Conditional Recurrent Neural Networks for Enhancing Throughput Prediction and Slow File Transfers Detection in Large Science Workflows

Boyu Fan[1], Alex Sim[2], Kesheng Wu[3], Jinoh Kim[4]

[1] University of California, Berkeley, david_fan1@berkeley.edu
[2] Lawrence Berkeley National Laboratory, asim@lbl.gov
[3] Energy Sciences Network, johnwu@es.net
[4] Texas A&M University-Commerce, Jinoh.Kim@tamuc.edu

***Abstract*—Efficient data transfer across scientific computing facilities is critical for enabling timely scientific discoveries. In this work, we explore the options of anticipating extremely slow data transfers to enable preventive actions. However, the dynamic nature of the large distributed scientific workflows driving these data transfers presents significant challenges for predicting network throughput. This study introduces a Conditional Recurrent Neural Network (CondRNN) model, specifically utilizing Conditional Long Short-Term Memory (CondLSTM), to integrate both static and dynamic features for enhanced throughput prediction. By leveraging historical transfers as proxy features, more than 60% of predictions achieved an absolute percentage error (APE) of less than 20%, and slow transfers were detected with a precision of 91.7% and recall of 100%, outperforming traditional RNN models. Implementing CondLSTM in scientific computing environments can optimize network resource utilization, ensuring efficient data transmission, thereby supporting the continuous progression of scientific research.**

***Index Terms*—Network Throughput Prediction, Conditional Recurrent Neural Network (CondRNN), Slow Transfer Detection, Time Series Analysis, TCP Connections**

I. Introduction

In the big data era, data transfer is one of the challenges with the increasing scale of data volumes [1]. In particular, scientific discoveries rely heavily on access to massive datasets generated by geographically dispersed scientific user facilities from experiments and observations [2]. Remote transfer is a key factor in data access cost, given the increasing demand for networking and data transfer in and out of scientific facilities [3], [4]. Many scientific applications, such as climate modeling [5], bioinformatics [6], and particle physics [7], exchange a large amount of data across computing facilities. Moreover, the integration of wireless-networking-enabled Internet of Things (IoT) sensors and artificial intelligence (AI) applications requires federated and distributed processing, and this introduces frequent, large-scale data transfers across scientific networking infrastructures [8], [9]. Since most of these users would not have the storage resource to replicate the entire dataset, they rely on dynamic mechanisms to retrieve the necessary data records [10]. To effectively support such remote data accesses, it is essential to ensure that appropriate networking and storage resources are allocated as needed for individual transfers.

While previous studies have made strides in network traffic monitoring and throughput prediction, gaps remain. Existing work on network traffic monitoring, such as identifying elephant flows [11], [12], primarily focuses on estimating traffic volumes without addressing the dynamic prediction of throughput under varying conditions. Throughput prediction research, utilizing models like Support Vector Regression (SVR) [13] and Autoregressive Integrated Moving Average (ARIMA) [14], has improved accuracy by incorporating static features such as file transfer history and path properties. However, these models often fail to capture the non-linear interactions between static and dynamic features in large-scale scientific networks.

These limitations are particularly pronounced in environments characterized by varying network conditions and the diverse nature of data transfer behaviors, such as those found in large-scale computational facilities like NERSC. Our initial study reveals that throughput does not inherently demonstrate predictable time series patterns, and thus, traditional models often struggle to capture the complex and dynamic characteristics of data flow across networks [15], [16].

To address these challenges, we investigated various properties of TCP connections using remote data transfer logs from a scientific computing facility and developed a prediction mechanism that estimates the performance of individual transfers from the past transfer information. Our exploratory data analysis reveals variations between two consecutive transfers from the same subnet. The autocorrelation for past transfers exhibits initial drops, followed by a steady and moderate correlation, suggesting that relying solely on the latest transfer record may not lead to optimal prediction quality. From this observation, we take a time series based prediction approach referring to a series of past observations to capture time-varying characteristics. Additionally, we find that some static features (e.g., IP subnet and transfer size) exhibit a notable correlation with measured throughput. This implies that referring to the same subnet history may be beneficial for improving

prediction quality [17].

Building on these explorations, we design our prediction mechanism based on a Conditional Recurrent Neural Network (CondRNN) [18], which integrates static features as conditional layers to enhance prediction performance. CondRNN architecture enables our model to adaptively learn from both historical transfer data and static network characteristics, capturing temporal patterns and non-linear interactions that traditional models overlook.

The rest of the paper is organized as follows: In Section II, we detail our exploratory data analysis (EDA), presenting key observations of the Tstat data features and their relationship with throughput. Section III introduces our proposed Conditional Recurrent Neural Network (CondRNN) model, explaining its architecture and the integration of static features as conditional parameters. In Section IV, we evaluate the performance of our model, compare it with other RNN architectures, and analyze the impact of conditional features. Section V focuses on the application of our model for detecting slow transfers and presents the evaluation metrics and results. Finally, Section VI concludes the paper by summarizing our findings and suggesting directions for future research.

II. Description of `tstat` Data and Exploratory Analysis

A. Data Description

Tstat data records network traffic flows, where each entry in the dataset is indicative of a distinct network flow. The dataset is obtained from the network traffic logs of NERSC's data transfer nodes, with the `start` and `end` columns providing timestamps for the beginning and end of each flow. Network performance metrics such as `tput` (throughput), `rtt` (average round trip time), and `retx` (number of retransmitted bytes) are also included. Additional columns, such as `to_country_label` and `to_ip_label`, provide geographical and IP-related information. At the prediction phase, the available data is limited to the IP addresses and the size of the file being transferred. This limitation poses a challenge in the predictive modeling of network performance, necessitating the creation of new feature sets that can be obtained before the predictive phase. In our study, we focus on analyzing the '`Inbound: S2c`' field, which measures the traffic coming into the network from external sources, known as server-to-client traffic.

B. Exploratory Data Analysis

Exploratory Data Analysis (EDA) forms the cornerstone of our approach. Previous research [19] has identified key features such as transfer size, round-trip time, and retransmission rate, which exhibit considerable correlations with throughput. To summarize, larger file transfers often achieve higher throughput, as substantial data movements utilize available bandwidth more effectively, though this relationship is not strictly linear due to potential network congestion or limitations in data source or destination capabilities. Additionally, a higher RTT generally correlates with lower throughput because longer acknowledgment delays can limit the amount of data 'in flight', thus reducing effective network utilization. The retransmission rate also plays a critical role. High retransmission rates have a negative impact on throughput. Our analysis confirms that throughput has an inversely proportional relationship with retransmission rates.

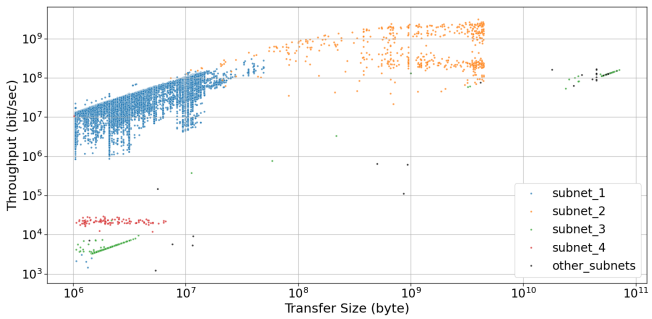

Fig. 1. Scatter Plot of Transfer Size vs. Throughput, showing distinct clusters based on different subsets of IP addresses known as IP subnets.

We further explored interactions between IP subnet, transfer size, and throughput. The IP subnet is identified by the first three digits of the IPv4 and IPv6 addresses, and we categorized the data based on these subnet groups. Figure 1 maps throughput against the transfer size, with the data color-coded by subnet group. It displays the varying throughput behaviors across different subnets and transfer sizes. For instance, data transfers from `subnet_3`, colored in green, exhibits two distinct clusters: one located on the lower left and the other in the upper right of the plot. By categorizing transfer sizes into bins, we can clearly delineate these groups, with each showing clear throughput bounds. This distinct separation suggests different network performance (throughput) within the same subnet based on the size of the transfer being transferred. The clarity of these divisions within Figure 1 encourages the inclusion of a conditional layer in our recurrent neural network architecture.

III. Proposed Method

In time series analysis, deep sequence methods have proven to be effective [20]; however, network performance data often contain static features that do not change over time. To effectively integrate these static attributes with dynamic data, we propose an architecture called the Conditional Recurrent Neural Network (CondRNN).

The key modification in CondRNN involves transforming static feature vectors (or conditions) through an independent neural network, typically using linear transformations. This transformation adjusts the dimensions of the static features to match those of the RNN's hidden layers. The restructured vector is then used to initialize the first hidden state of the RNN, embedding the static conditions directly into the network's memory, thus conditioning the prediction of future states ($p(y_t|x_0, \ldots, x_t, \textit{conditions})$ for $0 \leq t \leq T$) on both temporal and non-temporal inputs.

A. Definition of Features

In our network performance analysis, the static conditions considered include the IP subnet and the transfer size. As demonstrated in the EDA, these static features effectively constrain the range of throughput, which can be treated as conditional information that does not interfere with the temporal data at each time step. Additionally, we have included another condition indicating the presence of concurrent transfers. The feature `'concurrent'` was constructed as a binary variable, indicating whether more than one data transfer occurs simultaneously. This factor can affect network performance by potentially increasing congestion and reducing the available bandwidth per transfer, which could negatively affect throughput. Table I shows a summary of static features we embedded in the model.

Since round-trip time (RTT) and retransmission rates are only available after a data transfer is complete, we face the challenge of limited pre-transfer features, which restricts our information to transfer size and IP addresses. To overcome this, we propose using historical data as a proxy. We use metrics from the most recent transfer within the same IP subnet and the size bin we defined in Table I to generate proxy variables: `'prev_tput'`, `'prev_rtt'`, `'prev_retx'`, `'prev_size'`, as detailed in Table II. To address cases where these proxies are inaccurate, we introduce `'size_ratio'` as an indicator of the potential discrepancy between the proxy and the actual conditions of the current transfer.

B. Conditional Recurrent Approach

RNNs are ideal for learning sequential data due to the feedback loops, or connections within their layers, that provide an internal memory of past observations. In particular, RNNs with long short-term memory (LSTM) layers can handle longer sequences, where the sequential dependence of the data spans over a large number of observations [18]. This capability makes LSTMs especially well-suited for tasks involving long-term dependencies, such as throughptut prediction.

The Conditional Recurrent Neural Network (CondRNN) extends the standard RNN framework by incorporating static features as conditional parameters. The figure 2 illustrates an example of a model architecture for a CondLSTM, but the structure can be adapted similarly across other types of conditional RNNs by embedding contextual or static features directly into the initial states of these networks.

In the displayed architecture, the initial embedding layer handles categorical features of `'size_bin'`, `'ip_subnet'`, and `'concurrent'`. Each of these features is processed through embedding layers and weight matrices (W1, W2, W3), which transform these categorical inputs into dense vectors. These vectors are then flattened and concatenated, forming a unified feature vector that encapsulates the static context of the network's conditions. Lastly, this concatenated vector is subsequently reshaped and used to initialize the hidden state of the LSTM.

IV. Evaluation

A. Experimental Setting and Metrics

The data records we selected come from one of the Data Transfer Nodes (DTN01) at NERSC, encompassing over 20 million network streams. For our experimental timeline, we allocated data from January 1st to January 12th of 2022 for training and validation purposes. Subsequently, the data from January 13th to January 21st served as our evaluation set.

To assess performance, we focused on the Absolute Percentage Error (APE) and the Mean Absolute Percentage Error (MAPE) for our evaluations:

$$\text{APE} = \frac{|Actual_t - Predicted_t|}{Actual_t} \times 100\%$$

$$\text{MAPE} = \frac{1}{N}\sum_{t=1}^{N} \text{APE}_t.$$

These metrics are valuable given the broad range of throughput values, spanning from 10^3 to 10^8, providing a interpretable relative measure of the model's performance.

In addition, we have augmented our evaluation metrics with a novel measure called the Absolute Arctangent Percentage Error (AAPE) and its mean form, Mean AAPE (MAAPE), specifically designed for intermittent demand forecasts [22]. The AAPE for each observation t is defined as:

$$\text{AAPE}_t = \arctan\left(\left|\frac{Actual_t - Predicted_t}{Actual_t}\right|\right),$$

and the MAAPE is calculated as the average of these values:

$$\text{MAAPE} = \frac{1}{N}\sum_{t=1}^{N} \text{AAPE}_t.$$

MAAPE retains all the benefits of MAPE, being scale-independent and easily interpretable as an absolute percentage error. Its key advantage lies in the bounded nature of the arctangent function, which ranges from 0 to $\frac{\pi}{2}$. The bounded effect of the arctangent transformation renders MAAPE more robust than traditional MAPE by diminishing the impact of outliers. This robustness makes MAAPE particularly beneficial in environments like throughput prediction, where anomalies or drastic spikes in data can skew results. Implementing MAAPE in throughput prediction allows for a consistent evaluation of model performance, mitigating the influence of irregular, extreme data values.

B. Impact of Conditional Architecture

We utilized a series of machine learning models to predict throughput in network streams, employing both traditional and conditional architectures. The conditional models, which incorporated additional context about network conditions, consistently demonstrated enhanced performance over their non-conditional counterparts across various metrics.

The analysis involves six models: GRU, LSTM, Simple RNN, CondGRU, CondLSTM, and CondSimple. We assessed their performance based on the Absolute Percentage Error (APE) and Absolute Arctangent Percentage Error (AAPE) across five defined error ranges: 0%-20%, 20%-40%, 40%-60%, 60%-80%, and above 80%.

TABLE I
STATIC FEATURES AND DESCRIPTIONS

Feature	Description
`ip_subnet`	First three digits of IPv4 and IPv6 addresses, categorizing data by network subnet.
`size_bin`	Divides transfer sizes into small, medium, large groups using equal quantiles.
`concurrent`	Binary indicator (0 or 1) signifying overlap between the `start` of a transfer and the `end` of preceding ones.

TABLE II
DYNAMIC FEATURES AND DESCRIPTIONS

Feature	Description
`prev_tput`	Throughput from the most recent transfer within the same IP subnet and size bin.
`prev_rtt`	Average Round-Trip time from the most recent transfer within the same IP subnet and size bin.
`prev_retx`	Retransmission rate from the most recent transfer within the same IP subnet and size bin.
`prev_size`	Size of the most recent transfer within the same IP subnet and size bin prior to the current transfer.
`size_ratio`	Ratio of the size of the most recent previous transfer to the size of the current transfer within the same IP subnet and size bin.
`time_gap`	Interval in seconds between the start times of successive transfers

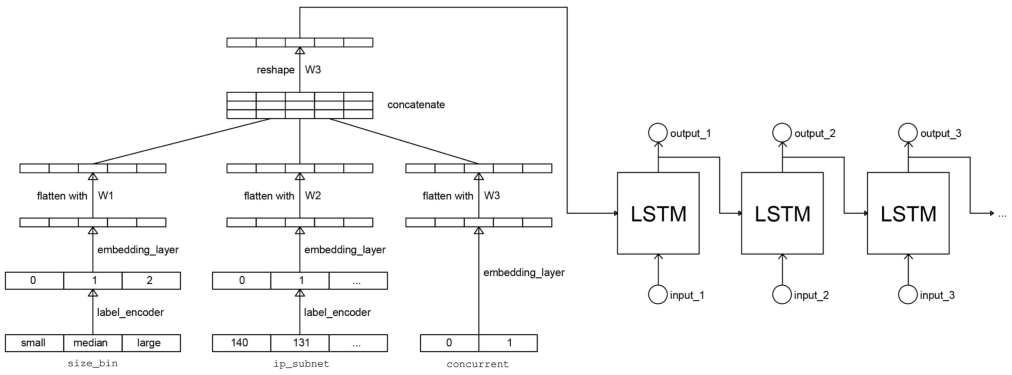

Fig. 2. Proposed model based on conditional recurrent network, inspired by Aumon and Adrien Andréas [21]

As shown in Table III, for lower error ranges (0%-20% and 20%-40%), conditional models generally recorded higher percentages of predictions within these favorable thresholds. For instance, CondLSTM captured 61.94% of its predictions within the 0%-20% error range for APE and 70.30% for AAPE. This performance outperformed to its non-conditional counterpart, LSTM, which registered 57.44% for APE and 62.16% for AAPE within the same range. This pattern was consistent across other models, where CondGRU and Cond-Simple also showed improvements in maintaining lower error percentages in these critical ranges. Remarkablely, in the highest error range (above 80%), the CondLSTM demonstrated significantly better control, with only 14.56% of its predictions falling into this category for APE and 0.12% for AAPE. This stark reduction compared to the LSTM, which saw 33.38% for APE and 24.77% for AAPE, demonstrates the enhanced capability of conditional models to avoid large errors.

The table IV summarizes the performance of six models with a focus on percentage metrics for evaluating their stability and precision in predictions. The CondLSTM model demonstrates outstanding performance with the lowest Mean Absolute Percentage Error (MAPE) of 35.95%. It also shows the smallest maximum APE and median APE, indicating its robustness against outliers and its reliability in typical scenarios. The low standard deviation in Relative Error for the CondLSTM (55.51%) compared to other models underscores its consistent performance across different network conditions.

C. Best Model Performance

The CondLSTM model emerged as the most effective in predicting network throughput, demonstrating superior accuracy and reliability across various metrics.

Shown in Figure 3, the line plot tracks throughput predictions over specific indices. Between prediction indices from 22600 to 22650, the plot reveals a significant discrepancy where the predicted throughput values drop sharply from around 10^8 to approximately 10^4, which illustrates the model's ability to detect and predict substantial deviations in throughput, specifically identifying periods of significantly slower network activity.

Figure 4 presents the cumulative Absolute Percentage Error (APE) of the data. The y-axis represents the cumulative percentage of the data, while the x-axis shows the ranked APE values. The orange vertical line indicates an APE threshold of 40%, which corresponds to about 72% of the data. This means that 72% of the predicted throughput values have an APE within 40%. The shape of the curve exhibits a rapid initial increase up to an APE of about 100%, followed by a gradual flattening. This pattern suggests that while a small number of predictions exhibit high errors (with a maximum APE reaching

TABLE III
PERFORMANCE COMPARISON OF MODELS BASED ON ERROR RANGES, APE, AND AAPE VALUES.

Model Name	Metric	Error Ranges: 0%-20%	20%-40%	40%-60%	60%-80%	Above 80%
GRU	Percentage of Predictions Within (% APE)	54.68	7.97	2.33	1.08	33.94
	Percentage of Predictions Within (% AAPE)	60.90	4.84	3.20	5.14	25.93
LSTM	Percentage of Predictions Within (% APE)	57.44	6.03	2.11	1.04	33.38
	Percentage of Predictions Within (% AAPE)	62.16	4.06	3.57	5.45	24.77
Simple	Percentage of Predictions Within (% APE)	51.27	10.42	2.74	1.19	34.37
	Percentage of Predictions Within (% AAPE)	59.24	6.03	3.14	5.25	26.34
cond_GRU	Percentage of Predictions Within (% APE)	56.06	6.68	2.63	1.42	33.22
	Percentage of Predictions Within (% AAPE)	61.15	5.13	4.23	7.30	22.19
cond_LSTM	Percentage of Predictions Within (% APE)	61.94	11.30	6.22	5.97	14.56
	Percentage of Predictions Within (% AAPE)	70.30	12.88	9.62	7.07	0.12
cond_Simple	Percentage of Predictions Within (% APE)	56.31	6.59	2.51	1.18	33.41
	Percentage of Predictions Within (% AAPE)	61.22	4.91	4.03	6.68	23.16

TABLE IV
QUANTITATIVE ASSESSMENT OF MODEL PERFORMANCES: APE (%)

Model	Max	Mean	Median	StdDev
GRU	628633.90	13643.85	14.94	41392.59
LSTM	126822.97	3124.94	12.60	8736.59
Simple	885657.73	27114.91	18.70	73765.52
cond_GRU	9573.59	385.58	13.63	857.37
cond_LSTM	**660.63**	**35.95**	**11.49**	**55.51**
cond_Simple	10866.23	561.13	13.26	1237.97

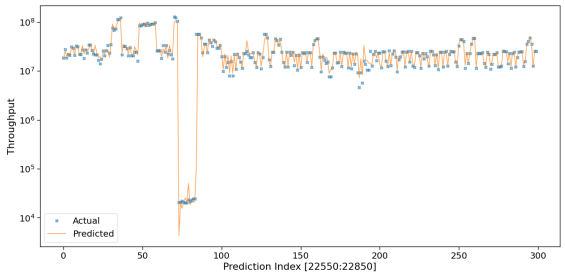

Fig. 3. Line Plot of Throughput Prediction from Indices 22550 to 22850, showcasing the model's adaptive capability in predicting slow transfers.

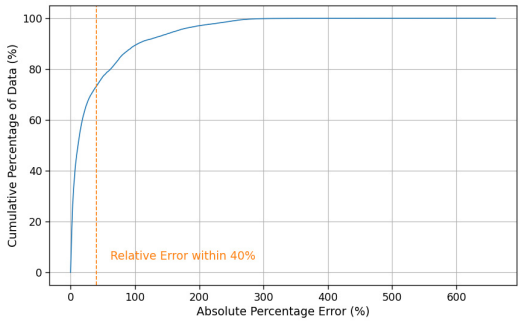

Fig. 4. Line Plot of Cumulative Distribution of APE, illustrating the proportion of predictions across different error rate ranges.

approximately 600%), the majority of the predictions are much more accurate, indicating that most of the predictions fall within an acceptable error range.

V. USE CASE: SLOW TRANSFER DETECTION

Detecting slow transfers in network data management systems is crucial to maintain efficient data transmission and optimize network resource utilization. Slow transfers can significantly impede the overall performance of a network, causing delays and increased latency, which in turn can affect the throughput of other ongoing transfers [23], [24]. Identifying these slow transfers promptly allows the system to make adjustments, such as rerouting data or allocating more bandwidth, thereby improving the effectiveness and reliability of the data infrastructure. This proactive approach in managing network performance ensures that service level agreements are met and operational costs are minimized.

However, the task of detecting slow transfers is challenging due to the severe class imbalance inherent in the dataset. Typically, the majority of transfers are normal, with slow transfers constituting only a small fraction of the total number of transfers. This imbalance makes it difficult for traditional binary classification models to accurately identify positive cases (slow transfers). Previous studies [19] have shown that when using a uniform random sample of the NERSC traffic logs, the models often exhibit low recall due to the scarcity of slow events.

A. Description of Slow Transfer Detection

For our experiment on slow transfer detection, we utilize the CondLSTM model, which has shown the best performance in prior tests, to predict and classify network transfers in real-time. The first step is to train a CondLSTM with the task of predicting throughput. Then, we apply it to the evaluation set to predict the throughput for each transfer. We define slow transfers as those with throughput less than 2×10^6 bps. The threshold is applied to both the actual throughput and the predicted throughput to classify each transfer in the evaluation set as either slow or normal.

After classifying throughputs into binary outcomes as either slow or normal, we evaluate the model's performance on this binary classification task using metrics including accuracy,

TABLE V
CONFUSION MATRIX

	Predicted Negative	Predicted Positive
True Negative	24893 (TN)	1 (FP)
True Positive	0 (FN)	11 (TP)

TABLE VI
EVALUATION METRICS

Precision	Recall	Accuracy	F1-Score
0.92	1.000	0.99	0.96

precision, recall, and the F1 score. Additionally, a confusion matrix is provided to summarize the number of correctly or incorrectly classified transfers.

B. Performance of Slow Transfer Detection

Shown in Table V, among a total of 24,905 data transfers, the model correctly classified 24,893 transfers as normal (True Negative) and 11 transfers as slow (True Positive), with only one instance of an incorrect classification as slow (False Positive) and no misclassified slow transfers (False Negative).

The evaluation metrics are presented in Table VI. The CondLSTM achieved an nearly perfect performance across all measures. These results confirm the effectiveness of our model in accurately detecting slow transfers, thereby optimizing network resource utilization and ensuring efficient data transmission in large-scale scientific computing environments.

VI. CONCLUSIONS

In conclusion, our study developed and evaluated a Conditional Recurrent Neural Network (CondRNN) model, particularly the Conditional LSTM (CondLSTM), to predict network throughput and detect slow transfers in large-scale scientific computing environments. By integrating static features, our model demonstrated superior performance compared to traditional RNNs, achieving lower error rates and higher predictive accuracy. The evaluation metrics, including APE and AAPE, confirmed the model's robustness in handling variable and skewed network throughput data. Furthermore, the CondLSTM model showed exceptional accuracy in detecting slow transfers, which is crucial for optimizing network performance and resource utilization. This work paves the way for further research into integrating additional contextual information and exploring other neural network architectures to continue improving network performance in scientific computing environments. Future work could involve the exploration of real-time implementation of the model and its application to different types of network traffic data, improving its adaptability and generalizability across diverse computational settings.

ACKNOWLEDGMENT

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, of the US Department of Energy under Contract No. DE-AC02-05CH11231, and also used resources of the National Energy Research Scientific Computing Center (NERSC).

REFERENCES

[1] A. L'heureux, K. Grolinger, H. F. Elyamany, and M. A. Capretz, "Machine learning with big data: Challenges and approaches," *Ieee Access*, vol. 5, pp. 7776–7797, 2017.

[2] E. W. Bethel, M. Greenwald, V. Dam, *et al.*, "Management, analysis, and visualization of experimental and observational data—the convergence of data and computing," in *2016 IEEE 12th International Conference on e-Science (e-Science)*, pp. 213–222, IEEE, 2016.

[3] R. Gerber, J. Hack, K. Riley, Antypas, *et al.*, "Crosscut report: Exascale requirements reviews," 1 2018.

[4] Y. Liu, Z. Liu, Kettimuthu, *et al.*, "Data transfer between scientific facilities–bottleneck analysis, insights and optimizations," in *2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pp. 122–131, IEEE, 2019.

[5] Z. Li, Q. Huang, Y. Jiang, and F. Hu, "Sovas: a scalable online visual analytic system for big climate data analysis," *International Journal of Geographical Information Science*, vol. 34, no. 6, pp. 1188–1209, 2020.

[6] T. Reiter, P. T. Brooks, L. Irber, S. E. Joslin, C. M. Reid, C. Scott, C. T. Brown, and N. T. Pierce-Ward, "Streamlining data-intensive biology with workflow systems," *GigaScience*, vol. 10, no. 1, p. giaa140, 2021.

[7] A. Alekseev, A. Kiryanov, A. Klimentov, T. Korchuganova, Mitsyn, *et al.*, "Scientific data lake for high luminosity lhc project and other data-intensive particle and astro-particle physics experiments," in *Journal of Physics: Conference Series*, vol. 1690, p. 012166, 2020.

[8] J. Ang, A. A. Chien, Hammond, *et al.*, "Reimagining codesign for advanced scientific computing: Report for the ascr workshop on reimagining codesign," 4 2022.

[9] K. Fagnan, Y. Nashed, G. Perdue, D. Ratner, A. Shankar, and S. Yoo, "Data and models: A framework for advancing ai in science," 12 2019.

[10] B. Enders, D. Bard, C. Snavely, L. Gerhardt, J. Lee, Totzke, *et al.*, "Cross-facility science with the superfacility project at lbnl," in *2020 IEEE/ACM 2nd Workshop*, pp. 1–7, 2020.

[11] R. B. Basat, G. Einziger, R. Friedman, and Y. Kassner, "Optimal elephant flow detection," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pp. 1–9, IEEE, 2017.

[12] A. Chhabra and M. Kiran, "Classifying elephant and mice flows in high-speed scientific networks," *Proc. INDIS*, pp. 1–8, 2017.

[13] M. Mirza, J. Sommers, P. Barford, and X. Zhu, "A machine learning approach to tcp throughput prediction," *IEEE/ACM Transactions on Networking*, vol. 18, no. 4, pp. 1026–1039, 2010.

[14] D. Lee, D. Lee, M. Choi, and J. Lee, "Prediction of network throughput using arima," in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 1–5, 2020.

[15] A. Lazar, K. Wu, and A. Sim, "Predicting network traffic using tcp anomalies," in *2018 IEEE International Conference on Big Data (Big Data)*, pp. 5369–5371, 2018.

[16] R. Shao, A. Sim, K. Wu, and J. Kim, "Leveraging history to predict infrequent abnormal transfers in distributed workflows," *Sensors*, vol. 23, no. 12, 2023.

[17] R. Shao, J. Kim, A. Sim, and K. Wu, "Predicting slow network transfers in scientific computing," in *Fifth International Workshop on Systems and Network Telemetry and Analytics*, pp. 13–20, 2022.

[18] S. Lauria and M. F. Saleh, "Conditional recurrent neural networks for broad applications in nonlinear optics," *Optics Express*, vol. 32, p. 5582, Feb. 2024.

[19] R. Shao, A. Sim, K. Wu, and J. Kim, "Leveraging history to predict infrequent abnormal transfers in distributed workflows," *Sensors*, vol. 23, no. 12, p. 5485, 2023.

[20] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 95–104, ACM, 2018.

[21] A. Aumon, *Exploratory and Predictive Methods for Multivariate Time Series Data Analysis in Healthcare*. Mathématiques (M. Sc.), Université de Montréal, 2022.

[22] S. Kim and H. Kim, "A new metric of absolute percentage error for intermittent demand forecasts," *International Journal of Forecasting*, vol. 32, no. 3, pp. 669–679, 2016.

[23] G. Juve and E. Deelman, "Scientific workflows and clouds," *ACM Crossroads*, vol. 16, pp. 14–18, 03 2010.

[24] T. Beermann, O. Chuchuk, A. Girolamo, M. Grigorieva, Klimentov, *et al.*, "Methods of data popularity evaluation in the atlas experiment at the lhc," *EPJ Web of Conferences*, vol. 251, p. 02013, 01 2021.