Gaussian Process Regression based Traffic Modeling and Prediction in High-Speed Networks

Abdolkhalegh Bayati, Vahid Asghari, Kim Nguyen, Mohamed Cheriet
École de Technologie Supérieure, University of Quebec, Canada
Email: {Abdolkhalegh.Bayati, vrasghari, knguyen, Mohamed.Cheriet}@synchromedia.ca

***Abstract*—Evolving nature of network traffic challenges existing models to fit and predict its behavior. In particular, real traffic modeling requires more flexible design that can adapt to long-range and short-range dependent traffic with dynamic patterns. Unfortunately, existing models cannot handle such requirements because various traffic behaviors such as periodic and self-similar are not taken into account. In this paper, Gaussian process regression (GPR) is adapted for traffic modeling and prediction. The connection between self-similarity as a traffic characteristic and GPR parameters has been driven and exerted to build of a new Hurst estimation method based on machine learning techniques. This led to propose *self-similar covariance functions* for enhancing prediction accuracy of GPR. The proposed GPR model has been applied for Hurst estimation as well as for traffic prediction on real traffic traces at different time-scales. The experimental results show the employment of self-similar covariance functions increases generalization ability of GPR for traffic modeling and prediction.**

I. Introduction

Network traffic modeling has widely been used for developing traffic characterization [1], performance modeling, traffic generation [2], and etc. An important application of traffic modeling is traffic prediction that plays an important role in quality of service (QoS) enhancement in computer networks. Traffic prediction is an efficient tool for making decisions to improve network performance [3]. For example, it has been proposed to utilize traffic prediction for predictive congestion control [4], proactive resource provisioning and allocation [5], and etc. Traditionally, in this type of applications, decisions are made based on current state of network traffic which result in unstable solutions. To address this issue, traffic forecasting techniques are exploited to reduce risk of decisions by using predicted state of network traffic instead of its current state.

An appropriate traffic prediction method must response to several requirements in order to be suitable for working in a real network environment. Firstly, it must be highly accurate to reduce the risk of consequence decisions. Clearly, an inaccurate prediction algorithm deteriorates the overall performance of the network and reduces resource utilization. The second is time constraint. In computer networks, decisions must be made almost in real-time. Another requirement is the amount of computational resources required for the prediction algorithm. Traffic predictors are usually implemented in network devices such as routers and switches which are limited in available computational resources (e.g. CPU, memory). All these restrictions make traffic prediction a challenging subject for researchers.

A traffic predictor estimates next values of a time series corresponding to network traffic given a history of previous samples. Traffic samples can be a sequence of instances of packets arrival times, or the number of arriving packets (or the total GG of bytes) in successive, non-overlapping time intervals of unit length (e.g. millisecond, second, etc). In this work, we focused on the prediction of load on high-speed links. Let $X = \{X(t_0), X(t_1), X(t_2), ...\}$ byte count process where $X(t_i)$ describes the total number of bytes in the time interval t_i. Whereas process X represents a traffic trace, it might have a highly evolving behavior since it includes different patterns at different times, links, and various time-scales. Meanwhile, dominant characteristics of traffic process such as self-similarity and long-range (short-range) dependency are definitely associated with its *autocorrelation* function [6]. These two clues led us to Gaussian Process (GP) framework [7] as a kernel-based method which allows to use particular kernels for treating certain patterns and, simultaneously enables direct access and manipulation of process autocorrelation using covariance functions. Hence, we used GPR framework to model and predict byte count process X. The contributions of our work are: 1) adapting GPR framework for traffic modeling and prediction by proposing self-similar covariance functions; and 2) using machine learning techniques to estimate Hurst parameter as a part of our prediction method.

The remaining of this paper is organized as follow: a brief summary of existing works in this field is presented in section II. Section III defines Gaussian Process Regression and its connection to self-similarity. Experimental results are reported in section IV and finally, in section V conclusions are drawn.

II. Related Works

Network traffic models have been employed for developing traffic prediction algorithms. Particularly, time-series analysis methods are common tools in this context. First examples of such time-series algorithms are autoregressive (AR), autoregressive moving average (ARMA), and autoregressive integrated moving average (ARIMA) models [1]. These models are simple and require only a small number of previous samples, thus are proper choices for traffic prediction. However, the major problem of these models is they cannot capture long-range dependency in real traffic traces [8]. The autocorrelation functions (ACF) of these models decay exponentially which is pretty different from ACF of real network traffic. Indeed, more complexity must be added to this kind of models to handle long-range dependent traffic.

As an effort to extend ARIMA in order to predict bursty and long-range dependent traffics, a single-step traffic prediction model called ARIMA/GARCH has been proposed in [9]. Also, FARIMA model has been employed in [10] for multi-step traffic prediction. They reduced the complexity of fitting procedure in FARIMA and thus reduced its time complexity. The flexibility in describing both types of short-range dependent (SRD) and long-range dependent (LRD) traffic is the main advantage of their method compared to former ARMA model [11]. However, to initialize parameters of their traffic forecasting algorithm, it is required to firstly determine Hurst exponent [6] value of traffic using one of the known Hurst estimation methods (e.g. periodogram, variance of residuals, R/S and etc [8]). Therefore, their performance is highly dependent on the accurate value of Hurst exponent which cannot be calculated definitively and only can be estimated [8]. Unfortunately, even well-known Hurst estimators may still produce conflicting results and there is no consistent robust Hurst estimator [8].

Unlike time-series models which are based on assumptions on the nature of network traffic (e.g. stationarity or self-similarity), artificial neural networks (ANN) do not require a prior knowledge about properties of data [3]. ANN models are widely used for traffic prediction which are able to handle real data characteristics such as non-linearity, non-stationarity, and non-Gaussianity [3]. Among all types of ANNs, feedforward neural network (also called multilayer perceptron) was one of the first tools for developing traffic predictors [3]. The simple implementation and well-understood learning algorithm (i.e. backpropagation) of Multilayer perceptron (MLP) network as well as its approximation capability of unknown function make it an attractive choice for prediction. However, MLP neural network is a very simplified version of biological neuron and it does not capture dynamic features of its biological counterpart, due to utilization of static scalar wights in synapses of MLP networks. Consequently, MLP provides only static mapping between input and outputs and it cannot handle temporal behavior of real data [3].

To address disadvantages of MLP traffic models, two different approaches have been followed in recent works. The first approach focuses on the ANN architectures and their combinations. For example, adaptive finite impulse response (FIR) linear filters have been used instead of static weights in MLP neural networks to introduce time delays into the synaptic structure of the ANNs [3]. FIR-based neural networks (FIRNN) use a modified version of backpropagation algorithm called temporal backpropagation to determine weights of FIR filters [3]. It has been reported that FIR neural networks gain better performance in time-series prediction compared to standard recurrent neural networks, linear predictors, Wiener filters, and MLPs [3]. In [12] an ANN architecture is proposed combining two individual ANNs to enhance prediction accuracy. They observed their architecture outperforms autoregressive (AR) model in one-step network traffic prediction with regards to four types of traffic (i.e. MPEG, and JPEG video, Ethernet and Internet traffic traces).

The second approach in ANN traffic prediction targets improving learning algorithms instead of working on the architecture. A popular example is multiresolution learning which significantly improves neural network's generalization and robustness [13] [3]. In fact, wavelet-domain methods can reveal traffic features that have direct network interpretation (e.g. round-trip time) [6]. Multiresolution learning paradigm and FIR neural network have been exploited to develop a time-series prediction algorithm in [3]. Their results showed that multiresolution learning paradigm improves neural network prediction capability. Performance of different ANN architectures and different learning algorithms for network traffic prediction have been studied in [13]. They applied the Real Time Recurrent Learning (RTRL) and Extended Kalman Filter (EKF) based training algorithm on MLP and also Radial Basis Function (RBF) and recurrent networks and compared their performance.

Although the ANN approaches efficiently handle nonlinear, nonstationary dynamic traffic traces, their main disadvantage is the large number of training samples they require (e.g. 1000 sample in the case of [3] and more than 1000 samples in the case of [12]). In fact, this is not a major problem when the time interval is small. But when the time scale is around second or more, this drawback leads to time consuming training phase and reduces applicability of ANN in real-time traffic prediction. Another disadvantage of ANN predictors is they are analytically intractable.

In addition to aforementioned approaches, there exist other algorithms based on the Support Vector Regression or SVR (e.g., see [14]), fuzzy models (e.g., [4]), signal processing methods and multi-resolution analysis (e.g., [15]), and etc. A complete survey is beyond the scope of this paper.

III. Gaussian Process Regression Framework

Gaussian Process Regression (GPR) [7] is a supervised machine learning technique that provides mapping function between input and (continuous) output data. Gaussian process framework has been used in different areas extended from classification to regression problems including: rural traffic prediction, time-varying systems [16] and etc. Consider n pairs of input and noisy output observations, $\mathcal{D} = \{(t_i, X(t_i)) | i = 0, 1, 2, ..., n-1\}$, and an unknown mapping function $f(t_i)$ while:

$$X(t_i) = f(t_i) + \varepsilon_i, \tag{1}$$

where ε_i is independent Gaussian noise with zero mean and variance σ^2, i.e. $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. In GPR framework, the mapping function is assumed to be a *stochastic Gaussian process* (GP) [7]. According to the definition, a Gaussian process (GP) is a set of random variables that any subset of them has joint Gaussian distribution. Gaussian process $f(t)$ is completely specified by its mean and covariance functions:

$$f(t) \sim \mathcal{GP}(m(t), k(t_i, t_j; \theta)), \tag{2}$$

where $m(t)$ is the mean function and $k(t_i, t_j; \theta)$ is the arbitrary covariance function and θ is the set of hyperparameters. Without loss of generality, we can assume that the mean function is equal to zero for stationary processes (i.e., $m(t) = 0$). Covariance function $k(t_i, t_j; \theta)$ plays an important role in GPR framework which will be discussed in the remaining of this section. According to Equation (1) and GP prior distribution of mapping function $f(t)$, the covariance function of target values $X = \{X(t_i),\ i = 0, 1, ..., n-1\}$ can be formulated as:

$$k_X(t_i, t_j; \theta) = k(t_i, t_j; \theta) + \sigma^2 \delta_{ij}, \tag{3}$$

where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ in other cases.

In a regression problem, the goal is prediction of the target $X(t_*)$ for new input data t_* which does not belong to the dataset $\mathcal{D}$. To achieve this, GPR framework uses the GP prior distribution on the mapping function $f(t)$ as well as the knowledge provided by dataset $\mathcal{D}$ to draw the posterior distribution over the mapping function and then to make inferences about the conditional distribution of the function value at t_*. The GP assumption implies that joint distribution of the observed target values X and the function value at t_* is a Gaussian distribution [7]:

$$\begin{bmatrix} X \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0,\ \begin{bmatrix} K + \sigma^2 I & K_* \\ K_*^\top & k(t_*, t_*; \theta) \end{bmatrix}\right), \tag{4}$$

where $f_* = f(t_*)$ and $t = \{t_i,\ i = 0, 1, ..., n-1\}$ is the input process in dataset $\mathcal{D}$. The element K is called *covariance matrix* of input data X and denotes a $n \times n$ matrix of covariances evaluated for all pairs in the input process, i.e., $[K]_{ij} = k_X(t_i, t_j; \theta)$, and the element K_* is $n \times 1$ matrix whose ith element is calculated as $[K_*]_{i,1} = k(t_i, t_*; \theta)$.

The conditional distribution of the function value f_* will be:

$$\begin{aligned} f_* | t, X, t_*, \theta &\sim \mathcal{N}\left(\bar{f}_*,\ \ cov(f_*)\right), & (5) \\ \bar{f}_* &= K_*^\top (K + \sigma^2 I)^{-1} X, & (6) \\ cov(f_*) &= k(t_*, t_*; \theta) - K_*^\top (K + \sigma^2 I)^{-1} K_*. & (7) \end{aligned}$$

Equation (6) provides predicted value for the f_* and is called mean predictor. Also, Equation (7) is an estimate of its variance. Clearly, the prediction given in Equations (6) and (7) depend on the covariance function $k(t_i, t_j; \theta)$ and noise variance (or noise power value) σ^2.

A. Model selection

In each GP-based modeling, the selected covariance function has to be able to properly reflect features of data in that particular context. In this framework, covariance function $k(t_i, t_j; \theta)$ is a positive semidefinite (PSD) function that determines the nearness or similarity of input pairs t_i and t_j. The accuracy of a model highly depends on the selected covariance function. Selecting appropriate covariance function is known as *model selection* problem. Despite existence of numerous number of covariance functions and the infinite number of their combinations, it is not possible to select the best one considering all of them in the model selection phase. In fact, we first have to select a small number of covariance functions which are better for traffic modeling and prediction.

In this work, our selected model has to be able to reflect two important behavior of traffic: *self-similarity*, and *periodicity*. We exploited stationary covariance functions for this purpose because based on our assumption, byte count process X_t is a stationary process especially in long duration time-scales (e.g. 5 minutes or longer). According to the definition, a stationary process is a function of $r = t_i - t_j$. Therefore, covariance function of such a stationary process can be denoted as $k(r; \theta)$ instead of $k(t_i, t_j; \theta)$.

1) Self-similarity: Network traffic exhibits self-similar or fractal-like behavior [6]. That is, the network traffic looks statistically similar at different time scales. The major reason of self-similarity in network traffic is heavy-tailed distribution corresponding to large file transfer over the network [8]. By definition, a stochastic process $X = (X_0, X_1, X_2, ...)$ is called self-similar with self-similarity parameter H if [17]:

$$X \overset{dis}{=} m^{1-H} X^{(m)}, \quad \forall m, \quad m \in \mathbb{N} \tag{8}$$

where $\overset{dis}{=}$ means equality in the sense of finite-dimensional distribution and $X^{(m)}$ is the aggregated process of X at aggregation level m:

$$\begin{aligned} X^{(m)} &= (X_0^{(m)}, X_1^{(m)}, X_2^{(m)}, ...), & (9) \\ X_i^{(m)} &= \frac{1}{m}(X_{im-m+1} + ... + X_{im}), \forall i \in \mathbb{Z}, i >= 0. \end{aligned}$$

The scalar parameter $H \in (0, 1)$ is called Hurst exponent and it quantifies level of self-similarity. Calculating Hurst parameter is not straightforward, it can only be estimated [8].

In addition to self-similarity, we also need to mention the define long-range dependency (LRD) and short-range dependency (SRD). By definition, a stationary process $X = (X_0, X_1, X_2, ...)$ is called a long-range dependent process if its autocorrelation function (ACF) $\rho(r)$ satisfies [8]:

$$\rho(r) \sim c r^{-\beta}, \tag{10}$$

for parameter $0 < \beta < 1$ where c is a constant (here, two functions $f(t)$ and $h(t)$ assumed to be approximately equal ($f(t) \sim h(t)$) if $f(t)/h(t) \to 1$ when $t \to \infty$). Long-range dependency means a process exhibits significant correlations across large time scales. Also a process is short-range dependent (SRD) if $1 < \beta < 2$ which implies a summable ACF [17].

LRD (or SRD) corresponds to the shape of autocorrelation function while self-similarity evaluates preserving statistical properties among different time scales. It has been proved that the following linear relationship holds between Hurst exponent H of a *stationary* self-similar process and parameter β of a LRD/SRD process [17]:

$$H = 1 - \frac{\beta}{2}. \tag{11}$$

For this reason, these two characteristics (self-similarity and LRD/SRD) might have been assumed the same.

The goal of our traffic modeling is capturing both types of LRD and SRD traffic. This improves generalization of proposed model. Our approach for obtaining such a goal is to select covariance functions which can lead to both strong dependent GPs with slowly decaying ACF as well as weak dependent GPs with fast decaying ACF by changing values of their hyperparameters. AFC of a LRD (or SRD) process must satisfy the condition given in (10) for certain values of β. Since the relation between covariance function and ACF is:

$$k(r) = \frac{\rho(r)}{k(0)}, \tag{12}$$

the condition given in (10) can be represented as the following condition for covariance functions:

$$k(r) \sim cr^{-\beta}. \tag{13}$$

In order to capture LRD or SRD, the covariance function in GPR has to satisfy (13) for $\beta \in (0,1)$ or $\beta \in (1,2)$ respectively. A Gaussian process with such a covariance function is able to model a self-similar traffic process X_t. Therefore, we call those covariance functions as *self-similar covariance functions*. An example of self-similar covariance functions is rational quadratic covariance function which is given by:

$$k_{RQ}(r;l,\alpha) = s^2.\Big(1+\tfrac{r^2}{2\alpha l^2}\Big)^{-\alpha},\ \alpha>0, l>0, \tag{14}$$

where s is variance parameter, l is known as length-scale parameter and α is magnitude parameter (or shape parameter) [7]. It can be shown that rational quadratic covariance function satisfies (13):

$$\lim_{r\to\infty} \frac{k_{RQ}(r)}{cr^{-\beta}} = 1, \tag{15}$$

iff:

$$\alpha = \frac{\beta}{2}. \tag{16}$$

Regarding (11) and (16) the relation between α, as a parameter of the model, and H, the Hurst parameter of the byte count process X_t, is elicited as:

$$\alpha = 1 - H. \tag{17}$$

This linear equality is very important because it proves that GPR-based modeling (using rational quadratic covariance function or any other self-similar covariance function) can capture traffic dependency and self-similarity while it does not rely on Hurst estimation algorithms. In fact, Hurst estimation is done as a part of training phase (i.e. selecting values of hyperparameters) using machine learning techniques. In compare to those previous regression-based models which relied on Hurst estimation algorithms (e.g., see [10]), this is a significant advantage of our work.

2) Periodicity: Periodicity is the other constantly observable behavior of traffic at different time scales. In short time units, network traffic exhibits periodicity because its behavior is mostly affected by the network and its running protocols and devices in those time intervals (e.g., few seconds or shorter time scales) [6]. On the other hand, users' traffic demands seem to be cyclical with a 24-hour cycle. This phenomenon begets periodic variation in traffic patterns in long time unit [5].

Cyclical behavior of network traffic can be exploited for improving results of the traffic predictor using a periodic covariance function. There are different periodic covariance functions. A prevalent instance is [7]:

$$k_{periodic}(r;l,p) = s^2.exp\left(-\tfrac{2sin^2(\pi r/p)}{l^2}\right), \tag{18}$$

with variance parameter s, period p and length scale parameter l. This periodic covariance function is created using squared exponential (SE) covariance function in u-space where SE kernel is formulated as [7]:

$$k_{SE}(r;l) = s^2.exp\left(-\tfrac{r^2}{2l^2}\right), \tag{19}$$

and function u is:

$$u(r) = \big(cos(r), sin(r)\big). \tag{20}$$

3) Self-similarity and periodicity: Although $k_{periodic}$ captures periodicity, it can be shown that it is not a self-similar covariance function because it cannot satisfy condition in (13). We need a covariance function which handles both traffic characteristics (periodicity and self-similarity) simultaneously. To this end, we propose creating a *semi-periodic self-similar (SPSS)* covariance function by combining previous ones:

$$\begin{aligned} k_{SPSS}(r;\theta) \quad &= \quad k_{periodic}(r;s_1,l_1,p).k_{SE}(r;s_2,l_2) \\ &\quad\quad +k_{RQ}(r;s_3,l_3,\alpha). \end{aligned} \tag{21}$$

where $\theta = \{s_1,l_1,p,s_2,l_2,s_3,l_3,\alpha\}$ is the set of hyperparameters. It can be shown that k_{SPSS} is a PSD and self-similar covariance function. This covariance function can handle periodic self-similar network traffic. For building such a SPSS covariance function, it is possible to use other periodic covariance functions instead of $k_{periodic}$ and also to employ other self-similar covariance functions instead of k_{RQ}. In section IV we will compare results of GPR-based traffic prediction using k_{RQ}, $k_{periodic}$, and k_{SPSS}. Also we compare the performance our model with other traffic predictors.

B. Selecting values of hyperparameters

Each covariance function has a set of parameters θ known as *hyperparameters*. The values of these hyperparameters have great effect on the model accuracy, so they should be selected carefully. We used Bayesian inference to estimate values of hyperparameters. This approach is based on maximizing the likelihood function [7]:

$$p(X|t,\theta) = \frac{1}{(2\pi)^{N/2}|K|^{1/2}} exp\left(-\tfrac{1}{2}X^TK^{-1}X\right) \tag{22}$$

This likelihood function simply obtained by considering $X \sim \mathcal{N}(0, K+\sigma^2 I)$. Equivalently, we can maximize *log-likelihood* function:

$$log\left(p(X|t,\theta)\right) = -\frac{1}{2}X^TK^{-1}X - \frac{1}{2}log(|K|) - \frac{n}{2}log(2\pi) \tag{23}$$

Search direction algorithms (e.g. gradient descent) are appropriate for maximizing log-likelihood function. This requires gradient (i.e. partial derivative with respect to hyperparameters) of log-likelihood function:

$$\frac{\partial}{\partial \theta_i} = \frac{1}{2} X^T K^{-1} \frac{\partial K}{\partial \theta_i} K^{-1} X - \frac{1}{2} tr \left(K^{-1}, \tfrac{\partial K}{\partial \theta_i} \right) \qquad (24)$$

where $tr(A)$ is trace of (square) matrix A.

IV. Experimental Results

Our experimental result is two-fold: 1) validation of relation between Hurst parameter H and parameter α given in (17), and 2) Comparison of the proposed model and previous work using the same benchmark.

A. *Hurst Estimation*

We evaluated the connection between Hurst exponent H and hyperparameter α in (17) as follows. First, we trained our GPR model (employing rational quadratic covariance function) using real traffic traces to estimate value of hyperparameter α. Then, we used well-known Hurst estimation algorithms on the same traffic traces to estimate value of Hurst exponent. Finally, we compared H and $1-\alpha$ to see the relation between them.

The training phase including maximum likelihood estimation (MLE) explained in section III-B uses 1500 traffic load samples in time-scale of 1 second (i.e., each sample represent number of byte transmitted on the link during 1 second) from two different sources. The first set of traces contains *Anonymized Internet Traces* from Center for Applied Internet Data Analysis (CAIDA) [18]. This set consists traffic traces collected from high-speed Internet backbone links at different days of different years (from 2010 to 2014). The average bit rates of those traces are in the range of Gbit/s and their duration is 1 hour. The second set includes traffic traces from Waikato VIII captured at network edge of the University of Waikato [19]. Each trace in the second set contains 24 hour of traffic and their average bit rates are in the range of Mbit/s.

Meanwhile, Hurst estimation algorithms have been applied on the same traffic data. There are two categories of Hurst estimators: time-domain estimators which investigate particular statistical properties in time domain (e.g., R/S, and variance of residuals), and frequency-domain estimators which operate in frequency or wavelet domain (e.g., periodogram, and Whittle) [8]. Regarding existing Hurst estimation methods, the corresponding results are often conflicting and no method is known as the most accurate and robust estimator. However, it has been shown in most cases, frequency-domain estimators (especially Whittle and Periodogram) seem to be more accurate [8]. We used the software package SELFIS to acquire results of different Hurst estimators [8].

Table I shows estimated values for hyperparameter α and Hurst exponent. The first two columns contain estimated values of α and $1-\alpha$ respectively. Other columns illustrate estimated value of H using different Hurst estimators. As shown $1-\alpha$ is close to estimated values of H for most of the traffic sequences, especially to estimation resulted from Whittle. This shows $1-\alpha$ is a good estimate for H. Therefore, the proposed GPR model with self-similar covariance function is able to handle self-similarity of traffic.

TABLE I
Comparison between estimated values for α and H

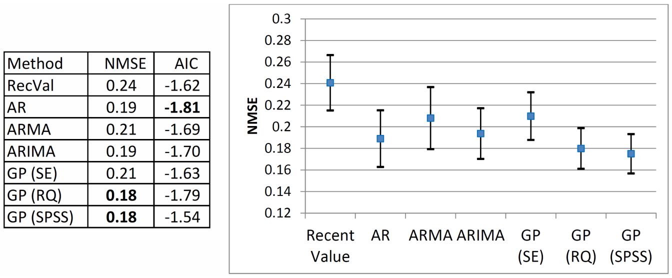
Traffic Data	α	$1-\alpha$	Estimated Hurst exponent (H)			
			R/S	Variance of Residuals	Periodogram	Whittle
CAIDA-01	0.06	0.94	0.79	1.06	0.99	0.97
CAIDA-02	0.13	0.87	0.8	0.97	0.92	0.85
CAIDA-03	0.02	0.98	0.8	1.16	1.13	0.97
CAIDA-04	0.22	0.78	0.77	0.9	0.84	0.87
CAIDA-05	0.15	0.85	0.77	0.97	0.79	0.79
CAIDA-06	0.38	0.62	0.79	0.87	0.74	0.92
CAIDA-07	0.13	0.87	0.82	1.08	0.99	0.88
CAIDA-08	0.2	0.8	0.81	1.0	1.06	0.88
CAIDA-09	0.1	0.9	0.82	1.03	0.98	0.97
CAIDA-10	0.01	0.99	0.7	1.19	1.11	1.0
Waikato-01	0.22	0.78	0.74	0.98	0.9	0.88
Waikato-02	0.11	0.89	0.78	1.04	1.04	0.97
Waikato-03	0.15	0.85	0.77	0.95	0.93	0.96
Waikato-04	0.08	0.92	0.8	1.13	1.13	0.99
Waikato-05	0.15	0.85	0.78	0.97	1.05	0.91
Waikato-06	0.05	0.95	0.82	1.04	0.85	0.87
Waikato-07	0.2	0.8	0.78	0.97	0.94	0.94
Waikato-08	0.13	0.87	0.84	1.0	1.07	0.94
Waikato-09	0.09	0.91	0.81	1.0	1.18	0.97
Waikato-10	0.24	0.76	0.73	0.88	0.74	0.9

B. *Traffic Prediction*

Like well-known regression-based methods, our proposed GPR model has been applied on real traffic traces to compare their prediction accuracy at various time-scales. We used bandwidth traffic traces monitored on different links of Abilene Internet2 Network in 5 minutes intervals over the period from 2005-10-16 to 2007-07-31 [20].

Four different traffic predictors have been employed for comparison. *Recent value prediction* is the simplest prediction method in which the recent observed value is used as the predicted value. The second model is *AR(4)* (i.e. autoregressive model of order 4) in which parameters are estimated using Burg's lattice-based method [11]. *ARMA(4,1)* and *ARIMA(4,1,1)* are other models utilized in this experiment and MLE has been used for estimation of their parameters.

The experiment has been done with more that 30 randomly selected points of the traffic traces from 10 different links. In each run, 100 data samples have been used for training (i.e. estimating model parameters) and 50 data points have been used for testing. The prediction accuracy is defined using two different metrics: Normalized Mean Squared Error (NMSE) [3], and Akaike Information Criterion (AIC) [11]. NMSE (also known as average relative prediction variance) measures



Method	NMSE	AIC
RecVal	0.24	-1.62
AR	0.19	**-1.81**
ARMA	0.21	-1.69
ARIMA	0.19	-1.70
GP (SE)	0.21	-1.63
GP (RQ)	**0.18**	-1.79
GP (SPSS)	**0.18**	-1.54

Fig. 1. Prediction error (NMSE) of different models (time-scale of 5 minutes)

prediction error:

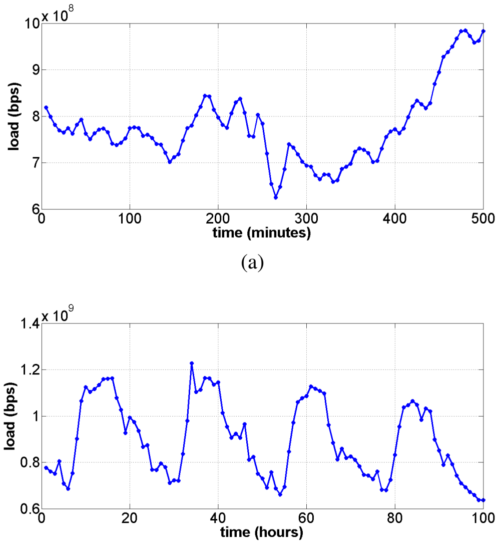
$$NMSE = \frac{1}{\sigma^2 N}\sum_{n=1}^{N}(x_n - \hat{x}_n)^2 \tag{25}$$

in which x_n is the bandwidth value of real traffic and $\hat{x}_n$ is the predicted value, and σ^2 is variance of the real traffic over N samples. A value of $NMSE = 0$ corresponds to the perfect predictor while $NMSE = 1$ is obtained when the average of data samples is used as the predicted value. On the other hand, AIC is a measurement for both prediction error and model complexity [11]:

$$AIC = ln(NMSE) + 2 * \frac{p}{N} \tag{26}$$

where p is number of parameters in the model (i.e. indication of model complexity). The value of AIC is $-\infty$ for a perfect predictor and its value increases when prediction error or number of parameters increase.

Results shown in Figure 1 indicate GPR with self-similar covariance functions have higher prediction accuracy compared to others. However, in the term of AIC, AR has the best result which shows that AR(4) is a simple model (with only four parameters) while GPR (with SPSS covariance function) has 8 hyperparameters. Nevertheless, GPR (with RQ covariance function) has a small error and at the same time its AIC is acceptable. Figure 1 also illustrates variance of NMSE values for GPR method (using self-similar covariance functions) is smaller than others which is an indication of generalization ability and stability of GPR.



(a)

(b)

Fig. 2. Two real traffic sequences with 100 data points monitored at time-scales of (a) 5 minutes and (b) 1 hour

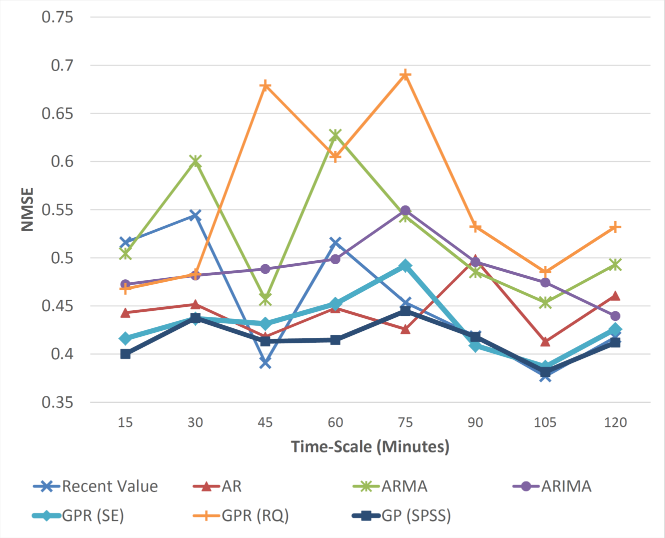


Fig. 3. Prediction error (NMSE) of different models at different time-scales

Traffic patterns are not the same at different time-scales. Figure 2 displays two traffic traces from the same link but monitored at different time-scales. Traffic has periodic pattern at time-scale of 1 hour while it is not periodic at time-scale of 5 minutes. Therefore we have to investigate performance of different models at different time-scales. One model has to be flexible enough to handle different patterns and behaviors to have a proper performance at different time-scales.

Figure 3 presents prediction error of different algorithms at different time-scales. We obtained these results by applying traffic predictors on data from one link at time-scales of 15, 30, 45, 60, 75, 90, 105, and 120 minutes. As seen, some prediction algorithms are not stable as their accuracy varies highly regarding different time-scales. This outcome can be explained through variable traffic behavior at various time intervals depicted in Figure 2. Overall, GPR (with SPSS covariance function) outperforms other algorithms in the terms of NMSE accuracy because its structure combines self-similar and periodic covariance functions. Therefore, it treats both periodic and self-similar patterns in the same time, so it is highly accurate and stable in most of time-scales.

V. Conclusion

Traffic exhibits variable behavior and various patterns in different time-scales analysis which challenges traffic models in making accurate prediction. The Kernel-based model (GPR) we proposed in this paper addresses this issue by capturing each behavior using a particular kernel and employing kernels combination to handle complex behaviors. As its major challenge, this method requires a precise model selection approach (i.e. making decision about number and types of covariance functions). We addressed this issue by proposing self-similar and SPSS covariance functions. Experimental results show that SPSS covariance function which focuses on traffic self-similarity and periodicity, is more stable and outperforms existing models at different time-scales. In future, the proposed model can be extended to a traffic generator considering aforementioned traffic characteristics (i.e., periodicity and self-similarity) and other characteristics such as burstiness.

Acknowledgement

The authors thank NSERC and Ciena for funding the project CRDPJ 461084. This research also receives support from the Canada Research Chair, Tier 1, hold by Mohamed Cheriet.

References

[1] A. Adas, "Traffic models in broadband networks," *Communications Magazine, IEEE*, vol. 35, no. 7, pp. 82–89, Jul 1997.

[2] J. Yin, X. Lu, X. Zhao, H. Chen, and X. Liu, "Burse: A bursty and self-similar workload generator for cloud computing," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 26, no. 3, pp. 668–680, March 2015.

[3] V. Alarcon-Aquino and J. Barria, "Multiresolution fir neural-network-based learning algorithm applied to network traffic prediction," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 36, no. 2, pp. 208–220, March 2006.

[4] B.-S. Chen, S.-C. Peng, and K.-C. Wang, "Traffic modeling, prediction, and congestion control for high-speed networks: a fuzzy ar approach," *Fuzzy Systems, IEEE Transactions on*, vol. 8, no. 5, pp. 491–508, Oct 2000.

[5] B. Krithikaivasan, Y. Zeng, K. Deka, and D. Medhi, "Arch-based traffic forecasting and dynamic bandwidth provisioning for periodically measured nonstationary traffic," *IEEE/ACM Trans. Netw.*, vol. 15, no. 3, pp. 683–696, Jun. 2007.

[6] W. Willinger, V. Paxson, R. Riedi, and M. Taqqu, *Theory and Applications of Long-Range Dependence*, P. Doukhan, G. Oppenheim, and M. Taqqu, Eds. Birkhauser Basel, 2003.

[7] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

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

[9] B. Zhou, D. He, and Z. Sun, "Traffic predictability based on arima/garch model," in *Next Generation Internet Design and Engineering, 2006. NGI '06. 2006 2nd Conference on*, 2006, pp. 8 pp.–207.

[10] Y. Shu, Z. Jin, L. Zhang, L. Wang, and O. Yang, "Traffic prediction using farima models," in *Communications, 1999. ICC '99. 1999 IEEE International Conference on*, vol. 2, 1999, pp. 891–895 vol.2.

[11] P. Brockwell and R. Davis, *Introduction to Time Series and Forecasting*, ser. Springer Texts in Statistics. Springer New York, 2013.

[12] A. Khotanzad and N. Sadek, "Multi-scale high-speed network traffic prediction using combination of neural networks," in *Neural Networks, 2003. Proceedings of the International Joint Conference on*, vol. 2, July 2003, pp. 1071–1075 vol.2.

[13] F. Vieira, V. Costa, and B. Gonalves, "Neural network based approaches for network traffic prediction," in *Artificial Intelligence, Evolutionary Computing and Metaheuristics*. Springer Berlin Heidelberg, 2013, pp. 657–684.

[14] Y. Qian, J. Xia, K. Fu, and R. Zhang, "Network traffic forecasting by support vector machines based on empirical mode decomposition denoising," in *Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on*, April 2012, pp. 3327–3330.

[15] N. Haghighat, H. Kalbkhani, M. G. Shayesteh, and M. Nouri, "Variable bit rate video traffic prediction based on kernel least mean square method," *IET Image Processing*, vol. 9, no. 9, pp. 777–794, 2015.

[16] J. Hu, X. Li, and Y. Ou, "Online gaussian process regression for time-varying manufacturing systems," in *Control Automation Robotics Vision (ICARCV), 2014 13th International Conference on*, Dec 2014, pp. 1118–1123.

[17] B. Tsybakov and N. D. Georganas, "Self-similar processes in communications networks," *Information Theory, IEEE Transactions on*, vol. 44, no. 5, pp. 1713–1725, Sep 1998.

[18] The CAIDA UCSD Anonymized Internet Traces 2010-2014. [Online]. Available: http://www.caida.org/data/passive/passive_2014_dataset.xml

[19] Waikato VIII 2011. [Online]. Available: http://wand.net.nz/wits/waikato/8/

[20] Abilene Internet2 Network. [Online]. Available: http://noc.net.internet2.edu/i2network/live-network-status/historical-abilene-data.html