A Gaussian mixture approach to model stochastic processes in power systems

Quentin Gemine, Bertrand Cornélusse, Mevludin Glavic, Raphael Fonteneau, Damien Ernst
Department of Electrical Engineering and Computer Science
University of Liège
Liège, Belgium
{qgemine,bertrand.cornelusse,mevludin.glavic,raphael.fonteneau,dernst}@ulg.ac.be

***Abstract*—Probabilistic methods are emerging for operating electrical networks, driven by the integration of renewable generation. We present an algorithm that models a stochastic process as a Markov process using a multivariate Gaussian Mixture Model, as well as a model selection technique to search for the adequate Markov order and number of components. The main motivation is to sample future trajectories of these processes from their last available observations (i.e. measurements). An accurate model that can generate these synthetic trajectories is critical for applications such as security analysis or decision making based on lookahead models. The proposed approach is evaluated in a lookahead security analysis framework, i.e. by estimating the probability of future system states to respect operational constraints. The evaluation is performed using a 33-bus distribution test system, for power consumption and wind speed processes. Empirical results show that the GMM approach slightly outperforms an ARMA approach.**

***Index Terms*—time series; stochastic process; stochastic modeling; Gaussian mixture model; lookahead simulation; synthetic trajectory; security analysis; distribution system.**

I. Introduction

The recent massive integration of renewable generation has increased the level of uncertainty in power systems, to the extent that probabilistic methods are emerging for operating electrical networks [1]. This is particularly true for the operation of distribution systems, which is progressively migrating from a fit-and-forget doctrine to Active Network Management (ANM) strategies [2]. This approach relies on short-term policies that control the power injected by generators and/or taken off by loads in order to avoid congestion or voltage issues and requires to solve sequential decision making problems [3]. An accurate model of this uncertain dynamical system is critical in order to take adequate control actions. Moreover, and contrary to wider power systems, the uncertainty about stochastic quantities (e.g. wind speed, solar irradiance, load consumption) is not softened by an averaging effect because of the local nature of distribution systems.

In this paper, we present an algorithm that models a stochastic process as a Markov process using a multivariate Gaussian Mixture Model (GMM). Such a parametric model learns the transition density of the process from time series of observations. For a given order of the Markov process (i.e. the length of the process history that is used to model the density of the next realization) and a given number of components in the mixture, the parameters of the GMM are learned from the data using a maximum likelihood approach. A model selection technique that relies on a multi-armed bandit framework [4] is used to search for the adequate order and number of components of the GMM.

We focus in this paper on the ability of stochastic models to perform reliable security analyses, i.e. lookahead security estimates of the operational state of a grid. It leads to the definition of a quality measure that compares the actual security state of a grid to the Monte Carlo simulations of a model. This measure is used both for the model section phase and for comparison purposes with other modeling approaches.

A. Related works

Existing approaches in the context of power system dynamic modeling and decision making include forecasting random variables (loads, PV and wind generation) based on the use of numerical weather prediction and time series models [5], [6], [7]. Reference [6] surveys existing approaches for wind power forecasting while [8] provides insight to short-term PV generation forecast. Numerical weather prediction uses meteorological data and models to forecast relevant variables such as wind, irradiance, etc, and further uses physical models or statistical techniques to forecast generation productions [5], [6], [8]. Artificial neural networks or fuzzy neural networks [5] were also considered for improved forecasting. Time series models use observed data values (historical data) to forecast future values of random variables. Auto-Regressive Moving Average (ARMA) models and its variants - auto regressive integrated moving average (ARIMA), ARMA with exogenous input (ARMAX/ARX) - are the most popular type in the time-series-based approaches for both load and generation forecasting [8], [7]. Neural networks and fuzzy neural networks were

This research is supported by the public service of Wallonia - Department of Energy and Sustainable Building within the framework of the GREDOR project. The authors give their thanks for the financial support of the Belgian Network DYSCO, an Inter-university Attraction Poles Program initiated by the Belgian State, Science Policy Office. Computational resources have been provided by the Consortium des quipements de Calcul Intensif (CCI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under Grant No. 2.5020.11

also considered in the context of the use of historical data [5], [7], [6].

In the context of power system problems, GMMs were only considered within static decision making where probabilistic power flow is a common tool to handle uncertainties [9]. GMM is used to approximate non-Gaussian probability density functions (Beta, Gamma, Weibull, Rayleigh) of loads [10] and generation [9]. One of the problems related to the use of GMM in this context is the choice of the number of Gaussian mixture components to accurately approximate the original non-Gaussian probability density function. The work presented in [9] compares three pair-merging methods to reduce the number of Gaussian mixture components and proposes a fine-tuning algorithm of integral square difference criterion for further improvements. On the other hand, a Markov process modeling approach was considered as an option to forecast load and wind generation. A Markov-based sensitivity model was proposed in [11] as a look-ahead capability approach for load and wind generation short-term forecasting.

To the best knowledge of the authors no work exist that uses GMMs combined with Markov process modeling for dynamic modeling and decision making in power systems. In a wider context, some developments on the use of GMMs for time series forecasting exist. Reference [12] reports on the initial results of the use of GMMs for time series but focuses exclusively on the forecasting abilities of the approach though the computation of conditional expectations.

II. Problem description

We aim at building models of stochastic processes that arise within power systems with the main motivation of sampling future trajectories of these processes from their last available observations (i.e. measurements). An accurate model that can generate these synthetic trajectories is critical for applications such as security analysis or decision making based on lookahead models.

A. Problem statement

Let $\mathcal{S}^*$ be a real-valued stochastic process and let

$$\mathcal{S} = \left\{ \begin{array}{c} (o_1^1, \ldots, o_T^1) \\ \vdots \\ (o_1^{n_\mathcal{S}}, \ldots, o_T^{n_\mathcal{S}}) \end{array} \right\}$$

be a set of $n_\mathcal{S}$ real-valued time series of length T that correspond to observations of $\mathcal{S}^*$. Given this set $\mathcal{S}$, we want to learn a Markov model $\mathcal{M}$ that approximates the probability density function $p : \mathbb{R}^L \to \mathbb{R}^+$ of the next realization x_t, conditional to the previous L realizations of the process, i.e. $p_\mathcal{M}(x_t|x_{t-1}, \ldots, x_{t-L})$.

B. Model evaluation

We do not focus on the forecast performance but instead aim at a model that is relevant to generate synthetic trajectories in a lookahead context. Models are discriminated based on their ability at producing good lookahead security estimates. Such an estimate corresponds to the probability, according to a model $\mathcal{M}$, that an electrical system $\mathcal{D}$ is secure (i.e. its operational constraints are respected) for some lookahead time horizon Δt and given an history $\boldsymbol{x}^{(\text{hist})} = (x_{t-1}, \ldots, x_{t-L})$. We consider that $\mathcal{S}^*$ is the only stochastic process that influences the electrical system $\mathcal{D}$ and that its state is fully determined (e.g. through a power flow simulation) given a time step $t \in \{1, \ldots, T\}$ and the realization x_t or an estimate $\hat{x}_t$ of $\mathcal{S}^*$. We denote thereafter this state $\mathcal{D}(t, x_t)$ or $\mathcal{D}(t, \hat{x}_t)$.

We introduce a score function $\eta_{\mathcal{D},\Delta t,M}(\mathcal{M}, \mathcal{S}) \in [0, 1]$ to assess the quality of lookahead security estimates produced through Monte-Carlo simulations of $\mathcal{M}$, when $\boldsymbol{x}^{(\text{hist})}$ takes as value every sequence of L successive observations in the set $\mathcal{S}$ of time series, and where M is the number of sampled trajectories for every Monte-Carlo simulation. The score function relies on a weighted Brier score [13] (i.e. the mean of squared differences), which is reversed and scaled so that the value of $\eta_{M,\Delta t}(\mathcal{M}, \mathcal{S})$ has the following interpretations:

- 0 means that $\mathcal{M}$'s security estimates are non-informative (i.e. the probability of every state to be secure is 0.5) at best;
- 1 means that $\mathcal{M}$'s security estimates match perfectly the actual security states observed in the set $\mathcal{S}$.

The following pseudo-code details how to compute this score function:

▷ *Overall procedure*
function GET_SCORE($\mathcal{D}, \mathcal{M}, \mathcal{S}, M, \Delta t$)
  $\text{dist}_\text{ok}, \text{dist}_\text{ko}, n_\text{ok}, n_\text{ko} \leftarrow 0$
  for $i \leftarrow 1, n_\mathcal{S}$ **do**
    for $t \leftarrow L, T - \Delta t$ **do**
      $\boldsymbol{x}^\text{hist} \leftarrow (o_t^{(\text{i})}, \ldots, o_{t-L+1}^{(\text{i})})$
      $\hat{p}_\text{ok} \leftarrow 0$
      for $m \leftarrow 1, M$ **do**
        $\hat{p}_\text{ok} \leftarrow \hat{p}_\text{ok} + \frac{\text{IS\_SECURE}(\mathcal{D},\mathcal{M},\boldsymbol{x}^\text{hist},t,\Delta t)}{M}$
      if $\mathcal{D}(t + \Delta t, o_{t+\Delta t}^{(i)})$ is secure **then**
        $\text{dist}_\text{ok} \leftarrow \text{dist}_\text{ok} + (1 - \hat{p}_\text{ok})^2$
        $n_\text{ok} \leftarrow n_\text{ok} + 1$
      else
        $\text{dist}_\text{ko} \leftarrow \text{dist}_\text{ko} + (\hat{p}_\text{ok})^2$
        $n_\text{ko} \leftarrow n_\text{ko} + 1$
  score $\leftarrow 1 - \left(\frac{\text{dist}_\text{ok}}{n_\text{ok}} + \frac{\text{dist}_\text{ko}}{n_\text{ko}}\right)$
  return max$(0, \text{score})$

▷ *Single security simulation*
function IS_SECURE($\mathcal{D}, \mathcal{M}, \boldsymbol{x}^{(\text{hist})}, t_0, \Delta t$)
  $t \leftarrow t_0$
  repeat
    $\hat{x} \leftarrow$ sample $p_\mathcal{M}(\cdot|\boldsymbol{x}^{(\text{hist})})$
    $\boldsymbol{x}^{(\text{hist})} \leftarrow (\hat{x}, x_1^{(\text{hist})}, \ldots, x_{L-1}^{(\text{hist})})$
    $t \leftarrow t + 1$
  until $t = t_0 + \Delta t$
  if $\mathcal{D}(t, \hat{x})$ is secure **then**
    return 1
  else
    return 0

III. Gaussian Mixture Model

We consider models $\mathcal{M}_{\boldsymbol{\omega}}(\boldsymbol{\theta}), \forall \boldsymbol{\omega} \in \Omega$ and $\forall \boldsymbol{\theta} \in \Theta$, that rely on a mixture of N Gaussian components to build the density function $p_{\mathcal{M}_{\boldsymbol{\omega}}}$, where $\boldsymbol{\omega} = (L, N)$ are the hyper parameters of the model and where $\boldsymbol{\theta}$ denotes the parameters of the Gaussian mixture. These latter parameters are the weight, mean, and covariance matrix of every component $i \in \{1, \dots, N\}$ of the mixture, they are further denoted by ϕ_i, $\boldsymbol{\mu}_i$, and $\boldsymbol{\Sigma}_i$, respectively. In particular, we have:

$$p_{\mathcal{M}_{\omega,\mathcal{L}(\boldsymbol{\theta})}}(x_t|x_{t-1},\dots,x_{t-L}) = \frac{p^{\cap}_{\mathcal{M}_{\omega,\mathcal{L}(\boldsymbol{\theta})}}(x_t,\dots,x_{t-L})}{\int_{\mathbb{R}} p^{\cap}_{\mathcal{M}_{\omega,\mathcal{L}(\boldsymbol{\theta})}}(x_t,\dots,x_{t-L})dx_t}\,, \quad (1)$$

where

$$p^{\cap}_{\mathcal{M}_{\omega,\mathcal{L}(\boldsymbol{\theta})}}(x_t, \dots, x_{t-L}) = \sum_{i=1}^{N} \phi_i \mathcal{N}(x_t, \dots, x_{t-L}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad (2)$$

is the approximation, by model $\mathcal{M}_{\boldsymbol{\omega},\mathcal{L}}(\boldsymbol{\theta})$, of the joint density function of $L{+}1$ successive realizations. One of the advantages of using Gaussian components is that the conditional density function in (1) can be easily determined given the values of the L previous realizations $\boldsymbol{x}_{\text{past}} = (x_{t-1}, \dots, x_{t-L}) \in \mathbb{R}^L$ of the process. The resulting density is also a Gaussian mixture [14] and each component $i \in \{1, \dots, N\}$ has the following parameters:

$$\phi_i^{t|L} = \frac{\phi_i \mathcal{N}(\boldsymbol{x}_{\text{past}}; \boldsymbol{\mu}_i^L, \boldsymbol{\Sigma}_i^{LL})}{\sum_{j=1}^{N} \phi_j \mathcal{N}(\boldsymbol{x}_{\text{past}}; \boldsymbol{\mu}_j^L, \boldsymbol{\Sigma}_j^{LL})}\,, \quad (3)$$

$$\mu_i^{t|L} = \mu_i^t - (\Lambda_i^{tt})^{-1} \boldsymbol{\Lambda}_i^{tL} \left(\boldsymbol{x}_{\text{past}} - \boldsymbol{\mu}_i^L\right)\,, \quad (4)$$

$$\Sigma_i^{t|L} = (\Lambda_i^{tt})^{-1}\,, \quad (5)$$

with

$$\boldsymbol{\mu}_i = (\mu_i^t, \boldsymbol{\mu}_i^L)\,,$$

$$\boldsymbol{\Sigma}_i = \begin{pmatrix} \Sigma_i^{tt} & \boldsymbol{\Sigma}_i^{tL} \\ \boldsymbol{\Sigma}_i^{Lt} & \boldsymbol{\Sigma}_i^{LL} \end{pmatrix}, \boldsymbol{\Sigma}_i^{-1} = \begin{pmatrix} \Lambda_i^{tt} & \boldsymbol{\Lambda}_i^{tL} \\ \boldsymbol{\Lambda}_i^{Lt} & \boldsymbol{\Lambda}_i^{LL} \end{pmatrix}.$$

For a given hyper parameter $\boldsymbol{\omega} = (L, N)$, learning model $\mathcal{M}_{\boldsymbol{\omega},\mathcal{L}}(\boldsymbol{\theta})$ consists in determining the mixture's parameters $\boldsymbol{\theta}^*$ that approximate at best the density function $p^{\cap}_{\mathcal{M}_{\omega,\mathcal{L}(\boldsymbol{\theta})}}(\cdot)$ of the set $\mathcal{L}$ of $n_{\mathcal{S}}$ time series, The following procedure allows to compute $\boldsymbol{\theta}^*$ as the maximum likelihood estimate (MLE) of $\mathcal{L}$:

1) build a set $\mathcal{L}'$ of L+1-length tuples:

$$\mathcal{L}' = \left\{(o^l_{t-L}, \dots, o^l_{t-1}, o^l_t),\, (l,t) \in \{1, \dots, n_{\mathcal{L}'}\} \times \{L, \dots, T-1\}\right\}\,;$$

2) produce the MLE $\boldsymbol{\theta}^*$ by solving:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta} \in \Theta} \sum_{\boldsymbol{x} \in \mathcal{L}'} \log p^{\cap}_{\mathcal{M}_{\omega,\mathcal{L}(\boldsymbol{\theta})}}(\boldsymbol{x})\,,$$

which is the classical MLE equation that can be solved using an expectation-maximization (EM) algorithm [15].

IV. Model selection

In this section, we describe our sampling-based procedure to navigate within the space of hyper parameters $\Omega = \{\omega_1, \dots, \omega_K\}$, $K \in \mathbb{N}$, using a multi-armed bandit approach. Our approach relies on the following assumptions: first, we assume that, for a given hyper parameter $\omega \in \Omega$, and a given set of learning data $\mathcal{L}$, we have access to a procedure, which may not be deterministic, that allows to generate a model $\mathcal{M}_{\omega,\mathcal{L}}$ (e.g. see Section III). Then, we assume that we have access to a score function η (e.g. see Section II-B) to compute noisy empirical evaluations of any model. Finally, we also assume that we have access to a selection strategy which allows us to iteratively select which hyper parameter to sample from based on noisy evaluations observed so far, and progressively converge towards an optimal hyper parameter. In the following, a standard UCB-1 algorithm [4] plays the role of this selection strategy.

Our procedure works as follows. Initially, all index values are set to $+\infty$:

$$\forall k \in \{1, \dots, K\}, B_k^{(0)} = +\infty$$

Then, at every iteration $i \in \{1, \dots, N\}$,

1) Select a hyper parameter $\omega^{(i)} \in \Omega$ according to a UCB-1 strategy; let $k^{(i)}$ be the index of $\omega^{(i)}$ in the set Ω:

$$k^{(i)} = \arg\max_{k \in \{1,\dots,K\}} B_k^{(i-1)}$$

2) Perform a random partition of the set of trajectories into two subsets of trajectories, a learning set $\mathcal{L}^{(i)}$ and a test set $\mathcal{T}^{(i)}$, formalized as follows:

$$\mathcal{L}^{(i)} = \left\{ \begin{array}{c} \left(o_0^{(i),1}, \dots, o_{T-1}^{(i),1}\right) \\ \vdots \\ \left(o_0^{(i),n_{\mathcal{L}}}, \dots, o_{T-1}^{(i),n_{\mathcal{L}}}\right) \end{array} \right\}$$

$$\mathcal{T}^{(i)} = \left\{ \begin{array}{c} \left(\mathfrak{o}_0^{(i),1}, \dots, \mathfrak{o}_{T-1}^{(i),1}\right) \\ \vdots \\ \left(\mathfrak{o}_0^{(i),n_{\mathcal{T}}}, \dots, \mathfrak{o}_{T-1}^{(i),n_{\mathcal{T}}}\right) \end{array} \right\}$$

3) Using the learning set $\mathcal{L}^{(i)}$ and the hyper parameter $\omega^{(i)}$, compute a model $\mathcal{M}_{\omega^{(i)},\mathcal{L}^{(i)}}$;
4) Compute a new noisy evaluation $\eta_k^{(i)}$ of model $\mathcal{M}_{\omega^{(i)},\mathcal{L}^{(i)}}$ as:

$$\eta_k^{(i)} = \eta(\mathcal{M}_{\omega^{(i)},\mathcal{L}^{(i)}}, \mathcal{T}^{(i)})\,,$$

and update the UCB-1 index values as follows:

$$\forall k \in \{1, \dots, K\}, B_k^{(i)} = \bar{\eta}_k + \sqrt{\frac{2\ln(i)}{n_k^{(i)}}}$$

where $\bar{\eta}_k$ denotes the empirical average of evaluations of hyper parameter ω_k observed so far and $n_k^{(i)}$ denotes the number of times the hyper parameter ω_k has been evaluated so far.

V. Numerical results

We present the results obtained for two different datasets, with the GMM approach presented in Section III and compared with ARMA models, which were also fitted using a MLE algorithm. For both approaches, the hyper parameters were selected using the model selection technique presented in Section IV. These hyper parameters are:

GMM: the Markov order L and the number of components N;

ARMA: the autoregressive order L_{ar} and the moving-average order L_{ma}.

Both datasets consist in observations acquired every quarter of hour, and every time serie spans a period of six weeks (i.e. 576 observations). The first dataset has 14 time series (i.e. 8064 observations) of the aggregated power consumption of 200 residential consumers, while the second dataset has 182 time series (i.e. 104832 observations) of wind speed measurements[1]. The electrical system $\mathcal{D}$ used for security analyses is the IEEE 33-bus distribution test system with three additional wind farms, as illustrated in Figure 1. The considered operational constraints are the voltage limits at the buses and the thermal limits of the links. Note that the consumption was assumed to be deterministic when evaluating the wind speed models, and conversely. When the stochastic process is the load consumption, the consumption at each bus is defined as a scaling factor times the consumption process. For the wind speed process, the production of wind farms is determined from the wind speed through a usual cubic power curve.

The performance estimations of the different models for the wind speed dataset, a lookahead time horizon of 4h (i.e. 16 time steps), and $M = 50$, are presented in Figure 2 for the GMM approach and in Figure 3 for the ARMA approach. The performance of these two sets of models was estimated by UCB-1 runs of 40h and 20h for the GMM and ARMA approaches, respectively. Several observations can be made from these results:

- the best GMM outperforms the best ARMA model;
- the performance of the GMMs is less sensitive to the choice of the hyper parameters than for the ARMA models;
- the computational budget (i.e. the average time required to make one evaluation) is higher for the GMM approach than for the ARMA approach.

We also report the performance estimations for the consumption dataset and a lookahead time horizon of half an hour (i.e. 2 time steps). The performance of the two sets of models was estimated by UCB-1 runs of 3h and 2h for the GMM and ARMA approaches, respectively. The results are presented in Figure 4 for the GMM approach and in Figure 5 for the ARMA one. We now observe that the performance of the best GMM and best ARMA model is very close. However, the sensitivity

[1]Both datasets were standardized (i.e. $\mu = 0$ and $\sigma = 1$) and diurnal seasonality was remove.

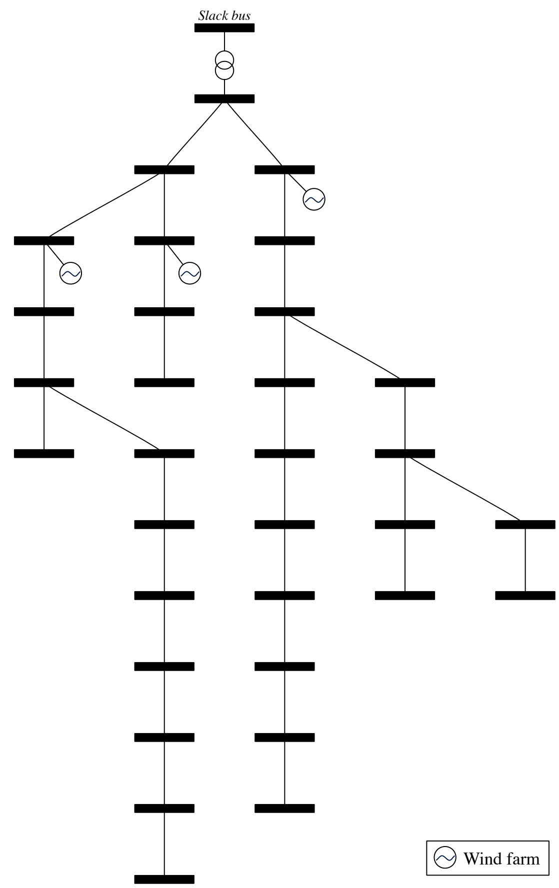


Figure 1: Electrical system $\mathcal{D}$, the IEEE 33-bus test system with three additional wind farms.

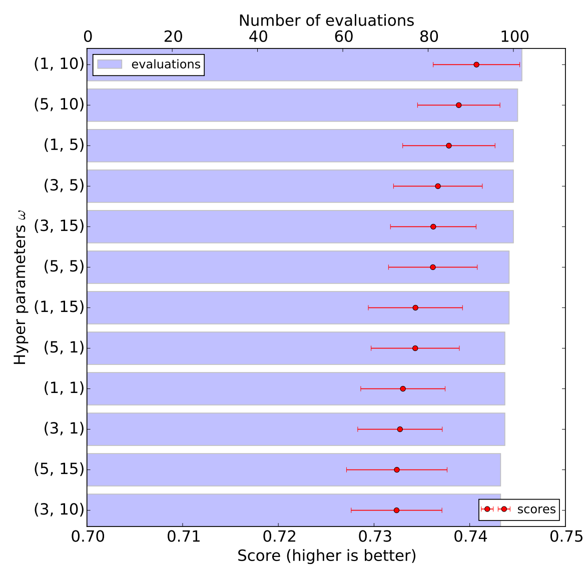


Figure 2: Performance estimations for GMMs for the wind speed dataset, a lookahead time horizon of 4h (i.e. 16 time steps), and $M = 50$, after a UCB-1 run of 40h.

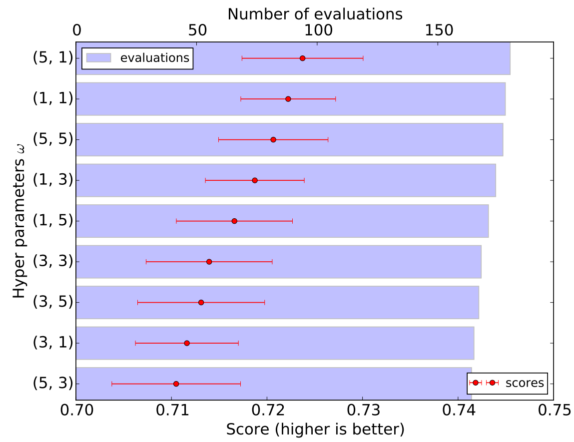


Figure 3: Performance estimations for ARMA models for the wind speed dataset, a lookahead time horizon of 4h (i.e. 16 time steps), and $M = 50$, after a UCB-1 run of 20h.

of the expected score to the hyper parameters is again higher for the ARMA models.

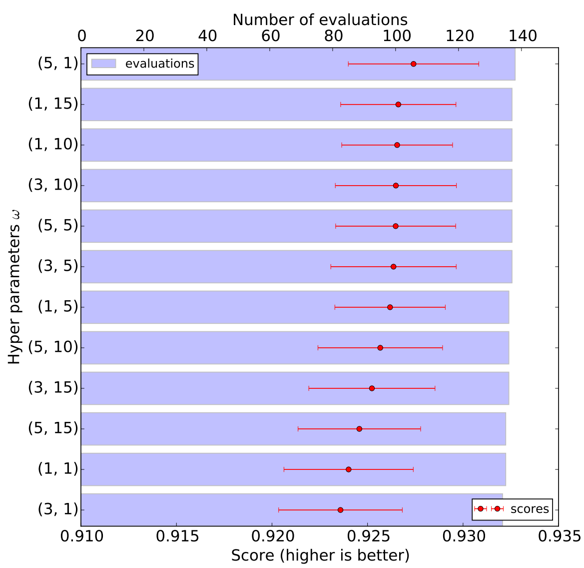


Figure 4: Performance estimations for GMMs for the consumption dataset, a lookahead time horizon of half an hour (i.e. 2 time steps), and $M = 50$, after a UCB-1 run of 3h.

We report in Figures 6 and 7 the expected performance of every candidate model for both approaches, both datasets, and for lookahead time horizons of 15min, 30min, 1h, 2h, and 4h. The gray scales are defined column-wise and the darkest cell of a column indicates the best expected score for the associated lookahead time horizon. We also illustrate in Figures 8 and 9 how the best model of the GMM approach performs comparing the best model of the ARMA approach, for the wind speed dataset and consumption dataset, respectively. We observe that,

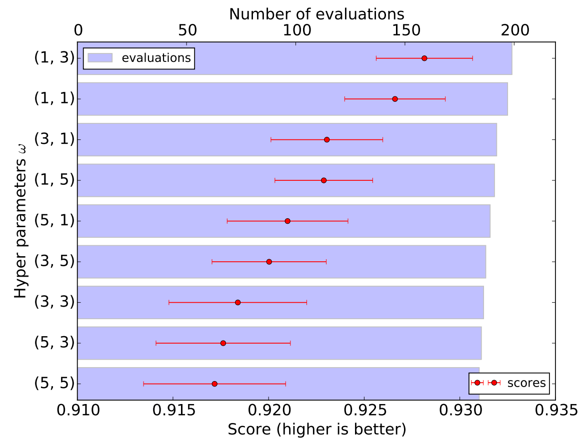


Figure 5: Performance estimations for ARMA models for the consumption dataset, a lookahead time horizon of half an hour (i.e. 2 time steps), and $M = 50$, after a UCB-1 run of 2h.

with the exception of the lookahead time horizons up to half an hour, the GMM approach outperforms the ARMA one. In addition, the lead of the GMM approach seems to get larger as the lookahead time horizon increases.

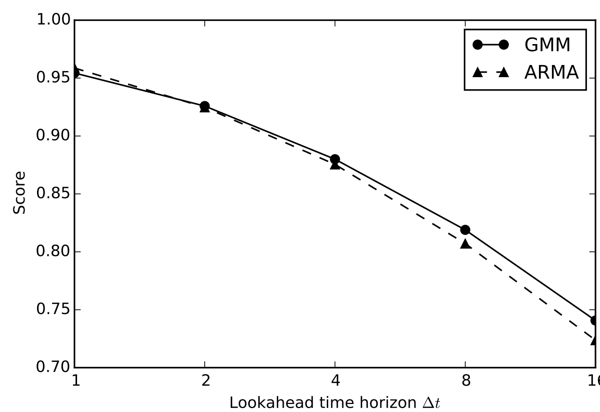


Figure 8: Performance comparison of each approach for the wind speed dataset, as a function of the lookahead time horizon (in time steps).

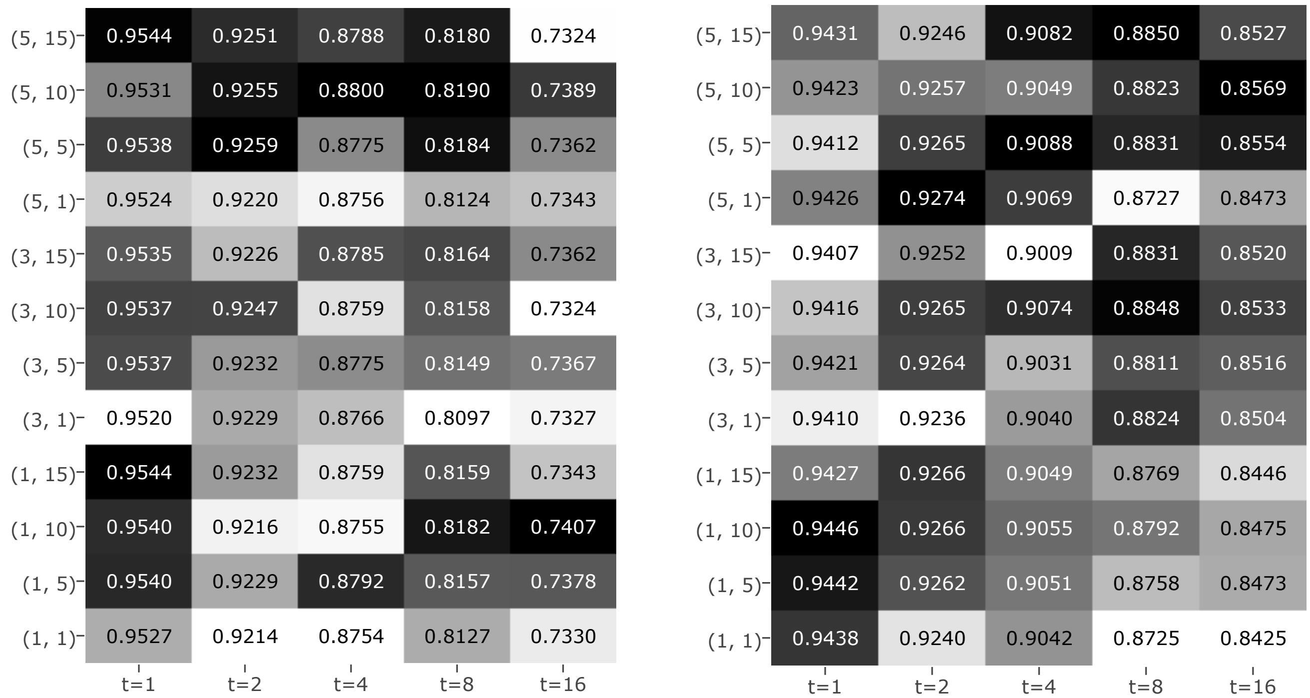
Finally, the values of the parameters that were used for the simulations are reported in Table I.

TABLE I: Parameters used for the simulations.

	$n_{\mathcal{L}}$	$n_{\mathcal{T}}$	M
$\mathcal{S}_{\text{cons}}$	$0.8\, n_{\mathcal{S}_{\text{cons}}}$	$0.2\, n_{\mathcal{S}_{\text{cons}}}$	50
$\mathcal{S}_{\text{wind}}$	$0.9\, n_{\mathcal{S}_{\text{wind}}}$	$0.1\, n_{\mathcal{S}_{\text{wind}}}$	50

VI. Implementation details

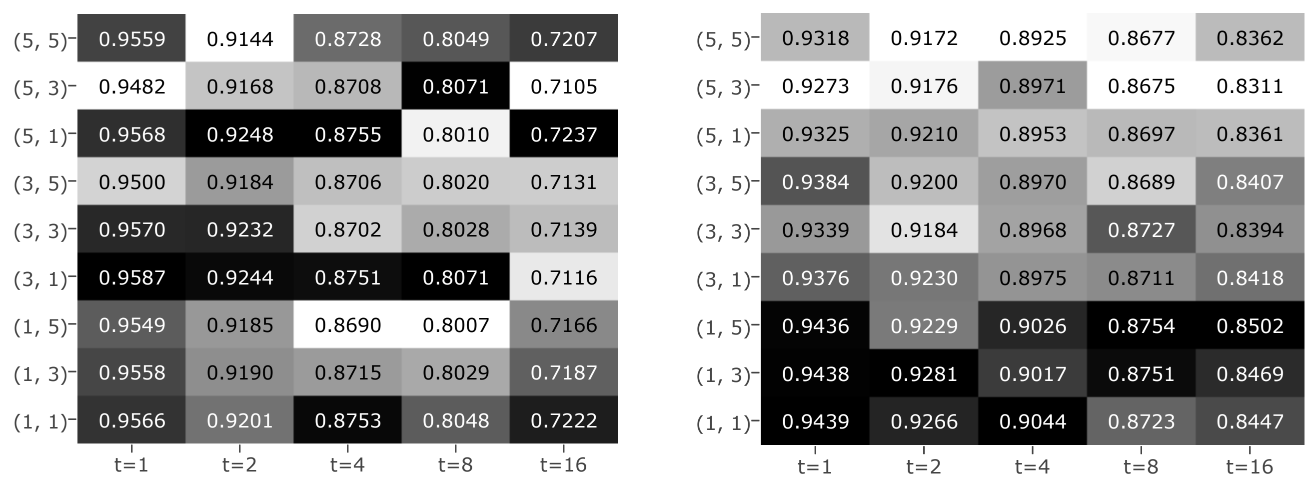
We benefited from the parallelization abilities of Monte-Carlo methods by running the model selection algorithm in a



(a) Wind speed dataset

(b) Consumption dataset

Figure 6: Expected score of every candidate GMM for the both dataset, as a function of the lookahead time horizon (in time steps).



(a) Wind speed dataset

(b) Consumption dataset

Figure 7: Expected score of every candidate ARMA model for the both dataset, as a function of the lookahead time horizon (in time steps).

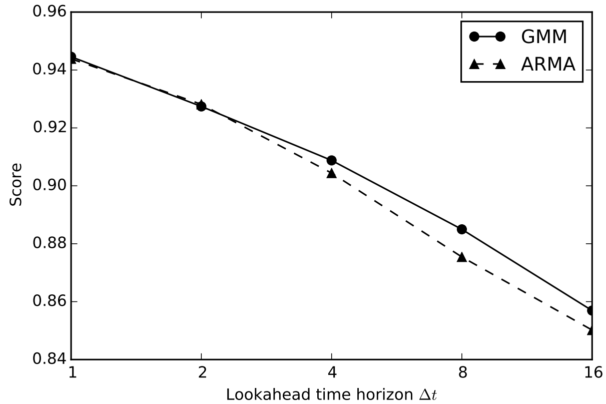


Figure 9: Performance comparison of each approach for the consumption dataset, as a function of the lookahead time horizon (in time steps).

HPC environment. A distinct computing core was dedicated for every trajectory $m \in \{1, \dots, M\}$ of the Monte-Carlo simulations. The program is written in Python and relies on the Scikit-Learn library [16] to learn Gaussian mixtures, while ARMA models where fitted to data using R's arima function through a R-to-Python interface.

VII. Conclusion and Future Work

We presented a novel approach that relies on Gaussian mixtures to model a stochastic process from a set of time series of observations. The hyper parameters of the model, i.e. the Markov order and the number of mixture components, are determined using a multi-armed bandit technique while the mixture parameters are learned from the data using an EM algorithm. Empirical results show that the proposed approach outperforms an ARMA approach for the considered application of lookahead security analysis, for datasets of residential power consumption and of wind speed.

As future work we consider several extensions of the present work, including simulations for other processes (e.g. solar irradiance), comparison with alternative modeling approaches (e.g. ARIMA and GARCH), as well as different test systems.

References

[1] P. Panciatici, G. Bareux, and L. Wehenkel, "Operating in the fog: Security management under uncertainty," *Power and Energy Magazine, IEEE*, vol. 10, no. 5, pp. 40–49, 2012.

[2] H. Farhangi, "A road map to integration: Perspectives on smart grid development," *Power and Energy Magazine, IEEE*, vol. 12, no. 3, pp. 52–66, 2014.

[3] Q. Gemine, E. Karangelos, D. Ernst, and B. Cornélusse, "Active network management: planning under uncertainty for exploiting load modulation," in *Bulk Power System Dynamics and Control-IX Optimization, Security and Control of the Emerging Power Grid (IREP), 2013 IREP Symposium*. IEEE, 2013, pp. 1–9.

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[5] P. Pinson, C. Chevallier, and G. N. Kariniotakis, "Trading wind generation from short-term probabilistic forecasts of wind power," *Power Systems, IEEE Transactions on*, vol. 22, no. 3, pp. 1148–1156, 2007.

[6] A. M. Foley, P. G. Leahy, A. Marvuglia, and E. J. McKeogh, "Current methods and advances in forecasting of wind power generation," *Renewable Energy*, vol. 37, no. 1, pp. 1–8, 2012.

[7] P. Chen, T. Pedersen, B. Bak-Jensen, and Z. Chen, "Arima-based time series model of stochastic wind power generation," *Power Systems, IEEE Transactions on*, vol. 25, no. 2, pp. 667–676, 2010.

[8] P. Bacher, H. Madsen, and H. A. Nielsen, "Online short-term solar power forecasting," *Solar Energy*, vol. 83, no. 10, pp. 1772–1783, 2009.

[9] G. Valverde, J. Q. Tortós, and V. Terzija, "Comparison of gaussian mixture reductions for probabilistic studies in power systems," in *Power and Energy Society General Meeting, 2012 IEEE*. IEEE, 2012, pp. 1–7.

[10] R. Singh, B. C. Pal, R. Jabr *et al.*, "Statistical representation of distribution system loads using gaussian mixture model," *Power Systems, IEEE Transactions on*, vol. 25, no. 1, pp. 29–37, 2010.

[11] N. Abdel-Karim and M. Ilic, "Modeling uncertain load and wind power in the electric energy systems," in *Power and Energy Society General Meeting, 2012 IEEE*. IEEE, 2012, pp. 1–8.

[12] E. Eirola and A. Lendasse, "Gaussian mixture models for time series modelling, forecasting, and interpolation," in *Advances in Intelligent Data Analysis XII*. Springer, 2013, pp. 162–173.

[13] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Monthly weather review*, vol. 78, no. 1, pp. 1–3, 1950.

[14] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[15] R. Redner and H. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review*, vol. 26, no. 2, pp. pp. 195–239, 1984.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.