Electric Load Forecasting using Multiple Output Gaussian Processes and Multiple Kernel Learning

Alireza Ghasempour and Manel Martínez-Ramón
Department of Electrical and Computer Engineering
The University of New Mexico

2023 IEEE Symposium on Industrial Electronics & Applications (ISIEA) | 979-8-3503-4749-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISIEA58478.2023.10212291

***Abstract*—Electric load forecasting refers to forecasting the electricity demand at aggregated levels. Utilities use the predictions of this technique to keep a balance between electricity generation and consumption at each time and make accurate decision for power system planning, operations, and maintenance, etc. Based on prediction time horizon, electric load forecasting is classified to very short-term, short-term, medium-term, and long-term. In this paper, a multiple output Gaussian processes with multiple kernel learning is proposed to predict short-term electric load forecasting (predicting 24 load values for the next day) based on load, temperature, and dew point values of previous days. Mean absolute percentage error (MAPE) is used as a measure of prediction accuracy. By comparing MAPE values of the proposed method with the persistence method, it can been seen that the proposed method improves the persistence method MAPE up to 4%.**

***Index Terms*—Electric load forecasting, Gaussian processes, mean absolute percentage error, multiple kernel learning, multiple linear regression, persistence method, smart grids.**

I. Introduction

Power grid has some issues such as economics, efficiency, energy security, greenhouse gas and carbon emissions, reliability, and safety. Smart grid is introduced to address these issues. Smart grid as a two-way data communications network integrates with the power grid, uses advanced metering infrastructure to acquire real-time data from smart meters [1] and [2], and analyzes this data to predict generation capacity and electric load consumption. These predictions are very valuable information for utility providers and consumers to manage the power generation and consumption. Electricity generation and consumption should be predicted as accurately as possible. By underestimating load consumption, generated electricity will not be enough to fulfill requested power and there will be power outage and economic loss. Also, by overestimating the power consumption, extra generated electricity will be wasted that increases the cost of electricity for customers.

Electric load forecasting can improve reliability and efficiency of power grid and prediction accuracy. In electric load forecasting, electricity demand at aggregated levels is predicted. Power utilities use it to balance electricity generation and consumption at each time. It is also used for transmission and distribution system planning, demand side management, financial planning, revenue projection, rate design, generating and purchasing electric power, revenue projection, load switching, infrastructure development, power system planning, operations, and maintenance, etc. Electric utilities, independent system operators, regional transmission organizations, regulatory commissions, industrial and big commercial companies, financial institutes, trading firms, and insurance companies need electric load forecasting.

In this paper, a multiple output Gaussian processes (MOGP) with multiple kernel learning (MKL) is proposed and implemented to predict 24 load values for the next day based on load, dew point, and temperature values of previous days. Load consumption is analyzed by investigating the effect of using 1) different kernels, 2) load and temperature values of previous days, 3) load and dew point values of previous days, 4) load, temperature, and dew point values of previous days, on prediction accuracy of electric load values. To evaluate the performance of MOGP with MKL, load, dew point, and temperature values of 2019 and 2020 from the ISO New England database are used and electric loads in 2021 are predicted. Mean absolute percentage error (MAPE) is used as a measure of prediction accuracy.

The remainder of this paper is structured as follows. In section II, a literature review for electric load forecasting is provided. Section III presents Gaussian processes and multiple kernel learning. Section IV presents the results of using the Gaussian process and multiple kernel learning for day-ahead short-term electric load forecasting.

II. Literature Review

Electric load forecasting can be classified into four categories based on the prediction time horizon: 1) very short-term electric load forecasting from some minutes to 1 day, 2) short-term electric load forecasting from 1 day to 2 weeks, 3) medium-term electric load forecasting from 2 weeks to 3 years, and 4) long term electric load forecasting from 3 years to 30 years.

The proposed methods for electric load forecasting can be categorized into two groups: statistical methods and artificial intelligence methods. The difference between statistical and machine learning techniques is that statistical techniques are used to explore and formalize relationships between input and output variables while machine learning techniques are used to learn and understand the data without using explicit or rule-based programming.

This work is partially supported by the National Science Foundation EPSCoR Cooperative Agreement OIA-1757207 and the Felipe VI endowed Chair of the University of New Mexico.

In the following papers, statistical techniques have been used for electric load forecasting. In [3], authors applied autoregressive integrated moving average (ARIMA), autoregressive conditional heteroscedastic (ARCH), generalized ARCH (GARCH), and hybrid ARIMA-GARCH models to forecast the electricity load of network infrastructures in Ghana. Authors of [4] deployed seasonal ARIMA (SARIMA) and Holt-Winter methods for daily peak load energy forecast.

In the following papers, artificial intelligence techniques have been used for electric load forecasting. In [4], a feed forward back propagation neural network (as an artificial neural network-based model) is proposed in addition to Holt-Winter and SARIMA and then a nonlinear optimization model is formulated to determine the best combination of these 3 methods for peak electricity load forecasting. Some deep neural networks and support vector regression model are proposed in [5] for short term load forecasting. Fuzzy time series and probabilistic neural network were considered for the load demand prediction in [6].

III. Gaussian Processes and Multiple Kernel Learning

Gaussian processes regression model with Gaussian noise can be expressed as follows [7]:

$$y_i = \phi(x_i)^T w + n_i, i = 1, \ldots, M, \tag{1}$$

where y_i is the i-th output, x_i is the i-th input vector in a training data set, w is a weight vector that is assumed to have a Gaussian distribution with zero mean and covariance matrix Σ_w, n_i is an additive white Gaussian noise that has independent, identically distributed Gaussian distribution with zero mean and variance σ_n^2, $\phi(x_i)$ is a nonlinear transformation function that maps an input vector x_i into an N-dimensional feature space, M is the number of observations.

It can be shown that the posterior predictive distribution $p(y_*|x_*, X, y)$ has a Gaussian distribution with mean $\bar{y}_*$ and variance $\sigma_{y_*}^2$ as follow:

$$\begin{aligned} \bar{y}_* &= k(x_*, X)\alpha, \\ \sigma_{y_*}^2 &= k(x_*, x_*) - k(x_*, X)(K + \sigma_n^2 I)^{-1} k(x_*, X)^T + \sigma_n^2, \end{aligned} \tag{2}$$

where

$$\begin{aligned} k(x_*, X) &= \phi(x_*)^T \Sigma_w \Phi(X) \\ \Phi(X) &= [\phi(x_1)\ \phi(x_2)\ \ldots\ \phi(x_M)] \\ \alpha &= (K + \sigma_n^2 I)^{-1} y \\ K &= \Phi(X)^T \Sigma_w \Phi(X) \\ k(x_*, x_*) &= \phi(x_*)^T \Sigma_w \phi(x_*), \end{aligned} \tag{3}$$

$k(x_i, x_j)$ is called a covariance function or kernel and it is an inner product with respect to Σ_w. If $\varphi(x_i)$ is define as $\varphi(x_i) = \Sigma_w^{1/2}\phi(x_i)$, then, $k(x_i, x_j)$ can be calculated as a dot product of two functions, $k(x_i, x_j) = \varphi(x_i)^T\varphi(x_j) = \varphi(x_i) \cdot \varphi(x_j)$.

To predict 24 values of the electric load for the next day, 24 GP (as MOGP) are used.

In this paper, the following kernels are used:

1) **Linear kernel:**

$$k(x_1, x_2) = \sigma^2 x_1^T x_2, \tag{4}$$

2) **Matérn kernel:**

$$k(x_1, x_2) = \frac{\sigma^2}{\Gamma(v)2^{v-1}} \left(\frac{\sqrt{2v}\|x_1 - x_2\|_2}{l} \right)^v K_v \left(\frac{\sqrt{2v}\|x_1 - x_2\|_2}{l} \right) \tag{5}$$

3) **Radial basis function (RBF):**

$$k(x_1, x_2) = \sigma^2 \exp\left(- \frac{\|x_1 - x_2\|_2^2}{2l^2} \right), \tag{6}$$

4) **Rational quadratic (RQ):**

$$k(x_1, x_2) = \sigma^2 \left(1 + \frac{\|x_1 - x_2\|_2^2}{2\alpha l^2} \right)^{-\alpha}, \tag{7}$$

where σ^2 is a variance parameter, $\Gamma(.)$ is the gamma function, v is a parameter that controls the smoothness of Matérn kernel, K_v is a modified Bessel function, $\|x\|_2$ denotes the l_2 norm of vector x and l ($l > 0$) is a length-scale parameter, and α ($\alpha > 0$) is a scale mixture parameter.

To select a suitable kernel (among some kernels) and its hyper parameters, cross validation has been used on a validation data set different from training data set. As an alternative, multiple kernel learning techniques are introduced that use a set of kernels (instead of using one kernel and its hyper parameters) and try to learn linear or nonlinear combinations of those kernels. Multiple Kernel Learning has the following benefits: 1) dealing with heterogeneous sources of data (e.g., load, temperature, and dew point), 2) Merging or fusing different heterogeneous information sources, 3) Feature combination (data fusion) and training is done simultaneously, 4) Reducing bias due to kernel selection, 5) Different kernels can measure different notions of similarity and multiple kernel learning method can pick one kernel or combination of kernels which works best.

Multiple kernel learning methods can be classified into 3 categories based on different ways of combining kernels as follows [8]:

1) Methods that combine base kernels linearly and have 2 categories: A) unweighted sum or mean of base kernels to produce combined kernel, B) weighted sum of base kernels which can be defined as follows:

$$k(x_i, x_j) = \sum_{l=1}^{p} \beta_l k_l(x_i^l, x_j^l), \tag{8}$$

where β_l is a coefficient for the l-th base kernel, $k_l(x_i^l, x_j^l)$ is the l-th base kernel function, p is the number of different sources or different kinds of data, $x_i = \{x_i^l\}_{l=1}^p$ is a set of input vectors from p different sources, x_i^l is the i-th input vector of source l, and $k(x_i, x_j)$ is the combination kernel.

2) Methods that use nonlinear functions of base kernels (e.g., multiplication, power, exponentiation, etc.):

For example, polynomial combination of base kernels can be expressed as [9]:

$$k(x_i, x_j) = \sum_{i_1+\cdots+i_p=d} \beta_1^{i_1} \ldots \beta_p^{i_p} k_1^{i_1}(x_i^1, x_j^1) \ldots k_p^{i_p}(x_i^p, x_j^p), \tag{9}$$

where d is a polynomial degree, β_m is a coefficient for the m-th base kernel, and $k_m(x_i^m, x_j^m)$ is the m-th base kernel.

3) Methods that use data-dependent combination of base kernels and assign specific weights to base kernels for each data instance.

In this paper, linear combination of 2 or 3 mentioned kernels (linear, Matérn, RBF, and RQ) is used depend on how many different kind of data is used, e.g., 2 kernels for load and temperature or 3 kernels for load, dew point, an temperature.

IV. Simulation Results

Load, dew point, and temperature data from the ISO New England database for Northeastern Massachusetts load zone are used in this paper. load values in 2021 are used for the test. Normalizing the training data is done by subtracting their means and then dividing the resulting data by their standard deviations (section IV-A provides a comprehensive information about different normalization techniques). The results of the proposed method is compared with the persistence method (as a benchmark) to verify its ability to improve the predictions of the persistence method. The persistent method uses the load values of today as predicted load values for tomorrow.

To evaluate performance of the proposed method, a prediction error metric should be used. Reference [7] provides comprehensive information about prediction error metrics, their categories, formulas, advantageous and drawbacks. In this paper, mean absolute percentage error has been used.

A. Normalization techniques

Normalization methods have been used to scale or transform data to a similar range so that all data features have uniform contributions. These methods can improve the performance of machine learning methods by reducing effects of dominant features and outliers that slow down the learning process in machine learning techniques. Also, gradient descent method converges faster when the data is normalized [10].

Some of the normalization techniques are:

1) **Mean centering normalization**:
This method removes the mean from data as follows:

$$\hat{x}_i = x_i - \mu, \quad i = 1, \ldots, N \tag{10}$$

where $\hat{x}_i$ is the normalized data value, x_i is the unnormalized data value, N is the number of samples in data, and μ is the mean of data defined as follow:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i. \tag{11}$$

2) **Power transformation**:
This technique reduces the effects of heteroscedasticity and transforms the data into homoscedasticitya and makes skewed distribution more symmetric. It is used when standard deviation of the data is proportional to the root of mean of the data [11]. It is defined as follows:

$$\hat{x}_i = \sqrt{x_i} - \frac{1}{N} \sum_{i=1}^{N} \sqrt{x_i}. \tag{12}$$

If feature values are negative, first, $x_i = x_i - \min(x_i)$ should be used and then (12) should be calculated.

3) **Log transformation**:
Log transformation compresses a wide range of feature values to a narrow range of values by applying logarithmic function on feature values, changes the feature distribution, and makes skewed distribution more symmetric,. It removes heteroscedasticity, but it cannot handle zero feature values [12]. It is defined as follows:

$$\hat{x}_i = log_{10}(x_i) - \frac{1}{N} \sum_{i=1}^{N} log_{10}(x_i). \tag{13}$$

4) **Max normalization**:
This normalization scales feature values to $[a, 1]$ or $[-1, b]$ where $a = min(x_i)/max(|x_i|)$ and $b = max(x_i)/max(|x_i|)$. It is sensitive to outliers and can be defined as follows:

$$\hat{x}_i = \frac{x_i}{\max(|x_i|)}, \tag{14}$$

5) **Decimal scaling normalization**: This method is useful when the feature values have logarithmic variations, otherwise, it is not useful. It moves the decimal points of feature values and can be defined as follows:

$$\hat{x}_i = \frac{x_i}{10^j}, \tag{15}$$

where $j = \lfloor log_{10}(max(|x_i|)) \rfloor + 1$ is the smallest integer such that $max(|\hat{x}_i|) < 1$ and $\lfloor x \rfloor$ is a floor function that maps x to the greatest integer less than or equal to x. It is similar to max normalization except it doesn't transfer negative feature values to positive values. For negative values, first another method should be used to map negative values to positive ones.

6) **Adjusted decimal scaling normalization**:
This method is similar to decimal scaling normalization except $j = log_{10}(max(|x_i|)) + 1$.

7) **Unit length normalization**:
This method scales a vector of feature values to have a length one and is defined as follows:

$$\hat{x} = \frac{x}{\|x\|_2}, \tag{16}$$

where x is a feature vector, $\|x\|_2$ is a l_2 norm of x, and $\hat{x}$ is a normalized feature vector.

8) **Min-max normalization**:
This technique converts feature values into a common range, e.g., $[0, 1]$ or $[-1, 1]$. It is also called range

normalization. The following equation maps x_i to the range $[0, 1]$:

$$\hat{x}_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}. \quad (17)$$

To scale feature values to a range of $[a, b]$, the following equation can be used:

$$\hat{x}_i = \frac{(x_i - \min(x_i))(b - a)}{\max(x_i) - \min(x_i)} + a. \quad (18)$$

Calculating minimum and maximum can be affected by outliers. So, min-max normalization is sensitive to outliers.

9) **Adjusted min-max normalization**:
This techniques is defined as follows:

$$\hat{x}_i = \frac{x_i - \min(x_i)}{(\max(x_i) - \min(x_i))^2}. \quad (19)$$

10) **Mean normalization**:
This method is similar to min-max normalization except in the numerator there is μ as follows:

$$\hat{x}_i = \frac{x_i - \mu}{\max(x_i) - \min(x_i)}. \quad (20)$$

11) **Level normalization**
This normalization can be defined as follows:

$$\hat{x}_i = \frac{x_i - \mu}{\mu}. \quad (21)$$

12) **Median and median absolute deviation normalization**
This normalization method is defined based on median and median absolute deviation as follows:

$$\hat{x}_i = \frac{x_i - median(x_i)}{median(|x_i - median(x_i)|)}. \quad (22)$$

Since median absolute deviation (denominator of the ratio) and median are not sensitive to outliers, so this method is robust regarding outlier effect. The normalized data has a zero median.

13) **Robust normalization**
This normalization method is defined based on median and interquartile range (IQR). If data has $2n$ number of values or $2n + 1$ number of values, the first quartile, Q_1, is the median of the n smallest data values, the 2nd quartile, Q_2, is the media of data values, and the 3rd quartile, Q_3, is the median of the n largest data values. $IQR = Q_3 - Q_1$ shows skewness of the data and it can be used to identify outliers. Outliers are defined as feature values that are less than $Q_1 - 1.5IQR$ and/or higher than $Q_3 + 1.5IQR$. Robust normalization shifts the data to have a zero median and is robust to outliers and can be defined as follows:

$$\hat{x}_i = \frac{x_i - median(x_i)}{IQR} = \frac{x_i - Q_2}{Q_3 - Q_1}. \quad (23)$$

14) **Z-score normalization**:
Sometimes, Z-score is called standardization or auto normalization. This technique is useful when there are a few outliers, otherwise, clipping technique should be used before this normalization method. Z-score normalization is used to ensure that the distributions of the normalized features have zero mean and unit variance if the original features have Gaussian distribution. Z-score normalization can be defined as follows:

$$\hat{x}_i = \frac{x_i - \mu}{\sigma}, \quad (24)$$

where σ is the standard deviation of x_i.
In clipping technique, maximum and minimum threshold values are defined for the feature values and values of outliers are changed to these threshold values. For normalization techniques which are sensitive to outliers, clipping technique should be used before them, for other normalization methods, this technique can be used before or after them.

15) **Pareto normalization**:
Svante Wold introduced the concept of this normalization in 1993. This normalization is defined similar to Z-score normalization except in its denominator, it has square root of standard deviation as follows [11]:

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma}}, \quad (25)$$

16) **Variable stability normalization**:
This normalization is also called vast normalization and defined as follows [11]:

$$\hat{x}_i = \frac{x_i - \mu}{\sigma} \cdot \frac{\mu}{\sigma}, \quad (26)$$

17) **Tanh based normalization**: This normalization is based on Hampel estimators, was introduced by Hampel et al. in 1986, and defined as follows [11]:

$$\hat{x}_i = \frac{1}{2}\left(tanh\left(0.01\frac{x_i - \mu^H}{\sigma^H}\right) + 1\right), \quad (27)$$

where μ^H and σ^H are the mean and standard deviation of the Hampel estimators, respectively.

18) **Variant of tanh based normalization**:
This method is similar to tanh based normalization except μ^H and σ^H are replaced by μ and σ, respectively as follows [11]:

$$\hat{x}_i = \frac{1}{2}\left(tanh\left(0.01\frac{x_i - \mu}{\sigma}\right) + 1\right). \quad (28)$$

19) **Logistic sigmoid normalization**:
This method is also called softmax normalization and is based on the logistic sigmoid function as a squashing function to limit data in range of $[0, 1]$ as follows [11]:

$$y = \frac{x_i - \mu}{\sigma}, \quad \hat{x}_i = \frac{1}{1 + \exp(-y)}. \quad (29)$$

20) **Hyperbolic tangent normalization**:
This method uses hyperbolic tangent function to map data in range $[-1, 1]$ as follows [11]:

$$y = \frac{x_i - \mu}{\sigma}, \quad \hat{x}_i = \frac{1 - \exp(-y)}{1 + \exp(-y)}. \quad (30)$$

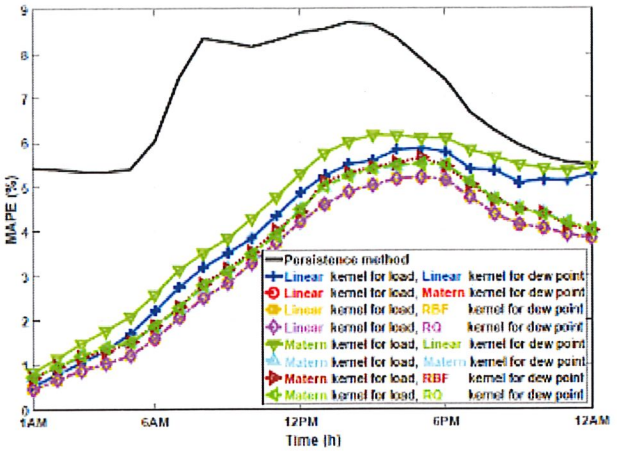

Fig. 1. MAPE values for MOGP with MKL and persistence methods using linear or Matérn kernels for load data and using linear, Matérn, RBF, or RQ kernels for dew point data.

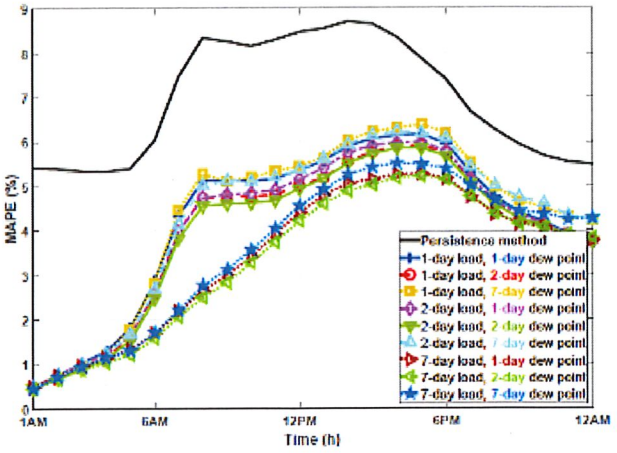

Fig. 3. MAPE values for MOGP with MKL and persistence methods using linear kernel for load data and RQ kernel for dew point data with different number of days of load and dew point data.

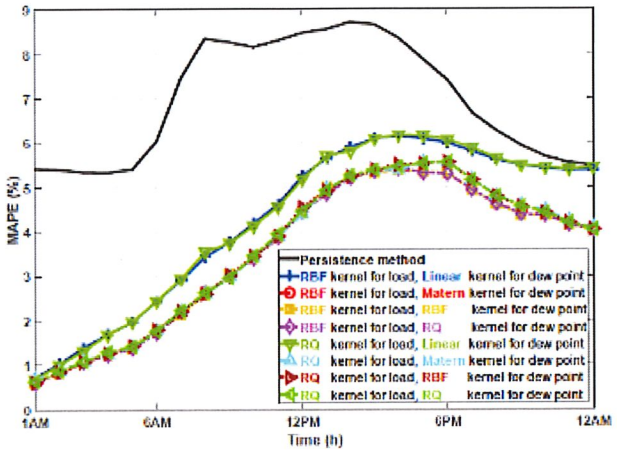

Fig. 2. MAPE values for MOGP with MKL and persistence methods using RBF or RQ kernels for load data and using linear, Matérn, RBF, or RQ kernels for dew point data.

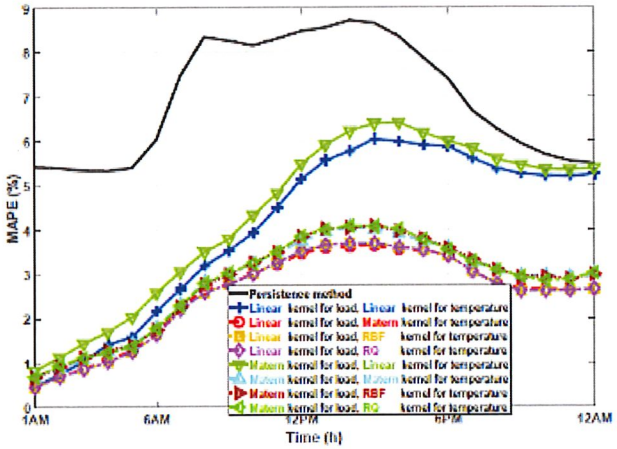

Fig. 4. MAPE values for MOGP with MKL and persistence methods using linear or Matérn kernels for load data and using linear, Matérn, RBF, or RQ kernels for temperature data.

B. Using previous load and dew point data for prediction

The following experiments show the results of our simulations when 1) load and dew point values of 2019 and 2020 are used for training and 2) load values of 7 consecutive days and dew point values of 2 consecutive days are used to predict 24 load values of the next day. Fig. 1 and 2 show the effect of using linear, Matérn, RBF, or RQ kernels on MAPE values of MOGP with MKL and persistence methods. Based on Fig. 1 and 2, linear kernel for load data and Matérn, RBF, or RQ kernel for dew point data should be used. Fig. 3 shows the effect of using different number of days of load and dew point data on MAPE values of MOGP with MKL and persistence methods when linear kernel for load data and RQ kernel for dew point data are used. Based on Fig. 3, load values of 7 consecutive previous days (7-day in legend) and dew point values of 2 consecutive days (2-day in legend which includes dew points values of the prediction day and the day before that) should be used to get the best result of predicting load values of the next day.

C. Using previous load and temperature data for prediction

In this section, the results of our simulations are discussed when 1) load and temperature values of 2019 and 2020 are used for training and 2) load values of 7 consecutive days and temperature values of 2 consecutive days to predict 24 load values of the next day, are used. Fig. 4 and 5 show the effect of using linear, Matérn, RBF, and RQ kernels on MAPE values of MOGP with MKL and persistence methods. Based on Fig. 4 and 5, a linear kernel for load data and Matérn, RBF, or RQ for temperature data should be used.

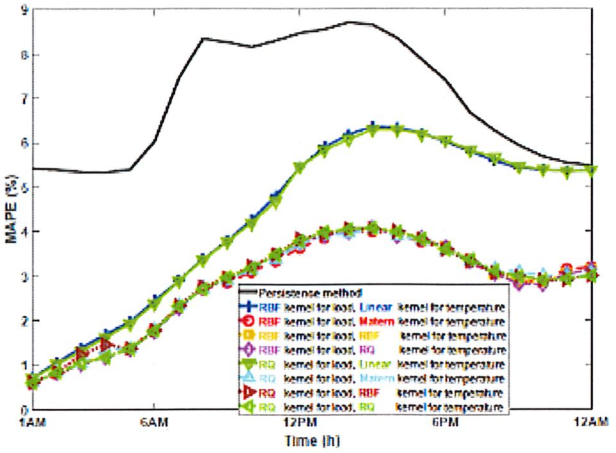

Fig. 5. MAPE values for MOGP with MKL and persistence methods using RBF or RQ kernels for load data and using linear, Matérn, RBF, or RQ kernels for temperature data.

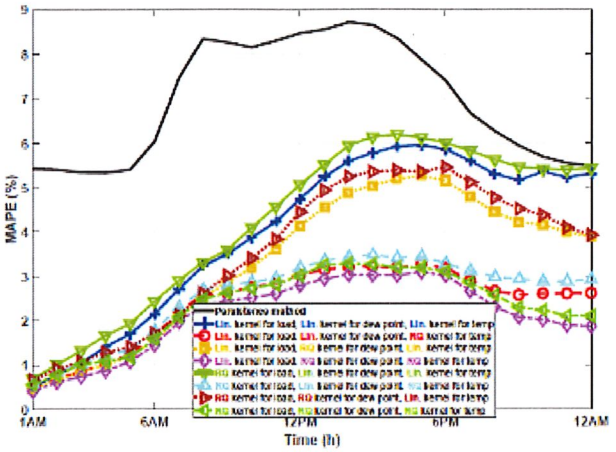

Fig. 6. MAPE values for MOGP with MKL and persistence methods using linear and RQ kernels.

D. Using load, dew point, and temperature data for prediction

In this section, the results of our simulations are discussed when 1) load, dew point, and temperature values of 2019 and 2020 are used for training and 2) load values of 7 consecutive days, dew point values of 2 consecutive days, and temperature values of 2 consecutive days are used. Fig. 6 shows the effect of using linear and RQ kernels on MAPE values of MOGP with MKL and persistence methods. From Fig. 6, it can been seen that the performance of MOGP with MKL when a linear kernel for load data and RQ kernel for dew point and temperature data are used is better than other kernel combinations. Fig. 7 shows the effect of using linear and RQ kernels and load, dew point, and/or temperature values on MAPE values of MOGP with MKL and persistence methods. From Fig. 7, it can been seen that the performance of MOGP with MKL when a linear kernel for load and a RQ kernel for dew point and temperature data are used is better than other

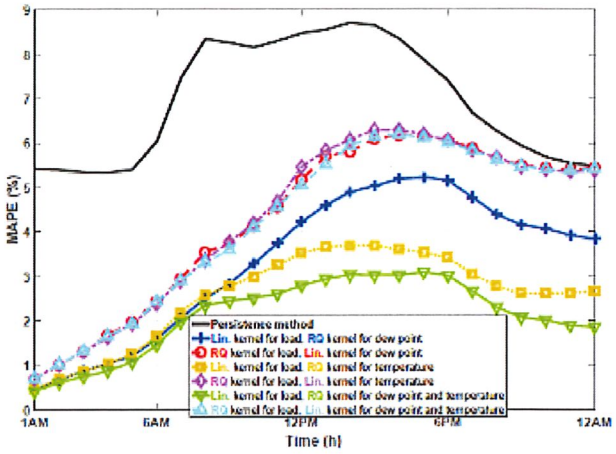

Fig. 7. MAPE values for MOGP technique and MKl and persistence method using 2 different kernels and different types of data.

kernel combinations. When a linear kernel for load data and a RQ kernel for temperature data are used, a better result is obtained than using a linear kernel for load data and a RQ kernel for dew point data. This means temperature data has more effect on MAPE values of MOGP than dew point data. Using suitable kernels with 3 types of data (load, dew point, and temperature) gives the better result compared to using suitable kernels with only types of data.

REFERENCES

[1] A. Ghasempour, "Optimized scalable decentralized hybrid advanced metering infrastructure for smart grid," IEEE International Conference on Smart Grid Communications (SmartGridComm), 2015, pp. 223-228.

[2] A. Ghasempour and J. H. Gunther, "Finding the optimal number of aggregators in machine-to-machine advanced metering infrastructure architecture of smart grid based on cost, delay, and energy consumption," IEEE Annual Consumer Comm. & Networking Conf., 2016, pp. 960-3.

[3] F. K. Oduro-Gyimah, M. A. Boateng, U. Abdallah, K. O. Boateng, D. M. O. Adjin and J. Q. Azasoo, "Forecasting Electricity Load of Network Infrastructure Sharing Mobile Sites in Ghana," Int. Conf. on Cyber Security and Internet of Things, 2021, pp. 37-42.

[4] A. A. Sleem, A. R. Mohammed, S. A. Shkoor and H. Saleh, "Peak Forecasting for Electricity Loads in Jordan Using a Weighted Combination of Feed Forward Back Propagation Neural Network and Holt-Winter," Int. Conf. on Intelligent Engineering and Management, 2022, pp. 226-231.

[5] S. Dua, S. Gautam, M. Garg, R. Mahla, M. Chaudhary and S. Vadhera, "Short Term Load Forecasting using Machine Learning Techniques," International Conference on Intelligent Technologies, 2022, pp. 1-6.

[6] K. Rama and N. G. Sahib-Kaudeer, "A Predictive Analysis of Residential Electrical Load Demand in Mauritius," IEEE International Energy Conference, 2020, pp. 1035-1040.

[7] A. Ghasempour and M. Martínez-Ramón, "Short-Term Electric Load Prediction in Smart Grid using Multi-Output Gaussian Processes Regression," IEEE Kansas Power and Energy Conference, 2023, pp. 1-6.

[8] M. Gönen and E. Alpaydin, "Multiple Kernel Learning Algorithms," J. of Machine Learning Research, vol. 12, no. 64, pp. 2211–2268, 2011.

[9] C. Cortes, M. Mohri, and A. Rostamizadeh, "Learning non-linear combinations of kernels," International Conference on Neural Information Processing Systems, 2009, pp. 396–404.

[10] J. Watt, R. Borhani, and A. K. Katsaggelos, Machine Learning Refined: Foundations, Algorithms, and Applications. Cambridge Uni. Press, 2016.

[11] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," Applied Soft Computing, vol. 97, 2019.

[12] Robert A. van den Berg et al., "Centering, scaling, and transformations: improving the biological information content of metabolomics data," BMC Genomics, vol. 7, 2006.