On Performance Prediction of Big Data Transfer in High-performance Networks

Wuji Liu*, Daqing Yun†, Chase Q. Wu*, Nageswara S.V. Rao‡, Aiqin Hou§, and Wei Shen¶
*Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102, USA
†Computer and Information Sciences Program, Harrisburg University, Harrisburg, PA 17101, USA
‡Computational Sciences and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
§School of Information Science and Technology, Northwest University, Xi'an, Shaanxi 710127, China
¶School of Informatics Science and Technology, Zhejiang Sci-Tech University, Hangzhou, Zhejiang, 310018, China
Email: {wl87, chase.wu}@njit.edu, dyun@harrisburgu.edu, raons@ornl.gov, houaiqin@nwu.edu.cn, shenwei@zstu.edu.cn

***Abstract*—Big data generated by large-scale scientific and industrial applications need to be transferred between different geographical locations for remote storage, processing, and analysis. High-speed dedicated connections provisioned in High-performance Networks (HPNs) are increasingly utilized to carry out such big data transfer. HPN management highly relies on an important capability of performance (mainly throughput) prediction to reserve sufficient bandwidth and meanwhile avoid over-provisioning that may result in unnecessary resource waste. This capability is critical to improving the resource (mainly bandwidth) utilization of dedicated connections and meeting various user requests for data transfer. Conventional methods conduct performance prediction by fitting prior observed transfer history with predefined loss functions, without considering unobservable latent factors such as competing loads on end hosts. Such latent factors also have a significant impact on the application-level data transfer performance, which may result in an inaccurate prediction model. In this paper, we first investigate the impact of latent factors and propose a clustering-based method to eliminate their negative impact on performance prediction. We then develop a robust machine learning-based performance predictor by: i) incorporating the proposed latent factor elimination method into data preprocessing, and ii) adopting a customized domain-guided loss function. Extensive experimental results show that our predictor achieves significantly higher prediction accuracy than several other state-of-the-art methods.**

***Index Terms*—Performance prediction, big data transfer, machine learning, latent variables, high-performance networks.**

I. Introduction

Next-generation scientific applications and extreme-scale industrial data analytics require fast and reliable networking services to support big data transfer. High-performance Networks (HPNs) featuring high-speed dedicated connections and advance bandwidth reservation have been developed and deployed in a rapidly expanding scope to provide such services. For example, OSCARS [2] provides advance reservation of secure virtual circuits with guaranteed bandwidth within ESnet [1], and AL2S enables similar reservation services within Internet2 [4] and across other networks. Google's B4 [20] is a private software-defined application-friendly wide-area network (WAN) platform that can be leveraged for big data sciences and industrial applications at the planet scale. Computer systems such as Data Transfer Nodes (DTNs) in Science DMZ [5] have also been widely deployed and used to support geographically distributed scientific applications to reap the benefits brought by dedicated connections of HPNs.

The provisioning agents of such network service providers typically ask end users to submit networking requirements such as desired bandwidth in their data transfer requests in advance, and then allocate resources accordingly over dedicated end-to-end network connections that are dynamically computed and established. During the decision-making process for planning resource allocation and granting usage permission, a resource scheduling strategy is needed to ensure adequate bandwidth while minimizing resource waste. Such a strategy critically relies on the ability to accurately predict real-time data transfer performance. Typically, in HPNs, once the connections with reserved bandwidths are allocated and granted, they become unavailable to other user requests during a certain time frame. This motivates us to build a robust performance predictor for big data transfer in HPNs since a proper resource allocation scheme should not only meet the bandwidth requirements of data transfer requests but also minimize the chance of over-provisioning resources that would be otherwise wasted.

Data transfer is a complex process whose throughput performance is affected by many factors, including not only hardware specifications of both network segments and end hosts, but also software configurations of operating systems and data transfer applications [7]. In addition, there exist a plethora of data transfer methods, including TCP variants such as CUBIC [10], UDP-based protocols such as UDT [9], and utility- or learning-based approaches such as Remy [21], which employ different congestion control mechanisms and are designed for their respective scenarios. Therefore, developing an accurate performance prediction model, even in HPNs where network conditions are relatively more stable in comparison with shared IP networks, is still very challenging.

Most existing methods for performance prediction fit historical performance measurements of data transfer under certain predefined loss functions [11], [12], [14]. Although achieving remarkable success, these methods typically use a limited set of static properties of network paths, end hosts, and applications as predicting features without considering dynamic latent factors such as competing loads and system dynamics on end hosts. These latent factors may also have a significant impact on the performance of big data transfer and are the main cause of inaccuracy for these prediction models in HPN environments.

In this work, we conjecture that the performance prediction model could be biased or overfitted if not excluding the abnormal behaviors of data transfer caused by latent variables.

Therefore, we conduct in-depth analysis of the effects of latent factors based on extensive performance measurements collected in the past several years from a large number of data transfer tests using different protocols and applications between various end sites in several real-life HPN testbeds. We then propose a clustering-based method to eliminate the negative impact of latent factors on performance prediction. We further develop a robust performance predictor by incorporating the proposed elimination method into data preprocessing and customizing domain-specific loss functions. Extensive experimental results show that our predictor achieves significantly higher accuracy in comparison with several state-of-the-art methods.

The rest of the paper is organized as follows. In Section II, we present a brief survey of related work. In Section III, we describe in detail the transport performance prediction problem in HPN environments. In Section IV, we conduct a thorough analysis of the impact of latent factors on transport performance and propose a clustering-based method to eliminate their negative impact on performance prediction. We design a performance predictor and evaluate its performance in Section VI. Section VII concludes our work and sketches a plan for future research.

II. Related Work

The significance of high-speed dedicated connections provisioned by HPNs has been widely recognized in both research and industrial communities due to the rapidly growing big data transfer needs of data- and network-intensive applications. In the past decade, a great deal of research efforts have been made to predict data transfer performance using different methods.

A. Profiling-Based Performance Prediction

Transport performance profiling employs an empirical approach to study the behaviors of different data transfer applications and their underlying transport protocols. A profile of transport performance in response to control parameters of transport methods and network environments is obtained by running data transfer tests with a sweep of the parameter space and collecting corresponding performance measurements. Such profiles can help us understand the network behaviors, facilitate the design of an effective performance predictor, and also be used as benchmarks.

Rao *et al.* in [16] provided large-scale TCP measurements over a set of 10 Gbps dedicated connections with emulated delays ranging from 0 ms to 366 ms, and further in [17] showed that TCP throughput is very sensitive to the connection delay and behaves in a combination of concave and convex functions. Performance profiling of UDT [9], another widely-used data transfer protocol in HPN community [6], is conducted in [7], where UDT behaviors with respect to various application settings and protocol socket options are measured and analyzed. These measurements and analyses show that control parameter settings also significantly affect throughput performance of big data transfer in HPNs. Unfortunately, such effects are not taken into consideration in conventional performance prediction methods. Liu *et al.* conducted similar research on performance profiling of data transfer methods in [19]. While profiling-based approaches offer better interpretability and explainability as they provide a deeper insight into the behaviors of data transfer methods under various circumstances, they typically incur high overhead. For example, to obtain a fully-covered transport profile of a given protocol over a given connection, an exhaustive sweeping of the entire parameter space may take hours or even days to complete.

B. Learning-Based Performance Prediction

Along with the emergence of HPN technologies and the accumulation of performance measurements of big data transfer, machine learning has been increasingly used to investigate and reveal the behavioral patterns of data transfer protocols and the underlying host and network infrastructures.

Mirza *et al.* in [14] considered a set of properties of historical data transfer over network paths as features to train machine learning models, and then used various combinations of subsets of these features for evaluation. Although this work was focused on predicting TCP performance in shared networks, some important features in such environments such as cross traffic were not directly considered when building the model. Liu *et al.* employed regression models to explain the observed performance patterns extracted from the log files of disk-to-disk wide-area file transfer powered by GridFTP [11]. They further in [12] expanded the feature set and developed a model selection strategy for performance prediction of file transfer in wide-area networks. Based on a retraining process, their approach showed promising prediction accuracy, which is verified by a comparative evaluation using Globus logs [3].

III. Problem statement

The throughput performance y of a data transfer over a dedicated connection is considered as a function f of a vector of feature variables $\mathbf{x}$ involving different segments including end hosts, network connections, and applications, i.e., $y = f(\mathbf{x})$. The analytical form of f is typically unknown, and thus we propose to employ machine learning to build a model to approximate f based on historical performance measurements of big data transfer.

More formally, we collect a set of measurements used as the training dataset $\mathcal{T} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i$ $(i = 1, 2, \ldots, n)$ is a specific set of values of the feature vector $\mathbf{x}$ that collectively determine the corresponding throughput y_i. We aim to estimate f based on $\mathcal{T}$, i.e., $\hat{f}(\mathbf{x}_i) \approx f(\mathbf{x}_i)$ such that $\hat{f}(\mathbf{x}_i)$ is close enough to the "true" value y_i for all training examples in $\mathcal{T}$ and can be used to predict y_i with high accuracy given a future arbitrary $\mathbf{x}_i$.

The feature vector $\mathbf{x}$ in this context is in the form of a list of observable variables in the three segments of an end-to-end data transfer path: i) end host configurations such as CPU speed, RAM size, etc.; ii) connection properties such as round-trip time (RTT), connection bandwidth, etc.; and iii) control parameters of data transfer applications such as socket buffer size, number of data streams, etc. However, there exist certain unforeseeable and unobservable latent factors including competing loads (since the end hosts are usually shared by multiple users), system dynamics on end hosts,

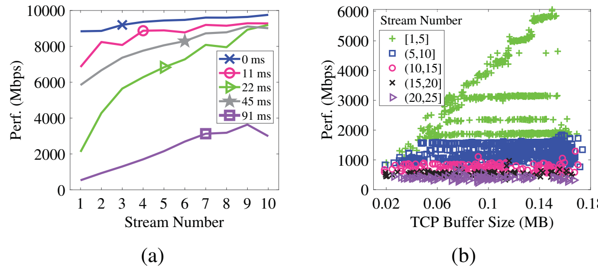


Fig. 1. Illustration of impact of application-accessible parameters on TCP performance: (a) performance vs. stream number and RTT; and (b) performance vs. stream number and buffer size.

and instabilities along the network connection, all of which may also significantly affect the end-to-end data transfer performance. This is mainly because when the network speed reaches a certain high rate, as in the case of HPNs, the speed of (mainly incoming) traffic may keep the end hosts (mainly the receiver) constantly busy and any perturbation under such conditions caused by any latent factor may overwhelm the end host, leading to unpredictable performance.

From the perspective of performance prediction, the above "abnormal" behaviors may result in large noise in the training dataset $\mathcal{T}$. Considering the number of observable factors in $\mathbf{x}$ and the complexity and randomness caused by the latent factors and other unknowns, it is extremely difficult to build a robust performance predictor for HPN resource management, which is critical to satisfying bandwidth requirements of user requests and minimizing resource waste.

Hence, in addition to normal (well-behaving) performance measurements y, the training dataset $\mathcal{T}$ typically also contains some "corrupted" performance measurements y' under the effects of both feature vector $\mathbf{x}$ and unobservable factors α, i.e., $y' = f(\mathbf{x}) + f'(\alpha)$, where $f'(\alpha)$ represents the collective (negative) effects imposed by the latent variables and other unknowns. In other words, the performance measurements in $\mathcal{T}$ are sampled from a combined set of $\{y\}$ and $\{y'\}$.

Our work has two technical components: i) use a generic clustering-based method in data preprocessing to eliminate the "latent-variable-corrupted" data points from the training set; and ii) employ machine learning methods to build an accurate performance predictor based on the cleaned training set.

IV. Analysis of Feature Variables and Latent Factors

Our performance dataset is collected by measuring the throughput performance of repeated data transfer tests over dedicated connections with different application-accessible parameter values. Many of the properties of a given dedicated connection such as delay and capacity can be considered as constants, and we study their impact on transport performance by running the same set of data transfer tests over different connections of different delays and capacities. We first analyze the effects of application-accessible variables, and then identify the (negative) impact of unobservable latent factors based on comparative experimental studies.

A. Effects of Application-Accessible Parameters

Here, we focus on three representative control parameters, i.e., buffer size, number of streams, and round trip time (RTT), and show their impact on throughput performance. More comprehensive profiling results are provided in [16], [19] for both TCP and UDT [9].

The throughput performance measurements with respect to number of streams and RTT are plotted in Fig. 1(a), which shows that, over a 10 Gbps dedicated connection, using multiple streams help achieve better transport performance, especially for a long connection delay. This observation actually has motivated the design of many data transfer toolkits and services such as Globus GridFTP [3] that are being widely used for big data transfer. The behavior under different RTTs indicates that achieving satisfactory performance over a long-haul connection is difficult even for a dedicated channel with sufficient bandwidth.

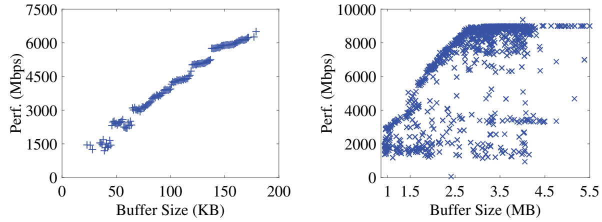
The performance measurements in response to buffer size and number of streams are plotted in Fig. 1(b), which shows that, over short connections, using a large number of parallel streams only brings a limited performance gain in comparison with an appropriately set buffer size. In such cases, they jointly dominate the throughput performance of TCP: the performance generally increases as the buffer size increases; however, as the number of streams increases, the performance gains from increasing the buffer size are diminishing, which is probably due to the resource demand of a large stream number overwhelming the end host system.

B. Effects of Unknowns

The results presented in Sec. IV-A suggest the use of machine learning methods for performance predication of big data transfer in HPNs. This is because: i) the performance patterns are qualitatively consistent and stable across different connections, e.g., the throughput increases as the buffer size and number of streams increase and the achievable throughput decreases as the connection delay increases; and ii) such patterns cannot be modeled analytically, e.g., the slope of performance increase with respect to buffer size increase may vary across different connections, and the optimal number of data streams may depend on not only the network environments but also the end host system configurations. Such multi-dimensional accessible control parameters or features make it difficult, if not at all possible, to derive an analytical form to describe their relationship with the throughput performance.

However, during our extensive experimentation, we found that there may exist certain latent variables that also significantly affect the throughput performance. These latent variables are not easily observable while the data transfer is being performed due to the data transfer application's limited access to the end host system and other unpredictable factors such as competing loads and system dynamics. Such latent effects, if not excluded, could make machine learning-based prediction biased or overfitted.

To show such latent effects, we compare the performance measurements of the same set of data transfer tests conducted on two different testbeds: i) a production HPN testbed where



(a) Without latent effects (b) With latent effects

Fig. 2. TCP performance vs. buffer size without and with latent effects.

the end hosts of data transfer are computing servers shared by many users (thus competing loads are significant), and ii) our local testbed where the experimental conditions are strictly controlled and competing loads from other users are forbidden.

Fig. 2 plots the difference in TCP performance with respect to buffer size with and without the effects of latent variables. As shown in Fig. 2(a), TCP throughput almost linearly increases with the increase of buffer size before reaching the peak. In Fig. 2(b), the maximal achievable TCP performance follows a similar pattern, i.e., increases linearly as the buffer size increases till reaching the peak. In addition, there also appear a non-negligible number of performance measurements below the maximal ones, which may cause inaccuracy in performance prediction for bandwidth scheduling in HPNs.

Figs. 2(a) and 2(b) show that in addition to the impact of data transfer applications and network environments, there are a certain number of unobservable factors such as system dynamics and competing loads from "hidden" users that could undermine the performance. Such latent factors originate from unknown features and are very difficult to estimate due to their unpredictable nature and randomness. They may generate a bias towards the "abnormal" data points during the training process, leading to inaccurate prediction eventually.

In this work, we propose to use machine learning methods to eliminate the (negative) effects of such latent variables during data preprocessing and further build a robust machine learning model for big data transfer performance prediction in HPNs.

V. Elimination of Latent Effects using Clustering

We first describe our approach to eliminate latent effects using clustering-based methods, and then compare different clustering algorithms.

A. Rationale on the Use of Clustering Algorithms

In our prediction problem, the throughput y of a data transfer over a dedicated connection is determined by a combined setting of accessible control parameters and network properties, i.e., the feature vector $\mathbf{x}$, and there is a function $f(\mathbf{x})$ that maps $\mathbf{x}$ to the achievable throughput. The performance observations are data points sampled from a certain Gaussian distribution with mean $\overline{f}(\mathbf{x})$ and variance $\sigma(\mathbf{x})$. For a specific $\mathbf{x}$, the corresponding variance of $\sigma(\mathbf{x})$ from a number of repeated measurements should be limited to some bounded scale. In other words, significant differences observed in repeated measurements with the *same* set of parameter values and network environments may be caused by latent variables and the corresponding effects must be eliminated for accurate prediction.

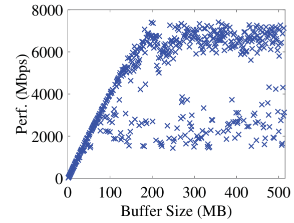


Fig. 3. UDT performance corresponding to buffer size diverges due to latent effects.

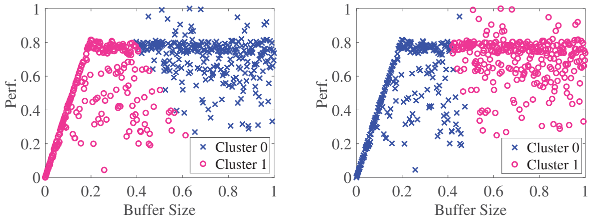
These effects are also illustrated by UDT performance in Fig. 3, where we fix all other parameters and only vary the buffer size. It shows that the UDT throughput with respect to buffer size diverges to two different patterns with and without latent effects. These latent factors would seriously impair the quality of a prediction model. This phenomenon motivates us to use a clustering-based method to eliminate the measurements that are observed under the conditions with significant latent effects. Other research (e.g., [12]) also pointed out the negative effects of such latent factors and a threshold-based method is adopted in [12] to eliminate the effects, which, although simple, may introduce an unexpected bias into the performance prediction model.

In addition, due to the nature of the problem, $f(\mathbf{x})$ is considered to be smooth. In other words, with a slight change to any parameter, e.g., buffer size, the change in the throughput performance should be bounded. If we have a sufficient number of data points for different values of control parameters, we are able to see a smooth pattern of throughput performance, as shown in Fig. 3.

As stated previously, our dataset is a combination of performance observations including both y and y', which are subject to different mapping functions. Our goal is to differentiate the divergence of different performance patterns and rule out the one that is manifested by the "abnormal" data points and is thus less frequently observed, as the regular pattern (exhibited by the normal data points) would appear more frequently in real-life bandwidth scheduling. This can be achieved by using clustering-based methods to categorize y and y' into different clusters with a certain distance measure such that the data points within the same cluster are closer to each other and thus are more likely to be measured under similar conditions with similar latent effects.

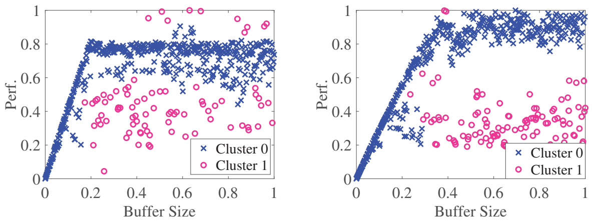
B. Comparison of Different Clustering Algorithms

To choose an appropriate clustering algorithm to separate "abnormal" data points from normal ones, we compare several well-known and commonly-used clustering algorithms, as shown in Fig. 4. Conventional clustering methods based on Expectation Maximization (EM) such as K-means and Gaussian Mixture Model (GMM) aim to maximize the log-likelihood derived from previous estimates. As shown in Fig. 4, K-means and GMM perform poorly in differentiating the data points under different levels of latent effects, since they simply divide the data points into two groups with a roughly equal radius. Therefore, we propose to use the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [13] to eliminate the data points with latent effects and hence fa-



(a) K-means (b) GMM

Fig. 4. Comparison of different clustering algorithms (values are normalized).



(a) Block size: 8955 bytes (b) Block size: 26867 bytes

Fig. 5. Clustering results of DBSCAN (values are normalized).

cilitate accurate performance prediction. DBSCAN categorizes the data points into different clusters based on their densities, where tightly-packed points are grouped together and those in low-density regions are classified as outliers. The clustering results of DBSCAN on the same datasets as used in Fig. 3 and Fig. 4 are presented in Fig. 5(a) and Fig. 5(b), respectively, which show the effectiveness of DBSCAN in differentiating data points with latent effects.

VI. Prediction of Big Data Transfer Performance in High-performance Networks

In this section, we first describe a customized loss function used for building a performance predictor and then present prediction results using various machine learning models. The performance predictors are all implemented in Python based on the `scikit-learn` library [8].

A. Customized Loss Function

Different from traditional methods of supervised learning [12], [14] that seek an optimal label for a given feature vector, for bandwidth scheduling, we aim to build a model that provides a loosened prediction with a reasonable range.

Together with DBSCAN-based data preprocessing, which eliminates negative latent effects, we build our performance predictor based on a customized loss function, as motivated by the domain knowledge of HPN management that requires the reserved bandwidth over a dedicated connection to match the bandwidth requirement of a data transfer request with minimal waste. Therefore, the optimal predicted transfer performance $\hat{y} = \hat{f}(\mathbf{x}_i)$ should lie within the range $[y_i, \epsilon y_i]$, where $\epsilon \geq 1$ is a small tunable positive number and y_i is the ground truth of the achievable performance with feature vector $\mathbf{x}_i$. The predicted value should be slightly higher than what a data transfer can utilize to satisfy the request and also minimize the waste. Inspired by the ϵ-insensitive loss used by Support Vector Regression (SVR) and other work [18], we customize

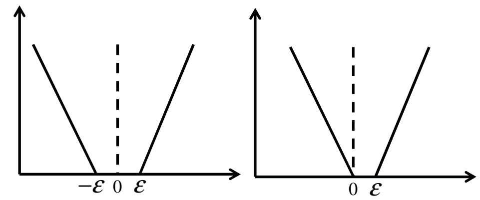


Fig. 6. Loss functions: ϵ-intensive loss [18] (left), the customized loss (right).

the ϵ-insensitive loss function in Fig. 6 (left) by restricting the tolerable errors to be positive only. As shown in Fig. 6 (right) the optimal value is parameterized by an error tolerance ϵ and our objective is to minimize the following loss

$$\mathcal{L}(\theta,\epsilon) = \sum_{i=1}^{n}\Big\{\max(y_i-\hat{f}_\theta(\mathbf{x}_i),0)+\max(\hat{f}_\theta(\mathbf{x}_i)-\epsilon\cdot y_i,0)\Big\}, \tag{1}$$

where the loss $\mathcal{L}(\theta,\epsilon)$ is 0 if the prediction $\hat{f}_\theta(\mathbf{x}_i)$ is larger than the observed value y_i but is within the tolerable range bounded by ϵ; otherwise, $\mathcal{L}(\theta,\epsilon)$ is the distance between $\hat{f}_\theta(\mathbf{x}_i)$ and the tolerable loss range of ϵ.

B. Evaluation Metric

We define a domain-oriented performance evaluation metric, denoted by γ, similar to the Mean Absolute Percentage Error (MAPE), which is a commonly-used accuracy metric in statistics. Unlike MAPE that counts the absolute error, γ counts only the positive errors that fall out of the range governed by ϵ as in Eq. 1. This Customized Mean Absolute Percentage Error (CMAPE) is defined as $\gamma = \frac{1}{n}\mathcal{L}(\theta,\epsilon) = \frac{1}{n}\sum_{i=1}^{n}\big\{\max(y_i-\hat{f}_\theta(\mathbf{x}_i),0)+\max(\hat{f}_\theta(\mathbf{x}_i)-\epsilon\cdot y_i,0)\big\}, \epsilon \geq 1$, where $\hat{f}_\theta(\mathbf{x}_i)$ is the predicted value given feature $\mathbf{x}_i$, y_i is the corresponding ground truth, n is the total number of test cases. A proper bandwidth allocation should satisfy the user requirement with only an inevitable (as governed by ϵ) amount of waste. In addition to γ, we also count the Effective Prediction Percentage (EPP, denoted by β) among all test cases, i.e., $\beta = \frac{1}{n}\sum_{i=1}^{n}\mathcal{I}\{y_i \leq \hat{f}_\theta(\mathbf{x}_i) \leq \epsilon \cdot y_i\}$, where $\mathcal{I}(\psi)$ is an indicator function that is equal to 1 if ψ is true, and 0, otherwise.

C. Models in Comparison

We compare four models [15] with the customized loss function defined in Eq. 1: i) linear models as represented by Ridge Regression (RR); ii) non-linear models as represented by Support Vector Regression (SVR); iii) Neural Networks (NN), where we use a standard three-layer neural network with ReLU as the activation function; and iv) ensemble models as represented by Random Forest Regression (RFR).

D. Dataset

Our dataset contains about 100,000 records of throughput performance measurements that are collected from big data transfer tests conducted over local back-to-back connections and in several other HPNs managed by different institutions.

E. Results

Our performance evaluation includes two parts: i) compare the prediction results of SVR on the original dataset with and without the DBSCAN-based preprocessing as introduced in Sec. V-B; and ii) compare the performance of the four models in Sec. VI-C and select the best one with data preprocessing.

1) SVR Prediction Accuracy With and Without Preprocessing

We first run the DBSCAN-based preprocessing to filter out data points that are heavily affected by latent factors and then perform prediction using SVR. This process is repeated without such preprocessing for comparison. As shown in Fig. 7, the accuracy of SVR-based performance prediction based on the cleaned dataset consistently outperforms the prediction accuracy based on the original dataset. Note that the dataset used in these tests is collected from a production HPN, where high-end servers used as the sender/receiver of data transfer tests are concurrently used by many other scientists to run their scientific computing jobs.

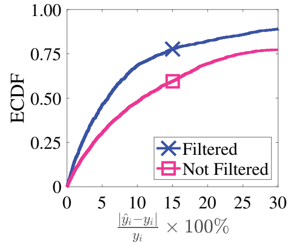


Fig. 7. Comparison of SVR prediction accuracy with and without DBSCAN-based preprocessing.

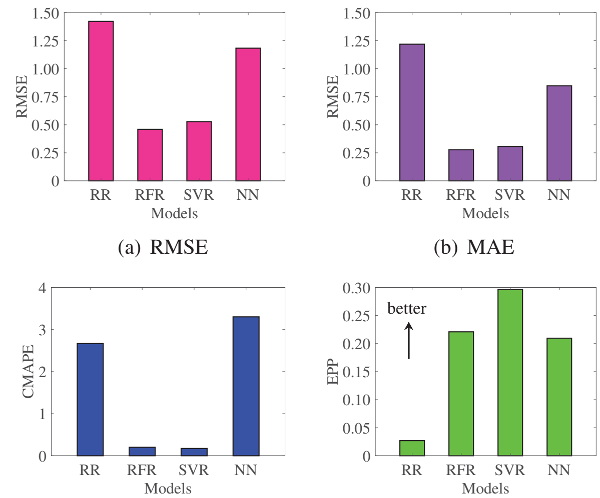
2) Prediction Accuracy of Various Models

We use the filtered dataset to compare the prediction accuracy of various models as mentioned in Sec. VI-C in terms of different performance criteria including Root Mean Square Error (RMSE), Mean Absolute Error (MAE), EPP (i.e., β), and CMAPE (i.e., γ). As shown in Fig. 8, the linear RR model performs poorly for all four criteria due to the limited richness of its hypothesis function set. The NN model performs even worse than RR for RMSE and MAE, which indicates extensive tuning or more network layers (thus higher overhead) are needed. RFR and SVR perform almost equally well for RMSE and MAE. SVR has the best overall performance since it outperforms all other models for both of the metrics defined for bandwidth scheduling, i.e., CMAPE and EPP. Therefore, we choose SVR with the customized loss function for performance prediction in HPNs.

VII. Conclusion

In this work, we studied the prediction problem of big data transfer performance in HPNs. We first identified the latent factors and analyzed their negative impact on performance prediction based on comparative experiments. We then proposed a clustering-based method to eliminate such negative impact and developed a performance predictor using various machine learning algorithms with a domain-guided customized loss function. Experimental results show that the SVR-based predictor achieved significantly higher accuracy in comparison with several other state-of-the-art methods in terms of various performance evaluation criteria.

We plan to investigate other methods for latent effect elimination and refine our models accordingly to further improve prediction accuracy. It is also of our interest to establish a performance bound of the selected machine learning model for data transfer performance prediction in HPNs.



(a) RMSE (b) MAE (c) CMAPE (γ) (d) EPP (β)

Fig. 8. Performance comparison of various models in terms of different metrics.

Acknowledgments

This research is sponsored by U.S. National Science Foundation under Grant No. CNS-1828123 with New Jersey Institute of Technology and the Presidential Research Grant of Harrisburg University.

References

[1] Energy Sciences Network. http://www.es.net.
[2] ESnet OSCARS Service. https://bit.ly/2Ou9qVe.
[3] Globus Data Transfer. https://bit.ly/2m10qLI.
[4] Internet2 AL2S. https://goo.gl/4iAbQn.
[5] Science DMZ DTNs. https://bit.ly/2k3qBQM.
[6] UDT-Powered Projects. https://bit.ly/2JZtA7n.
[7] D. Yun *et al.* Profiling optimization for big data transfer over dedicated channels. In *Proc. of ICCCN '16*.
[8] F. Pedregosa *et al.* Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
[9] Y. Gu and R. Grossman. UDT: UDP-based data transfer for high-speed wide area networks. *Computer Networks*, 51(7):1777–1799, 2007.
[10] S. Ha, I. Rhee, and L. Xu. CUBIC: A new TCP-friendly high-speed TCP variant. *ACM SIGOPS Oper. Syst. Rev.*, 42(5):64–74, 2008.
[11] Z. Liu, P. Balaprakash, R. Kettimuthu, and I. Foster. Explaining wide area data transfer performance. In *Proc. of HPDC '17*, pages 167–178.
[12] Z. Liu, R. Kettimuthu, P. Balaprakash, and I. Foster. Building a wide-area data transfer performance predictor: An empirical study. In *Proc. of the 1st Int'l Conf. on Machine Learning for Netw.*, 2018.
[13] M. Ester *et al.* A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of KDD '96*, pages 226–231.
[14] M. Mirza, J. Sommers, P. Barford, and X. Zhu. A machine learning approach to TCP throughput prediction. *IEEE/ACM Trans. on Networking*, 18(4):1026–1039, 2010.
[15] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2nd edition, 2018.
[16] N.S.V. Rao *et al.* TCP throughput profiles using measurements over dedicated connections. In *Proc. of HPDC '17*, pages 193–204.
[17] N.S.V. Rao *et al.* Learning concave-convex profiles of data transport over dedicated connections. In *Proc. of the 1st Int'l Conf. on Machine Learning for Netw.*, 2018.
[18] P. Ye *et al.* Customized regression model for Airbnb dynamic pricing. In *Proc. of KDD '18*, pages 932–940.
[19] Q. Liu *et al.* Measurement-based performance profiles and dynamics of UDT over dedicated connections. In *Proc. of ICNP '16*.
[20] S. Jain *et al.* B4: Experience with a globally-deployed software defined WAN. *ACM SIGCOMM Comput. Commun. Rev.*, 43(4):3–14, 2013.
[21] K. Winstein and H. Balakrishnan. TCP ex machina: Computer-generated congestion control. *ACM SIGCOMM Comput. Commun. Rev.*, 43(4):123–134, 2013.