Predicting Slow Network Transfers in Scientific Computing

Robin Shao at UC Berkeley
Jinoh Kim at TAMUC
Alex Sim, John Wu at LBNL

Background

- **data access performance**: a key metrics in scientific computing (scientific computing often requires last-mile access of data)
- **"elephant" connections**: consume a significant fraction of network bandwidth over a long period time
- many studies on **high bandwidth connections**
- not a lot on **low performance network** (low throughput)
- **throughput** is the **actual** measure while **bandwidth** is the **theoretical** measure of how much data could be transferred

Background

Motivations:

- predicted information can be used for **optimizing network resources**
- **understanding low performance network** can help identify the underlying problems and prevent unnecessary slow transfers

The Main Goal

Build a model to predict the low performance transfers to NERSC

Main sections:

- data exploration
- prediction mechanism
- experimental setting and results

Source of Data

- network data collected between Jan 2019 and May 2021 at NERSC
- Tstat is the format (http://tstat.tlc.polito.it/measure.shtml)
  - Tstat is a log format, in which each row corresponds to a different flow and each column is associated with a specific measure.
  - Client-to-Server (C2S) and Server-to-Client (S2C)
  - combine them together and only care about the origins for incoming connections and destinations for outgoing connections
- DTN1-4 (currently use DTN1 and will extend to other DTNs)

Tstat

Tstat::Measure

TCP STatistic and Analysis Tool

Data Choices

only **2021**:

- the latest data we got
- follow 1 year cycle
- achieved our expectation

only **dtn01**:

- most transfers
- involved in more “random” transfers

only **size > 10^6 bytes**:

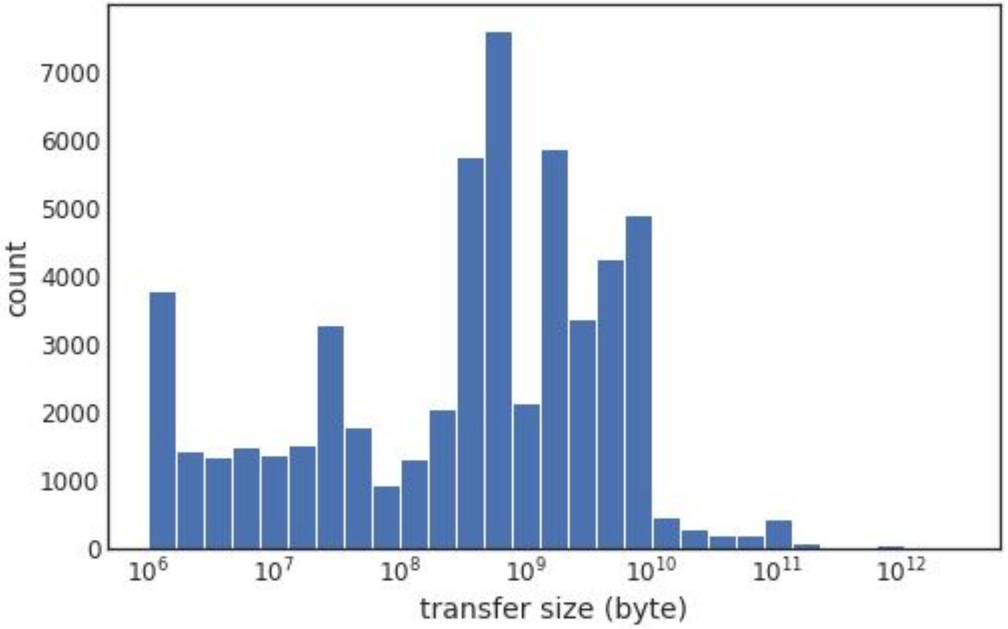
- smaller transfers can hardly be influential

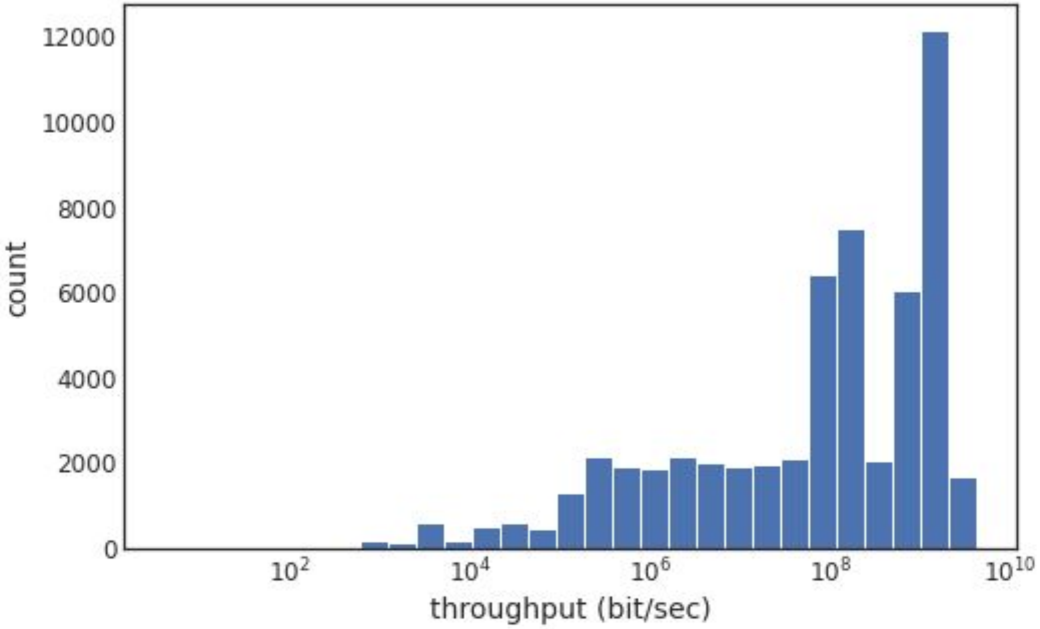
only **rtt_min > 1 ms**:

- internal transfers
- not record due to the transfer direction

Data Exploration

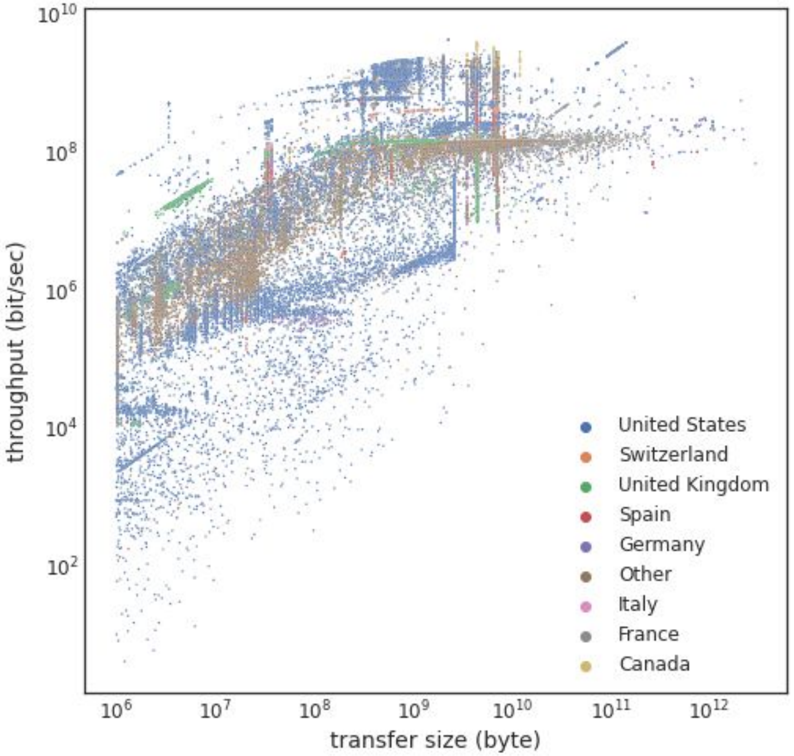
size & throughput

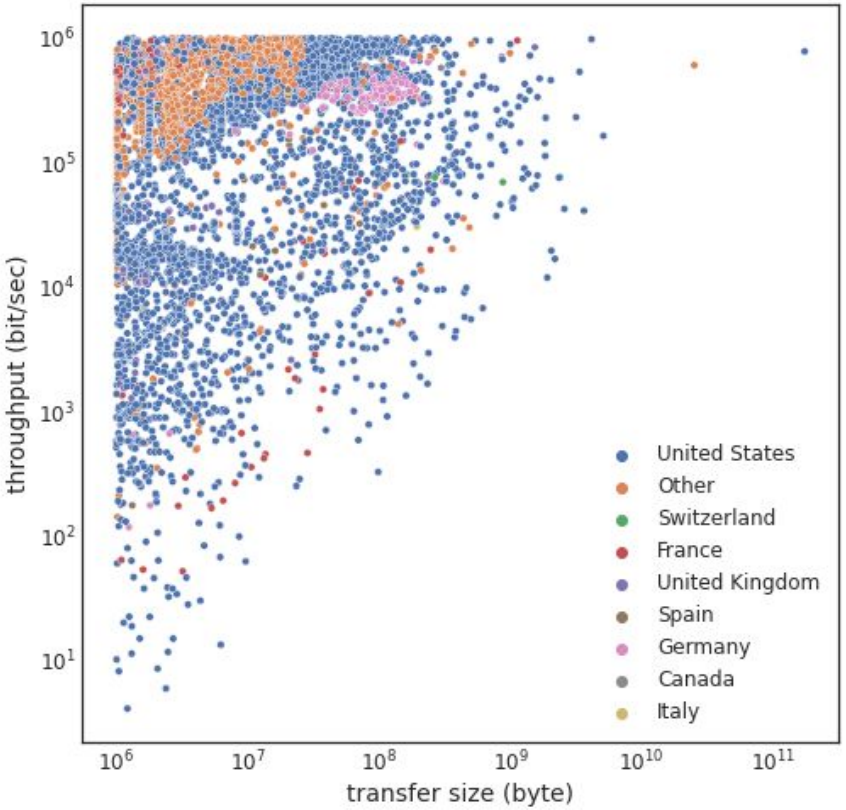




Data Exploration

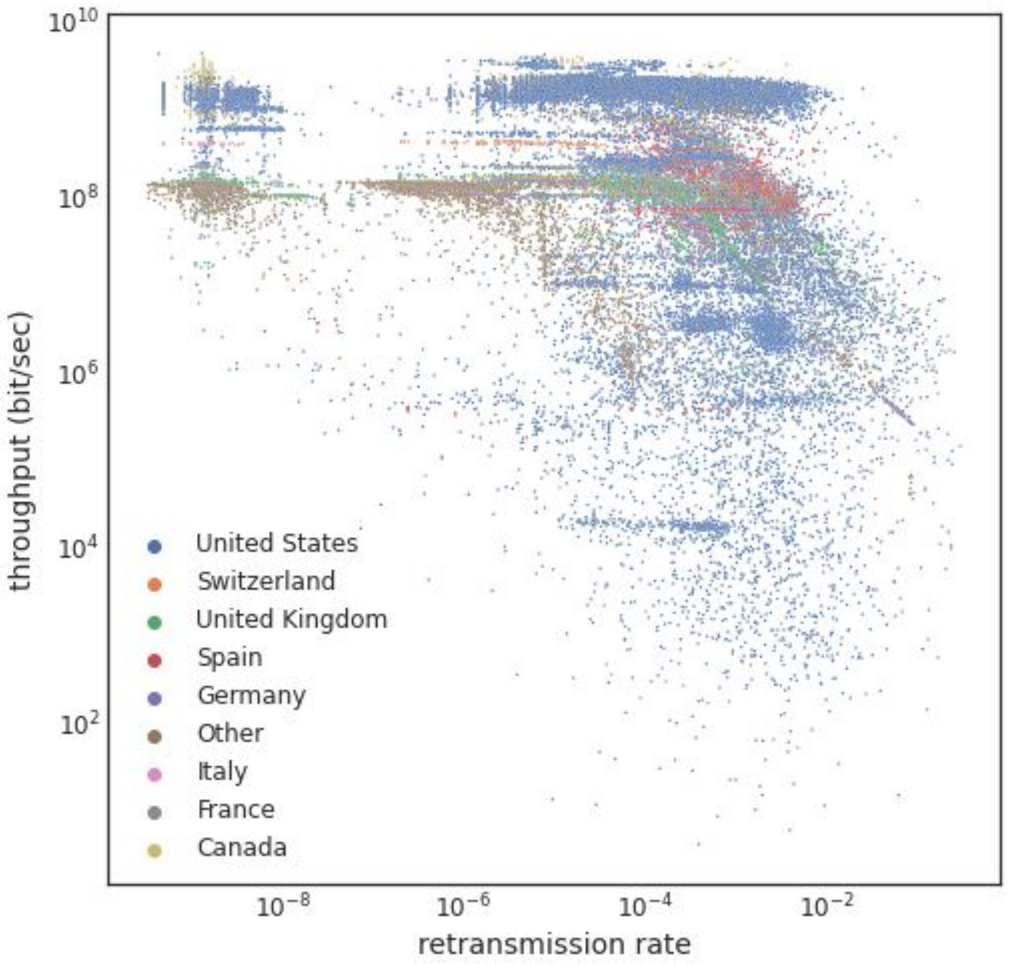
country derived from IP address (GeoIP2)
throughput vs. size





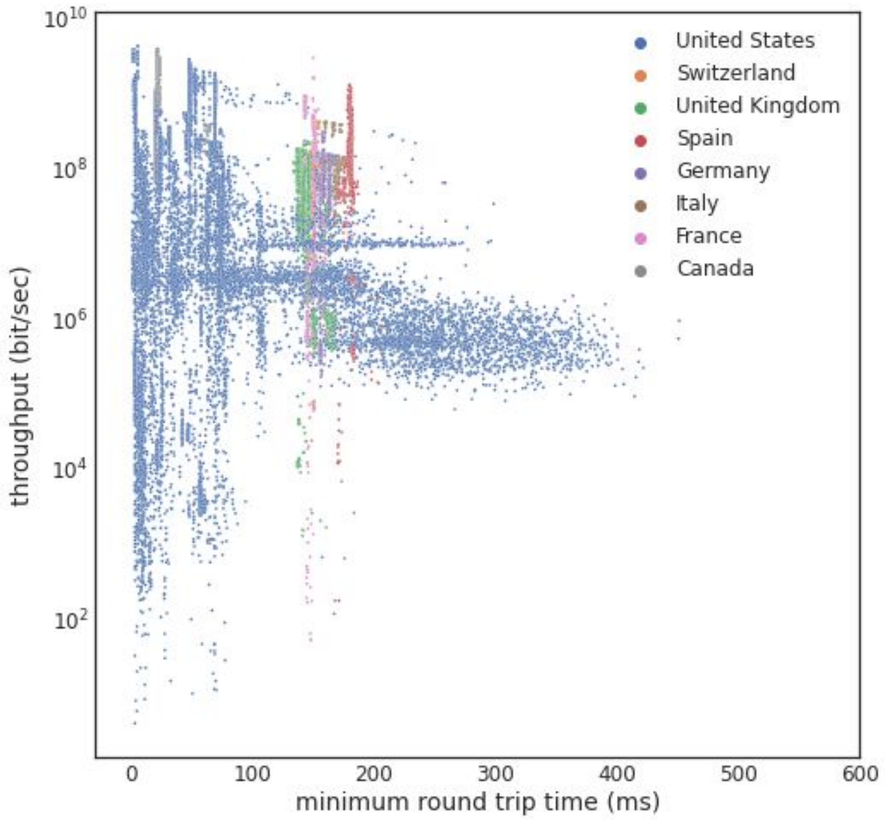
Data Exploration

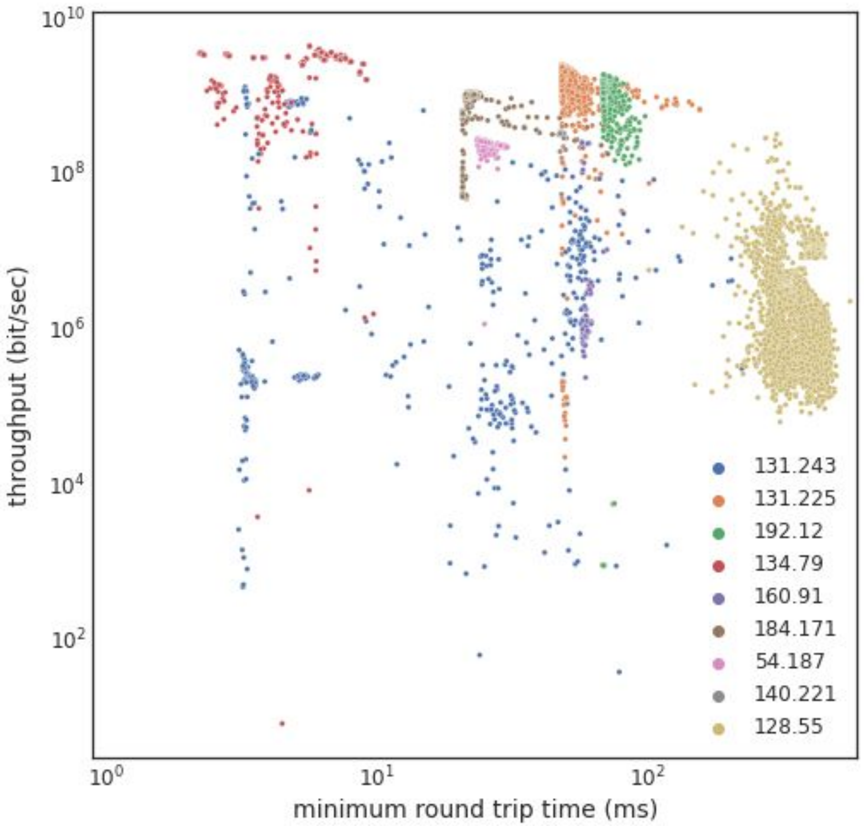
- calculate retransmission rate
- rate has higher correlation than size



Data Exploration

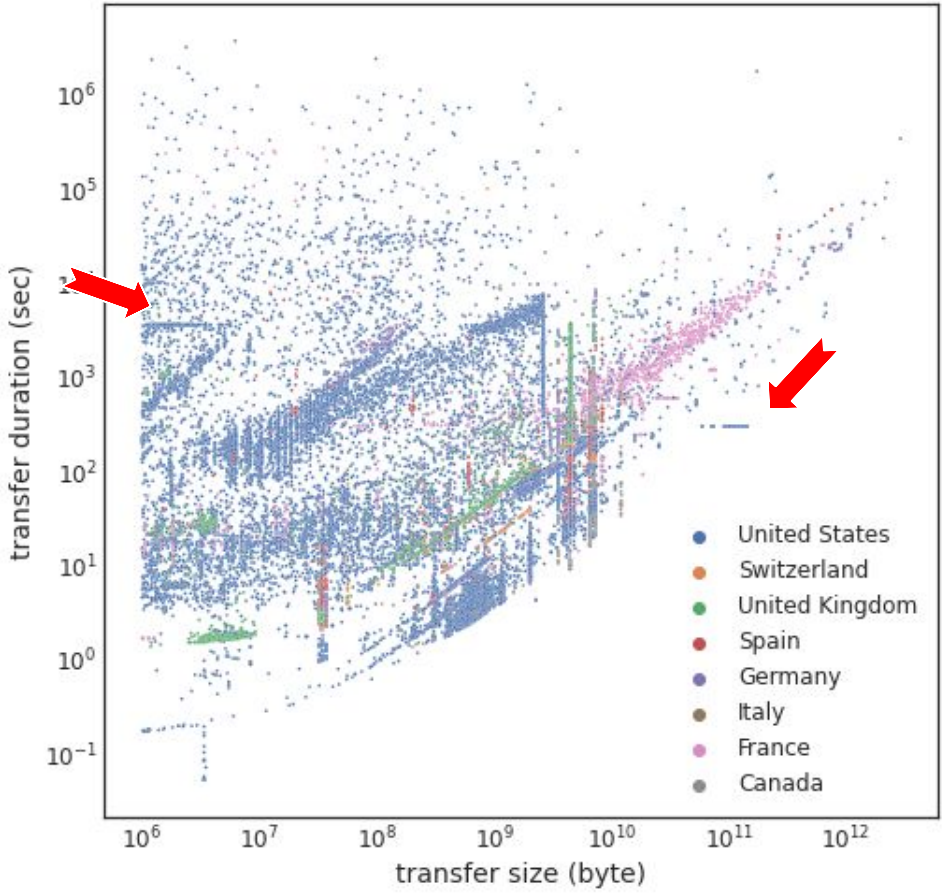
other countries are excluded
throughput vs. RTT min (RTT avg and max have lower correlation)

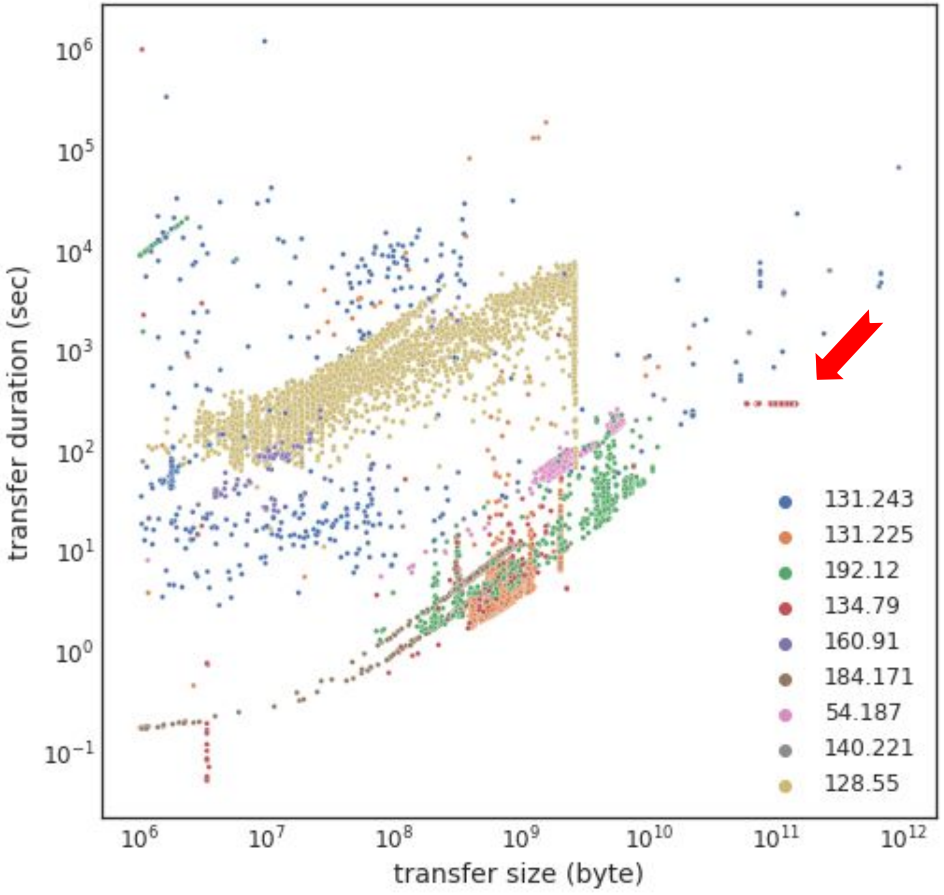




Data Exploration

duration vs. size





Low Performance

- **1 megabits per second** as the threshold
- **15.5%** of the transfers are categorized as low- performance transfers
- also considered duration and retransmission

Feature Engineering

All the new features created:

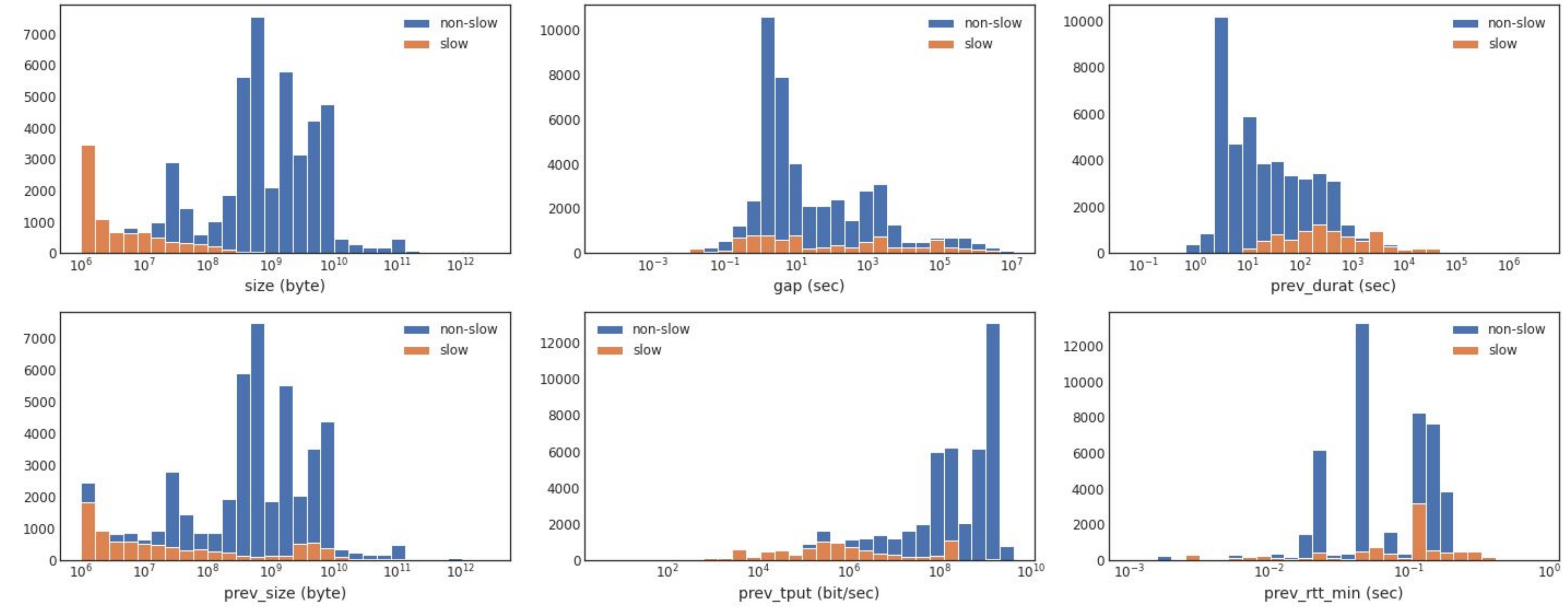
Table 1: Features defined for prediction

Feature	Description
`prev_tput`	Latest throughput measured between the same source and destination networks ("a.b.c.0")
`prev_size`	Latest transfer size (in bytes) between the same source and destination networks ("a.b.c.0")
`size_ratio`	Ratio between the latest transfer size (`prev_size`) vs. current transfer size
`prev_durat`	Latest transfer duration (in msec) between the same source and destination networks ("a.b.c.0")
`prev_min_rtt`	Latest minimum RTT between the same source and destination networks ("a.b.c.0")
`prev_rtt`	Latest average RTT between the same source and destination networks ("a.b.c.0")
`prev_max_rtt`	Latest maximum RTT between the same source and destination networks ("a.b.c.0")
`prev_retx_rate`	Latest retransmission rate between the same source and destination networks ("a.b.c.0")
`time_gap`	Time gap from the latest transfer to the current transfer between the same source and destination networks ("a.b.c.0")

only use latest transfer to maintain time efficiency
create size_ratio and time_gap to identify similarities between current and previous transfer

Feature Engineering

Some of the important features:



Feature Engineering

Labels:

- create labels with thresholds at 1mbps, 10bps, 100kbps

Features:

- have already had **size** and **country** (from IP address)
- created those previous features

Prediction Algorithm

Problem:

- **binary classification** models to **predict** the low-performance transfers from different data origins to NERSC

Algorithms tried:

- **decision trees**, random forests, SVM, **XGBoost**, and neural network

will consider deep learning methods like LSTM and transformer

Experiment settings

Metrics:

- **F1 score**
  - a harmonic mean of Precision and Recall
  - F1 = 2 * (precision * recall) / (precision + recall)

Train-test-split:

- train: Jan-April 21
- test: May 21

Baseline model

Intuition:

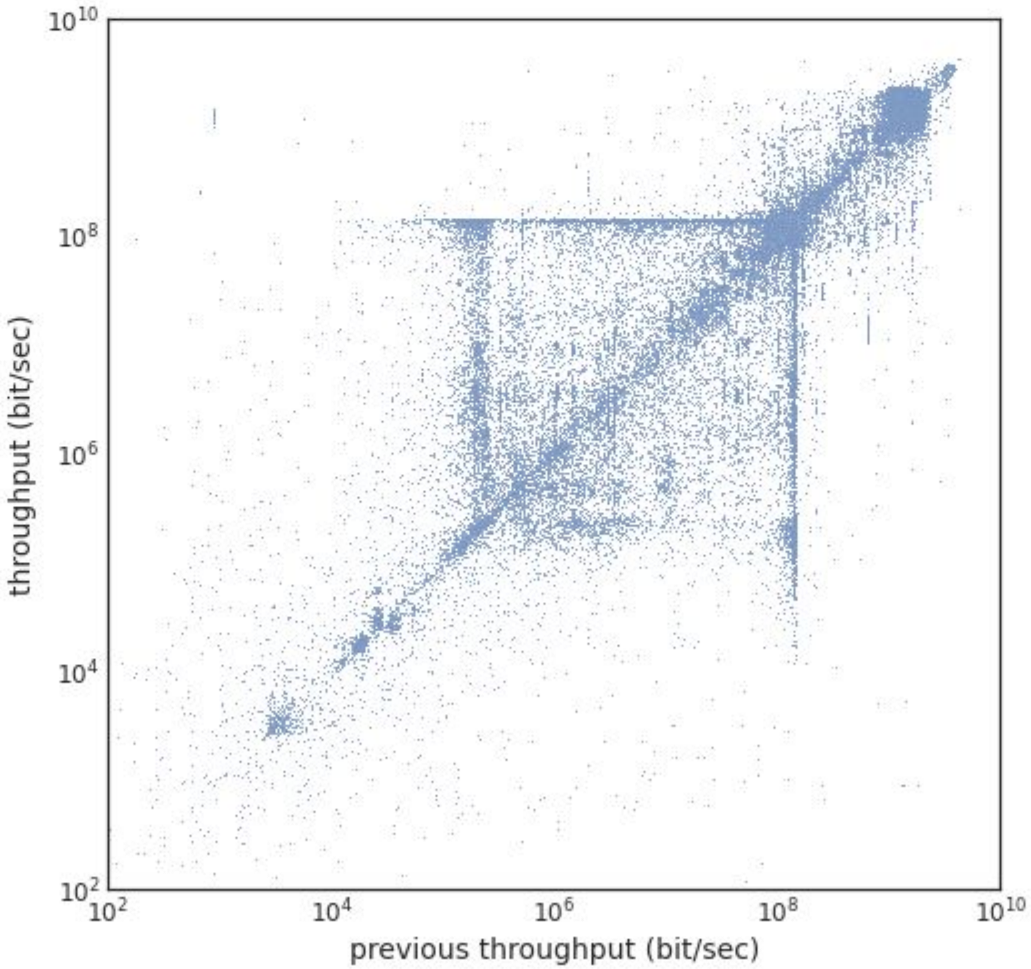
- If the latest transfer is slow, the current one may also be slow

Feature:

prev_tput

Performance:

- 0.77



Decision Tree

Hyperparameter tuning:

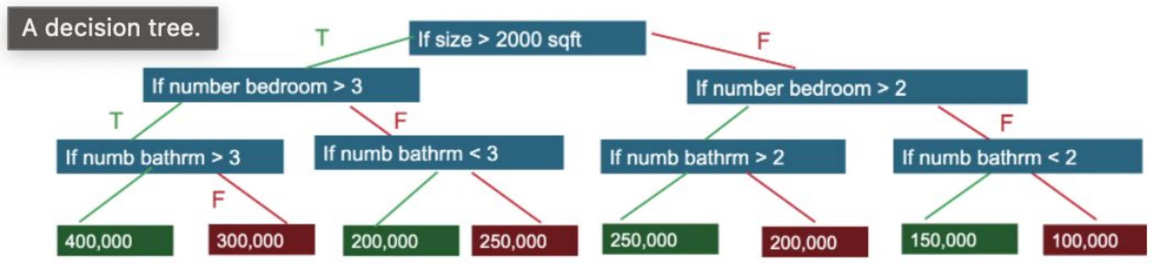
- **GridSearchCV**

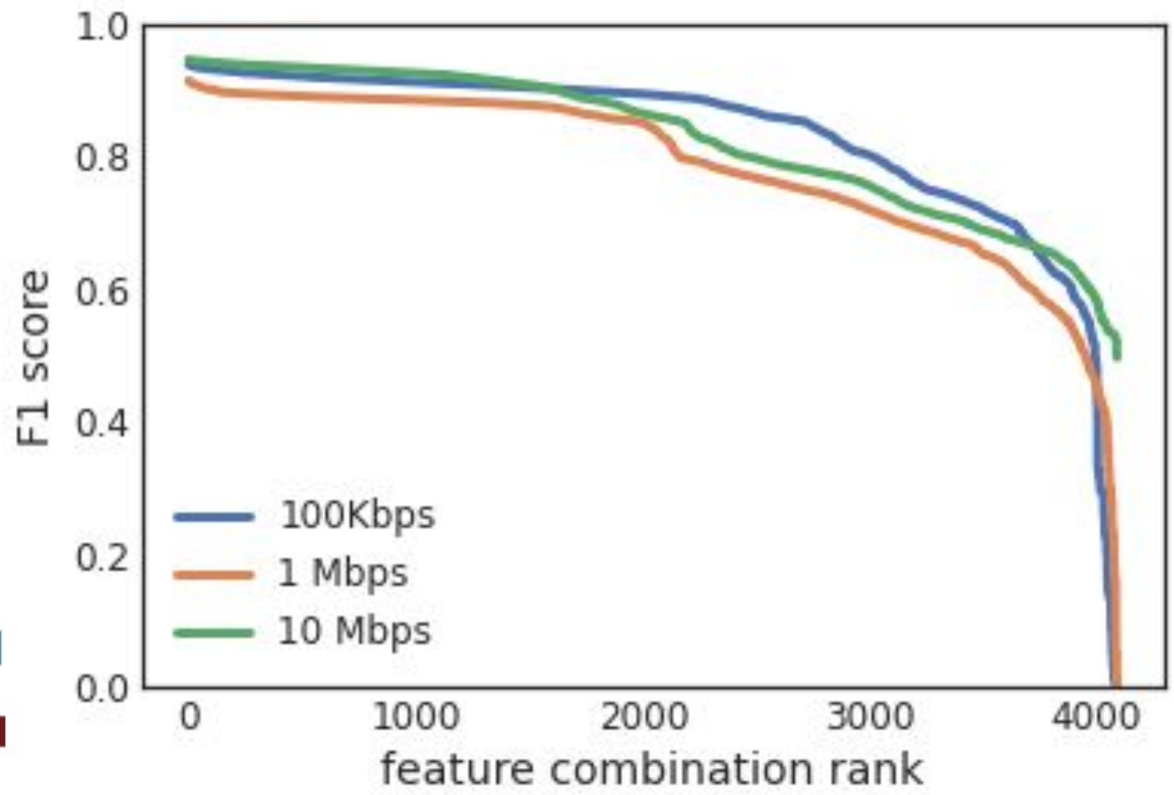
Feature selection:

- **Try-all feature selection**

Best performance:

- 0.913





Effectiveness of feature engineering & selection

Compared with **model without new features**:
Features: **size, country**
F1-score: **0.826** < 0.913

Compared with full **model with all the features**:
Features: **all the features available**
F1-score: **0.891** < 0.913

new features and feature selection improve the performance

XGBoost

Intuition:

- improving a single tree model by combining it with a number of other tree models in order to generate a collectively strong model

Algo Basics:

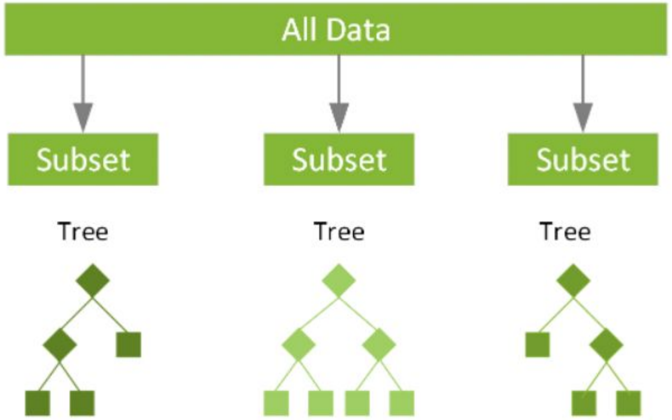
- use error residuals of previous model to fit the next model
- final prediction is a weighted sum of all the tree predictions

Performance (with tuning):

0.913

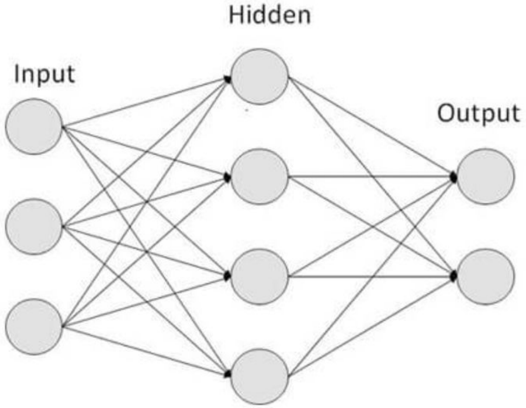
Other models

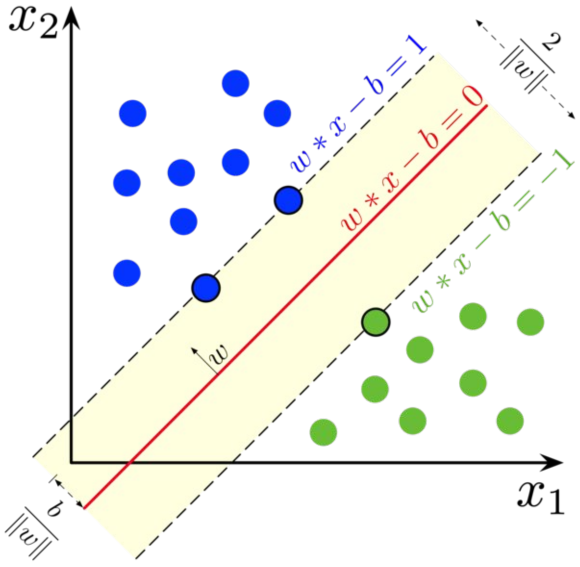
- Random Forest (with tuning)
- Fully Connected Neural Net
- SVM (Support Vector Machine)



Future:

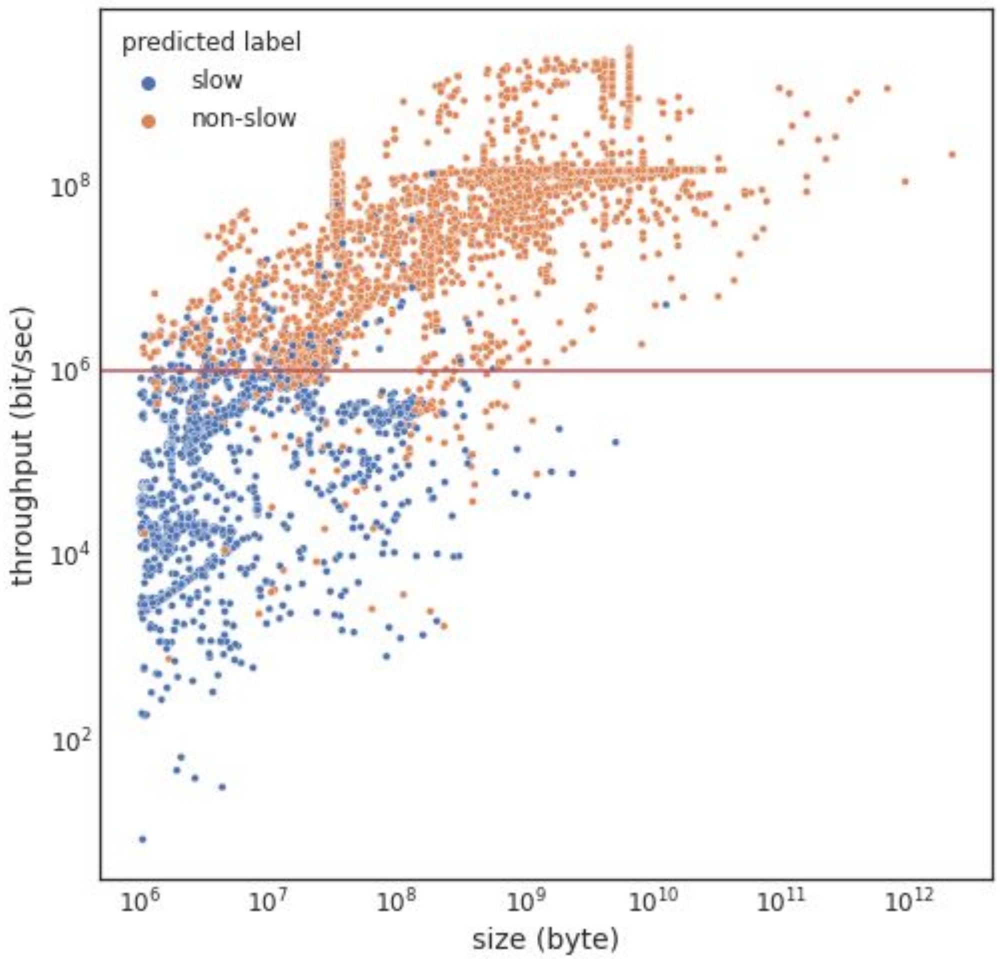
- LSTM (Long Short-term Memory)
- Transformer





Prediction Result

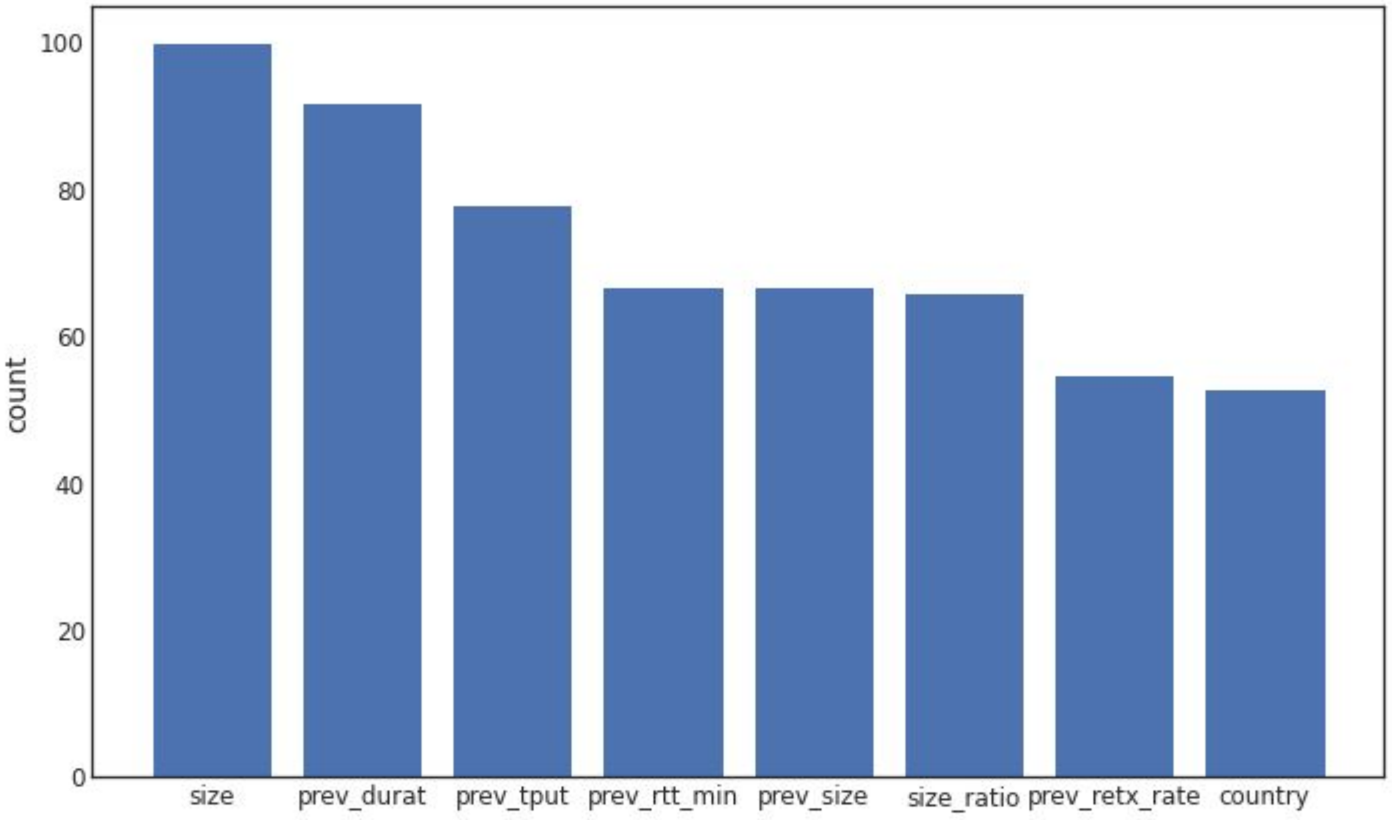
- the **prediction result** presented in throughput vs. size plot
- **orange** dots are predicted to be **non-slow** & **blue** dots are **slow**
- red line is the boundary



Important features

of occurrence in top100 combinations:

- size
- prev_durat
- prev_tput
- prev_rtt_min
- prev_size
- size_ratio
- prev_retx_rate
- country



Conclusion

Contributions:

- got interesting findings through data exploration
- achieved 0.913 F1-score for predicting low performance transfers at 1 Mbps threshold
- achieved as high as 0.945 F1-score at 10 Mbps threshold
- identified important features in determining low-performance transfers

Next Steps:

- explore more advanced algorithms
- explore performance differences on different DTNs in different years

Thanks