Residential probabilistic load forecasting: A method using Gaussian process designed for electric load data

Mahmoud Shepero⁎, Dennis van der Meer, Joakim Munkhammar, Joakim Widén

Department of Engineering Sciences, Uppsala University, P.O. Box 534, SE-751 21 Uppsala, Sweden

HIGHLIGHTS
• Probabilistic residential load forecasting using Gaussian and log-normal processes.
• Deterministic and probabilistic error metrics evaluated the proposed processes.
• Our results produced sharper forecasts compared with previous models.
• The log-normal process outperformed the Gaussian process in the forecast sharpness.
• The log-normal, unlike the Gaussian, process produced a varying forecast sharpness.

ARTICLE INFO
Keywords:
Gaussian process
Probabilistic load forecasting
Residential load

ABSTRACT
Probabilistic load forecasting (PLF) is of important value to grid operators, retail companies, demand response aggregators, customers, and electricity market bidders. Gaussian processes (GPs) appear to be one of the promising methods for providing probabilistic forecasts. In this paper, the log-normal process (LP) is newly introduced and compared to the conventional GP. The LP is especially designed for positive data like residential load forecasting—little regard was taken to address this issue previously. In this work, probabilistic and deterministic error metrics were evaluated for the two methods. In addition, several kernels were compared. Each kernel encodes a different relationship between inputs. The results showed that the LP produced sharper forecasts compared with the conventional GP. Both methods produced comparable results to existing PLF methods in the literature. The LP could achieve as good mean absolute error (MAE), root mean square error (RMSE), prediction interval normalized average width (PINAW) and prediction interval coverage probability (PICP) as 2.4%, 4.5%, 13%, 82%, respectively evaluated on the normalized load data.

1. Introduction
Electric load forecasting is vital for several businesses that are dealing with the operation, trading, and planning of energy, for example, banks, electric utilities, and insurance companies [1]. Probabilistic load forecasting (PLF) has become increasingly important in the last decade, as PLF provides a probability density function (pdf) rather than a point forecast, which might be more valuable to the stakeholders compared with the point forecast [1]. It was estimated that a one percent improvement in the mean absolute percentage error (MAPE) can save hundreds of thousands of dollars for utilities [2].

In the literature there is a plethora of PLF models, for example [3–6]. In the recent years, hybrid models which combine two forecasting methods have been gaining popularity [7]. For example, models using support vector quantile regression (SVQR) along with copula [8], genetic algorithm along with artificial neural networks (GA-ANN) [9], and other hybrid methods in [10–13] have recently been developed. However, there is an underrepresentation of non-parametric methods in the research literature [14]. One of the relatively under used non-parametric methods in the field of PLF is the Gaussian process (GP) [15]. This introduction focuses on reviewing previous implementations of the GP in PLF; for more information on other PLF methods the reader is directed to two recent literature reviews [1,15].

A GP, as defined in [16], is a collection of random variables, any finite number of which have a joint Gaussian distribution. A GP can be defined by the covariance function, or kernel, which is used to describe relationships or nearness between inputs [16]. GP models have been used previously to forecast wind power [17,18], solar power [19],...
electricity price [20,21], and temperature for electric load forecast [22]. It has also been used for clustering of electricity consumption profiles [23], and modeling of households’ response to demand response (DR) signal from an aggregator [24]. Rasmussen and Williams [16] provided a detailed mathematical description of the GP models and their implementation.

Forecasts can be categorized based on the resolution of the forecast. On the monthly resolution, Ploysuwan et al. [25] utilised the GP to forecast the monthly peak load in Thailand. The predictors, or explanatory variables, contained the load of the previous month, the gross domestic product (GDP) and gross national product (GNP). However, when performing forecasting, the authors set the GDP and GNP to zero. Though the authors did not explain how doing so resulted in an accurate forecast, it is likely that the GP model did not put high predictive weight on the GDP and the GNP. Atswathawichok et al. [26] also forecasted the monthly peak load in Thailand. The differences between [25,26] are that several kernels were incorporated in the latter model while the former one only used the Gaussian kernel, and that the latter model only used the previous lags in the time series as predictors while the former model used the GDP and the GNP—at least in training. Direct comparison of the performance between the two models is unfortunately not possible since they forecasted different years, and no error metrics for model evaluation were provided in [25]. Alamaniotis et al. [27] compared four different kernels to forecast the monthly electricity consumption using the monthly data from the four previous years as predictors, i.e., (12 x 4) values. The Matérn kernel demonstrated superior performance compared with the other kernels. Yan et al. [28] used a mixture of kernels to represent the monthly load trend in their GP forecasting model.

On the weekly resolution, Leith et al. [29] employed the GP to forecast the weekly electricity load. Two other models were compared, namely basic structural models (BSM), and seasonal auto-regressive integral (SARI)—described in detail in [30]. The authors concluded that the GP performed better than the other methods provided that the correct kernel is chosen. Moreover, the authors noted that the GP forecast decays to zero outside the space of the training data. The GP forecast decayed to the mean function in the cases of absence of knowledge learned from the training data [16]. This phenomenon can be avoided if (1) one of the predictors is the sequential index or time and the kernel have a periodic component, or (2) the prediction is not far outside the training data. Otherwise, the GP forecast decays to the mean function, which is commonly set to zero for simplicity [16,24].

GPs have also been used on daily resolutions. For example, Mori and Ohmi [31] showed that the GP outperformed multi-layer perceptron (MLP), support vector machine (SVM) and radial basis function network (RBFN) models in forecasting the daily maximum load. In their GP model, the predicted temperature of the forecasted day was the most important predictor for the forecast using the GP. In [32] historical load data from 1 month, and 1–3 years behind were used to predict the load of 30 days ahead. The authors used the Gaussian kernel. They also compared the GP, relevance vector regression (RVR) and autoregressive moving average (ARMA) models. The GP in this paper was the least accurate, i.e., worst performing, method. This bad performance might be attributed to the fact that GPs are designed for predicting single output rather than multiple output, i.e., one-step ahead rather than multi-step ahead prediction [33,34].

An analysis of hourly autocorrelation coefficients was done in [35] in order to select the most important time lags, as predictors, for the GP model. They developed the model for three different distribution feeders each representing a consumption profile: residential sector, non-residential sector, and service sector profiles. Lauret et al. [36] compared GP to ANN and Bayesian NN models in forecasting of the hourly load. The authors used the Gaussian kernel, and the GP model outperformed the other two methods on the test data set. In [37], the GP model suffered from overfitting when forecasting the hourly load using the Gaussian kernel. The GP could not achieve an MAPE of less than 15.9%.

Alamaniotis et al. [38] developed an ensemble GP using kernel machines for hourly forecasting. The authors employed a set of kernels to capture the different features of the demand patterns. A multi-objective optimization was used to find the linear combination between the Matérn, the NN and Rational Quadratic kernels that reduced the error metrics. One of the error metrics they used was the Theil inequality coefficient (Theil). The adoption of the Theil is sometimes accompanied by confusion since there are two versions of the Theil, and each has its own interpretation and parameters [39].

Yang et al. [40] used a hybrid GP quantile regression (GPQR) model to capture the relations between inputs and outputs of their probabilistic forecast. The authors used the current and previous temperature measurements as exogenous variables in their model. The model was able to achieve a prediction interval coverage probability (PICP) and prediction interval normalized average width (PINAW) of 99.4% and 23.8% on a hourly data. Such good results were achieved on a similar sinusoidal dataset using another hybrid forecasting method in [13].

Five minute forecasts were developed in [41] for providing a PLF for the energy intensive enterprises. The authors used the automatic relevance determination (ARD) version of the Gaussian kernel. They also used probabilistic error metrics, however, they did not normalize them. They could achieve a prediction interval with a width of around 100 MW in a 1150 MW peak load. On the minute resolution, Alamaniotis and Tsoukalas [42] compared GPs, with three different kernels, to the ARMA model. The Matérn kernel was ranked first as it performed best in 14 out of 24 trials.

The limitation of the GP to the forecasting of one-step ahead led to the development of the twin Gaussian processes (TGP) in [14,33,37] and other inference methods in [28,43]. Yan et al. [14] adopted the TGP for the PLF on a time series with hour resolution. They used the 24th and the 168th time lag besides six other predictors to perform multi-step ahead forecast of daily and hourly load. The authors showed that the percent error is higher in case of the hourly prediction compared with the daily prediction. On the other hand, Yan et al. [28] compared the recursive and direct multi-step ahead forecasting strategies using conventional GPs. In the recursive approach a single step ahead GP model was used to forecast multiple one-step ahead recursively. The direct approach trains several GP models each for forecasting one specific step. The direct approach outperformed the recursive approach. The uncertainty of the multi-step ahead prediction is supposed to increase with the index of the forecasted steps [43], i.e., in the multi-step ahead forecast the variance of the first step is supposed to be smaller than the variance of the 10th step. To model this phenomenon, Girard et al. [43] used the recursive approach, however, they viewed the lagged values as random variables. As a result, the uncertainty propagated from the forecasted first-step to the last-step ahead.

Learning the GP is based upon selecting the appropriate kernel for the problem at hand and learning the optimal hyperparameters [28]. The choice of kernel should be made by the modeller [16,24,29,38], and it is essential for ensuring the correct performance of the forecast [28]. In the reviewed papers, the Gaussian kernel was used in [25,27,31,32,35–37,40,42]. The ARD version of the Gaussian kernel was used in [41]. This kernel is smooth and might not be suitable for representing many physical processes, which is why the Matérn kernel is a better alternative [16]. The Matérn kernel was used in [27,42]. The NN kernel was employed in [27,42], linear kernel in [27]. Summation of several kernels was done in [29,38]. In [26,28] a similar kernel mixture was used. This mixture was used previously in [16] for a similar problem.

Alamaniotis et al. [38] used deterministic error metrics, like the mean square error (MSE), the root mean square error (RMSE), the mean absolute error (MAE), the MAPE, the maximum absolute percentage error (MAP) and the Theil, for the optimization of the hyperparameters of the kernel. The use of deterministic error metrics for learning the GP
is suboptimal because it only evaluates the accuracy of the predicted mean [16]. It, however, ignores the predicted probability distribution, thus one degree of freedom of the model is left unevaluated [16]. The usage of the marginal likelihood to optimize the hyperparameters is encouraged [16].

1.1. Research gaps and scientific contribution

From the previous review of the GP implementations in the PLF literature, three research gaps were identified. First, the GP by definition assumes that the data is normally distributed. The electricity consumption of households is, however, positively skewed and is better represented by the log-normal distribution [44,45]. Previous work in geostatistics has implemented a version of GP for log-normally distributed data called log-normal kriging which could be used here [46–48]. Kriging is the common name of the GP in the geostatistics field [16]. Second, in most of the previously reviewed papers deterministic error metrics were used to evaluate the performance of the GP [26–29,31,32,35–38,42]. No error metrics were used in [25], and only [40,41] used probabilistic error metrics though they were not normalized. The usage of deterministic error metrics with the PLF methods is a reason for the underdevelopment of the PLF methods, as the underperformance of the PLF methods compared with the deterministic counterparts when using the deterministic error metrics may have led to their underestimation [1]. The use of deterministic error metrics can, nonetheless, provide useful insight specially for PLF with less than weekly range [1]. Third, it was noticed that the prediction intervals, one of the features of PLF, were not presented nor assessed in [25–29,32,37,38,42], only presented in [31,35,36], and presented and assessed using an error metric in [40,41]. This showed that many papers treated the GP method as if it were a deterministic method and did not utilize its full potential. This paper complements the previous research by:

- Implementing the new method, the log-normal kriging, on the positively skewed load data and compare the results with the conventional GP. Here, we refer to the two methods by the names the log-normal process (LP) and GP, respectively.
- Comparing new combinations of kernels.
- Evaluating the performance of the two methods using probabilistic and deterministic error metrics.

Section 2 provides a description of the load data. The methodology of the PLF is presented in Section 3. Results are presented in Section 4 and conclusions are drawn in Section 5.

2. Data

The electric load of one random residential customer was obtained from the dataset presented in [49] and was used to train and test the model. In addition, the loads of five randomly selected customers were used only to perform sensitivity analysis—only described and used in Section 4.3—in order to ensure that the results are applicable to other customers. After removing the anomalous measurements, the load data contained 293 de-identified residential customers from Sydney, Australia. The data were recorded with 30 min resolution for a period of three years. Fig. 1 presents the data series and the probability distribution of the load. The probability distribution of the load shows that it is not normally distributed and is positively skewed. It can be noticed from Fig. 1 that there is an increase in the load during the Southern Hemisphere summer of 2011. During this time, high temperatures were recorded [49]. It can also be noticed that the base load suddenly decreased few months into 2012. The reason for this decrease is unknown.

In order to facilitate the comparison with future models, the data of all customers here were normalized to range between zero and one. Consequently, the error metrics calculated from the methods presented here are comparable with other normalized error metrics like MAPE and RMSPE provided that they were normalized based on the peak value in the data. There were few incidences where the load and the normalized load were zero. The log-normal distribution, which will be defined in Section 3.3, is undefined at zero. Hence, it was assumed that the normalized load data at these points to be a very small non-zero value of 0.01.

3. Methods

Fig. 2 depicts the flow chart of the GP as employed by the previously reviewed papers in the PLF. The core of the GP, box number II in Fig. 2, which is responsible for forecasting is described in Section 3.1. The newly implemented method for PLF, the LP, is described in Section 3.3. Box number I is described in Section 3.2. The error metrics, box number III, are presented in Section 3.4.

3.1. Gaussian process

A detailed description of the mathematical fundamentals of the GP was presented in [16]. Here a brief review of the fundamentals is provided.

A GP is a collection of random variables any finite set of which have a joint Gaussian distribution. Intuitively, a GP can be thought of as defining a distribution over functions, i.e., defining a set of functions that pass through the observations and are otherwise normally distributed. Inference from a GP takes place in the space of functions.

For the location \( \mathbf{x} \), the GP \( f(\mathbf{x}) \) can be defined by the mean function \( m(\mathbf{x}) \) and the kernel \( k(\mathbf{x},\mathbf{x'}) \) such that

\[
f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x},\mathbf{x'})).
\]

The value of the function \( f(\mathbf{x}) \) at \( \mathbf{x} \) can be represented by the random variables. The location \( \mathbf{x} \) need not be unidimensional. In fact, \( \mathbf{x} \in \mathbb{R}^D \) where \( D \) is the number of predictors. The mean function \( m(\mathbf{x}) \) is commonly set to zero [16].

Assuming that there is a set of training observations \( (\mathbf{X}, \mathbf{y}) \) and a set of test observations \( (\mathbf{X}, \mathbf{y'}) \) such that \( \mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)^T \), \( \mathbf{y} = (y_1, y_2, \ldots, y_n)^T \), \( \mathbf{x}_k = (x_{k1}, x_{k2}, \ldots, x_{kn})^T \), \( \mathbf{y}_k = (y_{k1}, y_{k2}, \ldots, y_{kn})^T \) where \( \mathbf{X} \in \mathbb{R}^{n \times D}, \mathbf{y} \in \mathbb{R}^{n \times D}, \) and \( \mathbf{y} \in \mathbb{R}^n \); and by denoting the function values of the training and test data by \( f(\mathbf{X}) \) and \( f(\mathbf{X}) \) respectively; we can incorporate the knowledge provided by the training data in order to perform predictions on the test data through defining the joint distribution

\[
\begin{pmatrix}
f(\mathbf{X}) \\
f(\mathbf{X})
\end{pmatrix} \sim \mathcal{GP} \left( 
\begin{pmatrix}
K(\mathbf{X},\mathbf{X}) & K(\mathbf{X},\mathbf{X'}) \\
K(\mathbf{X'},\mathbf{X}) & K(\mathbf{X'},\mathbf{X'})
\end{pmatrix}
\right)
\]

where \( K(\mathbf{X},\mathbf{X'}) \) is a \( n \times n \) matrix of covariances between all pairs of test and training points, and similarly the other entries \( K(\mathbf{X},\mathbf{X}) \), \( K(\mathbf{X},\mathbf{X'}) \), and \( K(\mathbf{X'},\mathbf{X}) \).

It can be shown through conditioning of the joint Gaussian prior on the observations that

\[
f(\mathbf{X}) \sim \mathcal{GP} \left( \mathbf{f}(\mathbf{X}), \text{cov}(f(\mathbf{X})) \right), \text{where}
\]

\[
\mathbf{f}(\mathbf{X}) = K(\mathbf{X},\mathbf{X}) K(\mathbf{X},\mathbf{X})^{-1} \mathbf{f}(\mathbf{X}),
\]

\[
\text{cov}(f(\mathbf{X})) = K(\mathbf{X},\mathbf{X}) - K(\mathbf{X},\mathbf{X}) K(\mathbf{X},\mathbf{X})^{-1} K(\mathbf{X},\mathbf{X}),
\]

where \( \mathbf{f}(\mathbf{X}) \) is a vector of length \( n \), and \( \text{cov}(f(\mathbf{X})) \) is a \( n \times n \) matrix. If we have a single test point, i.e., \( n_t = 1 \), the results of the GP prediction is a Gaussian distribution with a mean and a variance, estimated from Eqs. (3)–(5). Intuitively in a multi-test points, i.e., \( n_t > 1 \), the GP provide the mean and the variance of every test point, where the variance will be located in the diagonal of the matrix \( \text{cov}(f(\mathbf{X})) \). The variance, as shown in Eq. (5), only depends on the predictors and the kernels [16]. Conversely, the mean depends on the random variable of the observations \( \mathbf{f}(\mathbf{X}) \), as shown in Eq. (4). Eqs. (4) and (5) are
estimated for forecasting, see box II in Fig. 2.

3.2. Covariance functions

The covariance functions, or kernels, encode the relationship between the inputs in the GP (see: Eq. (2)), i.e., they define the nearness or the similarity [16]. Several kernels were compared in this paper. A detailed presentation of many possible kernels can be consulted in [16,50].

The distances \( r \) and \( r' \) between any two locations, defined before in Eq. (1), is estimated using
\[
r = (x-x')(x-x')^T,
\]
\[
r' = (u(x)-u(x'))(u(x)-u(x'))^T,
\]
where \( u(x) = (\sin(\pi x/p), \cos(\pi x/p)) \) and \( p \) is the period. Eq. (7) can be used to convert conventional kernels to periodic ones [50]. The kernels compared in this study are presented in Table 1. The Matérn kernels \( k^1(r) \) and \( k^2(r) \) were used here instead of the conventional Gaussian kernel. The Gaussian kernel is infinitely differentiable which makes it unsuitable for representing many physical processes, and the Matérn kernels represent a better alternative [16]. In addition to conventional kernels, kernels numbers \( i \in 1:6 \), we compared the ARD versions of the same kernels, kernel numbers \( i \in 7:12 \). The ARD kernels assume a vector of characteristic length scales with length \( D \) \((l_1, l_2, ..., l_D)\) instead of

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Mathematical representation</th>
<th>Number of hyperparameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k^1(r) )</td>
<td>( \sigma^2 \left( 1 + \frac{\sqrt{2} r}{l} \right) \exp \left( -\frac{\sqrt{2} r}{l} \right) )</td>
<td>2</td>
</tr>
<tr>
<td>( k^2(r) )</td>
<td>( \sigma^2 \left( 1 + \frac{\sqrt{2} r}{l} + \left( \frac{\sqrt{2} r}{l} \right)^{1/2} \right) \exp \left( -\frac{\sqrt{2} r}{l} \right) )</td>
<td>2</td>
</tr>
<tr>
<td>( k^3() )</td>
<td>( k^1(r) + k^1(r') )</td>
<td>5</td>
</tr>
<tr>
<td>( k^4() )</td>
<td>( k^2(r) + k^2(r') )</td>
<td>5</td>
</tr>
<tr>
<td>( k^5() )</td>
<td>( k^1(r) \times k^1(r') )</td>
<td>5</td>
</tr>
<tr>
<td>( k^6() )</td>
<td>( k^2(r) \times k^2(r') )</td>
<td>5</td>
</tr>
<tr>
<td>( k^7() )</td>
<td>( k_{ARD}(r) \times k_{ARD}(r') )</td>
<td>( D + 1 )</td>
</tr>
<tr>
<td>( k^8() )</td>
<td>( k_{ARD}(r) \times k_{ARD}(r') )</td>
<td>( D + 1 )</td>
</tr>
<tr>
<td>( k^9() )</td>
<td>( k_{ARD}(r) + k_{ARD}(r') )</td>
<td>( 4D + 2 )</td>
</tr>
<tr>
<td>( k^{10}() )</td>
<td>( k_{ARD}(r) + k_{ARD}(r') )</td>
<td>( 4D + 2 )</td>
</tr>
<tr>
<td>( k^{11}() )</td>
<td>( k_{ARD}(r) \times k_{ARD}(r') )</td>
<td>( 4D + 2 )</td>
</tr>
<tr>
<td>( k^{12}() )</td>
<td>( k_{ARD}(r) \times k_{ARD}(r') )</td>
<td>( 4D + 2 )</td>
</tr>
</tbody>
</table>

The single value characteristic length scale \( l \) assumed in the conventional kernels. The same applies for the period \( p \). This enables the kernel to better model the relations between similar predictors. The location \( x \in \mathbb{R}^D \) has \( D \) predictors, and better results might be achieved if each predictor is represented by a different characteristic length scale. More information regarding the number of hyperparameters of each kernel can be consulted in [50].

The summation of kernels, as in kernels 3, 4, 9, and 10, can be conceptualised as a summation of two hidden GPs [51]. Each GP is modelling a specific trend using a single kernel [52]. For example, if there is a GP \( f_i \sim \mathcal{GP}(0,k^1(r)) \) and another GP \( f_j \sim \mathcal{GP}(0,k^2(r')) \), then the summation of the two GPs is a third GP with the summation of the kernels \( f_i + f_j \sim \mathcal{GP}(0,k^1(r) + k^2(r')) \sim \mathcal{GP}(0,k^{10}(r)) \).

The multiplications of kernels, as in kernels 5, 6, 11, and 12, can be conceptualised by looking at kernel 5, for example. A larger similarity, encoded as high \( k^2() \), takes place if both \( k^1(r) \) and \( k^1(r') \) are large. The product between decay kernels, like the Matérn kernels used in this study, and a periodic kernel models a locally periodic trend [53]. This feature might be useful in modeling the trends in electricity load profiles.

For each kernel there are some parameters that determine the behavior of the kernel, namely the signal variance \( \sigma^2 \), the characteristic length scale \( l \), and the period \( p \) for the periodic kernels [16]. These
parameters are called hyperparameters \( \theta \) of the kernel [16]. It can be shown through Bayes’ theorem that in order to choose the hyperparameters the log marginal likelihood should be maximized [16]:

\[
\log(p(f(X)|X, \theta)) = -\frac{1}{2}f(X)^T K(X, X)^{-1} f(X) - \frac{1}{2} \log|K(X, X)| - \frac{n}{2} \log(2\pi),
\]

which is done in see box I in Fig. 2. As stated before, the usage of the marginal likelihood in the optimization of the hyperparameters is encouraged over the use of other deterministic metrics [16]. An initial guess for the hyperparameters need to be inputted externally from the modeller [16].

3.3. Log-normal process

As depicted before in Fig. 1, the load data are positively skewed and better represented by a log-normal distribution. The GP, on the other hand, works best with normally distributed data, and this is a fundamental assumption behind the derivation of the GP. Consequently, the log-normal kriging method has been developed before to forecast log-normally skewed data [46–48]. The method exploits the relation between the log-normal distribution and the Gaussian distribution.

If \( \tau \) is a random variable such that it is log-normally distributed \( \tau \sim LN(\mu, \sigma) \), the random variable \( v = \log(\tau) \) is a random variable that is normally distributed \( v \sim N(\mu, \sigma) \). This property is of particular interest, since taking the natural logarithm of the original data, after being normalized as described in Section 2, and performing conventional GP forecasting on the transformed data can produce more accurate results. Thus, the LP \( g(z) \) can be viewed as a GP on the natural logarithm of the location \( z \);

\[
\text{if } g(z) \sim \mathcal{N}(m(z), k(z, z)), \text{ then } g(\log(z)) \sim \mathcal{N}(m(\log(z)), k(\log(z), \log(z))).
\]

Fig. 3 depicts a flow chart of the LP code. In a conventional GP forecast the results are encoded in the mean and the variance. The variance is used to estimate a prediction interval. In the LP the back transformed results encoded in the expected value \( E[\tau] \), which is adjusted for bias such that \( E[\tau] = \exp(\mu + \sigma^2/2) \approx \exp(\mu)(1 + \sigma^2/2) \) (using Taylor series) [54,55]. The prediction interval borders were back transformed from the Gaussian scale to the log-normal scale, which is the original normalized scale of the data, by taking the exponent with no bias adjustment [55].

Fig. 4 depicts the transformed normalized load data. Note that the data after the transformation were no longer normalized. However, when the GP model produced forecast on the transformed data, the forecast predictions were back transformed to the original normalized scale, i.e., the comparison between the two methods through the evaluations of the error metrics was performed on the same normalized scale.

3.4. Error metrics

Deterministic and probabilistic error metrics were evaluated in this study. Both evaluate the model performance, see Fig. 2. A review on the common available metrics can be found in [15]. The MAE calculates the average error between the actual values and the forecasted values [15]:

\[
\text{MAE} = \frac{1}{n_f} \sum_{i=1}^{n_f} |\hat{y}(x_i) - y_i|,
\]

where the forecasted value \( \hat{y}(x_i) \) for the index \( i \) can be calculated from Eq. (4), and \( y_i \) is the actual value. In addition to the MAE, the RMSE defined by

\[
\text{RMSE} = \sqrt{\frac{1}{n_f} \sum_{i=1}^{n_f} (\hat{y}(x_i) - y_i)^2},
\]

was employed.

In order to assess the results of the two methods based on probabilistic grounds, the prediction interval from the two methods has to be assessed. The PICP evaluates whether the actual value lies within a certain prediction interval limits [57]:

\[
\text{PICP} = \frac{1}{n_f} \sum_{i=1}^{n_f} e_i,
\]

where

\[
e_i = \begin{cases} 1, & \text{if } y_i \in [L_i, U_i]; \\ 0, & \text{if } y_i \notin [L_i, U_i]. \end{cases}
\]

where \( L_i \) and \( U_i \) are the lower and the upper boundaries of the prediction interval for the index \( i \). In this paper, the prediction interval limits were assumed to cover 80% of the pdf of the forecast.

The evaluation of the PICP alone is misleading, since a forecast with very wide prediction interval can result in a high PICP. However, such forecast is of less value compared with another forecast that has a very small PICP.
narrow prediction interval and high PICP. In other words, the sharpness of the forecast is valuable. In fact, the width of the prediction interval measures the informativeness of the forecast [57]. For this the PINAW was used:

\[
PINAW = \frac{1}{nR} \sum_{i=1}^{n} (U_i - L_i),
\]

where \( R \) is the difference between the maximum and the minimum actual values, and it is intended for normalization [57–60]. This metric was often employed in the literature though with different names [57–60], and sometimes not normalized [41].

Different models were compared such that the better model was the one that minimized the MAE, RMSE, and the PINAW and at the same time maximized the PICP.

### 3.5. Model implementation

In this section, a description of the implementation of the model is provided. The GP was implemented using the GPML package developed for MATLAB [50]. This package was developed in [16], and it is publicly available.

Upon analysis of the autocorrelation function of the time series, it was decided to use only the first and the 48th time lags representing 30 min and 24 h lags, respectively, as they both represented the highest correlations. A decision was made to include the hour of the day as predictor, and it was encoded as a dummy variable \( H \) obtained using the `dummyvar` function in MATLAB. The reason for the inclusion of the hour is that the variations in night time load are expected to be small compared with the afternoon load, and the hour might help both the GP and the LP to capture this difference. The dummy variable \( H \) was not subject to any transformation when using the LP.

The data were divided into two parts. The first part, the first two years of recorded data, was used for the training and cross validation (CV) phase. This part of the data was used to compare the two methods, the GP and the LP, as well as the kernels and the two options of the predictors, presented in Table 2. The second part of the data, the last year of recorded data, was used for the testing phase. It was used for the testing of the performance of both the GP and the LP with the chosen kernel and option. The testing results were used to compare the methods to other forecasting methods in the literature.

In the training and CV phase, four weeks of half hourly data were used for learning the hyperparameters for the two methods, and one week was used for CV. In the CV week the trained model was used to forecast the load and the error metrics were evaluated. In other words, the length of the training data \( n \) and the length of the forecast \( n^* \) were 1344 (48 indices a day × 4 weeks × 7 days) and 336 (48 indices a day × 7 days), respectively. This procedure was described before in Fig. 2. Then a shift of one week was done and the previous procedure was repeated until the end of the first part of the data, see Fig. 5. Similar proportions between \( n \) and \( n^* \) were used for the testing phase. The testing results were used to compare the methods to other forecasting methods in the literature.

### Table 2

<table>
<thead>
<tr>
<th>Option</th>
<th>predictors</th>
<th>( D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Option 1</td>
<td>( x_i = (y_{i-1}, y_{i-2}, H) )</td>
<td>26</td>
</tr>
<tr>
<td>Option 2</td>
<td>( x_i = (y_{i-1}, H) )</td>
<td>25</td>
</tr>
</tbody>
</table>

Fig. 4. The transformed load data obtained by taking the natural logarithm of the normalized load data, described before in Fig. 1. On the right hand side, a probability distribution of the transformed load data is presented.

Fig. 5. A diagram representing the two phases: the training and CV phase, and the testing phase. The training and CV phase was performed on all kernels and options of predictors using the two methods, the GP and the LP, while the testing phase was done on one kernel and one option of predictors using the two methods, the GP and the LP. Every square represents one week of data, i.e., 336 observations. The process performed in each step was depicted in Fig. 2.
results of the error metrics were averaged among all the steps of the training and CV. The initial guess of the hyperparameters of the kernels was randomly assigned for each step.

In the testing phase, a similar procedure was followed, such that four weeks of data were used to learn, i.e., train, the hyperparameters and one week to evaluate the error metrics, here called testing. In this phase, only one kernel and one option of predictors were used.

Fig. 6 presents a flow chart for model implementation in MATLAB. Nested for loops were used to train and CV, described in Fig. 5, both methods using several kernels and the two options of predictors, both presented before in Table 1, 2. Then based upon the analysis of the error metrics one kernel and one option were chosen. Based on this choice both methods were tested, described in Fig. 5, and the results were then compared to other forecasting methods in the literature.

4. Results

4.1. Training and cross validation

In this section the results from the comparison between the GP and the LP, the various kernels, and the various options of predictors are presented. The results of the GP forecasting on the first part of the data are presented in Fig. 7. Similarly, the results of the LP on the same data are presented in Fig. 8.

As seen in Fig. 7, the variation in the MAE among all simulations for the GP method was less than 1%. However, there was a trend where the second option, the one that excluded the 48th time lag, of the predictors slightly outperforms the first option, the one that includes the 48th time lag, for all the kernels. This means that the extra information encoded in the 48th lag deteriorated the performance of the GP. Possibly because what was previously mentioned, the Pearson correlation coefficient between the time series and its first and 48th time lag was 0.72.

![Flow chart for the training and CV on all kernels and options of predictors, and for the testing on the selected kernel and option of predictors. The kernels were presented in Table 1, and the options of predictors were presented in Table 2. Both the training and CV phase, and the testing phase were depicted in Fig. 5.](image)

![Heat map of the error metrics resulting from the training and CV using the GP. The y-axis was encoded according to the kernel number—the option of the predictors presented before in Tables 1 and 2, respectively. The shading was scaled based on the specific error metric, i.e., the darker the shading, column wise, the worse the performance.](image)
and 0.43, respectively. A similar pattern of performance was noticed for the RMSE and the PICP metrics. The PINAW, on the other hand, had an inverse relation with the PICP. The difference between the two Matérn kernels was small, especially in the case of the second option for the predictors. Moreover, the ARD kernels seemed to behave worse than their conventional counterparts regarding the MAE and RMSE. This deterioration might be attributed to overfitting of the large number of the hyperparameters. However, they sometimes had a better performance regarding the PICP. Overall, kernel $k(·)$ combined with the second option of predictors produced the best PICP value. This choice resulted in 91.7% and 26.7% for the PICP and PINAW, respectively.

Regarding the LP, as depicted in Fig. 8, the difference between the two options of the predictors was smaller, in many cases negligible, considering the MAE metric. It can be seen that the best performance considering the PICP was for kernel $k(·)$ and the first option of predictors. For this case the PICP, PINAW were 88.7% and 26.1%, respectively. However, the MAE for this case was not the best. Conversely, kernel $k(·)$ and the second option of predictors improved the PINAW by 10.3% and decreased the PICP by 4%. The performance of the ARD kernels $k^{11}(·)$ and $k^{12}(·)$ in the first option of predictors revealed a poor performance in both PICP and PINAW. Though these kernels did not display a similar behavior in the GP, the authors think that these kernels might need careful initial guessing of the hyperparameters. It might have been difficult for the underlying optimization algorithm to find the optimal hyperparameters in the 106 dimensions field. More studies are needed to explain this phenomenon.

Fig. 9 compares the two alternative methods through subtracting the error metrics of the LP from the error metrics of the GP. In most of the cases, the two methods were comparable when considering the MAE and RMSE. However, in most of the cases, the PINAW results of the LP were superior to that of the GP. Consequently, it is encouraged to use the LP if prediction sharpness is paramount. This might be explained by the fact that a normally distributed data is an important assumption in the GP. A normally distributed input, like in the LP, results in a sharper forecast. Sharpness in prediction comes at the cost of PICP performance. Considering the PICP, the GP outperforms the LP.

### 4.2. Testing of the selected model

In this section the results of the performance of the GP and LP were evaluated on the second part of the data, the test data. In this section, only kernel $k^{1}(·)$ and the second option of the predictors were employed. Four weeks of training data were used, and testing of the learned hyperparameters was done on one week of data. Then a shift of one week and a similar procedure was repeated until the end of the testing data.

It can be noticed that there were negligible differences between the results of the MAE and the RMSE in Table 3 and Figs. 7 and 8. A 7.7% and 2.8% improvements in the PINAW were noticed for the GP and the LP, respectively. As stated before, from Eq. (5), the variance of the forecast depends on the predictors [16]. Upon inspection of the predictors, particularly $y_{i-1}$, as described in Table 2, it was found that the standard deviation of this variable was higher in case of the training and CV data compared with testing data, 0.075 compared with 0.06. Both measures were estimated on the normalized untransformed data. This reduction in the variance resulted in a sharper forecast from the
The difference between the improvement of the sharpness of the forecast of the GP and the LP and their respective training and CV results can be explained by the fact that the training data in the LP were distributed according to a more deformed normal distribution compared with the training data, see Fig. 10. Considering the PICP, there was a 4.7% and 2.2% deterioration in the performance for the GP and the LP, respectively.

Table 3 shows that, similar to the training and CV data, the LP is recommended if forecast sharpness is paramount. The LP improved the PINAW by 5.8% compared with the GP. The PICP, on the other hand, deteriorated by 5%.

The forecast of randomly chosen four weeks of the test data is presented in Fig. 11 for the GP, and in Fig. 12 for the LP. By comparing both methods it can be seen that both methods did not capture the sudden unexpected sharp increments in the load. Further research is needed to develop a more suitable implementation of the GP in the PLF in order to capture sudden sharp increments in the load. These implementations might try introducing the temperature as a predictor as it might improve forecasting the sharp increments. The improvement in the sharpness of the forecast through using the LP can be seen in Fig. 12 compared with Fig. 11. The GP provided a forecast with almost constant prediction interval width, see Fig. 13. This indicated that the GP had an almost constant uncertainty on the prediction regardless of the hour. On the other hand, the LP varied the width of the forecasted prediction interval based on the uncertainty. This phenomenon might be explained by the fact that GPs are more suited to model normally distributed data. Moreover, the lower boundary for the prediction interval in case of the GP sometimes become negative. Such a situation is impossible for a positively skewed data like the load.

4.3. Sensitivity analysis on the results of the model

In this section, the developed model was tested on the five remaining customers from the same dataset. Such testing was needed to ensure that the model is capable of producing comparable results even on different customer loads. Five random customers were selected for this analysis. The customers had varying standard deviation of their loads which ranged from 0.56 kW h to 0.13 kW h. In comparison, the standard deviation of the load data of the first customer which were used in the training and CV phase, in Section 4.1, and the testing phase, Section 4.2, was 0.28 kW h.

For each customer, four consecutive training and forecasting steps—the steps were described in Fig. 5—were performed. The starting date for the four steps was randomly selected for each customer. In this analysis and unlike Fig. 5, the load data was not divided into two parts, instead the starting date was randomly selected from the 3 years of

<table>
<thead>
<tr>
<th>Method</th>
<th>MAE</th>
<th>RMSE</th>
<th>PINAW</th>
<th>PICP</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP</td>
<td>0.025</td>
<td>0.049</td>
<td>0.19</td>
<td>0.87</td>
</tr>
<tr>
<td>LP</td>
<td>0.024</td>
<td>0.045</td>
<td>0.13</td>
<td>0.82</td>
</tr>
</tbody>
</table>

The difference of performance between the LP and the GP. The y-axis was encoded according to the kernel number—the option of the predictors presented before in Table 1 and 2, respectively. Shaded areas reflect deterioration in the performance, i.e., the LP is worse than the GP. Note that a negative difference means improvement of performance in all metrics but the PICP.
measured data.

An average of the error metric results and training time of the five customers was obtained for both methods as presented in Fig. 14, 15. Fig. 14 showed that for the GP the first and second kernels had the shortest training time which is less than 37 s per step. In addition, the results of the error metrics of the five customers were comparable with the results obtained on the training and CV phase, see Fig. 7. For the LP, Fig. 15 also showed that the first two kernels had the shortest training time. However, for the first two kernels the training time of the LP was slightly longer than that of the GP in 3 out of 4 cases. The results of the error metrics of the LP were also comparable with that of the training and CV phase, see Fig. 8. The first kernel with the second option of predictors provided satisfactory error results on the five customers and had the shortest training time.

As regards the forecast sharpness, the LP provided sharper forecasts in comparison with the GP. On the other hand the PICP of the GP was higher than that of the LP. These results were similar to the ones previously obtained from the training and CV and the testing phases performed earlier.

4.4. Comparison with other methods in the literature

Considering the probabilistic error metrics, Khosravi et al. [58] was able, using a NN model to forecast two days ahead of hourly load and

---

Fig. 10. Distribution of the values of the predictor $y_{i-1}$, defined in Table 2, for both the first and the second parts of the data. It can be seen that for the second part of the data, the data used for testing, in the LP is a deformed Gaussian distribution. This was not the case for the training and CV data, the first part of the data.

Fig. 11. Plot (a) presents the forecast of four weeks using GP. Plot (b) provides a three days excerpt of the forecast. The shading represents the boundaries of the 80% prediction interval.
with including information about the weather, to achieve 94.1% and 23.1% for the PICP and PINAW, respectively. On the other hand, Quan et al. [57] used a NN model on load data from Singapore and New South Wales, Australia. The authors could achieve a PICP of 91.8% and 90.4% respectively for Singapore and New South Wales on a much smoother aggregated load. Their results for the PINAW for the two load data were 16% and 23.7%, respectively. The authors, unfortunately, did not elaborate on this significant difference in the PINAW results between the two data sets which rather looked similar. Our results for the LP, presented in Fig. 15, surpassed that of [58] by 6.5% in the PINAW but lagged by 11.9% for the PICP. Data regarding the weather might improve the PICP of our forecast. Considering [57], even though their data was with 30 min resolution, they forecasted the aggregate load which was smoother than the individual load of a single household that was forecasted here. Still, the LP, developed here, produced a sharper forecast than they did.

4.5. Practical implementation of the results

Accurate PLF is vital for several stakeholders in the electricity sector, and both the GP and LP methods were shown to achieve competitive results. In addition, both methods are non-parametric and probabilistic. This study is of importance if the data which is supposed to be forecasted is not normally distributed which is the case for the electric load of residential customers [44,45].

The authors encourage the usage of the LP if the sharpness of the
forecast is of important value. This method can be implemented by taking the natural logarithm of the data and performing a GP on the transformed data, thereby making the data more normally distributed.

In case the practical application requires high PICP, the authors recommend using the GP method directly. In other words, data transformation should not be performed. Many publicly available packages, such as GPML package developed for MATLAB [50], can be used for developing the model.

5. Conclusion

GP models are capable of providing a prediction interval representing the uncertainty of the output. This important feature had not been exploited by many of the previous implementations of GP models for PLF. Moreover, the electric load of many residential customers was shown to be positively skewed and to follow a log-normal distribution [44,45]. Thus the implementation of the GP is suboptimal. Instead, the LP should be used. In this paper, the two methods, the GP and the newly implemented LP, were compared along with several combinations of kernels and predictors for PLF. Such a comparison was not done before.

Several conclusions can be drawn from this study:

1. The results from the GP and the LP were comparable with other models in the literature.
2. The Matérn kernel was able to produce satisfactory results.
3. The LP produced sharper forecasts compared with the GP. The sharpness of the forecast can also be improved by selecting a set of predictors with lower dispersion, i.e., smaller standard deviation.
4. The sharpness of the forecast varied throughout the day for the LP reflecting variation in the uncertainty of the forecasted value. Conversely, the sharpness of the forecast was almost constant throughout the day for the GP.
5. Conventional GPs produce prediction intervals that sometimes extend to negative values. In some physical processes negative values are impossible to occur. In our case the residential load, and in absence of local generation, cannot be negative. This phenomenon, of negative prediction intervals, did not and can not take place with the LP.
6. Both the GP and the LP methods were not able to capture the sudden sharp increments of the load and further research is needed in this area. The introduction of the temperature as predictor might enhance the ability of both methods to capture the sharp increments.

Further research can focus more on improving the PICP of the GP.
models. The PICP of the GP models seem to be lagging behind other forecasting methods in the literature.

Acknowledgments

This work was financially supported by SamspEL 2016–2020 in the project “Development and evaluation of forecasting models for solar power and electricity use over space and time”, financed primarily by the Swedish Energy Agency.

References


