Machine Learning for Interconnect Network Traffic Forecasting: Investigation and Exploitation

Xiongxiao Xu
Illinois Institute of Technology
Chicago, Illinois, USA
xxu85@hawk.iit.edu

Xin Wang
Illinois Institute of Technology
Chicago, Illinois, USA
xwang149@hawk.iit.edu

Elkin Cruz-Camacho
Rensselaer Polytechnic Institute
Troy, New York, USA
cruzce@rpi.edu

Christopher D. Carothers
Rensselaer Polytechnic Institute
Troy, New York, USA
chrisc@cs.rpi.edu

Kevin A. Brown
Argonne National Laboratory
Lemont, Illinois, USA
kabrown@anl.gov

Robert B. Ross
Argonne National Laboratory
Lemont, Illinois, USA
rross@mcs.anl.gov

Zhiling Lan
Illinois Institute of Technology
Chicago, Illinois, USA
lan@iit.edu

Kai Shu
Illinois Institute of Technology
Chicago, Illinois, USA
kshu@iit.edu

ABSTRACT

Interconnect networks play a key role in high-performance computing (HPC) systems. Parallel discrete event simulation (PDES) has been a long-standing pillar for studying large-scale networking systems by replicating the real-world behaviors of HPC facilities. However, the simulation requirements and computational complexity of PDES are growing at an intractable rate. An active research topic is to build a surrogate-ready PDES framework where an accurate surrogate model built on machine learning can be used to forecast network traffic for improving PDES. In this paper, we make the first attempt to introduce two representative time series methods, the Autoregressive Integrated Moving Average (ARIMA) and the Adaptive Long Short-Term Memory (ADP-LSTM), to forecast the traffic in interconnect networks, using the Dragonfly system as a representative example. The proposed ADP-LSTM can efficiently adapt to the ever-changing network traffic, facilitating the forecasting capability for intricate network traffic, by incorporating a novel online learning strategy. Our preliminary analysis demonstrates promising results and shows that ADP-LSTM can consistently outperform ARIMA with significantly less time overhead.

ACM Reference Format:

Xiongxiao Xu, Xin Wang, Elkin Cruz-Camacho, Christopher D. Carothers, Kevin A. Brown, Robert B. Ross, Zhiling Lan, and Kai Shu. 2023. Machine Learning for Interconnect Network Traffic Forecasting: Investigation and Exploitation. In *ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (SIGSIM-PADS '23), June 21–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3573900.3591123

SIGSIM-PADS '23, June 21–23, 2023, Orlando, FL, USA

ACM ISBN 979-8-4007-0030-9/23/06.
https://doi.org/10.1145/3573900.3591123

1 INTRODUCTION

High-performance computing (HPC) systems rely on efficient and scalable interconnect networks to support unprecedented system size at a reasonable cost. For example, the Dragonfly network is a hierarchical, high-radix, low-diameter topology that can reduce the network cost with high-bandwidth and low-latency service [15, 16]. Such a unique network topology has been widely adopted by various HPC facilities, e.g., the National Energy Research Scientific Computing Center and the Argonne Leadership Computing Facility.

Parallel discrete event simulation (PDES) has been successfully applied for modeling network data flows and hierarchical storage systems, including transportation and mobility applications, internet and cybersecurity simulations, and simulations for hardware co-design [23]. Although PDES modeling frameworks, such as ROSS [3] and CODES [22], can conduct simulations on Dragonfly networks, the high computational complexity hinders its deployment in practice. For example, PDES takes four hours to simulate the 4,096-node MILC running for 122 milliseconds [27].

To address the issue, high-fidelity *surrogate* models are desired to accelerate PDES due to their potential for forecasting network-level performance by capturing temporal correlation in the network traffic. We envision a surrogate-ready PDES where the simulation alternates between two phases: a detailed PDES phase where the application workload is simulated fully and a surrogate phase where the induced traffic is forecasted to fast-forward PDES through the time period forecatsed by the surrogate. However, designing effective surrogate models faces several challenges. First, when simulating multiple HPC applications on a large-scale interconnect network, the fierce contention among the applications for shared resources can have a detrimental effect on the forecasting accuracy of surrogate models [20, 28]. Second, the dynamically-changing network conditions pose additional challenges, requiring the surrogate model to be able to adaptively incorporate evolving network information. Third, PDES events are generated at a rapid pace. There is a pressing need to ensure forecast efficiency from surrogate models, in addition to maintaining effectiveness.

To tackle the above challenges, we make the first attempt to investigate the potential for machine learning frameworks to build surrogate models for interconnect network traffic forecasting. Specifically, we utilize ARIMA and ADP-LSTM, two representative time series forecasting models, as surrogate models to accelerate PDES models, with a 72-node Dragonfly simulation as a case study. The proposed algorithm Adaptive Long Short-Term Memory (ADP-LSTM) integrates the strength of offline learning and online learning to adapt to intricate and dynamic network traffic. In the experiments, we conduct a preliminary study to compare the performance and time overhead of ARIMA and ADP-LSTM. The experimental results show ADP-LSTM can obtain superior performance than ARIMA and ADP-LSTM requires significantly less time overhead.

2 RELATED WORK

In this section, we briefly describe the related work on (1) machine learning for traffic forecasting, and (2) traffic forecasting on interconnect networks.

2.1 Machine Learning for Traffic Forecasting

The research field of traffic forecasting has existed for many years [13, 17, 24, 29, 37]. The earlier researchers utilize statistical method ARIMA [1, 18, 19, 30, 31] to forecast traffic. Afterward, the researcher employ traditional machine learning, such as neural network [6, 8] and Support Vector Machine (SVM) [4, 14]. However, the classical statistical and traditional machine learning approaches are relatively weak due to their oversimplified assumptions and limited representation capabilities, respectively. Nowadays, deep learning methods are popular in traffic forecasting due to its strong expressive capabilities. The existing deep learning methods for traffic forecasting can be classified into Feedforward Neural Network (FNN) [9, 36, 40], Convolutional Neural Network (CNN) [33, 35, 39], Recurrent Neural Network (RNN) and LSTM [7, 21, 35], Graph neural network (GNN) [5, 34, 44] and Transformer [26, 38, 41, 43].

2.2 Network Traffic Forecasting

A few studies have been conducted on network traffic prediction. Huang et al. [12] propose a table-driven framework called Network Traffic forecasting Table (NTPT) to predict traffic in the on-chip interconnect network. Huang et al. [11] propose a table-based traffic predictor which can capture application traffic patterns. Zhou et al. [42] design a traffic forecasting strategy and propose three latency trade-off models. Other studies focus on traffic forecasting on software defined network. Azzouni et al. [2] propose the NeuTM, a model based on Long Short-Term Memory Recurrent Neural Networks (LSTM RNNs), for network traffic matrix forecasting. The existing studies mainly focus on tile-based network-on-chip or wide-area network at a small scale. Different from the existing research, we target the traffic forecasting problem in the Dragonfly interconnect network which features a hierarchical topology. In the case study shown in Section 4, we investigate the network traffic forecasting on a 252-port dragonfly network. To the best of our knowledge, this is the first paper to focus on network traffic forecasting in the Dragonfly network.

3 NETWORK TRAFFIC FORECASTING

In this section, we formally define the problem, and assess two machine learning aproaches based surrogate models.

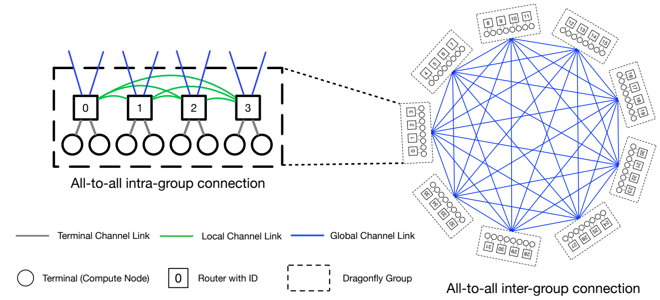


Figure 1: The illustration of the 1D Dragonfly network.

3.1 Problem Definition

We focus on the 1D Dragonfly network as shown in Figure 1 with n routers $R_i \in \{R_1, R_2, ..., R_n\}$. Each router R_i has m ports, including terminal ports, local ports, and global ports. We denote terminal ports as $T_{j_T} \in \{T_1, T_2, ..., T_{m_T}\}$, local ports as $L_{j_L} \in \{L_1, L_2, ..., L_{m_L}\}$, and global ports as $G_{j_G} \in \{G_1, G_2, ..., G_{m_G}\}$. The $t \in \{1, 2, ..., T\}$ is defined as the set of time intervals. In a time interval, application processes on the ports communicate with each other, creating traffic on the network. We define the traffic as the size of the packet in/out of a port during a time interval. Formally, we use $y^t_{i,j}$ to denote the total traffic of port j on the router i in a time interval t. Since we only focus on port-level forecasting in this paper, we ignore the i and j in the following descriptions. Our study considers *a k-step delay*, meaning at time t our forecast model can access the data at $t-k$ and before. We formally define the problem as follows:

Problem Statement: Given historical data $\mathcal{H}_{t-k} = \{y_1, y_2, \dots, y_{t-k}\}$ of port j on router i, the network traffic forecasting problem aims to forecast traffic y_t of port j on router i at time interval t (*k-step delay*).

3.2 Surrogate Modeling

To capture the complex network traffic, we investigate two types of surrogate methods: ARIMA and ADP-LSTM, due to success in traffic forecast scenarios such as road traffic forecasting [1, 35].

ARIMA (AutoRegressive Integrated Moving Average) [1] is a classical statistical time series forecasting method and has been widely adopted in traffic forecasting. The $ARIMA(p, d, q)$ has three parameters: p, d, and q. p is the order of the autoregressive model; q is the order of the moving-average model; d is the degree of differentiation, which enables ARIMA to model non-stationary signals.

ADP-LSTM (Adaptive Long Short-Term Memory) is developed based on the LSTM [10] framework. LSTM is shown to be effective in addressing the gradient vanishing problem and capable of capturing long-term dependency [25]. However, the network traffic on the interconnect networks such as Dragonfly networks is intricate since the Dragonfly network has a more complex topology. To capture the complex network traffic, we propose ADP-LSTM which can dynamically update parameters and adapt to the evolving network traffic, by incorporating an online learning strategy into the standard offline learning. Specifically, offline learning means the model is trained before the inference, and online learning denotes the trained model is dynamically adjusted based on the newly accessible data during the inference. Such flexibility enables the capacity of adapting dynamically to complex network traffic and further facilitating the traffic forecasting performance.

3.3 Model Training and Inference

ARIMA. During the training, ARIMA fits the data stream with a fixed-length historical data window; in the inference, ARIMA forecasts the traffic based on the fitted model. ARIMA forecasts one-step traffic at a time. By sliding the historical data window and iterating the above processes, ARIMA can continuously perform forecasting to get multi-step traffic.

ADP-LSTM. As shown in Algorithm 1, the training of ADP-LSTM includes two phases: an offline learning phase and an online learning phase. In the offline learning phase (lines 2 – 4), we obtain a small portion of data for the training of a desirable ADP-LSTM. The remaining data is for the inference (lines 5 – 11). In the online learning phase (lines 8 – 10), we tune ADP-LSTM with a single newly accessible data to adapt dynamically to the changing traffic. In the inference, ADP-LSTM forecasts one-step traffic at a time and continuously forecasts to obtain the multi-step traffic by sliding a fixed window.

Note that in the setting of a k-step delay, our surrogate models can only access the actual data with a delay of k intervals. The historical data window consists of the actual data when available and the subsequent forecasted data.

Algorithm 1 The algorithm of ADP-LSTM

Require: Historical data $\mathcal{H}_T = \{y_1, y_2, ..., y_T\}$ in a time period T and l denoting the fixed length of historical data window.
1: Split historical data $\mathcal{H}_T$ into $\mathcal{H}_{train}$ and $\mathcal{H}_{test}$
2: **while** offline learning phase **do**
3: Train ADP-LSTM with $((y_{t-l}, y_{t-l+1}, ..., y_{t-1}), y_t)$ in $\mathcal{H}_{train}$
4: **end while**
5: **while** inference **do**
6: Forecast traffic $\tilde{y}_t$ at time t based on a historical data window, concatenated by actual data $(y_{t-l}, y_{t-l+1}, ..., y_{t-k})$ and forecasted data $(\tilde{y}_{t-k+1}, \tilde{y}_{t-k+2}, ..., \tilde{y}_{t-1})$
7: Access a newly acquired data y_{t-k+1}
8: **while** online learning phase **do**
9: Tune ADP-LSTM with the newly accessible data $((y_{t-(k+l-1)}, y_{t-(k+l-2)}, ..., y_{t-k}), y_{t-k+1})$
10: **end while**
11: **end while**

4 EXPERIMENTAL EVALUATION

Network Simulator. We utilize CODES to generate the network traffic data. CODES [22] is a packet-level, high-fidelity simulation tool that can collect detailed network information on Dragonfly networks. We simulate both Jacobi 3D and MILC which run simultaneously on a 72-node Dragonfly system. Each application occupies half of all the compute nodes. We collect the aggregated traffic volume that each port routed in a series of time intervals. The time interval size is set to 300 microseconds.

Network Topology. The Dragonfly network (see Figure 1) has a hierarchical design, consisting of the all-to-all inter-group connection and intra-group connection. In the network, the 72 compute nodes and 36 routers are averagely divided into 9 groups, which are all-to-all connected. Within a group, the routers are also all-to-all connected. Each router has 7 ports, including 2 terminal ports, 3 local ports, and 2 global ports. The total number of ports is 252.

Applications. Our workload includes two widely-used applications [27]: (1) **Jacobi 3D** is a widely used scientific application. The processes of Jacobi 3D are formed in a $4 \times 3 \times 3$ Cartesian grid and every process communicates with 6 neighbors at each iteration, and (2) **MILC** is developed by the MIMD Lattice Computation (MILC) collaboration for quantum chromodynamics (QCD). It executes simulations of four-dimensional SU (3) lattice gauge theory.

Job Placement. We investigate two job placement policies [28]: (1) *Contiguous Placement* selects computer nodes consecutively for the processes of the job to occupy, and (2) *Random Placement* selects computer nodes randomly for the processes of the job to occupy.

Adaptive routing [28] is adopted in our simulation. It routes packets along the minimal path or the non-minimal path based on the network congestion state. When a non-minimal path is selected, the packets will be minimally routed into a randomly intermediate router, then minimally forwarded to their destinations. Accordingly, we denote the two settings as *cont-adp* and *random-adp*.

Model Training and Testing. The dataset consist of 1,580 and 1,630 data for *cont-adp* and *random-adp*, respectively. We use the first 200 data for training and the rest for testing. For ARIMA, the hyperparameters p, d and q are set to 13, 1 and 5, respectively. For ADP-LSTM, the hidden dimension and learning rate are set to 64 and 0.001, respectively. The historical data window lengths for ARIMA and LSTM are 200 and 13, respectively.

Evaluation Metrics. We use the following metrics to assess the effectiveness and efficiency of the aforementioned surrogate models.

- **MSE** (Mean Square Error) and **MAE** (Mean Absolute Error) are two metrics to evaluate the performance of the forecasting models in the forecasting problem [32, 35, 38]
- **Time Overhead** measures the overall time overhead of the forecasting models on a port. For ARIMA, it includes the time of calculating parameters and the time of estimating network traffic. For ADP-LSTM, it includes the offline training cost, the online training cost, and the inference overhead.

4.1 Experimental Results

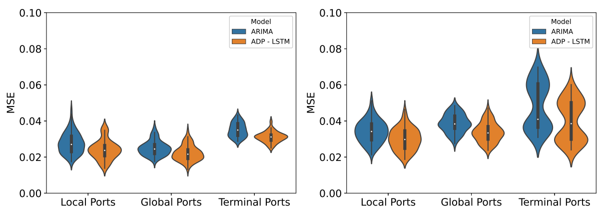
In the following, we describe our results in two aspects: port level analysis and network level analysis. In port level analysis, we analyze the results of some representative ports; in network level analysis, we present an overall perspective for all ports on all routers. Note that we consider the one-step delay (i.e., $k = 1$) for the following results and will perform a sensitivity analysis in section 4.2.

4.1.1 Port Level Analysis. We evaluate the performance and time overhead of ARIMA and ADP-LSTM on all ports of all routers. Due to page limits, we show the results of three routers and three ports for each router, and the results are shown in Table 1.

Performance. From Table 1, we can clearly observe that ADP-LSTM outperforms ARIMA consistently. The reason is that ADP-LSTM can leverage powerful capabilities of deep learning to capture complex patterns implied in network traffic while ARIMA based on the statistical algorithm is relatively limited in handling such intricate traffic. More importantly, ADP-LSTM can dynamically adapt to complex network traffic, which boosts the forecasting capability for network traffic.

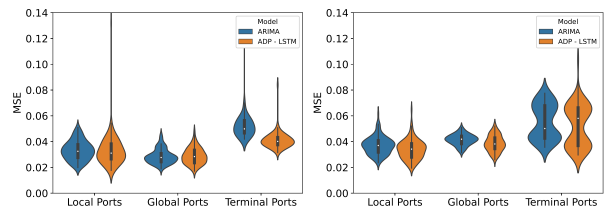
Table 1: Accuracy and time overhead results of port L_0, G_0 and T_0 on Router R_0, R_{18} and R_{35} when $k = 1$. T.O. means time overhead. IMPROVEMENT(Δ) indicates the improvement of ADP-LSTM compared to the ARIMA.

Setting	Router	Port	ARIMA			ADP-LSTM			IMPROVEMENT (Δ)		
			MSE	MAE	T.O. (s)	MSE	MAE	T.O. (s)	$\Delta(MSE)$	$\Delta(MAE)$	$\Delta(T.O.)$
cont-adp	R_0	L_0	0.0321	0.1410	2,255	0.0219	0.0939	26	46.72%	50.13%	85
		G_0	0.0271	0.1238	1,796	0.0234	0.0912	25	15.77%	35.81%	69
		T_0	0.0359	0.1411	1,998	0.0317	0.1127	25	13.32%	25.22%	78
	R_{18}	L_0	0.0321	0.1401	2,087	0.0284	0.1274	25	12.88%	10.00%	82
		G_0	0.0287	0.1339	1,925	0.0233	0.1154	25	23.49%	16.09%	76
		T_0	0.0333	0.1431	2,055	0.0313	0.1367	25	6.32%	4.70%	80
	R_{35}	L_0	0.0231	0.1145	1,983	0.0217	0.1009	26	6.86%	13.56%	75
		G_0	0.0226	0.1166	1,908	0.0171	0.0947	26	32.11%	23.14%	73
		T_0	0.0367	0.1506	2,076	0.0353	0.1474	26	3.97%	2.16%	79
random-adp	R_0	L_0	0.0332	0.1479	2,263	0.0298	0.1386	25	11.43%	6.66%	90
		G_0	0.0334	0.1468	2,241	0.0275	0.1301	25	21.54%	12.78%	89
		T_0	0.0621	0.2012	2288	0.0501	0.1800	25	23.85%	11.79%	91
	R_{18}	L_0	0.0407	0.1630	2,186	0.0372	0.1536	25	9.53%	6.10%	86
		G_0	0.0361	0.1539	2193	0.0313	0.1426	25	15.30%	7.91%	86
		T_0	0.0557	0.1938	2246	0.0477	0.1797	25	16.80%	7.83%	87
	R_{35}	L_0	0.0315	0.1403	2,192	0.0228	0.1087	26	38.18%	29.04%	85
		G_0	0.0328	0.1451	2,170	0.0265	0.1277	26	23.62%	13.60%	82
		T_0	0.0396	0.1505	2,148	0.0320	0.1161	26	23.79%	29.69%	81



(a) *con-adp* setting w.r.t. MSE (b) *random-adp* setting w.r.t. MSE

Figure 2: Forecast accuracy distributions ($k = 1$).



(a) *con-adp* setting w.r.t. MSE (b) *random-adp* setting w.r.t. MSE

Figure 3: Forecast accuracy distributions ($k = 5$).

Time Overhead. From Table 1, we observe that the time overhead of ADP-LSTM is significantly less than ARIMA. The great gap is due to the following reason. Although AIRMA does not require any offline training, it needs to calculate parameters by fitting a set of historical data per step. Such a model fitting process is more computationally expensive compared to ADP-LSTM which only needs to be tuned by a single newly accessible data per step.

4.1.2 *Network Level Analysis.* To assess the consistency of the forecast results across all ports and routers, we show the distributions of the forecast accuracy results, grouped by local ports, global ports, and terminal ports, as shown in Figure 2. From the figure, we can obtain consistent results with the port level analysis. The ADP-LSTM is more effective than ARIMA, on local ports, global ports, and terminal ports. It further demonstrates the effectiveness of ADP-LSTM as a surrogate model compared to ARIMA.

4.2 Sensitivity Analysis

We investigate the effect of the delay k on forecast accuracy. Due to the page limit, we only show the results when $k = 5$ in Figure 3. First, we observe that some ports have very large MSE e.g., in $cont-adp$ setting, R_8L_2 : 0.6339 and $R_{13}L_2$: 15.5494 for ADP-LSTM; in $random - adp$ setting, $R_{29}T_0$: 81457340.7567 for ARIMA, $R_{14}T_0$: 3.0814 and $R_{23}T_0$: 85.4369 for ADP-LSTM. Second, the proposed ADP-LSTM still outperforms ARIMA when $k = 5$. In addition, compared to Figure 2, the accuracy of ARIMA and ADP-LSTM both diminish. It may be because as k increases, the proportion of actual data in the historical data window decreases, resulting in reduced usage of the actual data for surrogate modeling.

5 CONCLUSION AND FUTURE WORK

In this paper, we investigate the potential of machine learning surrogate models to accelerate PDES, by forecasting the intricate traffic in the Dragonfly network. Several key research questions remain open: (i) how to improve forecasting accuracy by taking into account additional network features and the temporal/spatial correlation among ports in the dragonfly, (ii) how to improve long-term forecast accuracy as the delay step k grows, and (iii) how to integrate and coordinate Python based surrogate modeling with C/C++ based PDES modeling.

Acknowledgement: This work is supported by US National Science Foundation grant CCF-2109316, and the Director, Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-06CH11357 and the "Kronos: Enabling Long Timescale PDES Simulations via Multi-Resolution Methods" project.

REFERENCES

[1] Mohammed S Ahmed and Allen R Cook. 1979. *Analysis of freeway traffic time-series data by using Box-Jenkins techniques.* Number 722.

[2] Abdelhadi Azzouni and Guy Pujolle. 2018. NeuTM: A neural network-based framework for traffic matrix prediction in SDN. In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 1–5.

[3] Christopher D Carothers, David Bauer, and Shawn Pearce. 2002. ROSS: A high-performance, low-memory, modular Time Warp system. *J. Parallel and Distrib. Comput.* 62, 11 (2002), 1648–1669.

[4] Manoel Castro-Neto, Young-Seon Jeong, Myong-Kee Jeong, and Lee D Han. 2009. Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert systems with applications* 36, 3 (2009), 6164–6173.

[5] Di Chai, Leye Wang, and Qiang Yang. 2018. Bike flow prediction with multi-graph convolutional networks. In *Proceedings of the 26th ACM SIGSPATIAL international conference on advances in geographic information systems*. 397–400.

[6] Kit Yan Chan, Tharam S Dillon, Jaipal Singh, and Elizabeth Chang. 2011. Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg–Marquardt algorithm. *IEEE Transactions on Intelligent Transportation Systems* 13, 2 (2011), 644–654.

[7] Xingyi Cheng, Ruiqing Zhang, Jie Zhou, and Wei Xu. 2018. Deeptransport: Learning spatial-temporal dependency for traffic condition forecasting. In *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[8] Mark S Dougherty, Howard R Kirby, and Roger D Boyle. 1993. The use of neural networks to recognise and predict traffic congestion. *Traffic engineering & control* 34, 6 (1993).

[9] Zhixiang He, Chi-Yin Chow, and Jia-Dong Zhang. 2018. STANN: A spatio–temporal attentive neural network for traffic prediction. *IEEE Access* 7 (2018), 4795–4806.

[10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[11] Yoshi Shih-Chieh Huang, Kaven Chun-Kai Chou, and Chung-Ta King. 2012. Application-driven end-to-end traffic predictions for low power NoC design. *IEEE transactions on very large scale integration (VLSI) systems* 21, 2 (2012), 229–238.

[12] Yoshi Shih-Chieh Huang, Kaven Chun-Kai Chou, Chung-Ta King, and Shau-Yin Tseng. 2010. NTPT: On the end-to-end traffic prediction in the on-chip networks. In *Proceedings of the 47th Design Automation Conference*. 449–452.

[13] Weiwei Jiang and Jiayun Luo. 2022. Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications* (2022), 117921.

[14] Xuexiang Jin, Yi Zhang, and Danya Yao. 2007. Simultaneously prediction of network traffic flow based on PCA-SVR. In *Advances in Neural Networks–ISNN 2007: 4th International Symposium on Neural Networks, ISNN 2007, Nanjing, China, June 3-7, 2007, Proceedings, Part II 4*. Springer, 1022–1031.

[15] Yao Kang, Xin Wang, and Zhiling Lan. 2022. Study of Workload Interference with Intelligent Routing on Dragonfly. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis* (Dallas, Texas) *(SC '22)*. Article 20, 14 pages.

[16] John Kim, Wiliam J Dally, Steve Scott, and Dennis Abts. 2008. Technology-driven, highly-scalable dragonfly topology. *ACM SIGARCH Computer Architecture News* 36, 3 (2008), 77–88.

[17] Kyungeun Lee, Moonjung Eo, Euna Jung, Yoonjin Yoon, and Wonjong Rhee. 2021. Short-term traffic prediction with deep neural networks: A survey. *IEEE Access* 9 (2021), 54739–54756.

[18] Sangsoo Lee and Daniel B Fambro. 1999. Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. *Transportation research record* 1678, 1 (1999), 179–188.

[19] Moshe Levin and Yen-Der Tsao. 1980. On forecasting freeway occupancies and volumes (abridgment). *Transportation Research Record* 773 (1980).

[20] Boyang Li, Sudheer Chunduri, Kevin Harms, Yuping Fan, and Zhiling Lan. 2019. The effect of system utilization on application performance variability. In *Proceedings of the 9th International Workshop on Runtime and Operating Systems for Supercomputers*. 11–18.

[21] Xiaolei Ma, Zhimin Tao, Yinhai Wang, Haiyang Yu, and Yunpeng Wang. 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies* 54 (2015), 187–197.

[22] Misbah Mubarak, Christopher D Carothers, Robert B Ross, and Philip Carns. 2017. Enabling parallel simulation of large-scale HPC network systems. *IEEE Transactions on Parallel and Distributed Systems* 28, 1 (2017), 87–100.

[23] Kalyan Perumalla, Maximilian Bremer, Kevin Brown, Cy Chan, Stephan Eidenbenz, K Scott Hemmert, Adolfy Hoisie, Benjamin Newton, James Nutaro, Tomas Oppelstrup, et al. 2022. *Computer Science Research Needs for Parallel Discrete Event Simulation (PDES).* Technical Report. Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States).

[24] David Alexander Tedjopurnomo, Zhifeng Bao, Baihua Zheng, Farhana Murtaza Choudhury, and Alex Kai Qin. 2020. A survey on modern deep neural network for traffic prediction: Trends, methods and challenges. *IEEE Transactions on Knowledge and Data Engineering* 34, 4 (2020), 1544–1561.

[25] Igor V Tetko, David J Livingstone, and Alexander I Luik. 1995. Neural network studies. 1. Comparison of overfitting and overtraining. *Journal of chemical information and computer sciences* 35, 5 (1995), 826–833.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[27] Xin Wang, Misbah Mubarak, Yao Kang, Robert B Ross, and Zhiling Lan. 2020. Union: An automatic workload manager for accelerating network simulation. In *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 821–830.

[28] Xin Wang, Misbah Mubarak, Xu Yang, Robert B Ross, and Zhiling Lan. 2018. Trade-off study of localizing communication and balancing network traffic on a dragonfly system. In *2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 1113–1122.

[29] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. 2022. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125* (2022).

[30] Billy M Williams. 2001. Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling. *Transportation Research Record* 1776, 1 (2001), 194–200.

[31] Billy M Williams and Lester A Hoel. 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of transportation engineering* 129, 6 (2003), 664–672.

[32] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems* 34 (2021), 22419–22430.

[33] Yuankai Wu, Huachun Tan, Lingqiao Qin, Bin Ran, and Zhuxi Jiang. 2018. A hybrid deep learning based traffic flow prediction method and its understanding. *Transportation Research Part C: Emerging Technologies* 90 (2018), 166–180.

[34] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 753–763.

[35] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, and Zhenhui Li. 2019. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5668–5675.

[36] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, Yanwei Yu, and Zhenhui Li. 2018. Modeling spatial-temporal dynamics for traffic prediction. *arXiv preprint arXiv:1803.01254* 1 (2018), 9.

[37] Haitao Yuan and Guoliang Li. 2021. A survey of traffic prediction: from spatio-temporal data to intelligent transportation. *Data Science and Engineering* 6 (2021), 63–85.

[38] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2022. Are transformers effective for time series forecasting? *arXiv preprint arXiv:2205.13504* (2022).

[39] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-first AAAI conference on artificial intelligence*.

[40] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. 2019. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE transactions on intelligent transportation systems* 21, 9 (2019), 3848–3858.

[41] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 11106–11115.

[42] Li Zhou and Avinash Karanth Kodi. 2013. Probe: Prediction-based optical bandwidth scaling for energy-efficient nocs. In *2013 Seventh IEEE/ACM International Symposium on Networks-on-Chip (NoCS)*. IEEE, 1–8.

[43] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*. PMLR, 27268–27286.

[44] Jiawei Zhu, Xing Han, Hanhan Deng, Chao Tao, Ling Zhao, Pu Wang, Tao Lin, and Haifeng Li. 2022. KST-GCN: A knowledge-driven spatial-temporal graph convolutional network for traffic forecasting. *IEEE Transactions on Intelligent Transportation Systems* 23, 9 (2022), 15055–15065.