Machine Learning for Data Transfer Anomaly Detection

Sarah Cooper, Masud Bhuiyan, Engin Arslan
Department of Computer Science & Engineering
University of Nevada Reno
{sarahcooper,masud.hasan}@nevada.unr.edu, earslan@unr.edu

***Abstract*—Data transfer performance is critical for many science applications that rely on remote clusters to process the data. Despite the presence of high-speed research networks with up to 100 Gbps speeds, most data transfers obtain only a fraction of network bandwidth due to a variety of reasons. This project aims to pinpoint the underlying causes for performance anomalies by collecting and processing real-time performance metrics from file systems, data transfer nodes, and networks such that proper actions can be taken to mitigate the issues timely. As veracity and velocity of performance statistics is beyond what human operators can handle, we trained a Neural Network (NN) model to analyze the data in real-time and make high-accuracy predictions. The results indicate that NN can find the correct anomaly type with 93% accuracy.**

I. Introduction

Despite the continuous efforts to upgrade the networking infrastructure of research and education institutions to meet the large-scale data analytics needs and foster collaboration between scientists, the data transfers on these networks often perform very poorly, especially in the wide-area. The reason for the poor performance may not always be the lack of network bandwidth, but it may be due to other factors such as misconfigured servers, storage and I/O bandwidth limitations, network congestion, overloaded end systems, or other anomalies [1], [2].

Figure 1 shows the throughput of a 7GB file transfers between the Comet and Stampede clusters when repeated for four days. The observed throughput varied drastically between 150 Mbps and 3000 Mbps. While it is possible to correlate throughput fluctuation of memory-to-memory transfers to background traffic, doing so is much more complicated for disk-to-disk transfers, because it also depends on other factors such as storage and DTN load. As many science projects require predictable and stable network performance to streamline computation and communications tasks, it is critical to identify the root causes of performance bottlenecks such that proper actions can be taken to mitigate them to utilize high-cost research networks effectively.

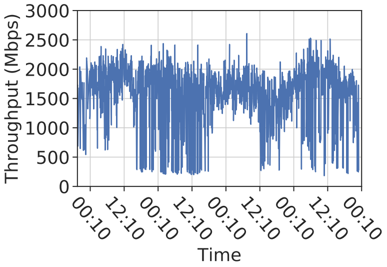


Figure 1. The performance of file transfers fluctuate significantly in high-speed networks.

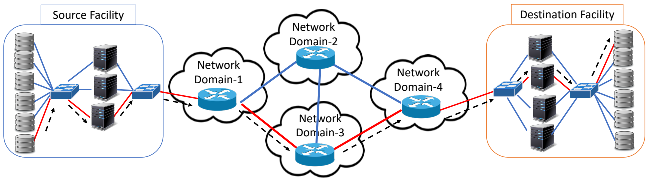


Figure 2. End-to-end transfer performance is determined by many system components.

A typical setup for data transfers in research networks is illustrated in Figure 3. Large-scale scientific data is generally stored in parallel file systems, causing transfers to interact with many storage servers across the cluster to read/write files and access/update metadata information. Moreover, high-performance computing clusters deploy several data transfer nodes (DTNs) to handle large numbers of concurrent transfers. Finally, transfers typically travel through many network domains before reaching their destination. Monitoring and troubleshooting the performance of high-speed data transfers that are reading/writing from/to many storage servers, using multiple DTNs, and passing through many network domains is challenging. The shared nature of these systems exacerbates the complexity and makes troubleshooting even harder.

For high-level network monitoring, PerfSonar [3] runs active measurements between research and education institutions to detect major network events such as node and link failures. It collects throughput, delay, packet re-transmission, and packet route information using measurement tools such as *traceroute* and *Iperf*. While it can provide useful information about network status, it can only detect persistent issues as its measurements are not continuous to avoid affecting actual science flows. Moreover, since it only collect few high-level performance metrics, it is unable to pinpoint the underlying reasons of poor transfer performance. Therefore, this project proposes online, comprehensive, and fine-grained solution to the problem of end-to-end scientific flow monitoring problem. To achieve this goal, we collect the set of key performance metrics that can capture performance of end-to-end transfers under various, commonly-observed circumstances.

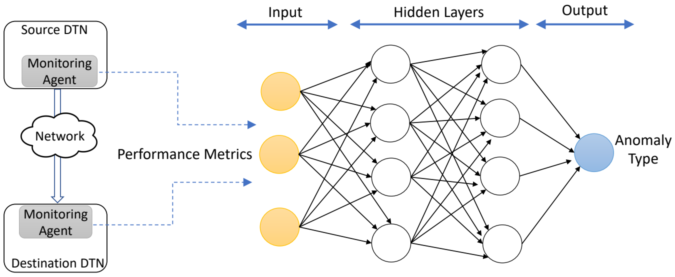


Figure 3. Proposed solution to collect performance metrics in real-time to detect performance anomalies.

II. Proposed Solution

We developed an anomaly detection framework to pinpoint the root causes of performance degradation as shown in Figure 3. Unlike earlier anomaly detection studies that categorize network traffic as *normal* or *abnormal*, we take a step further and also find the underlying reasons for abnormalities, such as congestion in a storage server or a faulty network interface card using the real-time performance metrics.

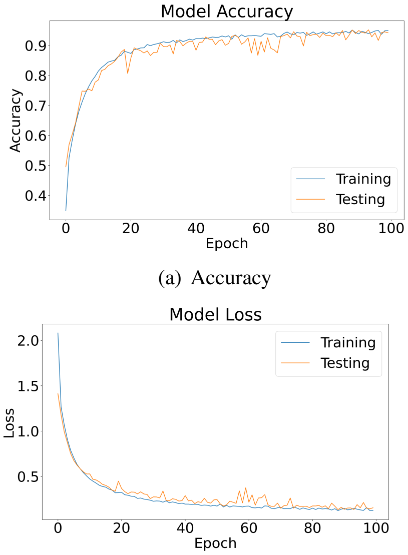
To train a model, we reproduced 10 common traffic anomalies with different severity levels in Chameleon Cloud network, where we created one instance in Chicago, IL and another instance in Austin, TX. The network bandwidth is 10Gbps and delay is 32ms. We collected performance metrics for file system, data transfer nodes, and TCP performance. Memory overload, disk congestion, CPU contention, packet loss, packet delay, packet duplicate, packet corrupt, packet reordering are among the anomaly types that we reproduces using the linux tools such as `tc` and `stress`. We then used `ss`, `sar`, and `iostat` utilities to collect real-time performance metrics in 100ms intervals.

Neural Network (NN) mimics human brain to detect patterns in datasets and is proved to be very effective in many areas. Therefore, we trained a NN model to process performance statistics and extract patterns between the given input metrics and anomaly types. We used a Rectified Linear Unit (ReLU) activation sequence for the hidden layers and a SoftMax activation sequence for the output layer. Activation sequences are used to signify if a neuron should be included in the next layer or not. In addition, we used the Adam optimizer as it is useful for large data sets and is optimized for large parameter sets. We utilized Tensorflow and Keras to create build and train NN and matplotlib to provide an accuracy and loss plot. Each run used 80% the total number of data points as the training size and 20% as the testing size with an epoch of 100.

III. Preliminary Results

The neural network ran for 100 epochs with a batch size of 270. Figure 4 shows the accuracy and loss rate of the model during the training phase. It shows that the model can achieve 93% accuracy rate and 0.3% loss rate after 100 epochs.

While the model accuracy is limited to 93% when it is tested with 30 different labels (10 anomalies with 3 severity level for each anomaly), its performance increases to 96% when the three different severity levels are grouped into same category as shown in Fig 5. For example, we reproduced three different packet loss anomalies by injecting 0.1%, 0.5%, and 1% losses and model performance improved slightly when these three anomaly types are marked with the same "packet loss" label. Similarly, the model performance improved to 99.96% when only four labels are used as *normal, server, network, and file system*. This high level categorization can be used to develop an hierarchical model where higher level predictions can be used to route the performance issues to the right the group in large organizations to facilitate ticket categorization.



(a) Accuracy

(b) Loss Rate

Figure 4. Accuracy and loss rate of Neural Network model during the training phase.

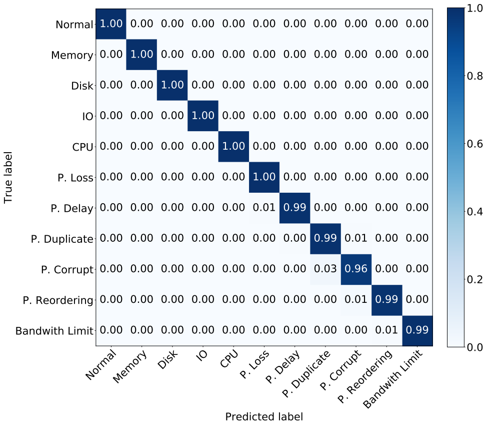


Figure 5. Confusion matrix of the neural network model. It is able to predict the correct label with 96% accuracy.

IV. Conclusion and Future Work

We proposed a neural network model to pinpoint the underlying reason for performance anomalies in data transfers by collecting and processing the performance metrics in real-time. The model can predict the correct anomaly type with 93% accuracy among 30 anomaly types. As a future work, we intend to apply other machine learning models to evaluate their performance for this problem. As integrity verification is critical yet costly operations for file transfers [4], [5], we also intend to extend the model to detect anomalies caused by the integrity verification process such as I/O contention or CPU overloading.

References

[1] E. Arslan, B. A. Pehlivan, and T. Kosar, “Big data transfer optimization through adaptive parameter tuning,” *Journal of Parallel and Distributed Computing*, vol. 120, pp. 89–100, 2018.

[2] E. Arslan and T. Kosar, “High-speed transfer optimization based on historical analysis and real-time tuning,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 6, pp. 1303–1316, 2018.

[3] “PerfSonar,” 2018, https://www.perfsonar.net.

[4] B. Charyyev, A. Alhussen, H. Sapkota, E. Pouyoul, M. H. Gunes, and E. Arslan, “Towards securing data transfers against silent data corruption,” in *IEEE/ACM International Symposium in Cluster, Cloud, and Grid Computing*, 2019.

[5] B. Charyyev and E. Arslan, “RIVA: Robust integrity verification algorithm for high-speed file transfers,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 6, pp. 1387–1399, 2020.