Evaluation of Deep Learning Models for Network Performance Prediction for Scientific Facilities

Makiya Nakashima: Texas A&M University-Commerce
Alex Sim: Lawrence Berkeley National Laboratory
Jinoh Kim: Texas A&M University-Commerce

Outline

- Introduction
- Dataset
- Deep learning models
- Experiments
- Conclusion

Introduction

- Large data transfers are getting more critical with the increasing volume of data in scientific computing
- To support large data transfers, scientific facilities manage dedicated infrastructures with a variety of hardware and software tools
- Data transfer nodes (DTNs) are dedicated systems to data transfers in scientific facilities that facilitate data dissemination over a large-scale network

Introduction

- Predicting network performance based on the historical measurement would be essential for workflow scheduling and resource allocation in the facility
- In that regard, the connection log would be a helpful resource to infer the current and future network performance, such as for change point and anomaly detection and for throughput and packet loss prediction

Introduction

- Analyze a dataset collected from DTNs
- Evaluate deep learning (DL) models with respect to the prediction accuracy of network performance for scientific facilities

DL models: Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Gated Recurrent Unit (GRU), and Long Short-Term Memory (LSTM)

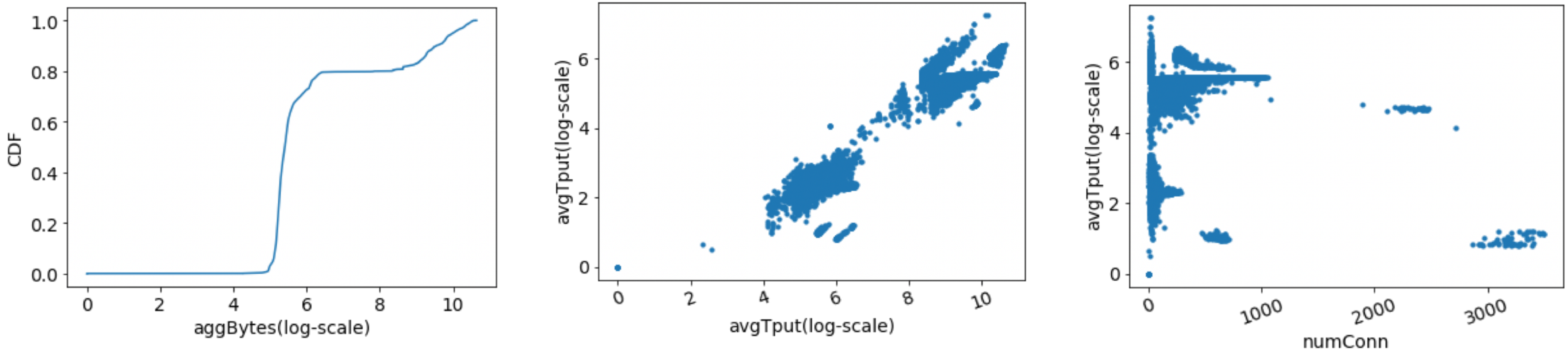
Dataset

- tstat tool collects TCP instrumentation data for each flow
- The tool measures the transport layer statistics, such as the number of bytes/packets sent and received, the congestion window size, and the number of packets retransmitted.
- Number of features: 107 features
  - aggBytes: Aggregated bytes
  - numConn: Number of connections
  - avgTput: Average throughput (=aggBytes/numConn)

  Note: avgTput is the prediction target

Data analysis (w = 1 min, January)

(a) CDF of aggBytes (b) Correlation of avgTput vs. aggBytes (c) Correlation of avgTput vs. numConn

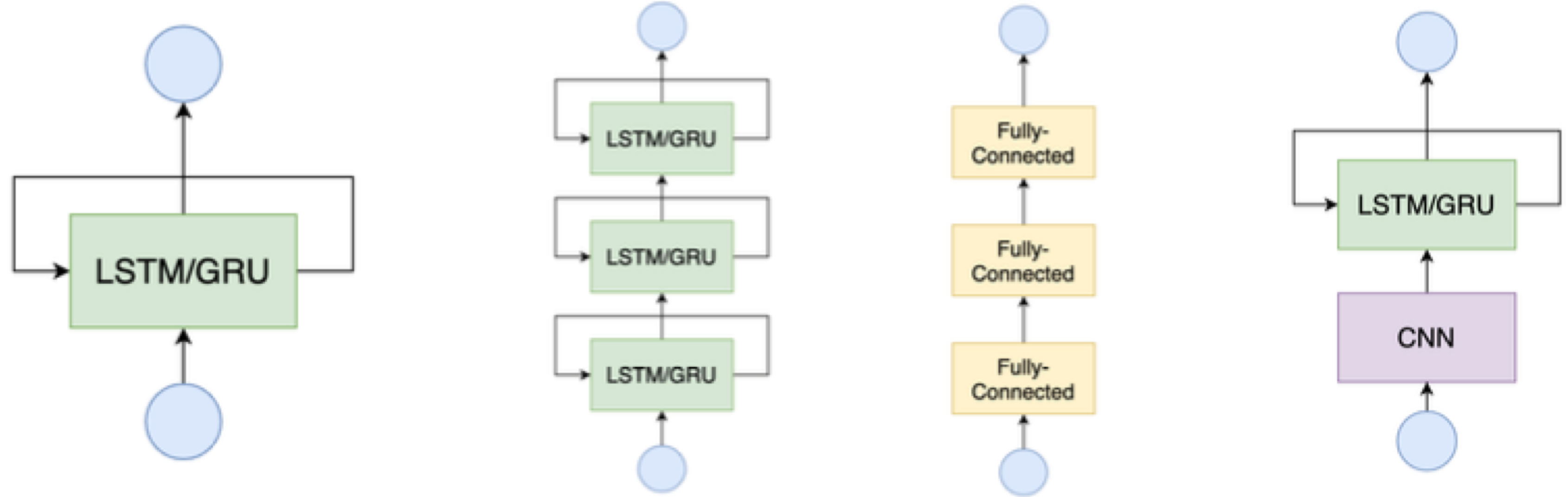


(a) Greater than 10GB downloading in one minute from roughly 20% of windows, while around 50% of time shows light traffic less than 1MB

(b) There is high degree of correlation between avgTput and aggBytes

(c) avgTput is inversely correlated to numConn

Deep learning models



- LSTM/GRU
- Stacked LSTM/GRU
- Stacked ANN
- Combination of CNN-LSTM

Experiments setting

- Normalization: standard feature scaling (0–1)
- Window size: w = 1 minute
- Sequence length: s = {5, 15, 30, 60}
- Training: First 60% of windows, Testing: the rest (40%)
- Metrix: Root Mean Squared Error, Relative Difference

$$RMSE = \sqrt{\frac{\sum_i (m_i - p_i)^2}{N}} \qquad RD(m_i, p_i) = \frac{|m_i - p_i|}{\left(\frac{|m_i + p_i|}{2}\right)}$$

Initial DL experiment (January)

- GRU or LSTM works well compared to the other structures.
- Using s = 5 works better than longer sequence lengths. Using s = 60 works better than s = 15 and s = 30

Note: C=CNN, D=DNN, L=LSTM, G=GRU
GGG = 3 layers GRU

Model	$s=5$	$s=15$	$s=30$	$s=60$
C(s)	118126	87321	125536	74340
D(s)	207915	193880	105496	207634
G(s)	58411	118167	110756	67322
L(s)	58183	108850	139787	80361
CCC(s)	195506	221347	296393	219396
DDD(s)	309117	346295	88380	68821
GGG(s)	81606	100447	163036	185395
LLL(s)	71197	133158	234786	72297

Top-10 testing performance for predicting (January)

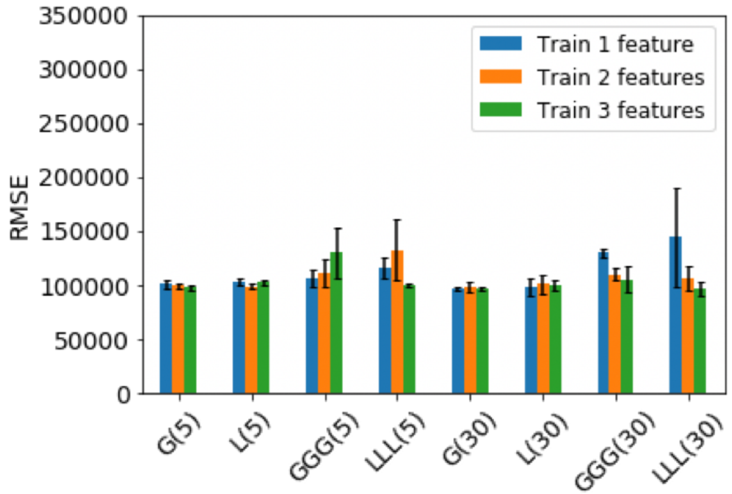
- Single-layer models with s = 5 quite work well, yielding better results than multi-layer models or with a longer sequence length

Model	Num. variables	RMSE (training)	RMSE (testing)
L(5)	1	98494	58183
G(5)	1	97292	58411
GGD(5)	1	107531	58504
G(15)	3	94028	59890
GD(60)	1	98966	61928
G(30)	3	94989	62309
LLL(5)	3	97940	62513
L(5)	3	99997	64686
G(60)	1	94740	67322
DDD(60)	1	161026	68821

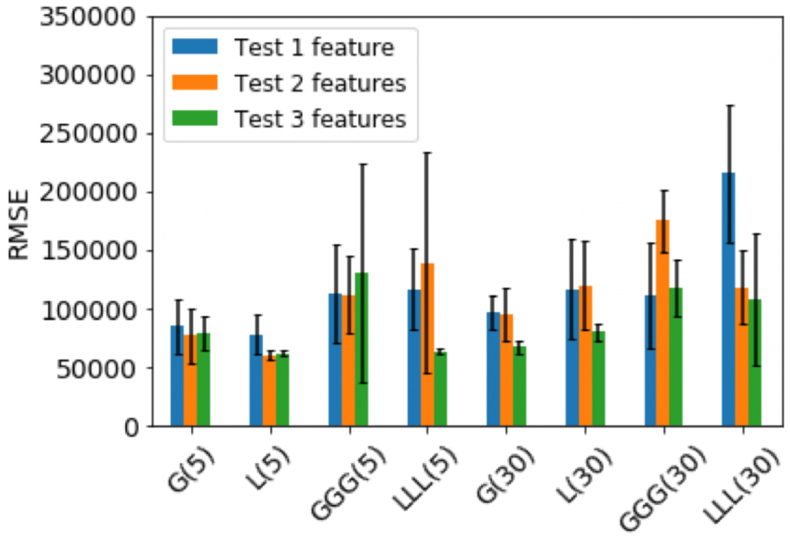
Experiments with DL models based on GRU and LSTM structures

1 feature: $avgT\ put$
2 features: $avgTput, numConn$
3 features: $avgTput, aggBytes, numConn$

Training RMSE for $avgT\ put$ (Jan)



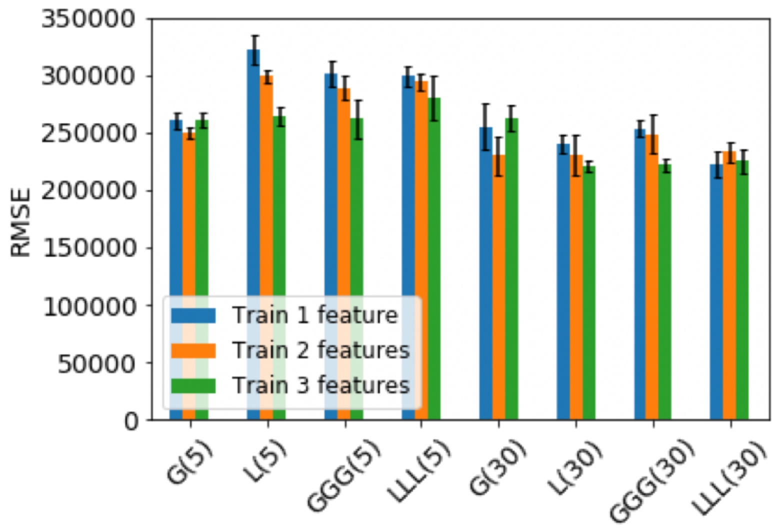
Testing RMSE for $avgT\ put$ (Jan)



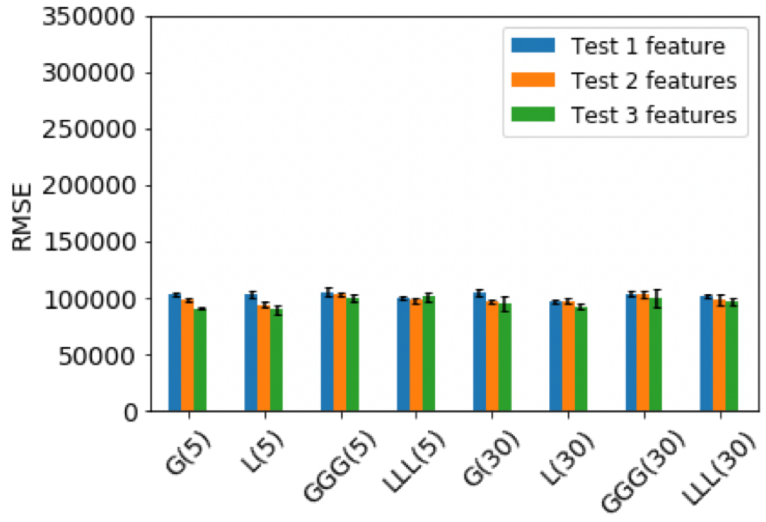
Using three features slightly works consistently compared to the use of the less number of features

Experiments with DL models based on GRU and LSTM structures

Training RMSE for $avgTput$ (Feb)



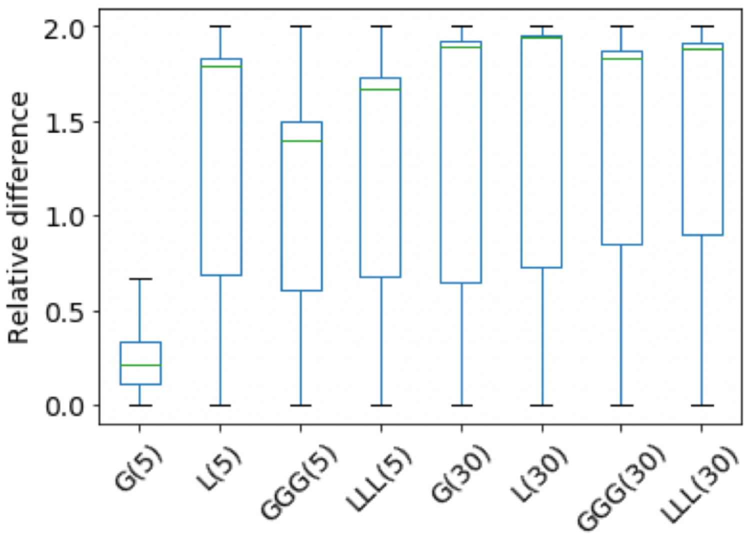
Testing RMSE for $avgTput$ (Feb)



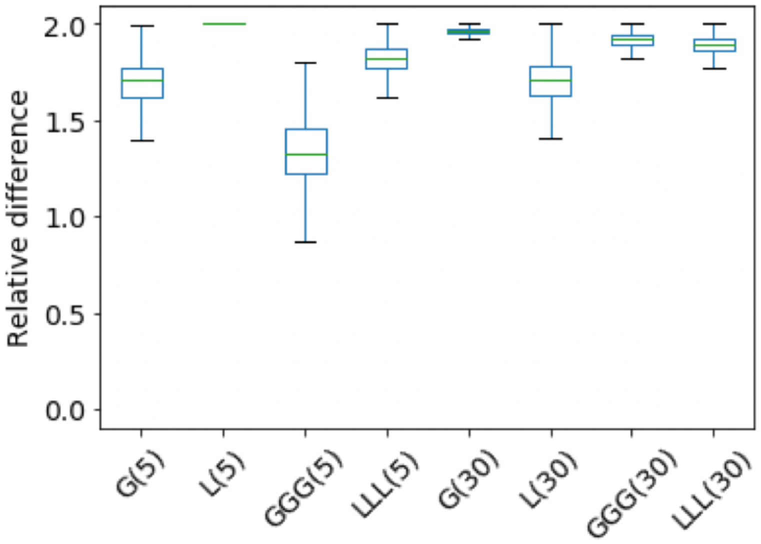
Training error is higher than January data, but testing error is lower

Comparison of DL models using the RD metric

Testing RD for $avgT\ put$ (Jan)



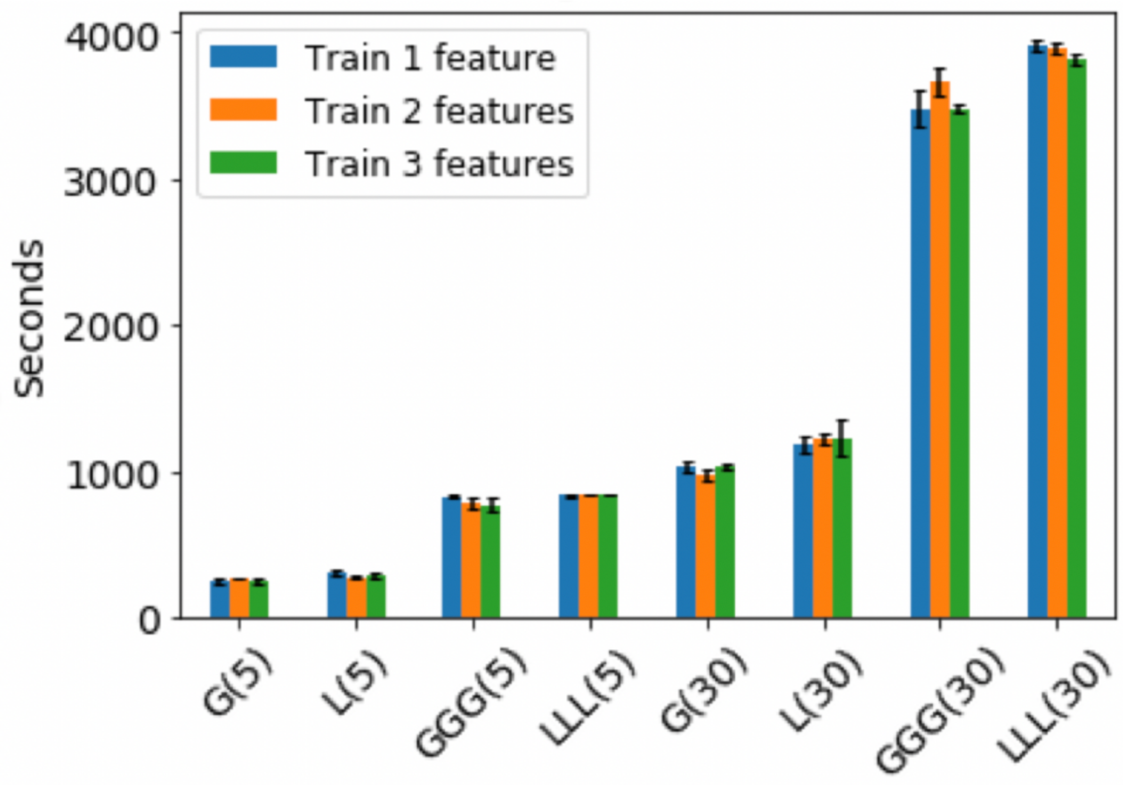
Testing RD for $avgT\ put$ (Feb)



G(5) and GGG(5) show much better results than the other models including the relevant LSTM models with much smaller relative difference values

Time complexity based on GRU and LSTM structures

- Using a smaller number of cells is beneficial for reducing the amount of time for learning data
- Using a smaller sequence length would require a less amount of time for executing



Conclusion

- Established a set of DL models based on ANN, CNN, GRU, LSTM, and combined DL models, to predict average throughput
- From the extensive experiments, our observations show that using recurrent DL models (based on GRU or LSTM) work better than non-recurrent models (based on CNN and ANN)
- Simple model with a single layer and a relatively small sequence length would have some benefits, given the significantly high timing complexity for complicated models