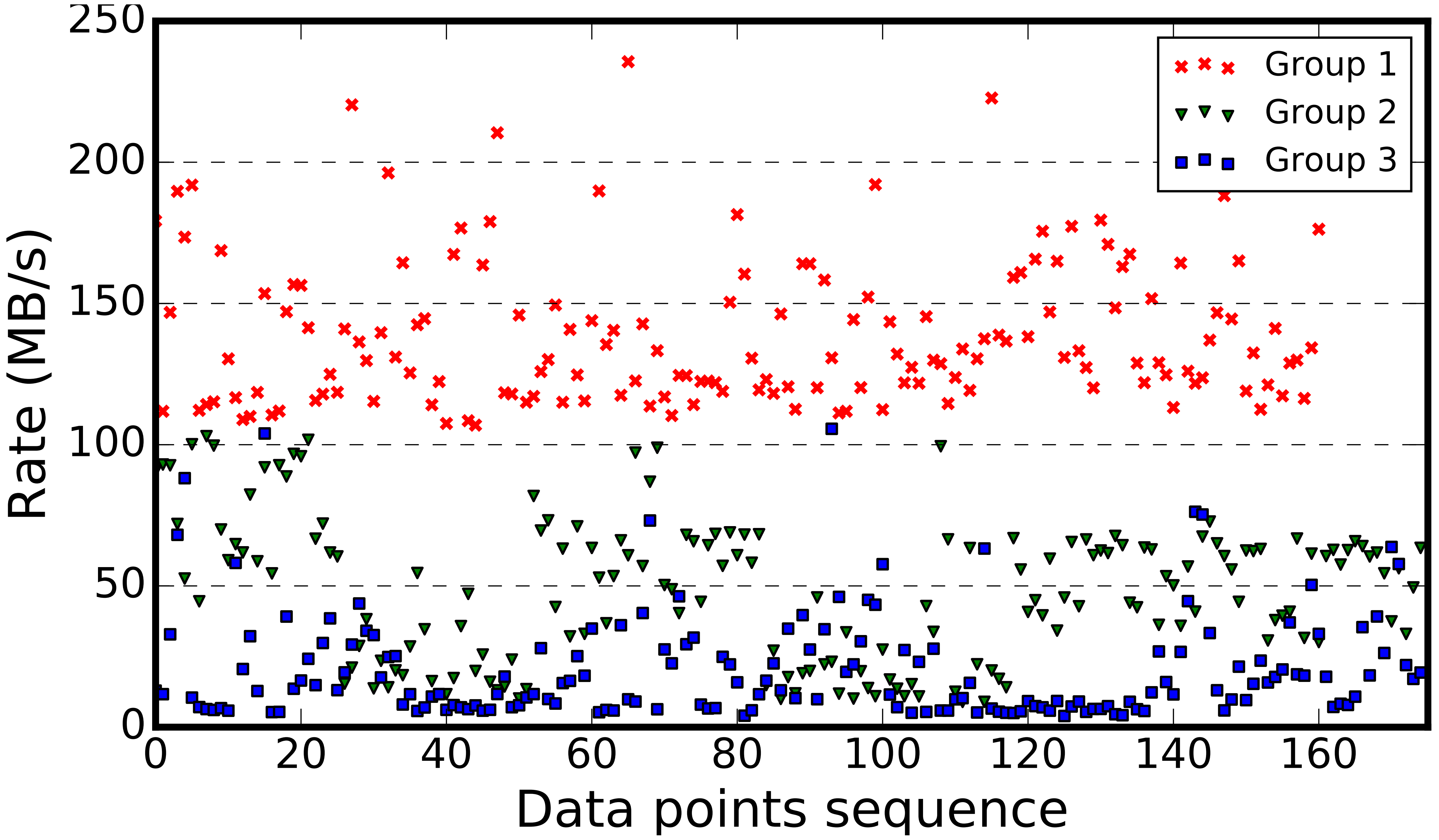
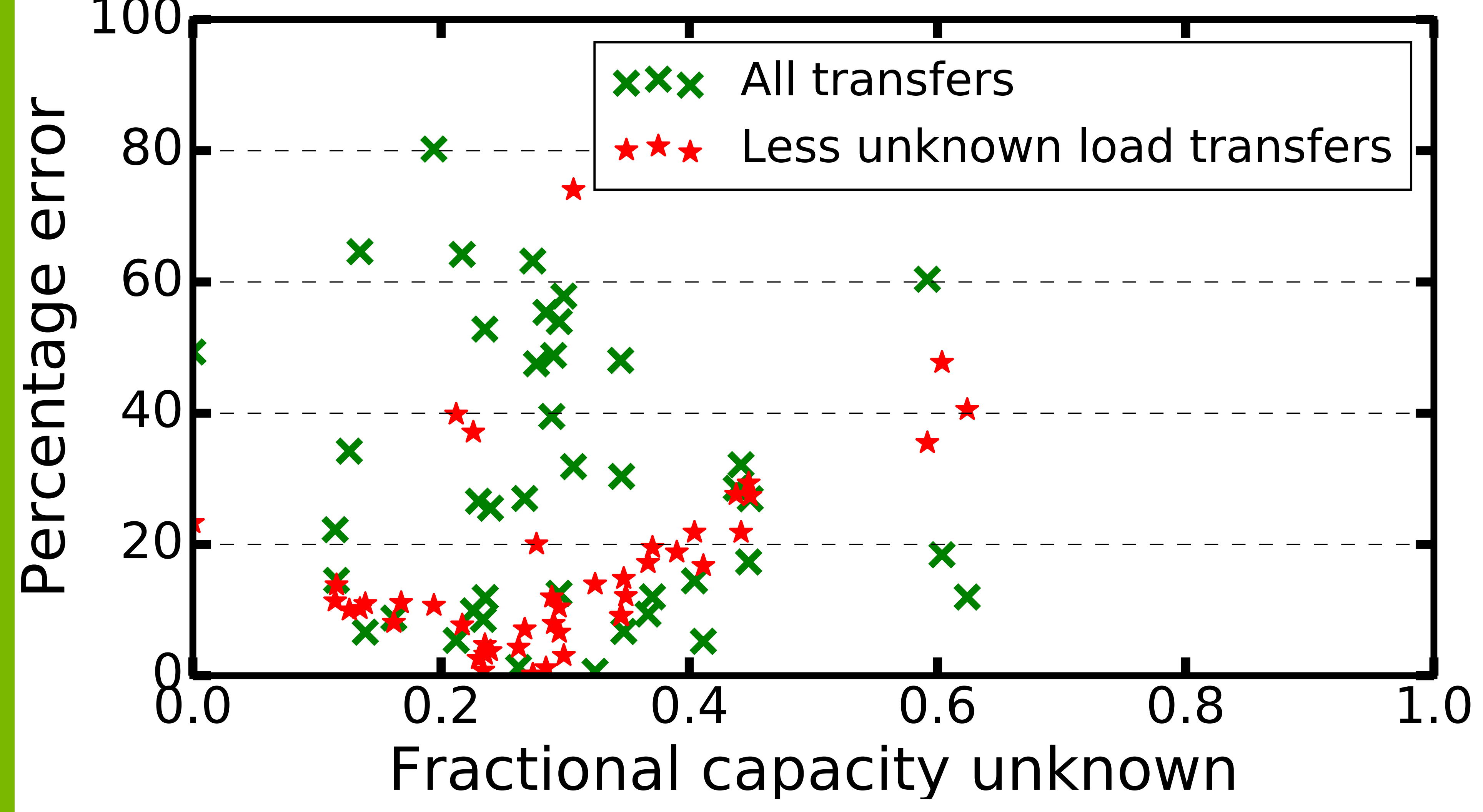


TABLE III: Prediction error (%) with different machine learning algorithm on the three groups.

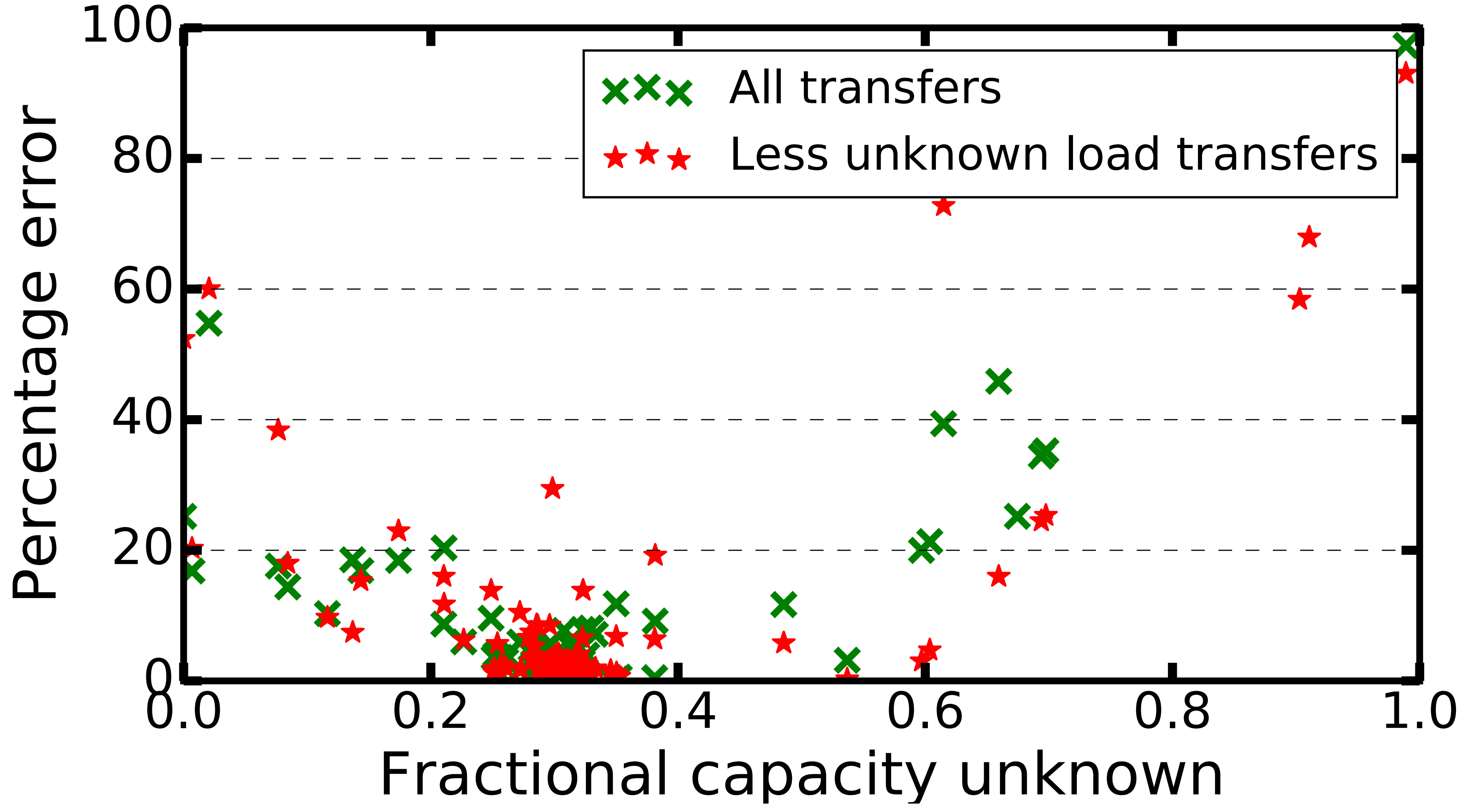
Algorithm	Group	Q50	Q75	Q90
Ridge Regression	1	11.24	18.19	22.63
	2	20.04	33.08	64.33
	3	35.37	126.54	223.29
XGBRegressor	1	11.85	22.91	25.20
	2	8.20	18.06	29.36
	3	27.16	51.02	72.49
BaggingRegressor	1	9.54	18.83	25.02
	2	9.46	14.81	32.64
	3	29.85	51.27	133.48

Further insights for the prediction error

Train model with clean data (transfers that are less likely to have unknown load)



harvard.edu to a Globus Connect Personal



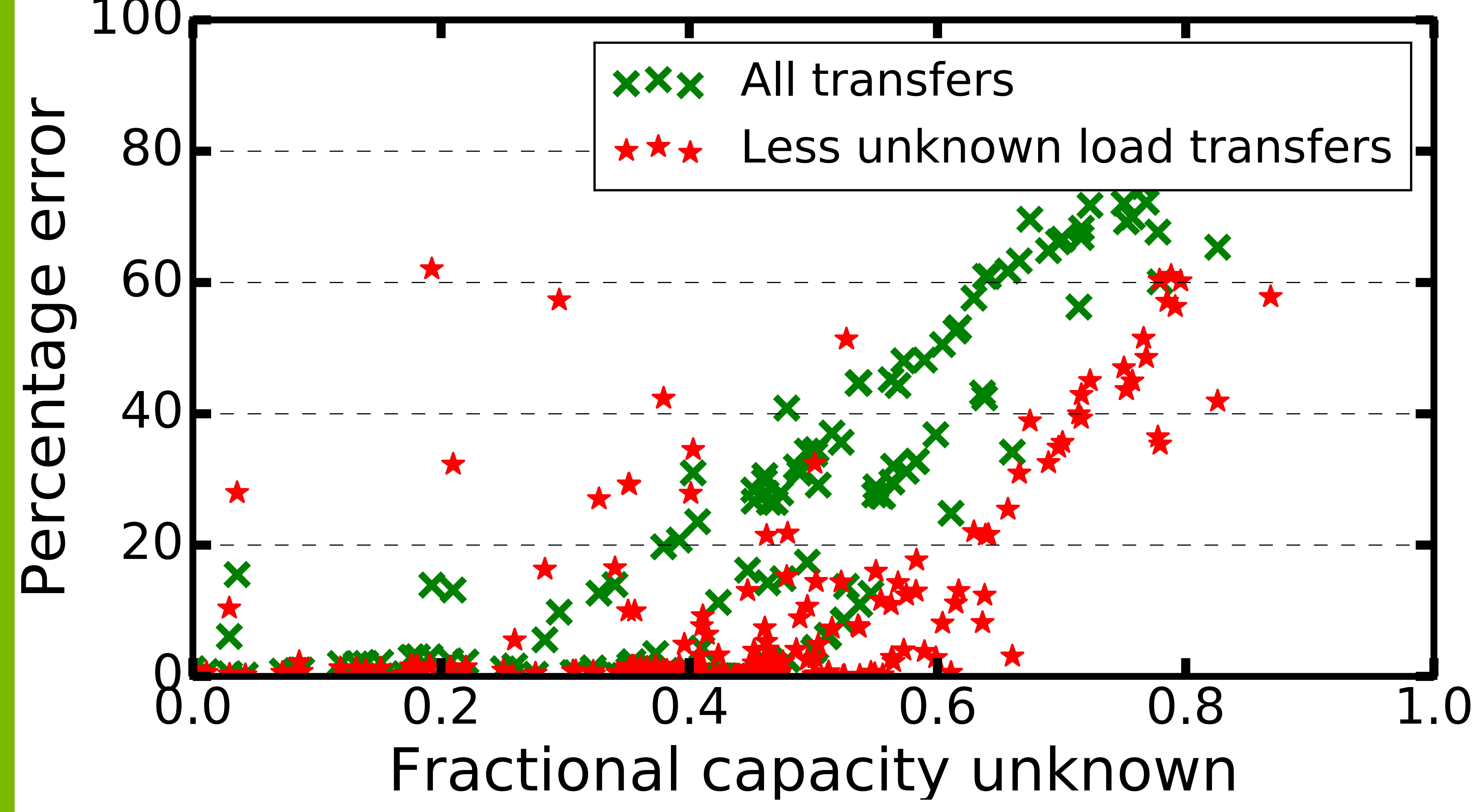
cuny.edu to a Globus Connect Personal

$$KL_k^{src} = \frac{K^{sout} + R_k}{DR^{max}}$$

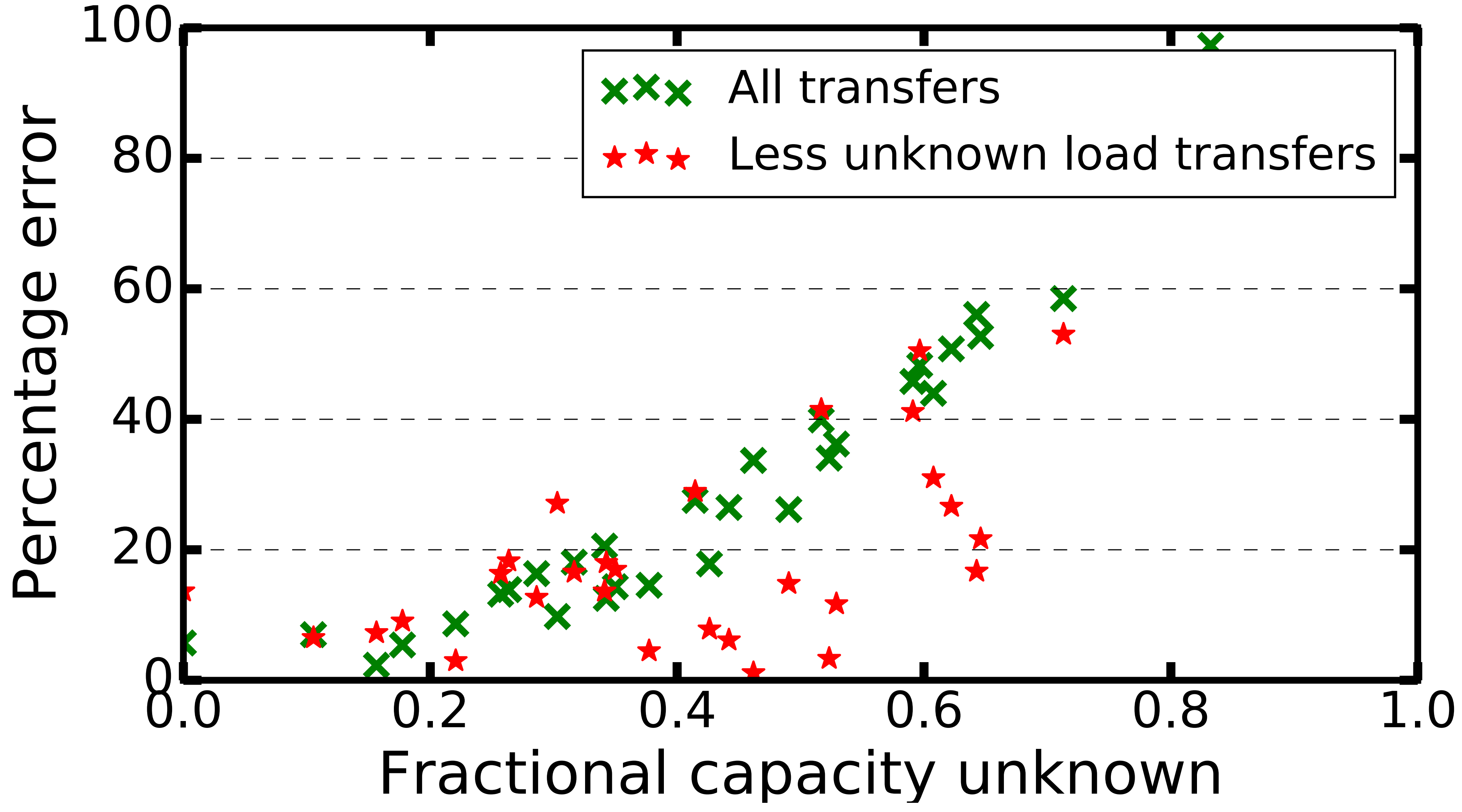
$$KL_k^{dst} = \frac{K^{din} + R_k}{DW^{max}}.$$

$$KL_k = \max\left(RL_k^{src}, RL_k^{dst}\right)$$

$$UC_k = 1 - KL_k$$



cwru.edu to a Globus Connect Personal



westgrid.ca to a Globus Connect Personal

Conclusion

- Machine based method are studied to build the wide area file transfer time predictor.
- Models perform well for many transfers, with a median prediction error < 21% for 50% of edges, and < 32% for 75% of the edges.
- For some edges, further insights are studied to understand the root cause of prediction error.
- Unknown load can interfere with model training, eliminating transfers with high unknown load from training data can improve prediction accuracy for transfers with less unknown load.
- Collecting more information about endpoint load can further improve the prediction accuracy.

THANKS!!!

QUESTIONS?