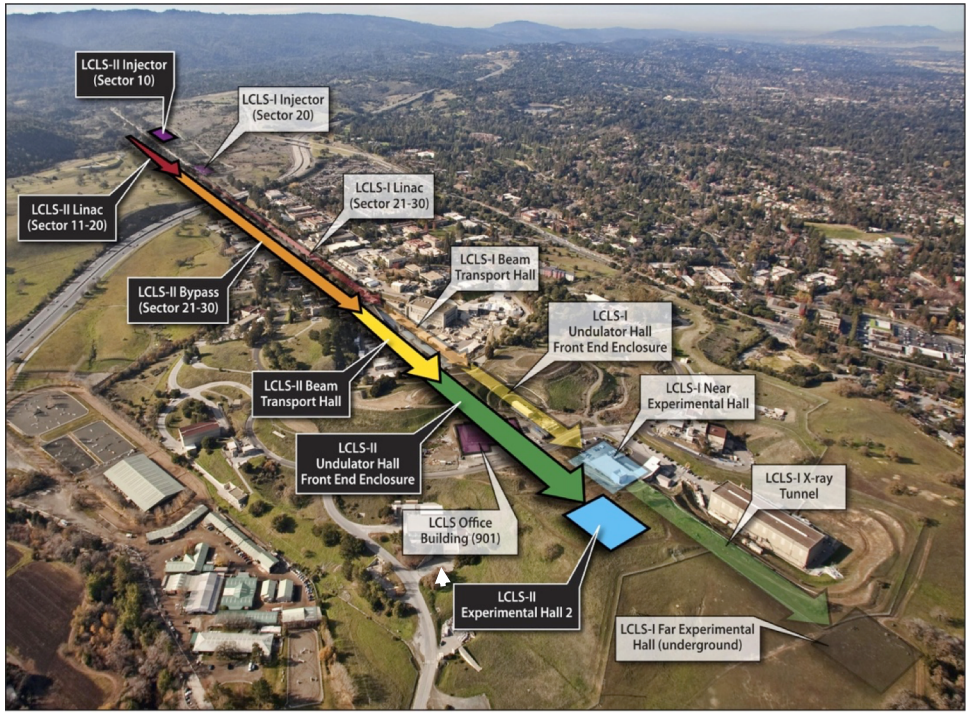
Performance Prediction for Data Transfers in LCLS Workflow

Mengtian Jin[1], Youkow Homma[1], Alex Sim[2],
Wilko Kroeger[3], K. John Wu[2]

1. Stanford University
2. Scientific Data Management Research Group
Computational Research Division
Lawrence Berkeley National Laboratory
3. SLAC National Accelerator Laboratory

LCLS Experiment Utilizes NERSC for Storage and Computing



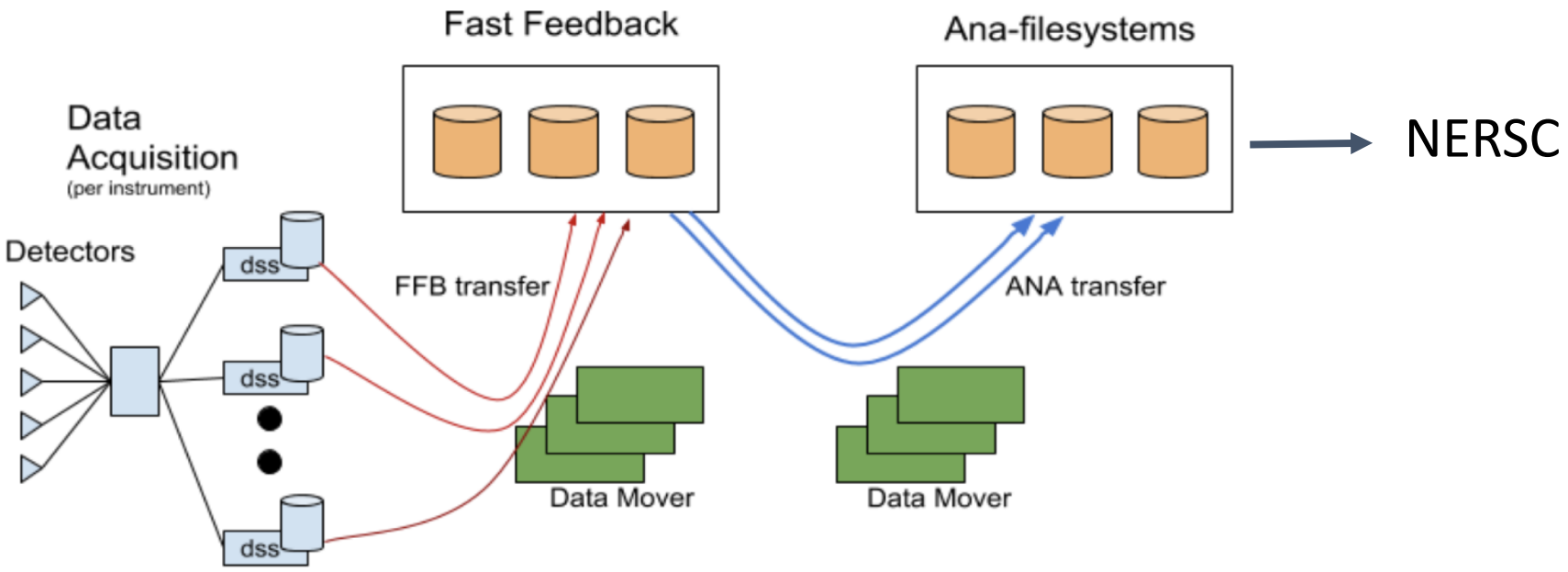
Linac Coherent Light Source (LCLS) @ Stanford CA

- A laser used to image molecules
- Produces terabytes data per experiment
- LCLS-II will produce 10,000 times as much

National Energy Scientific Computing Center (NERSC) @ Berkeley CA

LCLS Data Flow

- **Data pipeline involves Fast Feedback (FFB), Analysis (ANA) and NERSC**

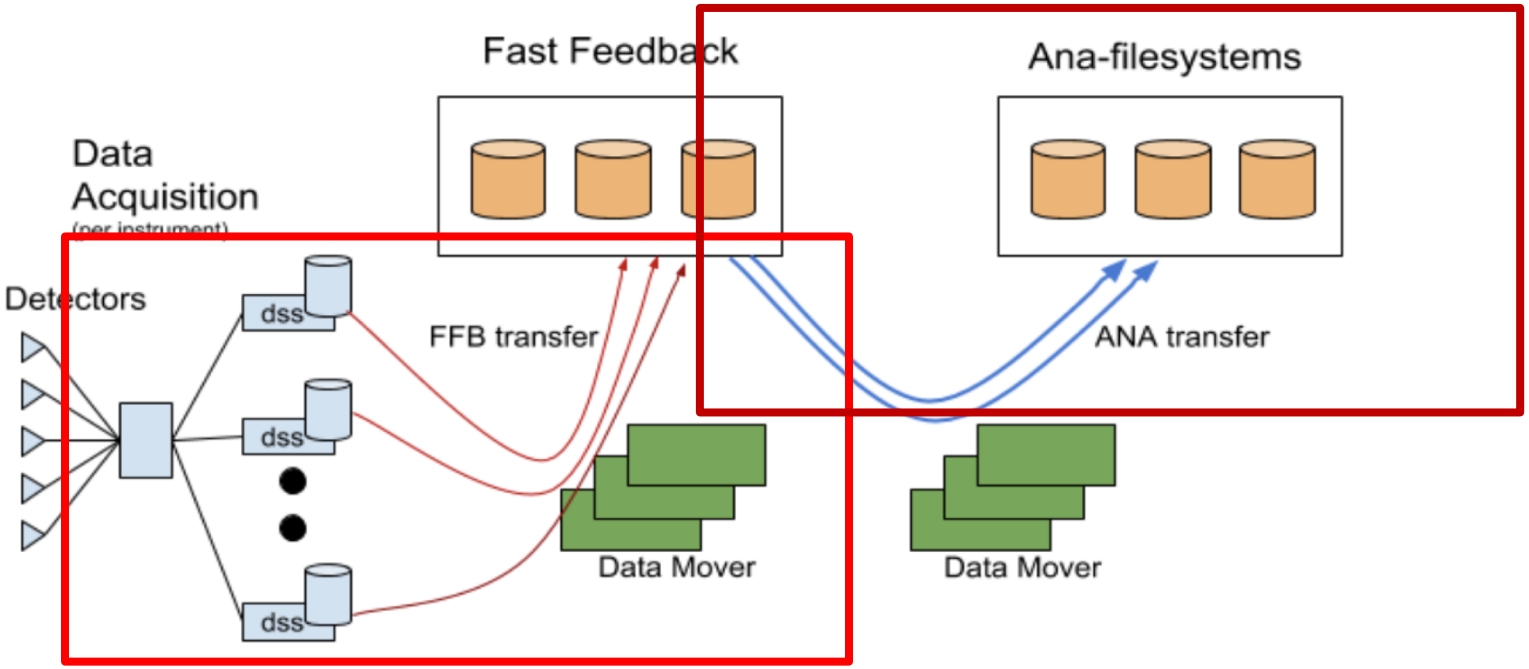


Problem Statement

- To handle the large amount of data expected at LCLS, what are the expected network and I/O performance requirements so that hardware systems could efficiently handle a majority of the workflows in the desired time frame.
- By predicting file transfer rates, we can help make better decisions about
  - When to transfer data from an experiment
  - Resource allocation for avoiding slow-downs
  - Capacity planning for future generation of LCLS

Specific Objectives

- Predict file transfer rate from DSS nodes to FFB
- Predict file transfer rate from FFB to ANA



Data Description

- June 2017 to January 2018
- Recorded information include:
  - time began, time stopped, file size, transfer rate (file-size / (stoptime - starttime)), ffbtrans, file name, instrument involved, source query, target file system, data mover host, localtrans, source host, target host

startt	stopt	fsize	frate	ffbtrans	fn	instr	srcquery	trgfs	dmhost	localtrans	srchost	trghost
1.504066e+09	1.504071e+09	1304.142128	257.4	0	e1061-r0421-s02-c00.xtc	xpp	daq-xpp-dss03-ana-ana	ana12	psexport08	1	psexport08	psexport08
1.504066e+09	1.504071e+09	1293.475224	278.8	0	e1061-r0421-s05-c00.xtc	xpp	daq-xpp-dss06-ana-ana	ana12	psexport07	1	psexport07	psexport07

CS Contribution 1: Feature Engineering

- Statistics on last finished job on same instrument, target file system, target host, node
  - Transfer rate, File Size
  - Time between last finished job's end time and this job's start
- Same statistics above for last finished job of same chunk, if available
- Time difference between the start time of the first job of the chunk and the start time of the current job.
- Number of total jobs and number of unique experiments running on the same trgfs, trghost, and node, respectively, when the current job is started
- Time of day and day of week
- Stream #: how many files are written in parallel
- Experiment #, Run #

CS Contribution 2: Hyperparameter selection through Nested CV

- Designed for time series data to temporal order between training data and testing data
- Select best hyperparameters to minimize RMSE

1: Inputs: Training set (X), number of hyperparameters to try (num_params), number of CV iterations (k), CV training and test widths ($train\_width$, $test\_width$), CV training and test set sizes ($train\_size$, $test\_size$)
2: Sort X in increasing order of start time
3: **for** i = 1 to num_params **do**
4: Set $\alpha \leftarrow$ randomly sampled hyperparameters
5: **for** j = 1 to k **do**
6: Set $train\_region \leftarrow$ random consecutive $train\_width$ rows of X
7: Set $train\_set \leftarrow$ random subset of $train\_region$ of size $train\_size$
8: Set $test\_region \leftarrow test\_width$ rows of X following $train\_region$
9: Set $test\_set \leftarrow$ random subset of $test\_region$ of size $test\_size$
10: Train Xgboost model on $train\_region$ and evaluate performance (RMSE) on $test\_region$, where the prediction is max(0, Xgboost prediction).
11: **end for**
12: **if** average of test RMSE's for α is lowest so far **then**
13: Set $\alpha_{best} \leftarrow \alpha$
14: **end if**
15: **end for**
16: **return** α_{best}

	Train width, subsample train size	Test width, subsample test size	

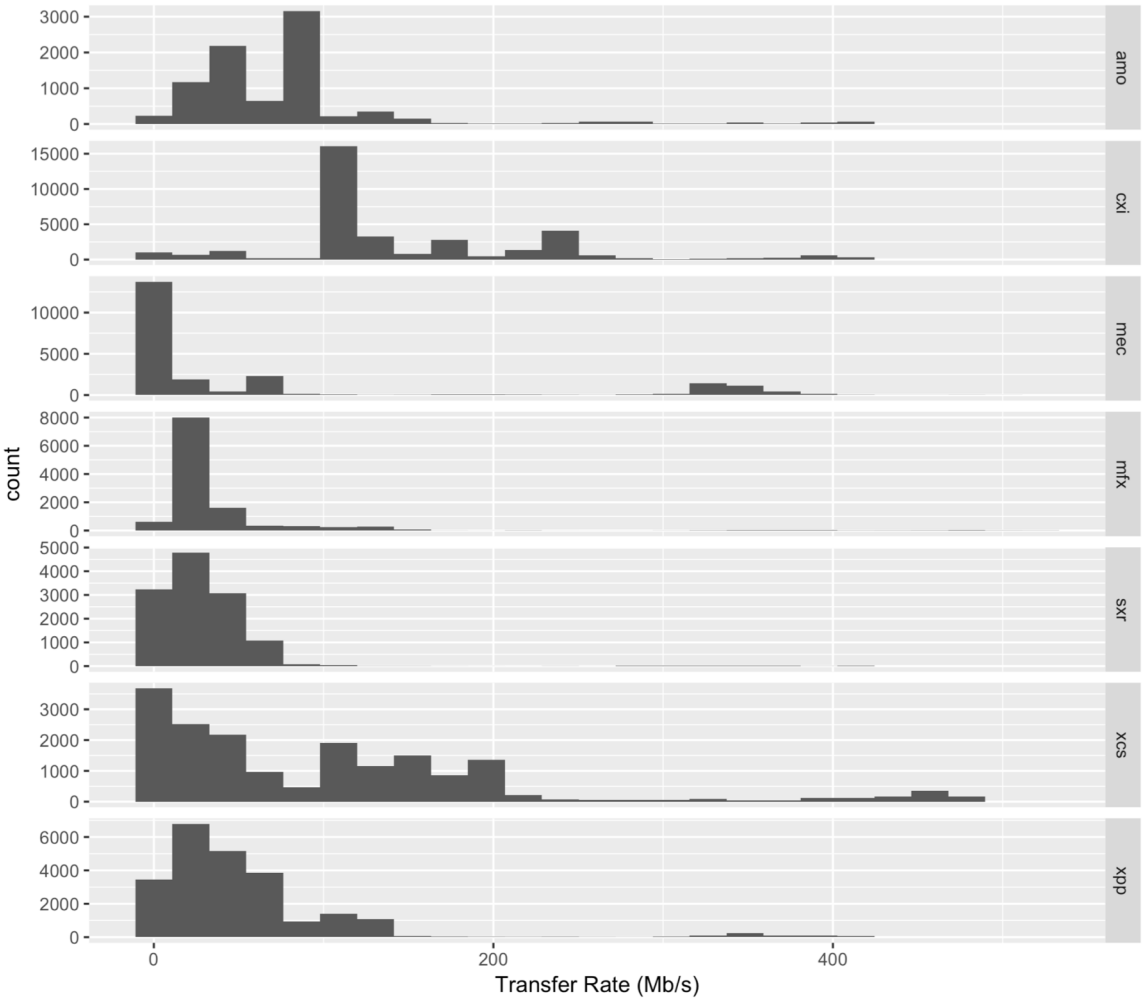
Structure of DSS -> FFB Transfers

- **Instrument location determines file system, host**

Location	Instrument	Host	File System
NEH	amo, sxr, xpp	ffb11	psana102, psana103
FEH	cxi, mec, mfx, xcs	ffb21	psana201, psana202, psana203

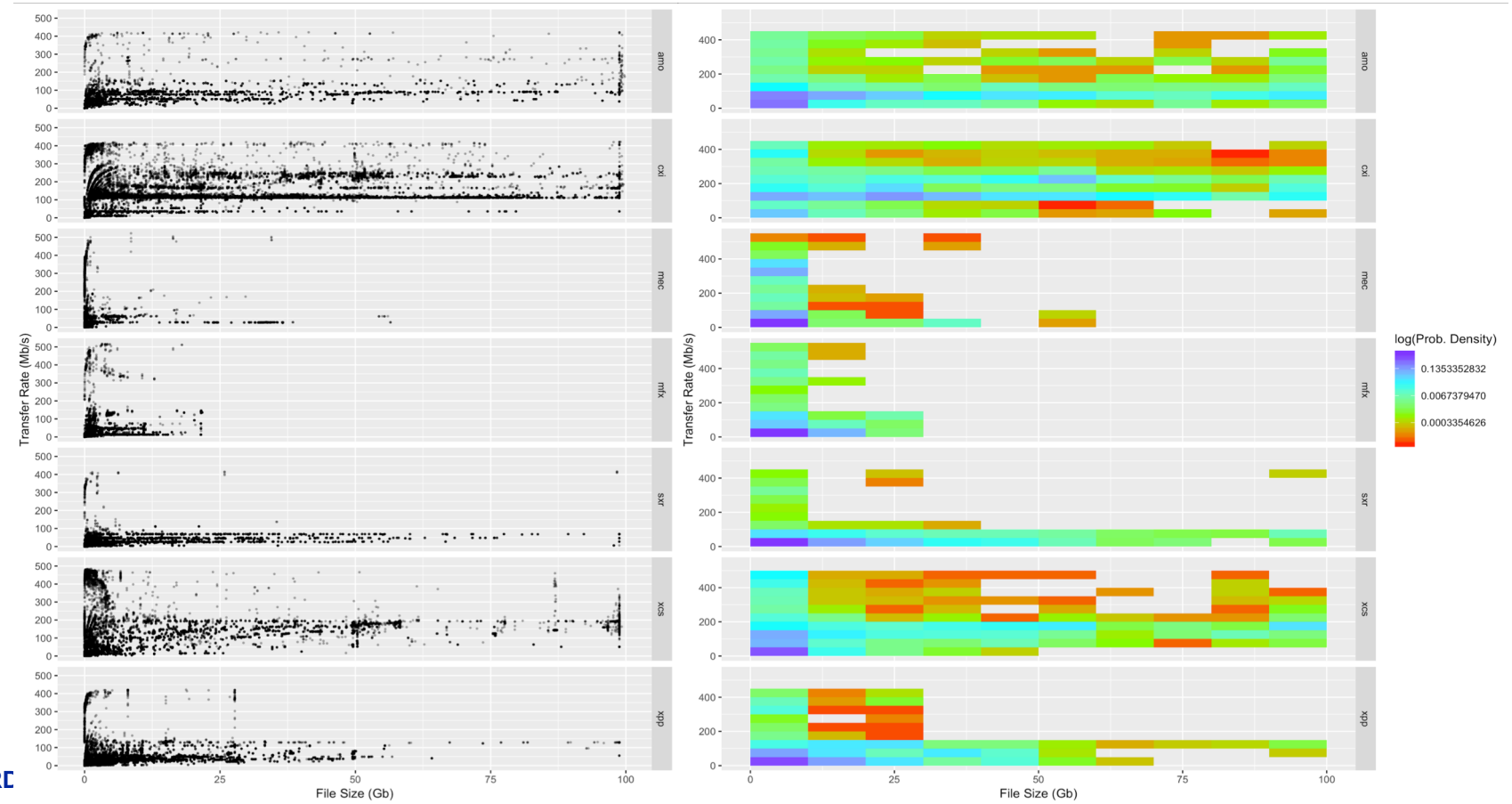
Transfer Rate by Instrument

- **Distributions of transfer rate vary significantly across instruments due to write speed of instrument**



Different Types of Experiments Show Different Transfer Rates to FFB

- The average transfer rates are consistent across file sizes with maybe a slight positive relationship
- Smaller files have greater variability in transfer rates
- Transfer rates of larger files tend to cluster more tightly

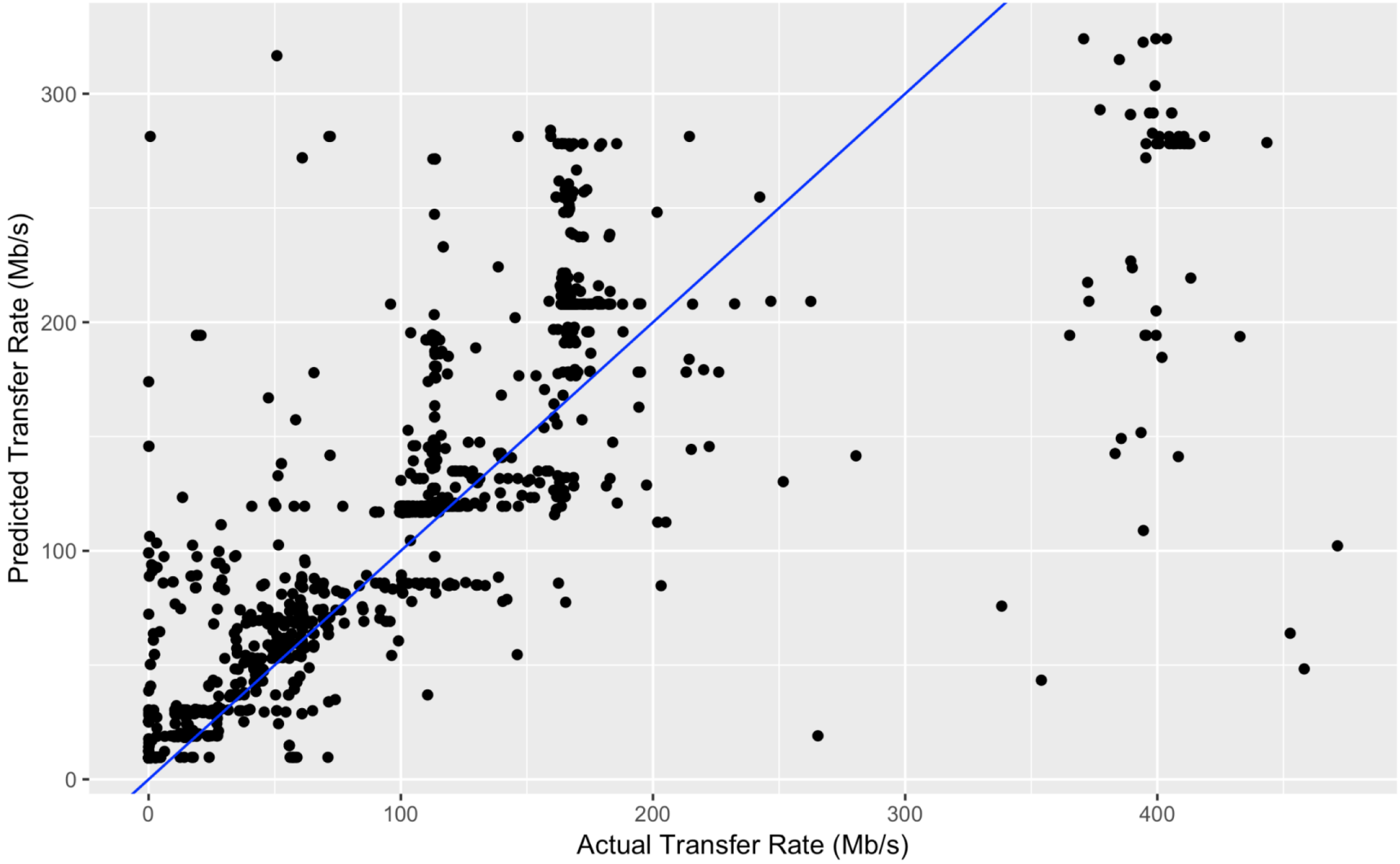


Model 1: Random Forest

- **Train a random forest to predict file transfer rates using raw features and all new features except:**
  - **Previous experiment data of same chunk since random forest does not handle missing values**
- **Random Forest (100 trees, max depth=4)**
  - **RMSE = 52.8**
  - **Normalized feature importance**
    - **Lagged transfer rates are the most important**

Feature	Importance
Last Job Instrument, Transfer Rate	70.4%
Last Job Host, Transfer Rate	7.9%
Last Job Node, Transfer Rate	7.0%
File Size	4.1%
Last Job Instrument, Stop Time Difference	4.0%

Random Forest: Actual vs Predicted

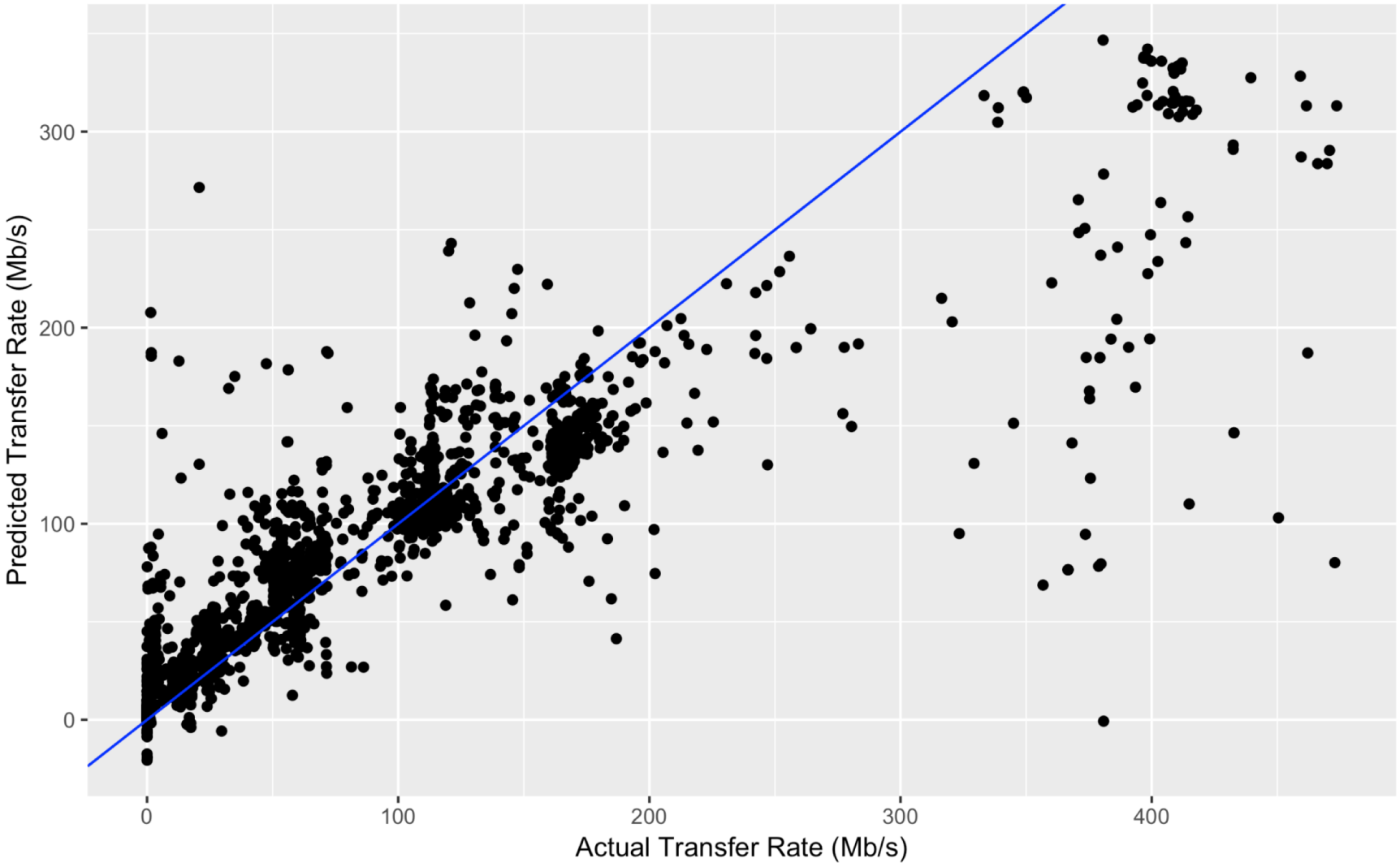


Model 2: Xgboost

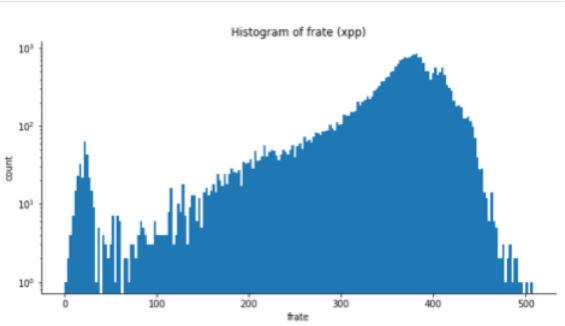
- **Train a random forest to predict file transfer rates using all raw and new features**
- **Test RMSE = 39.0 (13 point improvement on Random Forest)**
- **Feature Importance**
  - **Categorical features more pronounced (instrument, target host)**
  - **Day of week (seasonality) also in play**

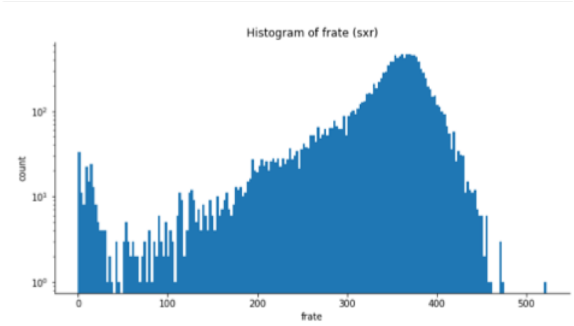
Feature	Importance
Instrument	12.9%
Last Experiment - File Size	9.5%
Day of the week	7.9%
Last Experiment - Stop Time	7.5%
Target host	6.2%
Last Experiment - Transfer Rate	5.3%
Last Node - Stop Time	4.4%
Last Instrument -Transfer Rate	3.9%

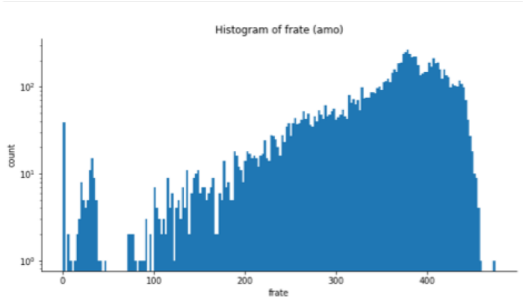
Xgboost: Actual vs Predicted

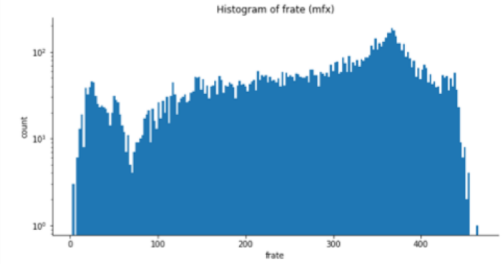


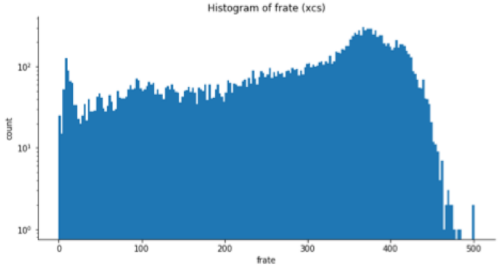
Statistics of Transfers to ANA

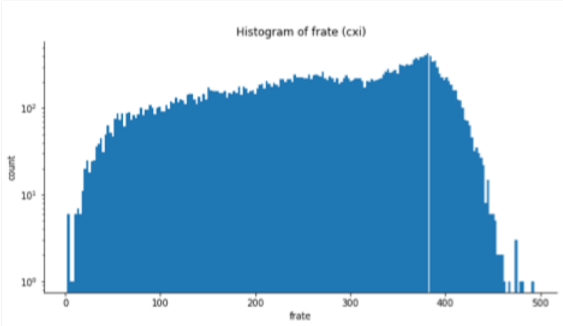


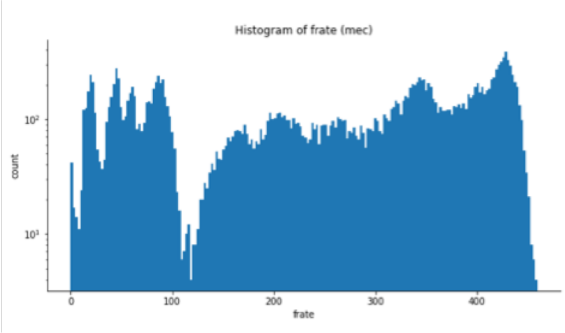












Histogram of file transfer rates by different Instruments

Model 1: Gradient Boosting Tree

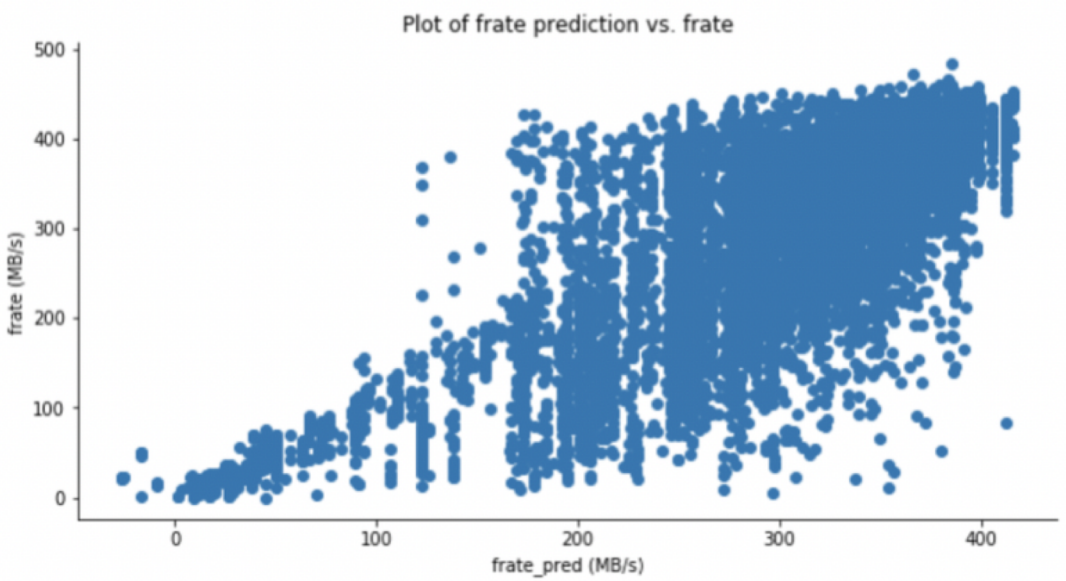
- Boosting methods + Regression trees
- Compute a sequence of simple trees, where each successive tree is built for the prediction residuals of the preceding tree.
- 90% Training
- 10% Testing
- Features Selected
  - File Size
  - Instruments
  - Sources (srcfs)
  - Target (trgfs)
  - Experiment #

Model 1: Gradient Boosting Tree

- **With Time Independent Feature Set**
  - File size
  - instr: one-hot-encoding
  - Srcfs: one-hot-encoding
  - Trgfs: one-hot-encoding
  - Experiment #: label-encoding

RMSE: 64.3MB/s

- File Size is the dominant factor
- Experiment number also shows high importance
- Cxi has highest importance among all instruments

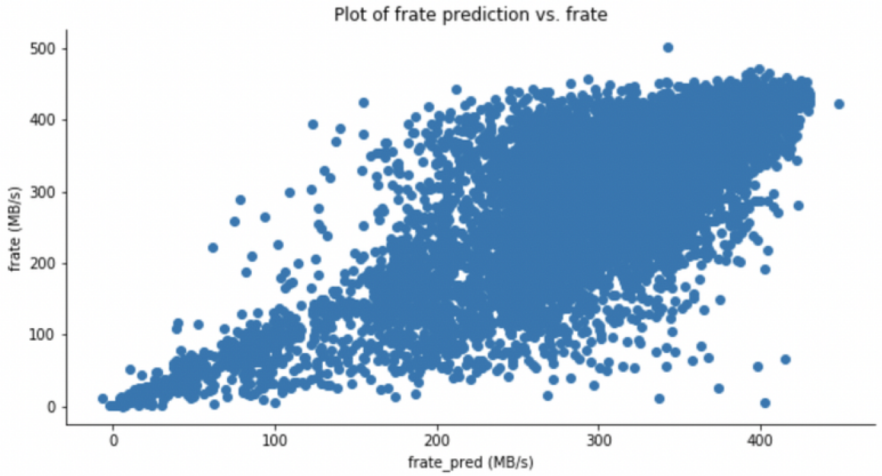


Feature	Importance
File Size	66.827%
experiment number	17.325%
cxi(instrument)	6.153%
mfx(instrument)	2.604%
mec(instrument)	0.945%
xpp(instrument)	0.219%
sxr(instrument)	0.119%
amo(instrument)	0.011%
ffb11(srcfs)	4.855%
ana01(trgfs)	0.491%
ana11(trgfs)	0.418%
ana12(trgfs)	0.033%

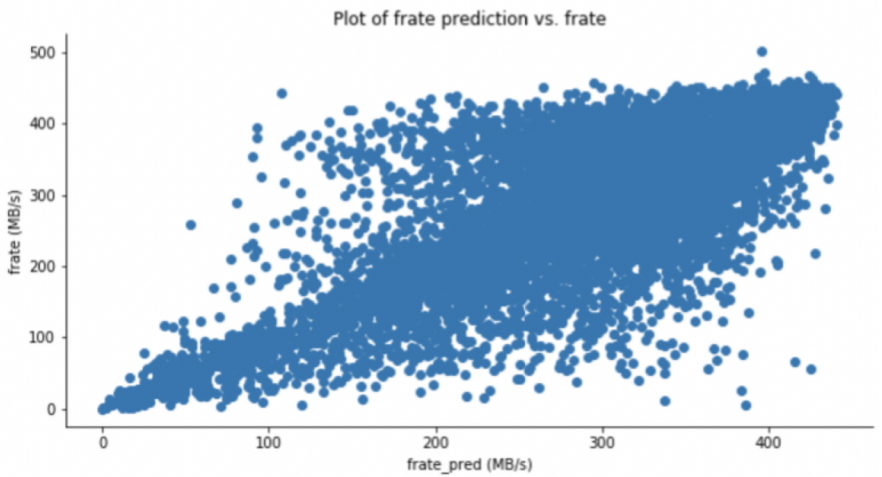
Model 2: Gradient Boosting Tree

- With Time Dependent Feature Set
  - Lag Variables
  - Lag 1 of transfer rate on the same instrument
  - Lag 1 of transfer rate from the same experiment
  - Lag 5 of transfer rate, overall
  - Lag 1 of file size, overall
- Features already in the model:
  - File size
  - instr: one-hot-encoding
  - Srcfs: one-hot-encoding
  - Trgfs: one-hot-encoding
  - Experiment #: label-encoding

Time Dependent Features Reduce Prediction Errors



Result from Gradient Boosting Tree (Model 2)
RMSE:58.6MB/s



Result from Random Forest (Model 3)
RMSE: 60MB/s

Model 1: 64.3MB/s -- 8.8% improvement

Looking into the Unusual Cases

- Sudden change in transfer rate
- Similar configurations as others
- Only noticeable different is transfer start time
- Hypothesis: gap in start time affects transfer rate

Example 1: file transfer with error > 300MB/s

fn	startt	stopt	fsize	frate	instr	trgfs	srcfs	experiment_num	run	stream	chunk
e1075-r0260-s04-c00.xtc	2017-10-02 20:32:17	2017-10-02 20:34:20	47.357902	394.1	cxi	ana11	ffb21	62	r0260	4	c00.xtc
e1075-r0260-s03-c01.xtc	2017-10-02 20:37:18	2017-10-02 20:39:39	47.355528	345.8	cxi	ana11	ffb21	62	r0260	3	c01.xtc
e1075-r0260-s01-c01.xtc	2017-10-02 20:37:20	2017-10-02 21:52:12	47.352599	10.8	cxi	ana11	ffb21	62	r0260	1	c01.xtc
e1075-r0260-s00-c01.xtc	2017-10-02 20:37:33	2017-10-02 20:39:37	47.365109	389.4	cxi	ana11	ffb21	62	r0260	0	c01.xtc
e1075-r0260-s04-c01.xtc	2017-10-02 20:41:20	2017-10-02 20:43:17	47.364330	416.2	cxi	ana11	ffb21	62	r0260	4	c01.xtc
e1075-r0260-s04-c02.xtc	2017-10-02 20:44:39	2017-10-02 20:46:39	47.362555	403.6	cxi	ana11	ffb21	62	r0260	4	c02.xtc
e1075-r0260-s03-c02.xtc	2017-10-02 20:44:40	2017-10-02 20:46:36	47.359036	419.0	cxi	ana11	ffb21	62	r0260	3	c02.xtc

Model 4: Gradient Boosting Tree

- Time difference between consecutive transfers
- Time Dependent Feature Set
  - Lag 1 of transfer rate on the same instrument
  - Lag 1 of transfer rate from the same experiment
  - Lag 5 of transfer rate, overall
  - Lag 1 of file size, overall
- Other Features
  - File size
  - instr: one-hot-encoding
  - Srcfs: one-hot-encoding
  - Trgfs: one-hot-encoding
  - Experiment #: label-encoding

Model 4: Gradient Boosting Tree

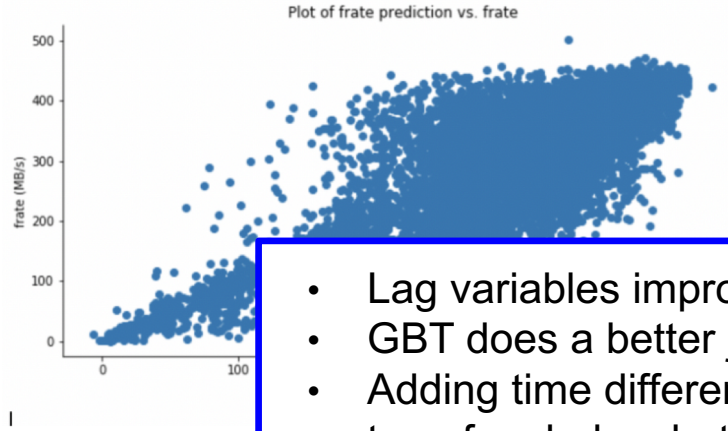
- **Cross Validation on GBT**
  - **learning rate = 0.1, n_estimators = 600, max features = 4.12, max depth = 11, min samples split = 700, min samples leaf = 10**
  - **RMSE:56.9 MB/s**

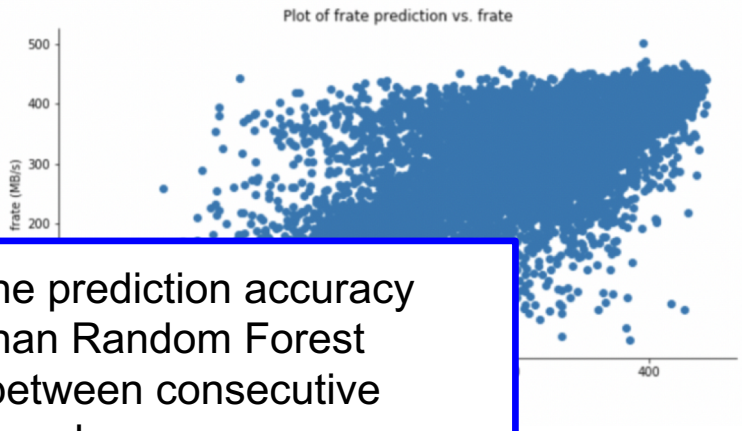
Feature	Importance
File Size	26.868%
lag1 from same experiment - transfer rate	22.614%
lag1 on same instrument - transfer rate	13.642%
lag1 overall - file size	9.254%
lag5 overall - transfer rate	8.766%
experiment number	6.496%
ffb11(srcfs)	3.599%
cxi(instrument)	1.937%
mfx(instrument)	0.574%
mec(instrument)	0.830%
xpp(instrument)	0.797%
sxr(instrument)	0.193%
amo(instrument)	0.298%
time difference between same experiment	2.940%
ana01(trgfs)	0.359%
ana11(trgfs)	0.538%
ana12(trgfs)	0.295%

Table 5: Feature Importance of GBT with Time Difference and Cross Validation

- Importance score of file size decreased by almost a half.
- Lag variables become dominant.

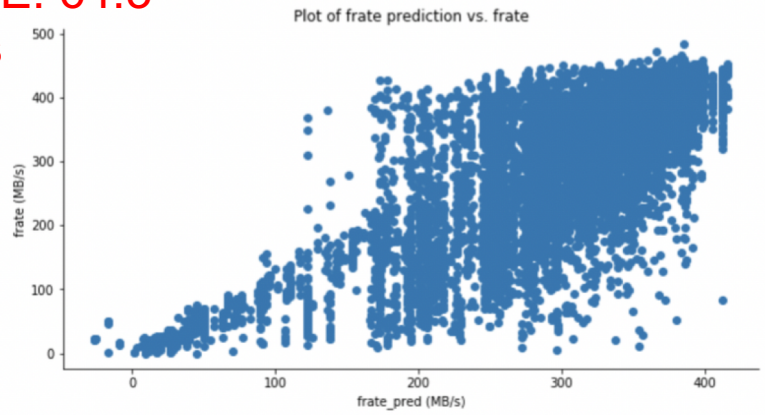
Model Comparison



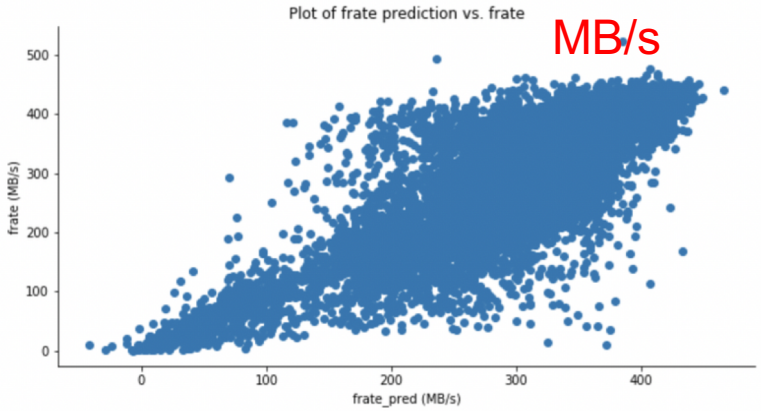


- Lag variables improve the prediction accuracy
- GBT does a better job than Random Forest
- Adding time difference between consecutive transfers helps, but not much

RMSE: 64.3 MB/s



RMSE: 56.9 MB/s



(**Top Left**: GBT in Time Dependent Model, **Top Right**: Random Forest in Time Dependent Model, **Bottom Left**: GBT in Time Independent Model, **Bottom Right**: GBT in Time Dependent Model with CV)

Summary

- **Improve transfer rate prediction for transfers to both FFB and ANA with**
  - **The status of the current system**
  - **Statistics related to recent transfers**
- **Data acquisition device capability dominates transfer rates to FFB**
- **Gradient Boosting Tree performs slightly better than Random Forest with the same feature set in both FFB and ANA process.**
  - **Random Forest emphasizes more on lag variables in the model**
  - **Lag variables are helping the prediction of most of the file transfers**
- **Still have trouble with predictions when the transfer rate experiences a dramatic change**