VoIP Traffic Modelling using Gaussian Mixture Models, Gaussian Processes and Interactive Particle Algorithms

Ana-Maria Simionovici[1], Alexandru Adrian Tantar[1], Pascal Bouvry[1],
Andrei Tchernykh[2], Jorge M. Cortés-Mendoza[2] and Loic Didelot[3]

[1]Faculty of Science, Technology and Communication, University of Luxembourg
6, rue Richard Coudenhove-Kalergi, Luxembourg
Email: {ana.simionovici, alexandru.tantar, pascal.bouvry}@uni.lu
[2]Computer Science Department, CICESE Research Center, Ensenada, Baja California, Mexico
Email: chernykh@cicese.mx, jcortes@cicese.edu.mx
[3]MIXvoip S.a., Z.I. Rohlach, Sandweiler, Luxembourg
Email: ldidelot@mixvoip.com

Abstract—**The paper deals with an important problem in the Voice over IP (VoIP) domain, namely being able to understand and predict the structure of traffic over some given period of time. VoIP traffic has a time variant structure, e.g. due to sudden peaks, daily or weekly moving patterns of activities, which in turn makes prediction difficult. Obtaining insights about the structure and trends of traffic has important implications when dealing with the nowadays cloud-deployed VoIP services. Prediction techniques are applied to anticipate the incoming traffic, for an efficient distribution of the traffic in the system and allocation of resources. The article looks in a critical manner at a series of machine learning techniques. We namely compare and review (using real VoIP data) the results obtained when using a Gaussian Mixture Model (GMM), Gaussian Processes (GP), and an evolutionary-like Interacting Particle Systems based (sampling) algorithm. The experiments consider different setups as to verify the time variant traffic assumption.**

I. INTRODUCTION

Voice over IP (VoIP) is a technology for creating voice communications and multimedia sessions over IP networks. It is widely used as a solution instead of traditional Public Switched Telephone Network (PSTN) systems due to cost reduction, communication improvement, extended functionality, etc. VoIP furthermore allows using different modern phone features, such as voicemails, conferencing, forwarding or music on hold, etc. An important requirement of VoIP providers is to ensure resource availability and to maintain a voice quality level comparable to the one of a traditional telephone system.

As a result of increased popularity of VoIP, voice traffic is becoming complex. To cope with peaks of the requests, overloading, load anomaly, resource availability etc. traffic prediction is beginning to emerge. Understanding the call model and predicting future events is highly important for capacity planning, anomaly detection and intrusion attacks. The arrival date of calls, their origin, destination, and duration are examples of data that can be used to build a prediction model, information usually stored in a Call Detail Records (CDRs) database, which can be used as an input when building such a model.

In this paper, we restrict our study to using an Interactive Particle Systems (IPS) [10] based algorithm, a Gaussian Mixture Model (GMM) [2] and a Gaussian Process (GP) [4]. They are able to provide a flexible modeling approach (IPS), traffic shaping, determined by clients (GMM), and scalable solutions with good prediction precision (GP). GPs approximate distributions obtained by Artificial Neural Network when the number of neurons tends to infinity for some specific cases [1].

GMM is one of the most widely spread clustering and density estimation methods. For the general case, a mixture model is a parametric probability density function represented as a weighted sum of component densities, which optimally fit real unknown distribution of the data. A cluster, in this model, can be mathematically represented by a Gaussian distribution [3]. When it is used for prediction, GMM gives the probability that a new value is generated from one of the Gaussian components that naturally group some data.

GP is another powerfull tool in machine learning statistics and one of the most important approaches for Bayesian learning. It relies on effective methods for placing a prior distribution over a space of functions, generalizing the multivariate Gaussian probability distribution. A GP is described by a mean function and a positive semi-definite covariance function. The use of Gaussian process models for prediction has become very attractive in a wide range of areas and problems, e.g. in geostatistics field.

In [9], the authors present the results of a predictive analysis study on incoming traffic data, with results that allow to outline different traffic patterns in a VoIP environment. With respect to this specific study, we rely on the logs of a real-life VoIP platform in order to test the proposed techniques. It should be mentioned that the data used for the previous study was generated during the delevopment phase of the MIXvoip company. These captures traits describe a small number of users, a limited service as well as only a marginal usage of the hardware resources. Meanwhile, MIXvoip extended its business and its coverage areas. Consequently, the traffic profile is significantly different, from where the need of a more

detailed study and more accurate algorithms, e.g. capable of capturing moving trends or data structure changes.

We hereby propose to extend these results to new prediction model that copes and adapts with the dynamic evolution of requests, anticipates the computation requirements, and scales resources as needed. One of the goals of this study is to minimize the infrastructure costs, while considering cost and energy-aware resource allocations.

The remainder of the paper is organized as follows. In Section 2, related work on using GMM and GP for VoIP is briefly introduced. In Section 3, information regarding the preliminary work conducted within the scope of this research is provided. In Section 4, new prediction model is described, experiments and the obtained results are analyzed. Finally, in Section 5, concluding remarks together with future directions of the work are presented.

II. RELATED WORK

This section surveys a series of previous works in VoIP traffic analysis, namely on the use of GMM classification and GP for modeling and predicting characteristics of VoIP calls. In [5], the authors present a GMM based text-dependent system for speaker identification, with a minor impact on the packet loss rate. Similar discussion of automatic speaker recognition over VoIP can be found in [6]. The authors study codec parameters and compressed packet streams over VoIP considering Probabilistic Stochastic Histogram algorithm with Vector Quantization Probabilistic Stochastic Histogram (VQPSH), and Gaussian Mixture Model Probabilistic Stochastic Histogram (GMMPSH).

The call traffic on VoIP networks under heavy network conditions is modeled as a linear GP in [6]. The authors provide an accurate (predictive) representation of different traffic patterns. The performance of a VoIP system for speech recognition at the receiver level, and a Gaussian algorithm for vector quantization are presented in [7]. The authors use matching Mel-Frequency Cepstral Coefficients features that represent raw speech signal. In [8], a model for call holding times that follow a Generalized Pareto Distribution along with a fractional Gaussian Noise Model for aggregated VoIP traffic are described.

In the previous studies, GMM and GP focus on a variety of models and are used to shape VoIP calls characteristics. These models consider speaker recognition for VoIP transmissions, call duration and call holding times.

The goal of our paper is to present an approach to predict the traffic during a time frame for shaping the future patterns and use it for capacity planning. Different predictors such as, Gaussian Mixture Models, Gaussian Processes including an evolutionary algorithm, Interactive Particle System (IPS) are presented and compared using real life data. We provide a study how prediction models fit to the field and analyze their deviations from real VoIP traffic behavior.

III. MOTIVATION AND PRELIMINARY STUDY

In this section, the motivation to conduct this study together with the preliminary work for traffic shaping and a model of the pattern prediction are presented. Anticipating the load

for a future time frame and proactively scaling the available resources can be used to optimize the distribution of the load and provider cost effectiveness. A VoIP system is composed by different voice nodes that handle incoming and outgoing calls. They communicate with a database, where the phone calls are recorded, as shown in Fig. 1. The Call-Detail-Record database stores information such as destination, prefix, duration, and arrival date. We use it for statistical analysis in order to outline the occurrence of peaks in the system along with a very low usage of the voice nodes.
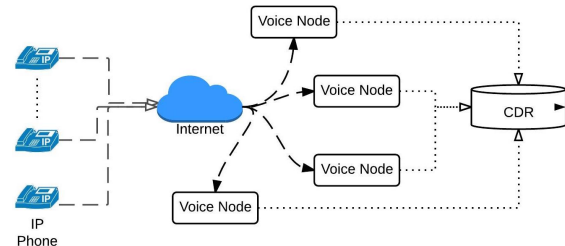

Fig. 1: Basic example of VoIP connectivity.

Due to the fact that, in our case, the VoIP customers are small and medium size companies, the highest load appears during weekdays and working hours. Fig. 2 (a), (b) show two examples of call distribution during working hours and weekdays. The ensuring availability of resources at the peak during rush hours without overprovisioning them becomes challenging.

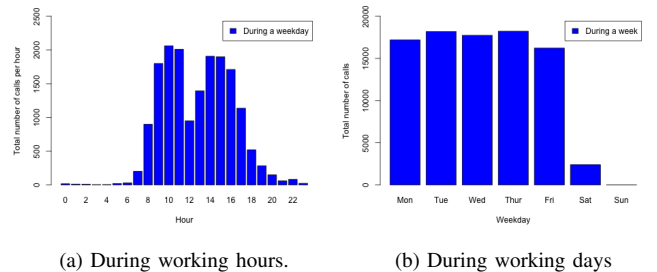(a) During working hours.

(b) During working days

Fig. 2: Phone calls distribution examples.

In [9] is proposed an Interacting Particle System (IPS) based algorithm which handles incoming traffic predictions in a VoIP system. Past observations are used to predict the number of calls arriving in the system during a future time frame. A single Gaussian model [11] is used to estimate parameters of the algorithm. The initial population is composed by particles that encode a mean vector and matrix sampled from a Wishart distribution [12]. The particles evolve according to a transition step. After they are perturbed, the likelihood of each particle is calculated. This step can be seen as the equivalent of mutation in an Evolutionary Algorithm (EA). The perturbed particle is considered for the selection step if the likelihood is improved.

In the current study we extend the previously proposed IPS by modifying the selection method by allowing only a percentage of particles to survive for the future generation. The perturbed particles are sorted by the likelihood values (Algorithm1 - Step3) and those with a better likelihood than $L^{threshold}$ will survive. Otherwise, they are replaced with a particle chosen randomly from the set of the surviving particles during that iteration. In this case, the old one is discarded on

the acceptance/rejection selection phase. This step is repeated and values of the parameters that improve the likelihood are recorded. The traffic for a new time frame is estimated by considering either the particle with the best likelihood or by calculating the weighted sum of the likelihoods of all particles from the final population (Algorithm 1).

The intuitive genetic mutation-selection type mechanism has been used in a diverse range of domains, e.g. rare event simulations, and genetic algorithms. The proposed interactive particle algorithm can be seen as a derived evolutionary algorithm, where mutations and selections are applied without considering crossover methods.

IV. MODELS, EXPERIMENTS AND RESULTS

All methods have been trained on three data sets, each having two time intervals; the dimension of the space is $d = 2$. The training set is a matrix with two columns that represent data in two time intervals. The test set is the data shifted to the time interval that is not used for training (Fig. 3). The output of the algorithms is the predicted number of calls during a future time frame. Different time frames (Table 1) and scenarios with static and dynamic setups for the predictors are considered and presented in the following.

Experimental Setup		
Number of time frames	*Training sets*	*Hours: 10 am/11 am 12 am/ 13 pm ; 20 pm/ 21 pm*
	Test set	*Hours: 10 am/11 am 12 am/ 13 pm ; 20 pm/ 21 pm*
Number of days	*Training sets*	594
	Test set	249
Interactive Particle System	*Number of particles*	1000
	Dimension of space	2
	Number of iterations (perturbation step)	100
Gaussian Mixture Model	*Maximum number of Gaussians for training*	10
	Number of iterations for training	100
Gaussian Processes	*Covariance function (Matrn form)*	@covMaterniso
	Likelihood function (Gaussian)	@likGauss
	Mode	exact
	Number of evaluations to perform	300

TABLE I: Input data set and parameters for IPS, GMM and GP.

A. Static scenario

A number of $N = 1000$ particles are generated using Interactive Particle System and each encodes a vector μ and a matrix Σ. Using the input samples, the likelihood of each particle is calculated (initial population). The perturbation step is applied for a number of times and only a percentage (20%) of particles will survive at each step(mutation - selection step). The final population of the algorithm is used to calculate the prediction by either considering it (weighted likelihood) or by selecting the particle with the maximum likelihood.
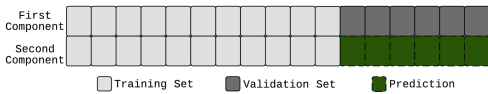

Fig. 3: Data extraction - Training and validation set, values to predict.

Algorithm 1 Interactive Particle System

Step 1 - Initial Population (size N) of Particles
for $i = 1 \to N$ **do**
    Generate $P_i(\mu_i, \Sigma_i)$,
    Where μ_i mean vector of size d, from $\mathcal{N}(0,1)$;
    $W_i(\Sigma, d, n)$, matrix sampled from Wishart distribution with n degrees of freedom
    Calculate likelihood $L_i = (2\pi)^{-N \times d/2} \times |\Sigma|^{-N/2} \times exp\sum_{i=1}^{N} \frac{-(x-\mu)^T \times \Sigma^{-1} \times (x-\mu)}{2}$
end for
return initial population

for $k = 1 \to steps$ **do**
Step 2 - Perturbation, mutation of the encoded parameters
    for $i = 1 \to N$ **do**
Perturb the encoded vector W_i
Perturb μ_i and $\mu_i^{new} \leftarrow \mu + val$, val generated from $\mathcal{N}(0,1)$
Calculate new Sigma, Σ_i^{new}
Calculate L_i^{new}, likelihood with new Σ_i^{new} and μ_i^{new}
        if $L_i^{new} > L_i$ **then**
            $P_i(\mu_i, \Sigma_i) \leftarrow P_i(\mu_i^{new}, \Sigma_i^{new})$
        end if
    end for

Step 3 - Selection of the particles for the next generation
Order particles based on likelihood. Select $L^{threshold}$ of the particle at the position given by percentage of particles to survive.
    for $i = 1 \to N$ **do**
        if $L_i \geq L^{threshold}$ **then**
            add P_i to the list of surviving particles after perturbation
        else
            $P_i(\mu_i, \Sigma_i) \leftarrow P_j(\mu_j, \Sigma_j)$, P_j chosen randomly from the list of the particles that have survived the perturbation.
        end if
    end for
end for
return final population

Step 4 - Prediction methods
Extract final population
if $IPS - MaximumLikelihood$ **then**
    Extract from final population P_j, where $L_j = max(L)$
    Extract μ_j, Σ_j that describe best the data
    Calculate prediction : $Z^\beta | Z^\alpha = z \sim \mathcal{N}(\mu^c, \Sigma^c)$
end if

if $IPS - AverageLikelihood$ **then**
    Use the weighted sum of the likelihood of each particle from final population:
    $$Z^\beta = \frac{\sum_{i=1}^{N} Z^\alpha \times L_i}{Ltotal}$$
end if
return prediction

In a second step, the Gaussian Mixture Model [14] is trained with the same input samples. Formally, a GMM can be described as a weighted sum of M component Gaussian densities as given by the equation:

$$p(x) = \sum_{i=1}^{M} w_i \frac{1}{2\pi^{\frac{D}{2}}|\Sigma_i|^{\frac{1}{2}}} exp\left\{\frac{-1}{2}(x-\mu_i)'\Sigma_i^{-1}(x-\mu_i)\right\}$$

- μ_i, mean;
- covariance matrix Σ_i;
- mixture weight w_i;
- where $\sum_{i=1}^{M} w_i = 1$;
- x a D- dimensional continuous-valued data vector.

We use the Akaike Information Criterion (AIC) score to measure the goodness and identify the best model out of a series of models with an increasing number of analyzed time intervals. The parameters given to the algorithm are the number of components and the maximum number of iterations.

After the training step, the parameters of the GMM are estimated using the maximum likelihood (ML) method. The parameters expected to maximize the likelihood of the GMM, are defined as $p(X|\lambda) = \prod_{t=1}^{T} p(x_t|\lambda)$, where T stands for the sequence of training vectors with λ parameters of the best GMM. At each iteration, the parameters are updated using an expectation-maximization-approach (Algorithm 2).

Algorithm 2 Update for parameters of GMM - EM [15]

---

for $i = 1 \rightarrow iterations$ **do**
$\bar{w}_i = \frac{1}{T}\sum_{t=1}^{T} Pr(i|x_t,\lambda)$, mixture weights;
$\bar{\mu}_i = \frac{\sum_{t=1}^{T} Pr(i|x_t,\lambda)x_t}{\sum_{t=1}^{T} Pr(i|x_t,\lambda)}$, means;
$\bar{\sigma}_i = \frac{\sum_{t=1}^{T} Pr(i|x_t,\lambda)x_t^2}{\sum_{t=1}^{T} Pr(i|x_t,\lambda)} - \bar{\mu}_i$, variances.
end for

We train and use the Gaussian Processes method to predict the traffic over a future time frame. The Gaussian Processes are defined by covariance function K with parameters that have to be estimated. The covariance function of the Gaussian Process is a function that shapes the similarity between two samples with, $k(\mathbf{x},\mathbf{x}') = E[f(\mathbf{x})f^*(\mathbf{x}')]$. For the scope of our study, we have opted for using a Matern kernel function, among others, allowing to infere smoothness from the data. The Matern class is defined by: $k_{Matern}(r) = \frac{2^{\ell-\nu}}{\Gamma(\nu)}(\frac{\sqrt{2\nu r}}{\ell})^\nu K_\nu(\frac{\sqrt{2\nu r}}{\ell})$, with $r = |\mathbf{x}-\mathbf{x}'|$ and positive parameters ν, ℓ, and a modified Bessel function K_v [16]. The algorithm runs with the parameters: training set, test set, number of iterations (300) and estimation of the parameters (exact).

Often used in statistics to express accuracy and compare fitted time series, the Mean Absolute Percentage Error (MAPE) is calculated as: $M = \frac{1}{N}\sum_{i=1}^{N}|\frac{A(i)-R(i)}{A(i)}|$, with actual set A, and the result R. MAPE has no restriction in terms of value on the upper level and it is equal to zero when the fit is perfect. The standard deviation of each classifier is shown in Table II. We identify GP, as the best performing for the first data set (10am - 11am) while GMM performs best on the second data set (12am - 13 pm). For the last set of data considered (20pm - 21pm) IPS-AL performs best.

The results obtained in the validation phase point out an evolving trend inside the data. Thus, a dynamic predictor capable of describing the time-changing structure of the data is required. In addition, in order to (i) assess the validity of our

	H10 am/ H11 am		H12 am/ H13 pm		H20 pm/ H21 pm	
Classifiers	MAPE Value	Standard Deviation	MAPE Value	Standard Deviation	MAPE Value	Standard Deviation
Interactive Particle System Maximum Likelihood	0.2283	0.1227	0.1623	0.0112	0.2441	0.0058
Interactive Particle System Average Likelihood	0.2335	0.1279	0.1644	0.0101	0.2432	0.0053
Gaussian Mixture Model	0.0831	0.0030	0.1527	0.0142	0.3332	0.0274
Gaussian Processes	0.0709	4.2345e-17	0.1544	8.4690e-17	0.2520	1.1292e-16

TABLE II: Mean Absolute Percentage Deviation and Standard Deviation for each classifiers, in average, after 30 runs.

assumption and (ii) construct a rough lower bound predictor, we consider a scenario where the full data set is shuffled before being split into training and validation data. Such a setup provides us an *a priori* knowledge on the number and evolution of calls (at training phase). While such an approach is not applicable in practice, it allows to construct a predictor which better captures the overall structure of the data. We will later use this predictor as a comparison basis for other predictors, when discussing both the static and the dynamic setups.

B. Static scenario with data shuffle

In this scenario the data is shuffled (Fig. 4) before splitting into training and validation set (Fig. 3). The predictors presented in the previous subsection are trained and validated in the same manner (Table 1).
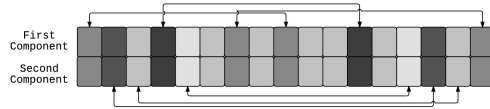

Fig. 4: Data shuffle before splitting into training and validation set.

In Table III, MAPE and the Standard Deviation are presented for this scenario. In this scenario, GMM performs best for two different data sets (10am - 11am and 12pm - 13pm) while the other three predictors (GP, IPS-AL and IPS-ML) give better results on the 20pm - 21pm data set.

	H10 am/ H11 am		H12 am/ H13 pm		H20 pm/ H21 pm	
Classifiers	MAPE Value	Standard Deviation	MAPE Value	Standard Deviation	MAPE Value	Standard Deviation
Interactive Particle System Maximum Likelihood	0.5815	0.3034	0.2183	0.0551	0.3641	0.0381
Interactive Particle System Average Likelihood	0.5979	0.3233	0.2227	0.0539	0.3630	0.0303
Gaussian Mixture Model	0.1047	0.0016	0.1423	0.0108	0.4928	0.0219
Gaussian Processes	0.1259	0	0.1921	5.6460e-17	0.3114	0

TABLE III: Mean Absolute Percentage Deviation and Standard Deviation for each classifiers, in average, after 30 runs.

C. Dynamic setup scenario

In this section, a dynamic setup for the predictors is presented. The training set is split and ordered into a collection of observations, each one being a record of the traffic that took place during a time interval. For a number of observations Q and a window size q, the predictors are trained for $m = Q - q + 1$ steps and m values are predicted(Fig. 5). At each iteration, only one value is predicted along with the calculated error, used for MAPE evaluation.
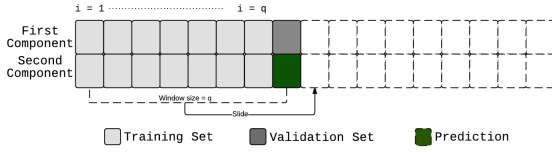

Fig. 5: Siliding window approach example.

IPS-ML and IPS-AL are trained and tested, as in the static setup scenario (Subsection 4.1), for a number of iterations $m = 100$. First, the N particles are generated and their likelihoods are calculated. The perturbation and acceptance phases are applied until the final population is resulted. Depending on the chosen method (IPS - ML or IPS - AL), one value is predicted. For a number of $m - 1$ times, these steps are repeated. The results are stored and compared with the real values.

GMM is trained with the first set of observations and the parameters of the model with the best AIC score, are estimated. In order to predict the first value, the parameters are updated using an expectation-maximization-approach (Algorithm 2). These steps are repeated for $m - 1$ times in order to complete the prediction and calculate the errors.

The GP is trained 100 times with the first set of observations. The first predicted value and the parameters of the GP are calculated. After building the first GP, its parameters together with the second set of observations are used to build the second GP. For $m - 1$ steps, the GPs are trained only 10 times, and the predicted values are calculated along with the errors given by the prediction.

In Table IV, the errors of the predictors in the dynamic scenario, MAPE and Standard deviation are presented. Compared with the previous scenarios, two predictors (GMM and GP) are improved.

Classifiers	H10 am/ H11 am		H12 am/ H13 pm		H20 pm/ H21 pm	
	MAPE Value	Standard Deviation	MAPE Value	Standard Deviation	MAPE Value	Standard Deviation
Interactive Particle System Maximum Likelihood	0.5105	0.0250	0.2586	0.0193	0.2857	0.0122
Interactive Particle System Average Likelihood	0.5024	0.0242	0.2502	0.0193	0.2847	0.0106
Gaussian Mixture Model	0.0857	0.0083	0.1487	0.0123	0.3389	0.0412
Gaussian Processes	0.0684	2.8230e-17	0.1238	4.2345e-17	0.2448	5.6460e-17

TABLE IV: Mean Absolute Percentage Deviation and Standard Deviation for each classifiers, in average, after 30 runs.

An ANOVA test [13] is used to validate the results by appling unpaired multiple comparisons between the predictors. In Table II, Table III and Table IV, the average of the errors of the predictors, after 30 iterations, is presented. The errors given by each predictor are the independent samples given as input for the statistical test. In the static case, GMM and GP performs better than IPS-AL and IPS-ML for the data sets: 10am - 11am and 12pm - 13pm (Fig. 6(a), (b)). In this scenario, for the 20pm - 21pm (Fig.6 (c)) data set GMM gives the lowest quality solution in terms of the prediction.

In the static scenario with shuffled data GMM and GP perform better than IPS-AL and IPS-ML for the data set: 10am - 11am (Fig. 7(a)). For the data set 12pm - 13pm (Fig. 7(b)) GMM works better than all the other predictors along with GP, while for the 20pm - 21pm data set GP gives the best solution followed by IPS-ML and IPS-AL (that are not significantly different), and GMM that performs worst (Fig. 7(c)).

For the dynamic scenario, the results of the test show that GP works best followed by GMM for the data sets: 10am - 11am (Fig. 8(a)) and 12pm - 13pm (Fig. 8 (b)), and the results given by IPS-AL are not signficantly different from IPS-ML. For the last scenario and the 20pm -21pm (Fig. 8(c)) data set the predictor that gives best solution in terms of quality is GP followed by IPS-AL and IPS-ML that are not significantly different and GMM that performs worst.

V. Conclusions

In this paper, we present and study different prediction models built for a real Voice over IP environment. Predicting the incoming voice traffic during a time frame is used for shaping the future patterns and capacity planning, can be used for the improvement of the VoIP qualtiy. Due to the dynamic evolution of requests, decisions concerning the distribution of the computational resources in a VoIP environment must be taken dynamically as well. The system can adapt and scale in order to ensure availabilty as the most important requirement in providing quality of service. The predictors considered in our study are Gaussian Mixture Models, Gaussian Processes and Interactive Particle System (IPS), trained and tested under different scenarios.

Our work includes traffic shaping and prediction. The behaviour of users that place VoIP calls is outlined. We provide insights on how particle algorithms can be used for optimization in conjunction with implicit learning models, strategies and scenarios. An overview the previous work where Gaussian Mixture Models and Gaussian Processes are used for VoIP calls characteritics shaping, such as call duration and call holding times, is provided. None of the previous work considers Gaussian Mixture Models or Gaussian Processes for modelling the amount of incoming calls placed during a time frame in a VoIP system.

For future work, we will extend the use of particle algorithms that will not only provide load estimation information but also act as actuators. The prediction models will be integrated with highly efficient load-balancing algorithms, that take into account the usage of resources and energy consumption.

(a) Data set 10am - 12am.   (b) Data set 12pm - 13pm.   (c) Data set 20pm - 21pm.

Fig. 6: ANOVA test for the results given by the predictors in Static setup scenario.

(a) Data set 10am - 12am.   (b) Data set 12pm - 13pm.   (c) Data set 20pm - 21pm.
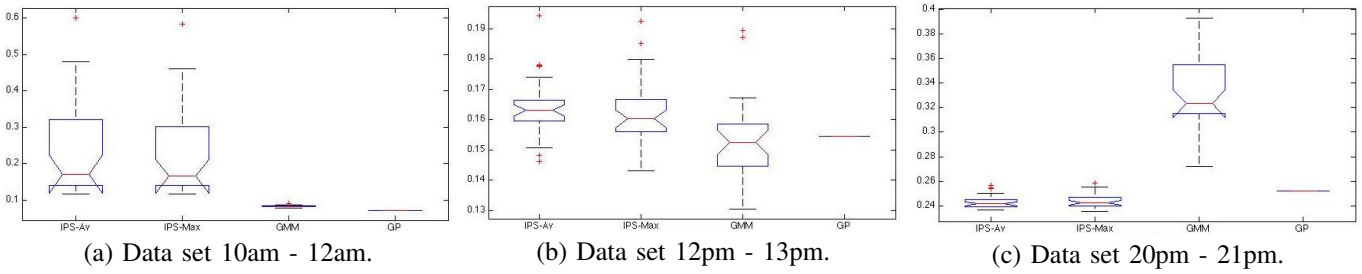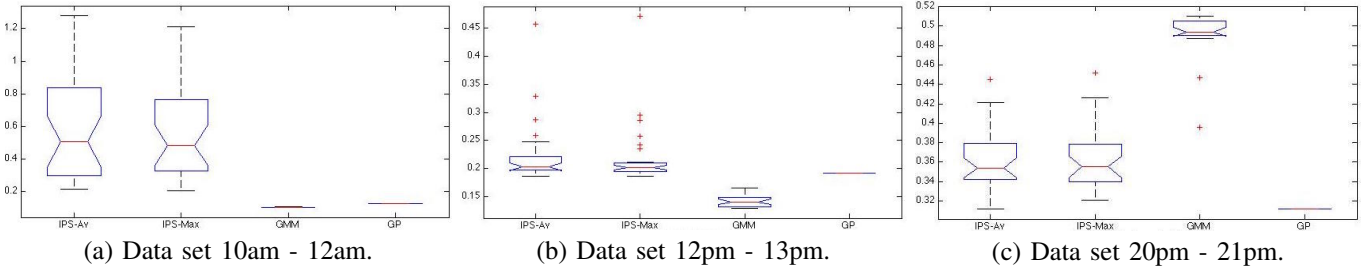
Fig. 7: ANOVA test for the results given by the predictors in Static setup scenario with shuffled data.

(a) Data set 10pm - 11pm.   (b) Data set 12pm - 13pm.   (c) Data set 20pm - 21pm.
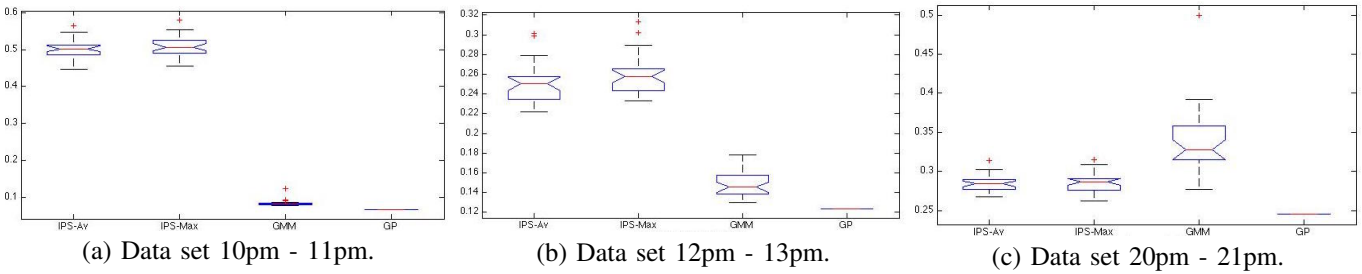
Fig. 8: ANOVA test for the results given by the predictors in Dynamic setup scenario.

REFERENCES

[1] R.-M. Neal, *BAYESIAN LEARNING FOR NEURAL NETWORKS*, PhD thesis, Department of Computer Science, University of Toronto, 1995.

[2] G.-J. McLachlan and K.-E. Basford, *Mixture Models: Inference and Applications to Clustering*, 1988.

[3] C. Ash, *Continuous Random Variables*, Wiley-IEEE Press, 1993.

[4] W. Davenport and W. Root, *The Gaussian Process*, Wiley-IEEE Press, 1987.

[5] Q. Dan, Y. Honggang, T. Hui and W. Bingxi, *Two Schemes for Automatic Speaker Recognition Over VOIP*, 2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application.

[6] D. Yessad and A. Amrouche, *Fusion Strategies for Distributed Speaker Recognition using Residual Signal Based G729 Resynthesized Speech*, 16th International Conference on Information Fusion Istanbul, Turkey, July 9-12, 2013.

[7] O. Nhway, *An investigation into the effect of security on reliability and voice recognition system in a VoIP network*, Advanced Communication Technology (ICACT), 2011 13th International Conference.

[8] T.-D. Dang, B. Sonkoly and S. Molnr, *Fractal Analysis and Modeling of VoIP Traffic*, Telecommunications Network Strategy and Planning Symposium. NETWORKS 2004, 11th International Conference.

[9] A.-M. Simionovici, A.-A. Tantar, P. Bouvry and L. Didelot, *Predictive modeling in a voip system*, Journal of Telecomunications and Information Technology, 2013.

[10] P. Del Moral and A. Doucet, *Particle Methods: An introduction with applications*, Springer LNCS/LNAI Tutorial book no. 6368, 2010-2011.

[11] P. Del Moral, A.-A. Tantar and E. Tantar, *On the Foundations and the Applications of Evolutionary Computing*, EVOLVE- A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation Studies in Computational Intelligence, vol. 447, pp. 3-89, 2013.

[12] S.-W. Nydick, *The Wishart and Inverse Wishart Distribution*, http://www.math.wustl.edu/ sawyer/hmhandouts/Wishart.pdf, 2012.

[13] R.-V. Hogg and J. Ledolter, *Engineering Statistics*, New York: MacMillan, 1987.

[14] D. Reynolds, *Gaussian Mixture Models*, http://www.ll.mit.edu/mission/communications/ist/publications/0802_-Reynolds_Biometrics-GMM.pdf.

[15] A. Dempster, N. Laird and D. Rubin, *Maximum Likelihood from Incomplete Data via the EM Algorithm*, Journal of the Royal Statistical Society, 1977.

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