Multiple-Step-Ahead Traffic Prediction in High-Speed Networks

Abdolkhalegh Bayati, Kim Khoa Nguyen, and Mohamed Cheriet

***Abstract*— Traffic in high-speed networks shows distinct patterns at different timescales. This characteristic should be taken into account to address the error propagation in the multiple-step-ahead traffic prediction. Based on this idea, we proposed an algorithm in which traffic is modeled at different timescales using Gaussian process regression (GPR). The prediction at a timescale is made using the data of that timescale as well as the prediction results at larger timescales. Experiments performed on two public traffic data sets show that our algorithm has lower error propagation than other algorithms, including ARIMA, FARIMA, LSTM, and Convolutional LSTM.**

***Index Terms*— Communication system traffic, traffic prediction, time-series prediction, Gaussian process regression.**

I. Introduction

In multiple-step-ahead traffic prediction, the goal is to predict the traffic performance measures (e.g., bandwidth, packet loss, and latency) forward in time, up to a particular horizon. A long horizon reveals traffic fluctuations in future steps and gives enough time to take proper decisions. However, it may also lead to traffic fluctuation in future steps. In the high-speed network management, there are many situations in which immediate changes in the network are expensive or not feasible. For example, the time required for establishing a wavelength (or lambda) in optical networks is often in the order of minutes, so it cannot be done instantly. Multiple-step-ahead traffic forecasting provides sufficient time for proactive management in such situations.

Two common strategies for multiple-step-ahead time-series prediction are *iterative* (or naive) and *direct* (or parallel) methods [1]. In the iterative approach, multiple-step-ahead time-series forecasting is achieved by making a repeated one-step-ahead prediction where the outputs in consecutive steps are used as the input for the next forecasting step [2]. The recursive one-step-ahead prediction can be repeated up to a required *time horizon*, and only a single forecasting model is used [3]. As its main drawback, the accumulation of prediction errors in the prior steps raises as the forecast horizon increases. On the other hand, in the direct approach, H different models are trained in parallel to perform H-step-ahead time-series forecasting where the learner h $(1 \leq h \leq H)$ predicts the h-th step-ahead value. The direct method requires H separate models to be trained and its time horizon is limited to H.

Our experimental results reveal that the traditional strategies fail to achieve an accurate prediction of high-speed traffics because they do not consider the traffic characteristics. Indeed, traffic of a high-speed link exhibits different patterns at different time-scales [4]. For example, at the small time-scales, traffic behavior is protocol dependent (e.g., TCP, UDP, HTTP) [5]. On the other hand, at the time-scale of hours, traffic samples represent the humans' daily activities. Also, traffic exhibits Long-Range Dependency (LRD) at large time-scales, but it has Short-Range Dependency (SRD) at small time-scales [5]. The traffic behavior at a time-scale is determined by a particular set of factors corresponding to that time-scale. This traffic characteristic is beneficial for multiple-step-ahead prediction. Consider three traffic samples which have been captured at time-steps 7:00 AM, 7:05 AM, and 8:00 AM when the sampling time-scale is 5 minutes. Now consider the factors that define the traffic behavior at the time-scale of 1 hour (e.g., humans' activity). The states of these factors are almost the same from 7:00 AM to 7:05 AM, however they change supposedly from 7:00 AM to 8:00 AM. In other words, for one-step-ahead prediction at 7:00 AM, the traffic behavior at higher time-scales can be ignored. However, for 12-step-ahead prediction at 7:00 AM (i.e., to predict 8:00 AM), the models of traffic behavior at higher time-scales are required. This traffic property should be taken into account in prediction algorithm.

Different algorithms have been used for traffic prediction including ARIMA [6], Artificial Neural Network [7], etc. In [8], Gaussian Process Regression (GPR) has been shown to be a powerful tool for single-step traffic prediction which can handle the traffic self-similarity and periodicity. This work enhances [8] by introducing a new multiple-step-ahead traffic prediction algorithm based on GPR. GPR framework allows studying the error propagation in the multiple-step prediction [3]. The proposed algorithm consists of H GPR experts where the expert f^h captures the traffic behavior at time-scale h $(1 \leq h \leq H)$. The prediction results of expert f^h are used to correct the prediction of the models at the smaller time-scales.

The algorithm is explained in Section II. Sections III and IV present the results and conclusion respectively. Throughout this letter, subscripts denote the index of variables (in vectors), and superscripts determine the time-scale. Also, the vectors are presented using boldface variables.

II. Algorithm

A. Gaussian Process Regression (GPR)

A Gaussian process [8] is a set of random variables such that any subset of them has a joint Gaussian distribution. GPR provides the mapping function between the input $\mathbf{X} = \{\mathbf{x}_i\}$ and (continuous) output $\mathbf{Y} = \{y_i\}$. Given the traffic samples $\mathbf{t} = \{t_i | \ 0 \leq i < n\}$, the feature vector $\mathbf{x}_i$ is a d dimensional vector created from time-series data

Manuscript received September 17, 2018; accepted October 2, 2018. Date of publication October 12, 2018; date of current version December 10, 2018. This research receives support from the Canada Research Chair, Tier 1, hold by Mohamed Cheriet. The associate editor coordinating the review of this letter and approving it for publication was D. Ciuonzo. *(Corresponding author: Abdolkhalegh Bayati.)*

The authors are with the École de Technologie Supérieure, University of Quebec, Montreal, QC H3C 1K3, Canada (e-mail: abdolkhalegh.bayati@synchromedia.ca; knguyen@synchromedia.ca; mohamed.cheriet@synchromedia.ca).

Digital Object Identifier 10.1109/LCOMM.2018.2875747

(i.e., $\mathbf{x}_i = [t_{i-1}, t_{i-2}, \ldots, t_{i-d}]$), and $y_i = t_i$. Consider n pairs of input and noisy output observations, $\mathcal{D} = \{(\mathbf{x}_i, y_i) | i = 1, 2, \ldots, n\}$, and the unknown mapping function $f(\mathbf{x}_i)$:

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \quad (1)$$

where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ is the independent Gaussian noise, and the function $f(\mathbf{X}) \sim \mathcal{GP}(m(\mathbf{X}),\ k(\mathbf{x}_i, \mathbf{x}_j; \theta))$ is defined with mean $m(\mathbf{X})$, covariance $k(\mathbf{x}_i, \mathbf{x}_j; \theta)$, and hyperparameters θ.

The goal is the prediction of target value y_* for new input data $\mathbf{x}_*$ which does not belong to the dataset $\mathcal{D}$. The GP assumption implies that joint distribution of the observed target values $\mathbf{Y}$ and the function value at $\mathbf{x}_*$ is a Gaussian distribution:

$$\begin{bmatrix} \mathbf{Y} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K + \sigma^2 I & K_* \\ K_*^\top & k_* \end{bmatrix}\right), \quad (2)$$

where $f_* = f(\mathbf{x}_*)$ and $k_* = k(\mathbf{x}_*, \mathbf{x}_*; \theta)$. The element K is called *covariance matrix* of $\mathbf{X}$ and denotes a $n \times n$ matrix of the covariance values evaluated for all pairs in the input data, i.e., $[K]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j;\ \theta)$. The element K_* is a $n \times 1$ matrix for which $[K_*]_i = k(\mathbf{x}_i, \mathbf{x}_*; \theta)$. The conditional distribution $f_* | D, \mathbf{x}_*, \theta \sim \mathcal{N}\left(\hat{f}_*, \hat{v}_*\right)$ leads to the mean and variance:

$$\hat{f}_* = K_*^\top (K + \sigma^2 I)^{-1} \mathbf{Y}, \quad (3)$$

$$\hat{v}_* = k_* - K_*^\top (K + \sigma^2 I)^{-1} K_*. \quad (4)$$

In this work, $\mathbf{t}$ is defined as the first difference of the bandwidth time-series:

$$t_i = b_i - b_{i-1}, \quad (5)$$

where b_i is the traffic bandwidth (measured in Bps) at time i. So, each traffic sample shows the difference of the monitored traffic bandwidth at two consecutive intervals. As the difference operator in Equation (5) removes trends in the traffic time-series, the mean function $m(\mathbf{X})$ can be taken to be zero.

B. Multiple-Step-Ahead Prediction

Our algorithm analyzes the traffic data at H time-scales and builds a GPR model for each time-scale. The traffic samples at each time-scale are calculated using the aggregate operation. The *aggregated process* of $\mathbf{t} = \{t_i | \ 0 \le i < n\}$ at the aggregation level h is called $\mathbf{t}^h = \{t_i^h | \ 0 \le i < n_h\}$ which is calculated by partitioning $\mathbf{t}$ into non-overlapping blocks of size h and calculating the sum of the blocks [5]:

$$t_i^h = \sum_{j=ih+1}^{(i+1)h} t_j, \quad (6)$$

$$n_h = \left\lfloor \frac{n}{h} \right\rfloor. \quad (7)$$

Aggregated process $\mathbf{t}^1$ is the same as the original process $\mathbf{t}$.

1) Training: Using aggregated versions of the traffic, H datasets are created where the dataset $\mathcal{D}^h = \{(\mathbf{x}_i^h, y_i^h)\}$ is employed to train the GPR model f^h. The $(d+1)$-dimensional feature vector and the output for $\mathcal{D}^h$ $(1 \le h < H)$ are:

$$\mathbf{x}_i^h = [w_i^h, t_{i-1}^h, t_{i-2}^h, \ldots, t_{i-d}^h], \quad (8)$$

$$y_i^h = t_i^h, \quad (9)$$

$$w_i^h = \sum_{j=ih+1}^{(i+1)h+1} t_j, \quad (10)$$

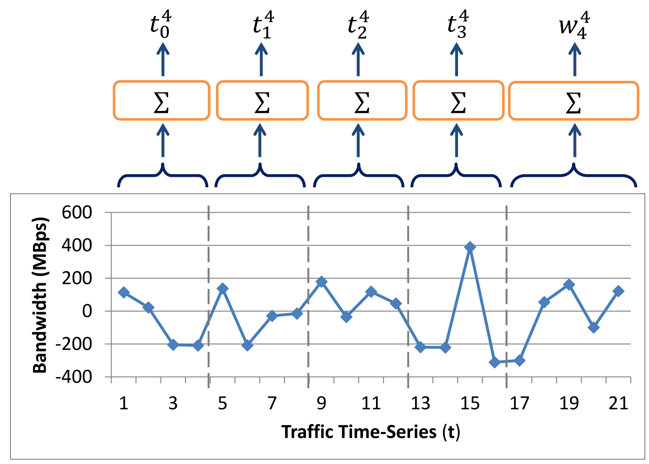

Fig. 1. Traffic aggregation at level 4. The time-series $\mathbf{t}$ is the first difference of the traffic bandwidth. The time interval between samples is 5 minutes.

where w_i^h is the next step of the traffic at aggregation level $h+1$. Figure 1 illustrates the traffic aggregation and the feature selection at aggregation level 4. The only exception is the dataset $\mathcal{D}^H$ for which:

$$\mathbf{x}_i^H = [t_{i-1}^H, t_{i-2}^H, \ldots, t_{i-d}^H] \quad (11)$$

which does not include any sample from the higher aggregation level.

For creating the dataset $\mathcal{D}^h$ with size n, $(n+d-1) \times h$ traffic samples are required. Since $\mathcal{D}^H$ requires the maximum number of traffic samples, the total number of traffic samples needed for creating the training data is $(n+d-1) \times H$.

The time complexity of standard GPR algorithm is $O(n^3)$ where n is the number of the training samples [3]. The proposed algorithm consists of H GPR experts that are trained separately; so, its time complexity is $O(H \times n^3)$. Since, H is a constant and $H \ll n$, the complexity of the proposed algorithm is the same as standard GPR.

2) Prediction: The prediction phase includes two steps as presented in Figure 2: (i) single-step prediction at all the aggregation levels, and (ii) iterative predictions using f^1.

In step 1, model f^h predicts value y_*^h given the input $\mathbf{x}_*^h$. However, the input $\mathbf{x}_*^h$ includes w_*^h which is unknown at the time of prediction. Considering Equations (8) and (10), w_*^h and t_*^{h+1} include the sum of the same traffic samples t_i at the prediction time, so they have the same value. According to Equation (9), y_*^{h+1} is the predicted value for t_*^{h+1}. Therefore, the value w_*^h can be estimated as:

$$w_*^h = y_*^{h+1}. \quad (12)$$

The feature vectors in $\mathcal{D}^H$ do not include w_i^H. Thus, the model f^H does not require the prediction results at the higher aggregation level, and its prediction is based on only the samples from aggregation level H. Model f^H provides y_*^H, the single-step-ahead prediction at aggregation level H. This predicted value is used instead of w_*^{H-1} in $\mathbf{x}_*^{H-1}$ which is the input to f^{H-1} to predict y_*^{H-1}. This process is repeated in step 1 from model f^{H-1} to f^1. The results is $\mathbf{Y}_* = \{y_*^h \mid h = 1, 2, \ldots, H\}$ which is utilized in step 2.

In step 2, the multiple-step-ahead predictions are achieved by iterative single-step estimations using f^1. For t_{s+h}, the

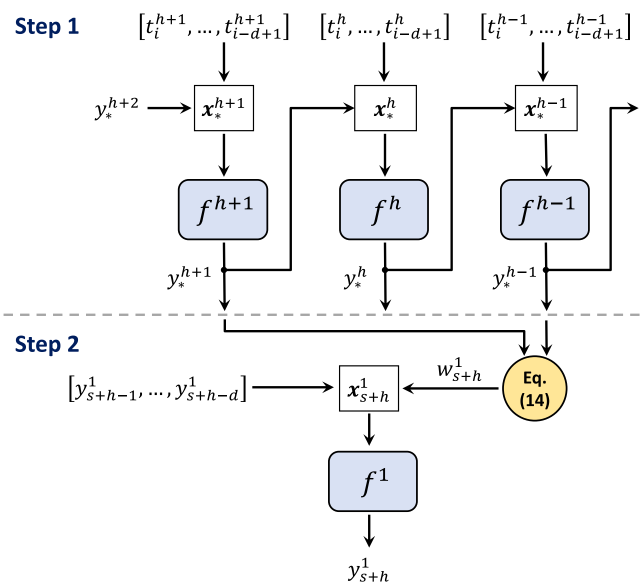

Fig. 2. The workflow model of the prediction algorithm.

$h-1$ previous predicted values and $\mathbf{Y}_*$ are required. The feature vector $\mathbf{x}_{s+h}^1$ is defined as:

$$\mathbf{x}_{s+h}^1 = [w_{s+h}^1, y_{s+h-1}^1, y_{s+h-2}^1, \dots, y_{s+h-d}^1], \tag{13}$$

where w_{s+h}^1 is estimated based on the values in $\mathbf{Y}_*$:

$$w_{s+h}^1 = y_*^{h+1} - y_*^{h-1}. \tag{14}$$

The value w_{s+h}^1 has a crucial role in the algorithm. It conveys the knowledge about the traffic behavior at longer time-scales to f^1. In the following section, the importance and role of w_{s+h}^1 have been analyzed.

C. Importance of w_{s+h}^l in the Feature Vector

The features in $\mathbf{x}_{s+h}^1$ do not contribute evenly to the prediction results. In a machine learning problem, *feature importance* measures the role of a feature in the prediction accuracy. Generally, the correlation between a feature and the target value in a prediction problem reveals the feature importance. According to the traffic autocorrelation function (ACF), it can be shown y_{s+h-1}^1 and w_{s+h}^1 are the most important features in $\mathbf{x}_{s+h}^1$. By definition, the traffic ACF satisfies [9]:

$$\rho(r) \sim c r^{-\beta}, \tag{15}$$

where c is a constant. Traffic shows LRD if $0 < \beta < 1$, and SRD if $1 < \beta < 2$. In both cases, ACF decreases as the time lag between traffic samples increases:

$$0 \le r \le q \quad \Rightarrow \quad cov(t_s, t_{s+r}) \ge cov(t_s, t_{s+q}). \tag{16}$$

According to (16), the importance of t_{s+r} as a feature for the prediction of t_s is greater than t_{s+q}. Equation (15) indicates the importance of a feature in $\mathbf{x}_{s+h}^1$ drops exponentially as the time lag between the element and the target value increases.

D. Effect of w_{s+h}^l on the Error Propagation

In the proposed algorithm, the predicted value at time-step h is used as one of the input features for forecasting the next time-steps. So, the error in prediction at time-step h (or the uncertainty in the feature vector) is propagated through the forecasts at next time-steps. It means the error propagation is reduced as the input uncertainty is minimized. This section illustrates w_{s+h}^1 reduces the uncertainty of the input feature vector and thus, lowers the error propagation.

First, the prediction uncertainty has to be formulated which can be done based on the *predictive variance*. Assume $\mathbf{x}_{s+h}^1$ as a random point with distribution $\mathcal{N}(m(\mathbf{x}_{s+h}^1), v(\mathbf{x}_{s+h}^1))$ where $v(\mathbf{x}_{s+h}^1)$ is a $(d+1)\times(d+1)$ matrix. The Gaussian assumption for $\mathbf{x}_{s+h}^1$ allows the analytical approximation for the predictive variance of y_{s+h}^1 [3]:

$$v(y_{s+h}^1) = v_{m(\mathbf{x}_{s+h}^1)} + V_h, \tag{17}$$

$$V_h = Tr\left\{ v(\mathbf{x}_{s+h}^1)\left(\frac{1}{2}\frac{\partial^2 \hat{v}_{\mathbf{x}_{s+h}^1}}{\partial \mathbf{x}_{s+h}^1 \partial \mathbf{x}_{s+h}^{1\,\top}} + \frac{\partial \hat{f}_{\mathbf{x}_{s+h}^1}}{\partial \mathbf{x}_{s+h}^1}\frac{\partial \hat{f}_{\mathbf{x}_{s+h}^1}}{\partial \mathbf{x}_{s+h}^1}^{\top} \right)\right\}. \tag{18}$$

The predictive variance $v(y_{s+h}^1)$ is the sum of two terms (i) $v_{m(\mathbf{x}_{s+h}^1)}$, and (ii) V_h. The first term is the GPR prediction uncertainty for input $m(\mathbf{x}_{s+h}^1)$ shown in (4). This term is larger at the inputs that are not similar (or close) to the training data compared to the point which are nearby the training data. The second term (in predictive variance) contains the variance (or uncertainty) of $\mathbf{x}_{s+h}^1$. This term is equal to zero when the input is not a random point. As the number of random elements in $\mathbf{x}_{s+h}^1$ raises, the value of V_h increases. So, the uncertainty of $\mathbf{x}_{s+h}^1$ is expected to be more than uncertainty of $\mathbf{x}_{s+h-1}^1$. Accordingly, $v(y_{s+h}^1)$ is expected to be greater than $v(y_{s+h-1}^1)$. This shows the errors are accumulated in y_{s+h}^1s as iterative prediction goes further ahead in time.

The level of uncertainty of w_{s+h}^1 is independent of the prediction time-step. Because, w_{s+h}^1 is the result of single-step predictions the second term of its predictive variance is equal to zero. Therefore, employing w_{s+h}^1 as a feature decreases the uncertainty of the feature vector.

III. Experimental Results

We performed our experiments on two well-known traffic datasets: (i) CAIDA Anonymized Internet Traces from 2008 to 2015 [10], and (ii) Abilene Network traffic data from 2007−01−01 to 2007−10−14 [11]. Abilene dataset has been used for prediction on the long time-steps (i.e., 5 minutes), and CAIDA dataset has been used for prediction on short time-steps (i.e., 30 seconds). In the experiment, each dataset has been divided into two non-overlapped subsets. The first subset has been used for the model selection (i.e., selecting the optimal size of the training set, and the optimal number of features d) for each algorithm using a cross-validation process. The second subset has been divided into 100 portions, and each portion has been employed to create a pair of the non-overlapped train and test sets (random train-test split). For each pair, the models have been fitted on the train set and then, evaluated on the test set. The reported results are

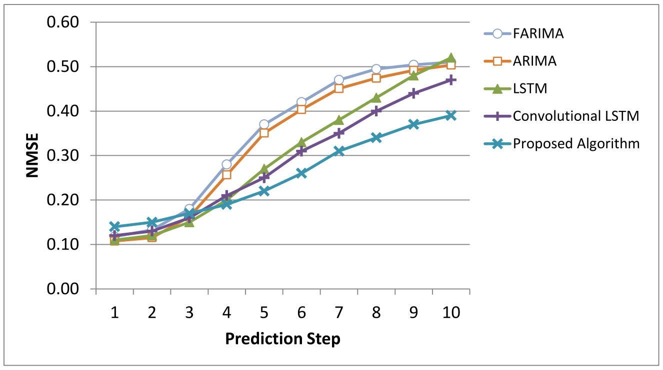

Fig. 3. 10-steps ahead prediction on the Abilene network traffic data (the time lag between steps is 5 minutes).

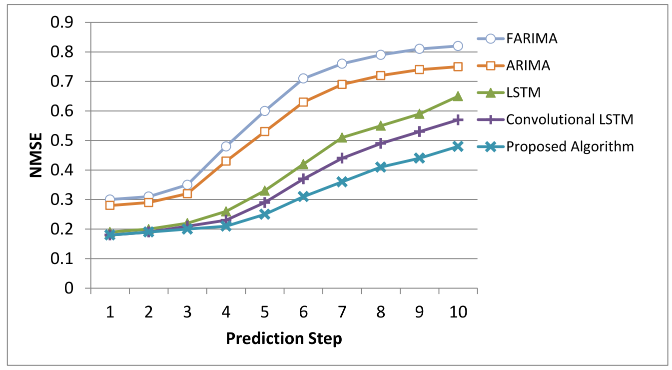

Fig. 4. 10-steps ahead prediction on the CAIDA traffic data (the time lag between steps is 30 seconds).

the average of 100 prediction error measurements which have been achieved from this process.

We compared our model with four multiple-step-ahead prediction algorithms: ARIMA, FARIMA, Long Short-Term Memory (LSTM) network, and Convolutional LSTM network [12]. ARIMA and FARIMA are two well-known time-series models. LSTM and Convolutional LSTM are results of recent advances in the deep learning algorithms. LSTM is a type of Recurrent Neural Networks (RNN), and Convolutional LSTM is based on the fully connected LSTM [12]. We employed the Rational Quadratic covariance function in our model because of its ability to capture traffic characteristics [8]. LSTM and Convolutional LSTM have been implemented as multivariate predictors. The number of units (or neurons) in the LSTM layer, and the order of ARIMA have been determined through the model selection. The order of FARIMA has been determined by maximum likelihood estimation.

The prediction error has been measured based on normalized mean-square error (NMSE):

$$NMSE = \frac{1}{\sigma^2 N} \sum_{i=1}^{N} (t_i^1 - y_i^1)^2 \tag{19}$$

where N is the size of the test set and σ^2 is the variance of $\mathbf{t}$. A value of $NMSE = 0$ corresponds to the perfect predictor.

Figure 3 illustrates the results of 10-steps ahead prediction on traffic data from Abilene network. The time difference between prediction steps is 5 minutes. As shown, the prediction accuracy of the iterative and direct method is less than others. In the first 3 steps, LSTM, Convolutional LSTM, and the proposed algorithm perform almost the same. However, the increase in the prediction error of the proposed algorithm is remarkably less than other models for the next steps.

Figure 4 represents the results of the traffic forecasting at the time-scale of 30 seconds using the traffic data from CAIDA. It is clear that the prediction errors at time-scale 30 seconds are bigger for all the algorithms compared to the prediction at time-scale of 5 minutes. In both cases, the prediction accuracy of the proposed model is higher than any other algorithm.

IV. Conclusion

In this work, we presented an algorithm for multiple-step-ahead traffic prediction based on the GPR in which the multi-scale traffic behavior is exploited to improve traffic prediction and to reduce the error propagation. We analyzed the error propagation in the proposed algorithm and showed that the traffic modeling at higher time-scales is essential and effective for an accurate multiple-step-ahead prediction. In the future work, we will investigate to use the traffic data from the same period of previous days. Also, we will employ the algorithm to improve the resource and traffic management in networks.

References

[1] S. B. Taieb, G. Bontempi, A. Atiya, and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 7067–7083, Jun. 2012.

[2] R. Palm, "Multiple-step-ahead prediction in control systems with Gaussian process models and TS-fuzzy models," *Eng. Appl. Artif. Intell.*, vol. 20, no. 8, pp. 1023–1035, Dec. 2007.

[3] A. Girard, C. E. Rasmussen, J. Q. Candela, and R. M.-Smith, "Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 545–552.

[4] Y. Xie, J. Hu, Y. Xiang, S. Yu, S. Tang, and Y. Wang, "Modeling oscillation behavior of network traffic by nested hidden Markov model with variable state-duration," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 9, pp. 1807–1817, Sep. 2013.

[5] W. Willinger, V. Paxson, R. H. Riedi, and M. S. Taqqu, "Chapter: Long-range dependence and data network traffic," in *Theory and Applications of Long-Range Dependence*. Berlin, Germany: Springer, 2003, pp. 373–407.

[6] B. Zhou, D. He, and Z. Sun, "Traffic predictability based on arima/garch model," in *Proc. 2nd Conf. Next Gener. Int. Design Eng. (NGI)*, Apr. 2006, pp. 207-1–207-8.

[7] Y. Li, H. Liu, W. Yang, D. Hu, X. Wang, and W. Xu, "Predicting inter-data-center network traffic using elephant flow and sublink information," *IEEE Trans. Netw. Serv. Manage.*, vol. 13, no. 4, pp. 782–792, Dec. 2016.

[8] A. Bayati, V. Asghari, K. Nguyen, and M. Cheriet, "Gaussian process regression based traffic modeling and prediction in high-speed networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–7.

[9] T. Karagiannis, M. Molle, and M. Faloutsos, "Long-range dependence ten years of Internet traffic modeling," *IEEE Internet Comput.*, vol. 8, no. 5, pp. 57–64, Oct. 2004.

[10] *The CAIDA UCSD Anonymized Internet Traces 2008–2015*. Accessed: Aug. 5, 2018. [Online]. Available: http://www.caida.org/data/passive/

[11] *Abilene Internet2 Network*. Accessed: Aug. 5, 2018. [Online]. Available: http://noc.net.internet2.edu/i2network/live-network-status/historical-abilene-data.html

[12] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 802–810.