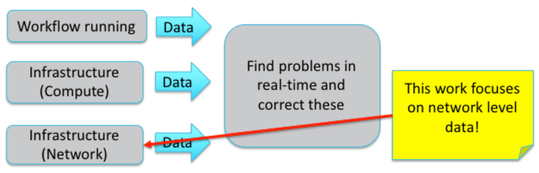
Detecting Anomalous Transfers in Workflows using Unsupervised Feature Extraction: PCA, Autoencoder and Isolation Forest

*Mariam Kiran, † Cong Wang, † Anirban Mandal

*Lawrence Berkeley National Lab,
† RENCI, University of North Carolina Chapel Hill

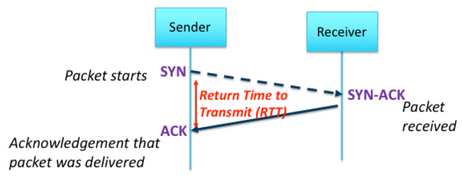
Panorama 360

The Panorama project aims to further understand the behavior of scientific workflows as they execute in heterogeneous environments. The project aims to develop a repository and associated capabilities for data collection, ingestion, and analysis for a broad class of DOE applications that span experimental and simulation science workflows.

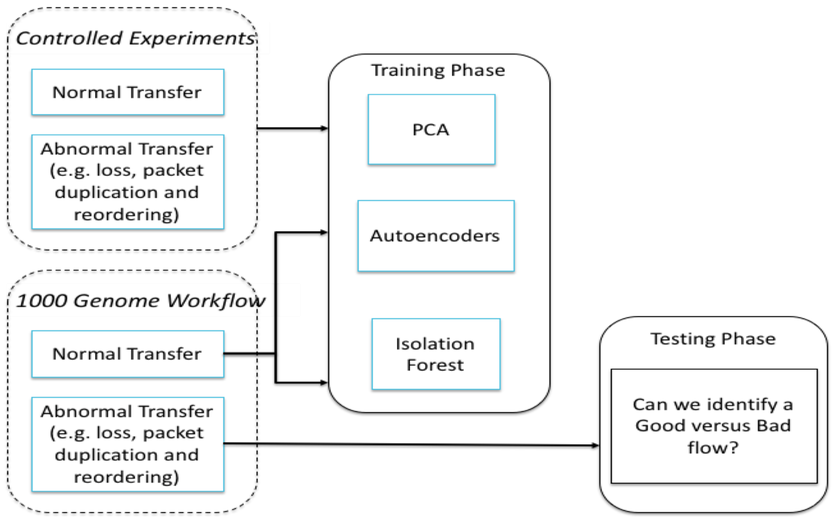


Network engineers record data about packets sent during File Transfers such as TCP statistics (Tstat files), Perfsonar Logs, SNMP data set and more. This work focuses on Just TSTAT data.
The aim of this experiment is that just by looking at Tstat data, can be find one of the following anomalies:

- Packet loss, Packet duplication or Synthetic reordering



Proposed Methodology

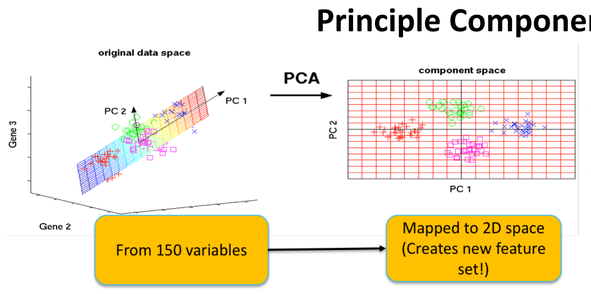


We perform anomaly detection by unsupervised feature extraction on a set of TCP traces collected from simple transfers between two nodes (Controlled Experiment) and a real workflow (1000 Genome Workflow).
We executed real file transfers using the iperf tool between two nodes set up on the Exogeni Testbed. The traces generated are used to inform the feature extraction algorithms of common features that can be witnessed in a reliable transfer and transfers with anomalies. Once trained, the feature extraction models are retrained with real workflow traces for two purposes: compare results of real versus controlled experiments and learn common features of normal transfers given what ever the infrastructure set up is.

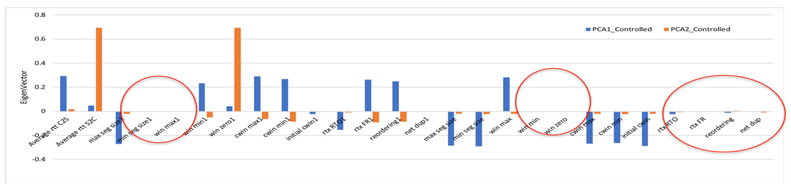
- Develop transfer learning for studying real workflow behavior, particularly for file transfer anomaly detection.
- Develop classification techniques for anomaly detection using various unsupervised feature extraction methods (e.g. PCA, Isolation Forest and Autoencoders).
- Experiments on real testbed (i.e. Exogeni) in controlled and real workflow network transfers. Each experiment was run for few hours with artificially introducing packet loss, duplication and reordering.

Unsupervised Feature Extraction Methods

Principle Component Analysis

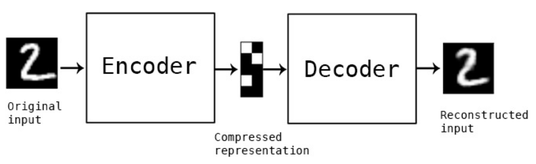


- Map any number of dimensions into a 2D space.
- Also reduces the exploration space by identifying the most influencing variables.

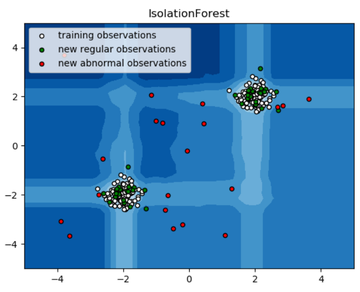


Autoencoders

- Learns to compress data from the input layer into a short code, and then uncompress that code into something that closely matches the original data.
- Forces dimensionality reduction, learning how to ignore noise.
- Use sparse autoencoder layers for image recognition.



Isolation Forest



- 'Isolates' observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature.
- Since recursive partitioning can be represented by a tree structure, the number of splittings required to isolate a sample is equivalent to the path length from the root node to the terminating node.
- This path length, averaged over a forest of such random trees, is a measure of normality and our decision function.
- Random partitioning produces noticeably shorter paths for anomalies.
- Hence, when a forest of random trees collectively produce shorter path lengths for particular samples, they are highly likely to be anomalies.

Training Data

Controlled Experiments

- Using a controlled network testbed called ExoGENI, to evaluate the effectiveness of our anomaly detection mechanisms. ExoGENI is a NSF funded IaaS cloud testbed that orchestrates a federation of independent cloud sites located across the US and circuit providers.
- Our topology consists of two VMs, sender and receiver. Each node/VM contains 2 cores, and 12GB RAM. The two nodes are connected via a 500 Mbps link.
- Each experiment lasts 1 hour, where the sender sends TCP traffic to the receiver using iPerf3.
- Generate synthetic network anomalies for different runs using the native Linux Traffic Control (TC) tool, with anomalies outlined in Table 3. We use TSTAT to collect the network measurement data, e.g., RTT, congestion window size, packet count, etc, on the data plane interfaces.

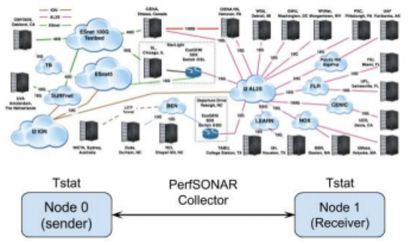
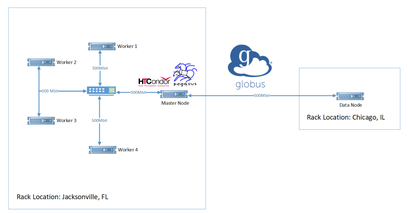


Figure 4: ExoGENI Experimental setup.



1000 Genome Workflow

- Using Pegasus we run the 1000 Genome workflow across 5 nodes.
- Located in Jacksonville and Chicago on different racks.
- Used the TC tool to inject same kind of anomalies and collect Tstat data.

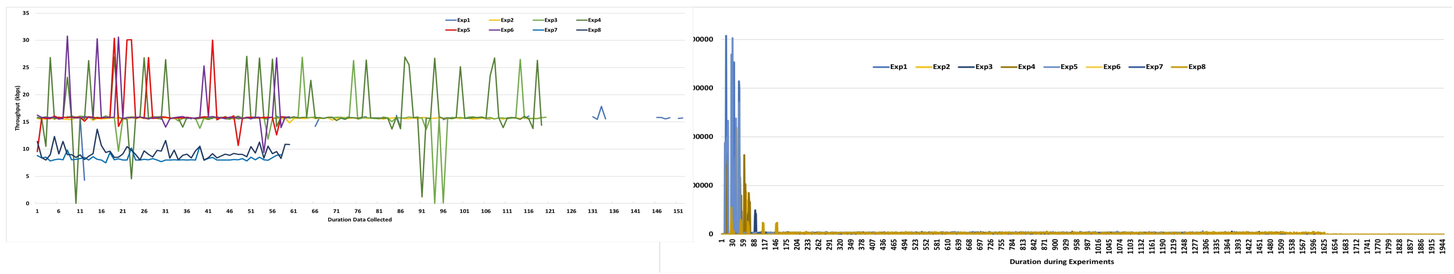
Table I: Controlled Experiment Details

No.	Experiment type (runs for 1 hour)	Tstat Sample
1	Random Traffic (iperf sends traffic at random)	152
2	No flow	120
3	Loss introduced 1%	120
4	Loss introduced 5%	120
5	Duplicate packets artificially 1%	60
6	Duplicate packets artificially 5%	60
7	Reordering packets artificially 25% - 50%	50
8	Reordering packets artificially 50% - 50%	60

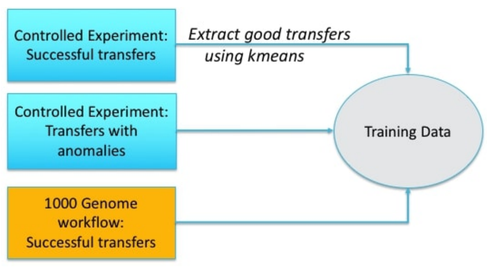
Table II: 1000 Genome Workflow Experiment Details

No.	Experiment type (runs for 1 hour)	Tstat Sample
1	Random Traffic (Workflow runs)	1475
2	Loss introduced 1%	1563
3	Loss introduced 2%	1636
4	Duplicate packets artificially 1%	1574
5	Duplicate packets artificially 5%	1527
6	Reordering packets artificially 25% - 50%	1491
7	Reordering packets artificially 50% - 50%	1592
8	Loss introduced 3%	1948

Results

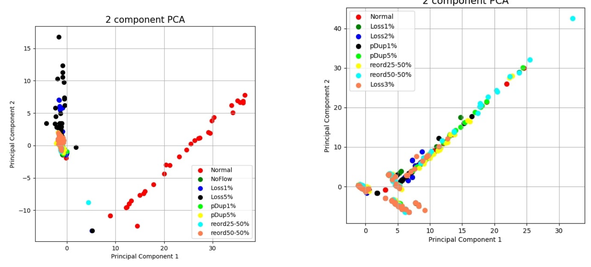


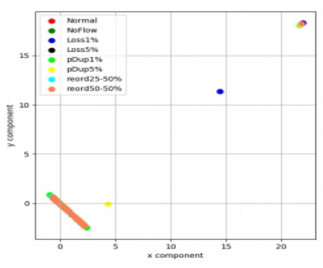
The throughput across both experiments show that they were very different in nature.



- The training data was prepared by removing any of the bad iperf transfers using k means method. K means was able to create two distinct clusters of good and bad transfers.

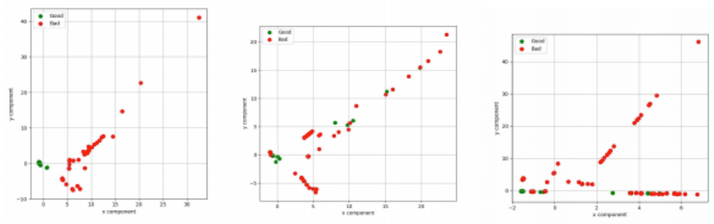
- PCA was run across both experiments:
  - ➢ Was able to distinguish normal transfers well in the controlled experiment
  - ➢ Did not perform well in the 1000 Genome workflow
  - ➢ Main different was that the RTT was higher in workflow experiment.



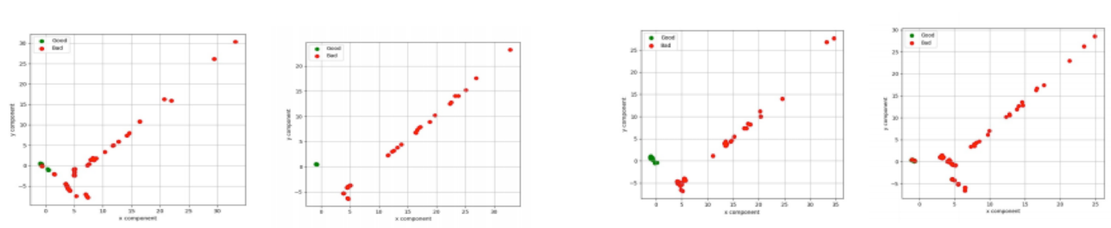


- Training the autoencoder:
  - ➢ Was not able to distinguish any good transfers in the real workflow experiment.
  - ➢ Accordingly to some literature, autoencoders do not perform well on statistical data sets and are most suited for images.

- Training the isolation forest:
  - ➢ Was able to distinguish good transfers in the real workflow experiment.
  - ➢ In experiments with anomalies it did count some as normal, but the error rate was between 0.2-0.4.
  - ➢ Further experiments are needed to see if it can actually identify the anomalies themselves.



(a) Packet Loss 1%. (b) Packet loss 2%. (c) Packet Loss 3%.



(d) Packet Duplication 1%. (e) Packet Duplication 5%. (f) Packet reordering 25%-50%. (g) Packet reordering 50%-50%.

Conclusions

- Why do we see clear clusters in controlled Exogeni experiment and not in Pegasus workflow?
  - ➢ Comparing PCA performance in experiment 1 and 2, the main variables that effect the eigenvectors are (Average RTT C2S, win min, win zero)
  - ➢ Average RTT is higher in the Pegasus workflow, ACK is slower
  - ➢ This is because the master and data node are in Jacksonville and Chicago respectively, whereas in experiment 1 there are on the same rack
- What is the difference between 'loss' and 'reordering'?
  - ➢ Win min and Win zero are higher in 'loss'
- Our results have shown that the way infrastructures are set up can vary the performance of the results

- **Isolation Forest is giving us better results in detecting anomalies.**
- We will be extending this work to actually work out what kind of anomalies and the cause and effect of them on the workflow performance.

http://sites.google.com/site/panoramaofworkflows

Panorama was funded by the US Department of Energy under Grant #DE-SC0012636