ML Approach to Log Anomalies

**What do we know?**

* Log data consists of words, phrases, and data corresponding to what is happening “under the hood”

**What did we find out deeper about log data?**

* Log data not only consists of these different words, phrases, data, but in the bigger picture, log data consists of patterns, activities, and operations.
* Using these different patterns to train an unsupervised model not only gives us a pretty accurate basis to use for detecting log anomalies, it gives the user more knowledge to figure out what is and isn’t normal just by appearance and gives the machine something to follow.
* Log parsing is not capable of consistently capturing the meaning of one single log message or the semantic meaning of a log message sequence

**What datasets can we use?**

* Datasets
  + Seems to have two popular types of log sources: distributed systems and supercomputers
  + Popular Distributed Systems generated datasets are…
    - Hadoop Distributed File System (HDFS)
    - Spark
    - ZooKeeper
    - OpenStack
  + Popular Super Computer generated datasets are…
    - Blue Gene/L super computer (BGL)
    - High Performance Cluster (HPC)
    - Thunderbird

## What approach do we plan on going with?

* There are viable supervised and unsupervised solutions, as well as several different datasets of log messages that we could use
* Neural Nets are viable, but require more work in optimizing runtime efficiency as well as accuracy, whereas other solutions are already optimized in their efficiency. Given that these systems are not as interpretable and not yet polished in the research world, we should avoid using these types of solutions
* Currently, clustering seems like an excellent approach to detecting anomalies in log data. We can use these different words, phrases, and patterns to train a model on without labeling data, while also being pretty accurate to what the log data should be and what patterns it should follow.
* Another approach will be using different kinds of trees to detect patterns in the data

## Potential Algorithms

* Shrunken Centroid Clustering: Utilized centroids in order to bring specific data points closer to centroids thus making it easier to identify outliers
* NeuralLog: uses subword tokenization + vectoring and then using transformer based classification to organize vectors relative to themselves
* Isolation Forest + 2 autoencoders
* LDA clustering

**Datasets we found**

* BGL Dataset: <https://github.com/logpai/loghub/tree/master/BGL>
* HDFS: https://hadoop.apache.org/docs/r1.2.1/hdfs\_design.html

Unsupervised clusters:

<https://www.zebrium.com/blog/using-machine-learning-to-detect-anomalies-in-logs>

Example programs that already do this: <https://neptune.ai/blog/machine-learning-approach-to-log-analytics>

Supervised using decision trees:

https://www.sciencedirect.com/science/article/pii/S2405959520300643