

## **Abstract**

Currently, there is no prediction provided to users for the amount of time a particular data transfer from one site in the Worldwide LHC Computing Grid to another will take to complete. To develop a time-to-complete prediction, network performance data and per-file information is gathered from two separate databases and fused, and the resulting cleaned data is fitted using random forest regression. Results are shown for two separate links: the link from CERN Data Centre to Brookhaven National Laboratory's ATLAS data center, and the link from CERN Data Centre to SARA-MATRIX in Amsterdam. A total RMS error of 25.93 minutes between predicted and test data is found for the CERN-PROD -> BNL-ATLAS link, while the CERN-PROD -> SARA-MATRIX link yields a total RMS error of 3.00 minutes.

## **Introduction**

The data taken at the Large Hadron Collider (LHC) are stored and distributed via the Worldwide LHC Computing Grid (WLCG), a network of computing centers and users across the globe. Currently, there is no prediction provided to users for the amount of time a particular data transfer from one site to another will take to complete. The goal of this project is to develop a model to predict the time-to-complete (TTC) of a file based on file size and type, including network status and performance at the time of file submission.

## **Data Acquisition**

The data for this model is taken from two different Elasticsearch databases, over a 4-week interval. Maintained by the University of Chicago, "rucio-events" is an ATLAS data management system that contains per-file information for all file transfers. The "ddm-metrics"

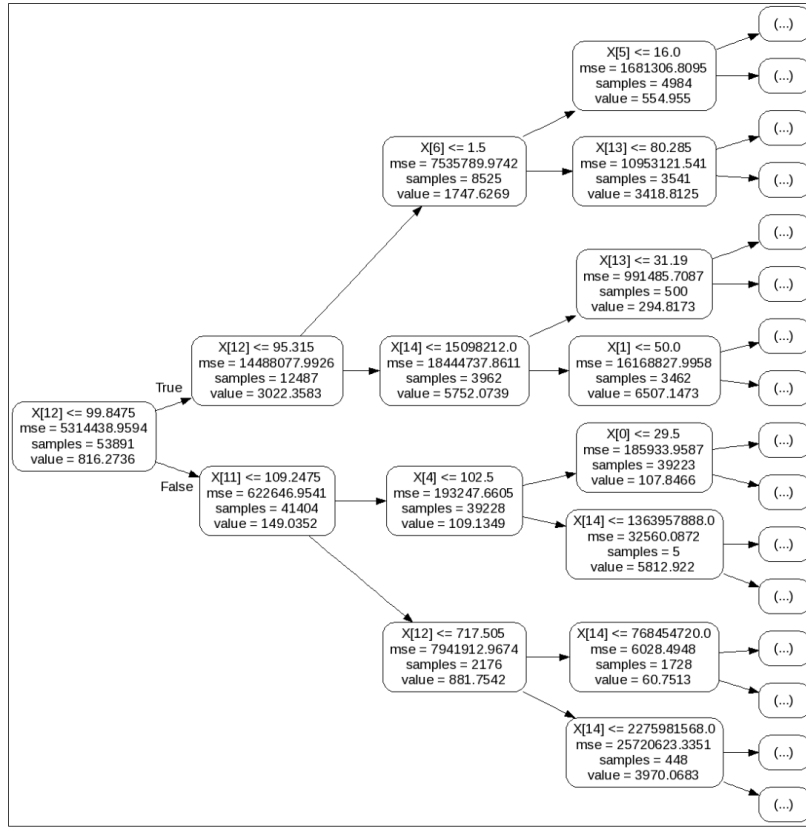
data contains network performance and status information, and is aggregated from a database maintained at CERN.

Instead of taking all files from the four weeks, a random sample of 100 files per 10 minutes is gathered from rucio-events. This is done to avoid overfitting on intervals that contain many files submitted in a short period. Applying this sampling reduces the total size of the data considered by about one-half.

Within rucio-events, each file has a size in bytes and an activity type which depends on the file's function. Finally, timestamps are recorded when the file is submitted to the queue, when the file begins transferring, and when the file has completed transferring. The TTC data that is used as the dependent variable within the regression model is computed from these timestamps.

Since certain activities are prioritized by the data transfer application, it is necessary to distinguish between activity type when constructing the model. In order to prepare the activity data for regression, each value is coded as an n-vector, where n is the number of distinct activities. For each observation, the index corresponding to the appropriate category holds a value of 1, and the rest of the indices are zero. For example, in the case of a variable with 3 possible categories the first is encoded [1,0,0], the second [0,1,0], and the third [0,0,1].

From ddm-metrics, throughput measures are acquired, as well as fields containing the number of files queued per activity. These fields are aggregated internally and updated at a rate of once every 5 or 10 minutes, depending on the field. For ease of comparison, these results are normalized to bin widths of 10 minutes, aggregating by sum for the queued fields and by mean for the throughput measures.

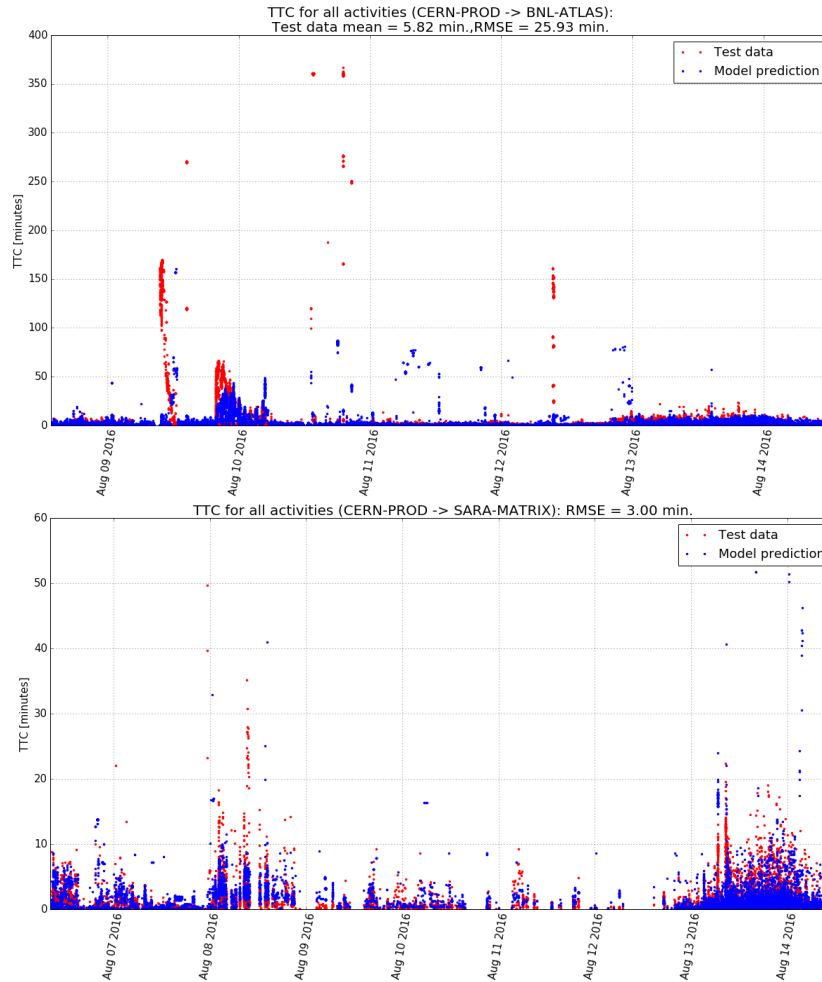


**Figure 1:** One of the regression trees (truncated) used for CERN-PROD->SARA-MATRIX.

To connect the rucio-events data to the ddm-metrics data, each file from rucio-events is associated with the appropriate aggregated data bin from ddm-metrics by the file's "submitted\_at" timestamp. The joined dataset matrix rows are then sorted according to increasing "submitted\_at" timestamp. Finally, any rows with missing values are not used for regression.

## Regression Model

Decision tree learning is an effective and efficient tool for classification and regression of large datasets, and may find nonlinear relationships between variables. Since decision trees are prone to overfitting, multiple random samples may be generated from the training data and fitted with separate decision trees. This method generates a forest of predictions that is

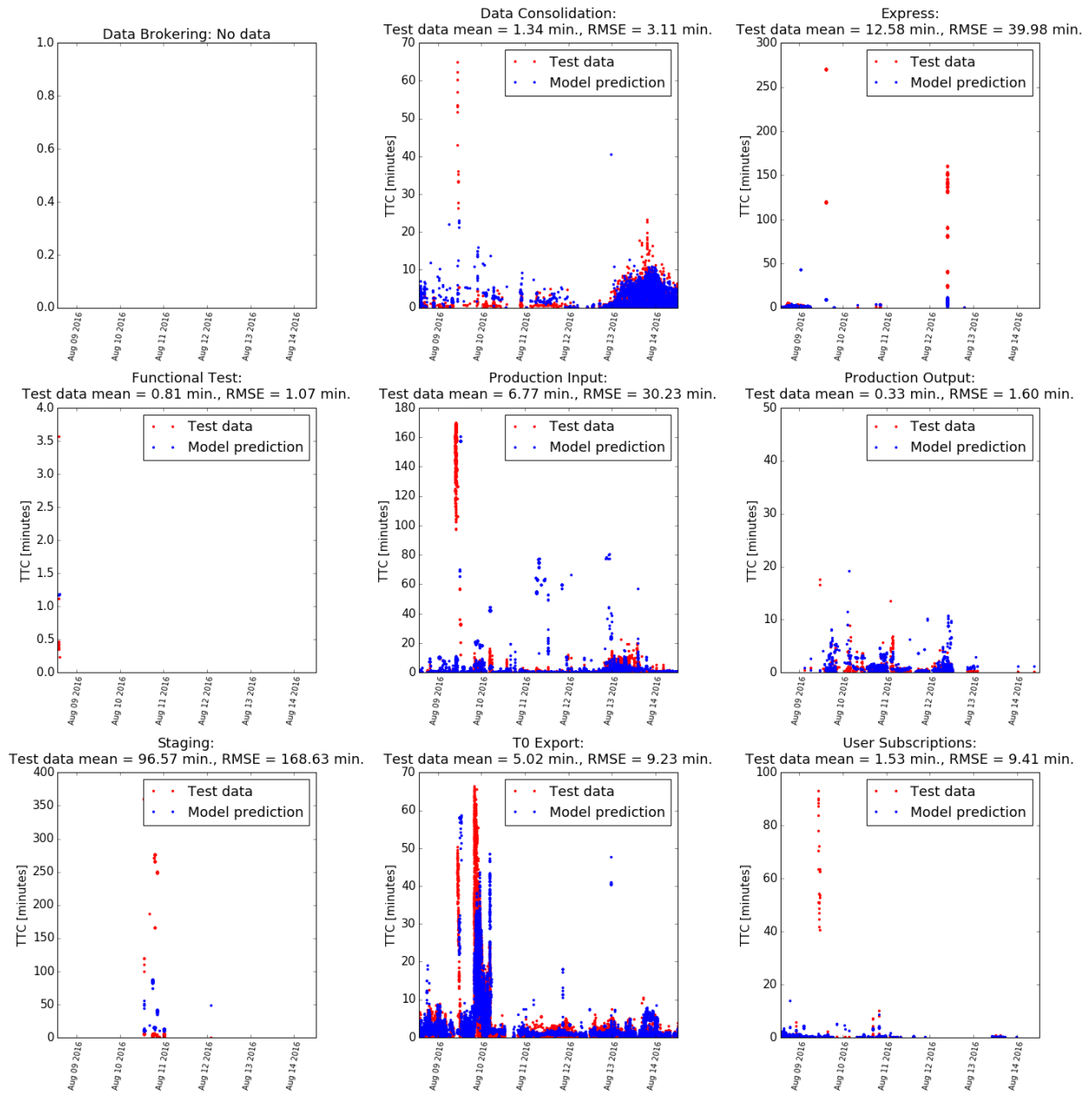


**Figure 2:** Data and model predictions for all activities. Mean and RMSE in subplot title.

averaged, producing a final prediction which is robust to outliers and noise, as shown by Breiman (2001). This project uses the Python library scikit-learn’s implementation of a random forest regressor, which produces continuous data as an output.

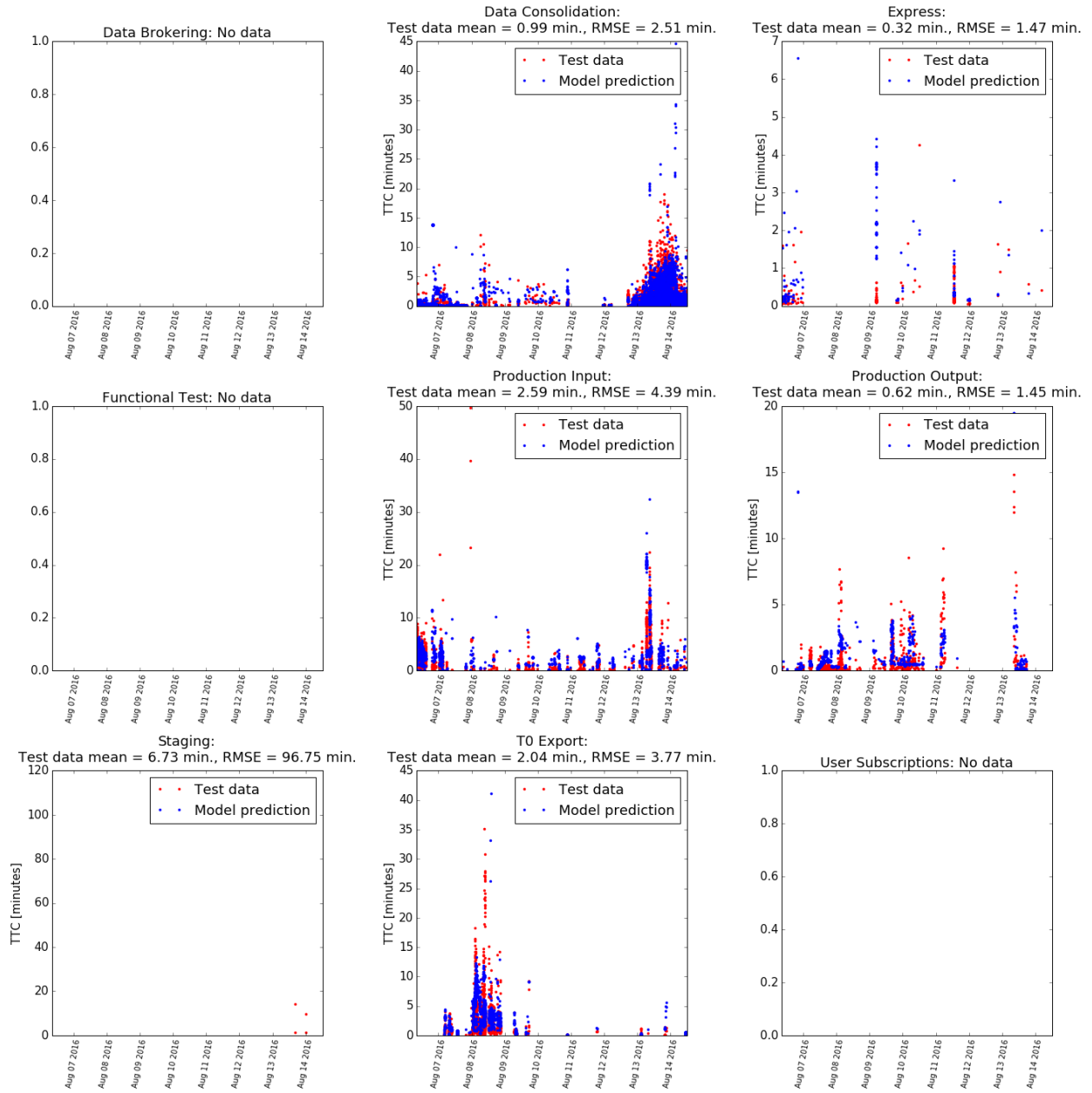
A training set is formed from the first 80% of the cleaned and sorted data, while the last 20% is reserved for validation. This choice of split is arbitrary: a 70/30 or 60/40 split could have been chosen (although preliminary results suggest that these choices cause an increase of outliers in predicted values). Fifty trees were chosen as the forest size; a truncated graph of one of the decision trees in the forest is displayed in Figure 1.

## CERN-PROD -> BNL-ATLAS: Model Performance by Activity

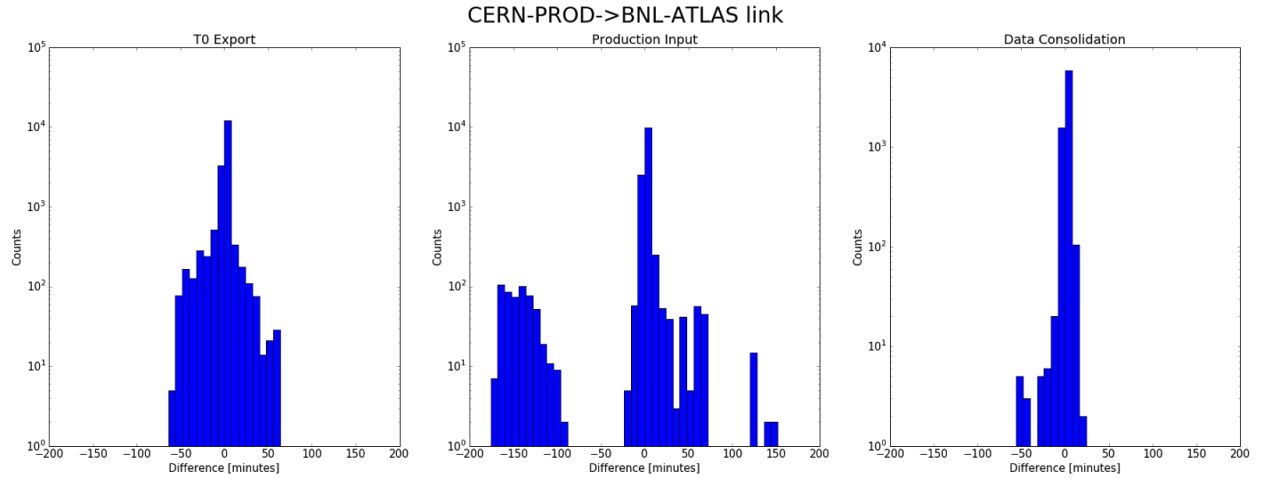


**Figure 3:** Per-activity comparison for CERN-PROD -> BNL-ATLAS. Mean and RMSE in subplot title.

## CERN-PROD -> SARA-MATRIX: Model Performance by Activity



**Figure 4:** Per-activity comparison for CERN-PROD -> SARA-MATRIX. Mean and RMSE in subplot title.



**Figure 5:** Histogram of absolute error (prediction - test data) for CERN-PROD -> BNL-ATLAS. Log y-axis to accentuate outliers.

|                           | CERN-PROD -> BNL-ATLAS |            | CERN-PROD -> SARA-MATRIX |            |
|---------------------------|------------------------|------------|--------------------------|------------|
|                           | Mean [min]             | RMSE [min] | Mean [min]               | RMSE [min] |
| <b>Data Consolidation</b> | 1.34                   | 3.11       | 0.99                     | 2.51       |
| <b>Express</b>            | 12.58                  | 39.98      | 0.32                     | 1.47       |
| <b>Functional Test</b>    | 0.81                   | 1.07       | N/A                      | N/A        |
| <b>Production Input</b>   | 6.77                   | 30.23      | 2.59                     | 4.39       |
| <b>Production Output</b>  | 0.33                   | 1.60       | 0.62                     | 1.45       |
| <b>Staging</b>            | 96.57                  | 168.68     | 6.73                     | 96.75      |
| <b>T0 Export</b>          | 5.02                   | 9.23       | 2.04                     | 3.77       |
| <b>User Subscriptions</b> | 1.53                   | 9.41       | N/A                      | N/A        |

**Table 1:** Summary of results for both links, presenting the mean value of the test data and the RMSE between predicted and test data, per activity and link.

## Results and Discussion

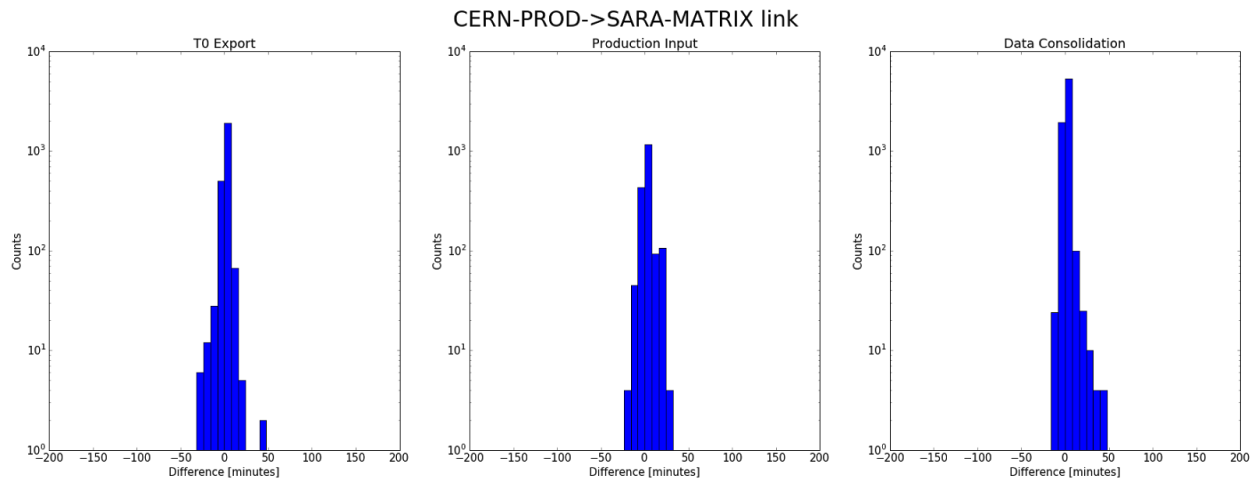
The root-mean-squared error (RMSE) is used to compare the model's predictions to the test dataset. Results are shown for two separate links, CERN-PROD -> BNL-ATLAS and CERN-PROD -> SARA-MATRIX, in Figures 2-4, and the RMSE values are summarized in Table 1.

The intent of this project is to give a "ballpark" estimate for the time-to-complete of a file. As such, the regression model developed for this project performs well for the two links considered. Because the regression model performs poorly on activities with low file counts, those activities often yield a considerably higher RMSE, e.g., staging.

There are significant deviations at times. For example, the central plot in Figure 3, corresponding to the Production Input activity, displays high TTC for some data that is not captured by the model. This is especially evident in a histogram of the error difference (Figure 5), visible as a peak centered at approximately -140 minutes in the tail of the Production Input activity.

Because of the exploratory nature of this project, it is impossible to determine whether all significant variables have been considered in the model. Indeed, since the spikes in the TTC data are not always captured by the model, and the model occasionally predicts spikes that do not occur, there is evidence that an important component is missing.

There are two potential improvements to this model that come to mind. First, it is known that the transfer of data traveling in one direction along a link can affect the queue time of data flowing in the opposite direction, because of the read-write capabilities of the storage



**Figure 6:** Histogram of absolute error (prediction - test data) for CERN-PROD -> SARA-MATRIX.

systems on either side. There is currently no measure for this influence, so it must be accepted as bias, but if in the future such a measure is implemented, it may be advantageous to add it to the model. Second, the current implementation of the regression model assumes that there is only one file being submitted at a given time. Knowledge of the number and size of files submitted simultaneously, or within a small time window, may improve the predictability of this model, especially for large batch submissions.

*This work was supported by the U.S. National Science Foundation under Grant No. 1358071.*

## References

Breiman, L. Machine Learning (2001) 45: 5.

## Appendix

The code for the time-to-complete prediction model is accessible via the following link:

[http://uct3-lx2.mwt2.org:9999/notebooks/Wesley/CERN-BNL\\_LinkRegressionForMeeting.ipynb](http://uct3-lx2.mwt2.org:9999/notebooks/Wesley/CERN-BNL_LinkRegressionForMeeting.ipynb)