INFORMS Journal on Data Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Spatio-Temporal Time Series Forecasting Using an Iterative Kernel-Based Regression

Ben Hen, Neta Rabin

To cite this article:
Ben Hen, Neta Rabin (2025) Spatio-Temporal Time Series Forecasting Using an Iterative Kernel-Based Regression. INFORMS Journal on Data Science 4(1):20-32. https://doi.org/10.1287/ijds.2023.0019

https://pubsonline.informs.org/journal/ijds

INFORMS JOURNAL ON DATA SCIENCE
Vol. 4, No. 1, January–March 2025, pp. 20–32
ISSN 2694-4022 (print), ISSN 2694-4030 (online)

Spatio-Temporal Time Series Forecasting Using an Iterative Kernel-Based Regression

Ben Hen,[a] **Neta Rabin**[a,*]

[a] Department of Industrial Engineering, Tel-Aviv University, Tel-Aviv 6139001, Israel
*Corresponding author
Contact: benhen@mail.tau.ac.il (BH); netara@tauex.tau.ac.il, https://orcid.org/0000-0002-1807-2446 (NR)

Received: September 19, 2023
Revised: February 26, 2024
Accepted: March 10, 2024
Published Online in Articles in Advance: April 16, 2024

https://doi.org/10.1287/ijds.2023.0019

Abstract. Spatio-temporal time series analysis is a growing area of research that includes different types of tasks, such as forecasting, prediction, clustering, and visualization. In many domains, like epidemiology or economics, time series data are collected to describe the observed phenomenon in particular locations over a predefined time slot and predict future behavior. Regression methods provide a simple mechanism for evaluating empirical functions over scattered data points. In particular, kernel-based regressions are suitable for cases in which the relationship between the data points and the function is not linear. In this work, we propose a kernel-based iterative regression model, which fuses data from several spatial locations for improving the forecasting accuracy of a given time series. In more detail, the proposed method approximates and extends a function based on two or more spatial input modalities coded by a series of multiscale kernels, which are averaged as a convex combination. The proposed spatio-temporal regression resembles ideas that are present in deep learning architectures, such as passing information between scales. Nevertheless, the construction is easy to implement, and it is also suitable for modeling data sets of limited size. Experimental results demonstrate the proposed model for solar energy prediction, forecasting epidemiology infections, and future number of fire events. The method is compared with well-known regression techniques and highlights the benefits of the proposed model in terms of accuracy and flexibility. The reliable outcome of the proposed model and its nonparametric nature yield a robust tool to be integrated as a forecasting component in wide range of decision support systems that analyze time series data.

History: Kwok-Leung Tsui served as the senior editor for this article.
Funding: This research was supported by the Israel Science Foundation [Grant 1144/20] and partly supported by the Ministry of Science and Technology, Israel [Grant 5614].
Data Ethics & Reproducibility Note: The code capsule is available on Code Ocean at https://codeocean.com/capsule/6417440/tree and in the e-Companion to this article (available at https://doi.org/10.1287/ijds.2023.0019).

Keywords: kernel regression • spatio-temporal • multiscale

1. Introduction

Regression methods provide a simple mechanism for evaluating empirical functions over scattered data points, allowing to forecast future values. In particular, kernel-based regressions, such as the Nadaraya-Watson estimator (Nadaraya 1964), are suitable for cases in which the relationship between the data points and the function is not linear. Successive applications of the Nadarya-Watson estimator, which is also known as iterative bias reduction (Cornillon et al. 2013), yield a multiscale model that generates a smoothed version of a function by using Gaussian kernels to smooth the residuals. This iterative model has been successfully applied for forecasting electricity prices (Cornillon et al. 2015), and shown to outperform Seasonal Auto-Regressive Integrated Moving Average and nonlinear additive autoregressive models (Wood 2017).

A slightly modified version of iterative bias reduction is the auto-adaptive Laplacian pyramids (ALPs), in which the scale of the kernel decreases at each iteration (Rabin and Coifman 2012, Fernández et al. 2020). Like in iterative bias reduction, Laplacian pyramids build a multiscale data representation, which can be extended to handle newly arrived data points at each scale by extending the averaged residuals. The iterations stop using an incorporated leave-one-out cross-validation method that avoids overfitting. The method is simple to implement, results in accurate extensions, and does not require carefully tuned parameters. ALP has been applied for function extension (Li et al. 2014), where it was shown to be a preferred technique for speech enhancement, prediction (Comeau et al. 2019, Rabin et al. 2023), imputation (Rabin and Fishelov 2019), and

other domains. Building on the previous multiscale regression, we propose a modification of ALP that is suitable for a data-fusion setting, with an emphasis on spatio-temporal forecasting.

Spatio-temporal time series analysis is a growing area of research. It is applied when data are collected across space and time and describes a phenomenon in a set of particular locations over a predefined time slot (Rao et al. 2012). Spatio-temporal data mining research includes different types of tasks, such as forecasting and prediction, clustering, visualization, and its applications span across various domains (Shekhar et al. 2015, Hamdi et al. 2022). Forecasting models for spatio-temporal time series can be coarsely separated into three categories: statistical models, machine learning models, and deep learning models. Statistical based methods offer a convenient framework for modeling spatio-temporal data. For example, Yang et al. (2015) performed solar radiance forecasting using a Lasso-based model; however, this method was developed for very short-term time series. The spatio-temporal auto-regressive moving average (STARMA) (Dambreville et al. 2014) and the spatio-temporal vector autoregression (Yang et al. 2014, André et al. 2016) has shown good predictive performance in predicting future solar irradiance based on past observations from relevant neighboring locations. Nevertheless, traditional Auto-Regressive Integrated Moving Average-based methods require careful optimization of the parameters, as the number of input modalities increase. This also results in more complex models and thus aggravates the model selection procedure.

Machine learning models, like tree-based methods, have achieved satisfactory forecasting results for spatio-temporal tasks. These models are convenient for studies in which the input data are tabular. For example, in Xia and Stewart (2023), a large number of variables were used as input for analyzing opioid-involved deaths during the COVID-19 pandemic. To incorporate the spatial information into tree models, one has to generate appropriate features and feed these into the tree-based model. Hengl et al. (2018) suggested adding the buffer distances from observation points as additional explanatory variables that allow the model to learn the spatial relationships. In Liu et al. (2020), an XGboost model for estimating the ozone level in China was proposed. The model relayed on a large number of input variables from multiple sources, and spatio-temporal information was coded as additional computed variables that were given as input to the model.

Deep learning techniques for time series forecasting have rapidly developed over the last few years. Recurrent neural networks (RNNs) along with modifications that deal with noisy data (Zhang et al. 2023) and architecture modifications that capture long-term dependencies (Shih et al. 2019) are two examples that demonstrate the models that are developed for single and multivariate time series forecasting. In Khaldi et al. (2023), the authors tested RNNs with different types of cell structures (such as Elman RNN, long-short–term memory (LSTM), or gated recurrent unit (GRU)) to see which best forecasts different types of time series (deterministic, random walk, nonlinear, long memory, and chaotic). They claim that there is a need for a guiding tool that would help the user fit the best type of RNN cell to the studied data.

Deep learning methods that aim to specifically tackle spatio-temporal forecasting often model the spatial interactions as a graph. A social spatio-temporal graph convolutional neural network was proposed in Mohamed et al. (2020) for the task of human trajectory prediction. The graph, coded by an adjacency matrix, captured the social interactions between pedestrians and their temporal dynamics. This was shown to improve previous methods; however, the application of such a model for a different type of task may require careful adjustments. Two known graph neural network libraries that are suited to deal with temporal graph learning problems, including forecasting, are PyTorch Geometric Temporal (PyGT) (Rozemberczki et al. 2021) and DynaGraph (Guan et al. 2022). Nevertheless, there are several open challenges in this area; these architectures may result in oversmoothing or oversquashing, there are only several works that explore the expressive power of such models (Souza et al. 2022, Beddar-Wiesing et al. 2024), and there currently does not exist a good evaluation benchmark. Although deep learning methods have been seen to produce state-of-the-art results, their black box nature (Wang et al. 2020), together with the required careful training parameters setting and configuration procedure, still remains a limitation.

Transformers have also gained attention in recent years due to their ability to learn relationship in sequential data like words, and they have also been successful for vision applications (Lin et al. 2022). Recent modifications of transformers for time series learning tasks are described in Wen et al. (2023), focusing on time series forecasting, spatio-temporal forecasting, and event forecasting. Spatio-temporal transformers have been adapted and applied to specific complex tasks like traffic forecasting (Cai et al. 2020, Xu et al. 2020), climate forecasting (Gao et al. 2022), and air pollution forecasting (Liang et al. 2023). Despite these notable achievements, the work of Zeng et al. (2023) questions the reliability of transformers for modeling the temporal relationships when the task is long-term time series forecasting. They show that simple models outperform sophisticated transformers in this case.

Building on successful applications of the single-modality ALP method for prediction (Hen et al. 2022), imputation (Rabin and Fishelov 2019, Rabin 2020), and out-of-sample extension (Comeau et al. 2017), we propose a natural extension of this method with multiple input modalities. A first attempt for such a model was

suggested in Rabin et al. (2023), where time trajectories of the derivative acted as a second modality to improve the predictions in oscillatory regions of the data. The idea of combining kernels for enhancing the performance of machine learning algorithms has been investigated in different contexts. Multikernel learning, which combines different kernel functions on the same input data, has been suggested for metric learning (Wang et al. 2011) and shown to outperform metric learning that uses a single kernel. For spatio-temporal applications, a multimodal kernel was recently suggested in Xu et al. (2023) for representing spatial features of thermal dynamics. Similar to this work, each kernel holds information from a different spatial location, and one global function is generated by combining these kernels. However, the scale of each kernel is fixed rather than multiscale. Our proposed extension of the single ALP model to the spatio-temporal auto-adaptive Laplacian pyramids (SALP) model can capture complex nonlinear structure both in time and in space.

The strengths of the SALP method are in its ability to approximate the data while capturing fine details; this derives from its successive multiscale construction. In addition, because the input data are coded by kernels, the framework provides a simple way to include several input modalities, which can be of different types and scales. Coding these inputs by kernels allows adding additional information on the pairwise distances in the data; this description is also suitable for data sets that behave in a nonlinear manner. Constructing a convex combination of these kernels expresses the desired contribution of each modality to the model. Our multiscale kernels-based approach differs from other spatio-temporal time series models in several aspects. Regression trees for example take the original data as input; therefore, they may fail to model the geometry of the data as coded by the kernels. SALP is suitable for processing large or small size data sets; the latter (small data sets) is often a limitation for deep learning models. As a nonparametric approach, there are no underlying assumptions of the type of model that describes the learned phenomena. This is an advantage when one compares the proposed approach to ARMA based techniques, which are parametric methods. Last, evoking SALP requires the user to define the desired influence of each modality to the model and an initial data-driven scale for the kernels. These input parameters are limited in number; moreover, setting the contribution of each spatial modality provides a clear understanding between the input and output of the model.

In this paper, we outline the general ALP framework for forecasting based on multiple input modalities and demonstrate this for a spatio-temporal setting. The model is implemented by constructing a convex combination of kernels, where each kernel carries information from one modality (here, one spatial location). We denote the model by SALP and demonstrate its performance on three different types of data sets. The first is solar energy prediction, the second is prediction of chickenpox cases, and the third is forecasting the number of future fire events. We demonstrate how the forecasting errors for a single location decrease when neighboring modalities are incorporated into the model and compare the proposed SALP with other spatio-temporal techniques. From a decision support perspective, the multiscale construction of the model allows the user to gain insight on the long- and short-term dynamics of the data in different scales. By setting different convex combination of the input modalities, one can learn the independence or dependence of the time series that is measured at a single location with respect to its spatial neighbors. Although not emphasized in this work, such an analysis can characterize a spatial-temporal data set in other means beyond forecasting (e.g., detection of anomalous locations) and provide such an analysis in different scales.

The rest of the paper is organized as follows. Section 2 describes the single modality ALP regression model. Section 3 describes our proposed extension of the model for a spatio-temporal setting. Experimental results including description of the tested data sets are described in Section 4. Finally, a discussion is provided in Section 5.

2. Mathematical Background

2.1. ALPs

ALPs (Fernández et al. 2020) are an iterative kernel-based regression model, suited for capturing the relationship between a set of scattered data points and a target function. Let $X = \{x_i\}_{i=1}^{N}, \quad x_i \in \mathbb{R}^M$ be the sample data set. The algorithm approximates a function f defined over X by constructing a series of functions f_0, $f_1, f_2, \ldots$ obtained by several refinements $d_1, d_2, \ldots$ over the approximation errors.

In more detail, a first Gaussian kernel with Euclidean distances and a wide initial scale σ is defined by

$$K_0 = k_0(x_i, x_j) = e^{\frac{-\|x_i - x_j\|^2}{\sigma^2}}, \text{ where } x_i, x_j \in X. \tag{1}$$

The smoothing operator P_0 is constructed as the row-stochastic normalized kernel matrix

$$P_0 = p_0(x_i, x_j) = \frac{k_0(x_i, x_j)}{\sum_{x_j \in X} k_0(x_i, x_j)}. \tag{2}$$

A first coarse of f is then generated by

$$s_0(x_i) = \sum_{x_j \in X} P_0(x_i, x_j) f(x_j). \tag{3}$$

This approximation captures the low frequencies of the function. We denote the first coarse approximation of f as $f_0(x_i) = s_0(x_i)$. The difference $d_1(x_i) = f(x_i) - f_0(x_i) = f(x_i) - s_0(x_i)$ is averaged by a finer kernel P_1 that is constructed with $\sigma = \sigma/2$. This yields with a finer

representation of f, $f_1(x_i) = f_0(x_i) + s_1(x_i)$, where $s_1(x_i) = \sum_{x_j \in X} P_1(x_i, x_j) d_1(x_j)$. In general, for $\ell = 1, 2, 3 \ldots$, we have $d_\ell = f - f_{\ell-1}$, and

$$f_\ell(x_i) = f_{\ell-1}(x_i) + s_\ell(x_i) = f_{\ell-1}(x_i) + \sum_{x_j \in X} P_\ell(x_i, x_j) d_\ell(x_j), \tag{4}$$

where

$$P_\ell = p_\ell(x_i, x_j) = \frac{k_\ell(x_i, x_j)}{\sum_{x_j \in X} k_\ell(x_i, x_j)}.$$

The variable K_ℓ is constructed similarly to K_0 in Equation (1) but with $\sigma = \sigma/2^\ell$.

In Fernández et al. (2020), it was suggested to set the initial value for σ as $\sigma = 10 \max(\mathcal{W}_{ij})$, where $\mathcal{W}_{ij} = \|x_i - x_j\|^2$, holds the Euclidean distances between pairs of data points.

Extension of the model to new points is straightforward. Given a new point $\tilde{x}$, the multiscale representations $f_0, f_1, \ldots, f_\ell$ are extended to evaluate $f_\ell(\tilde{x})$. First, s_0 is extended by

$$s_0(\tilde{x}) = \sum_{x_j \in X} P_0(\tilde{x}, x_j) f(x_j), \tag{5}$$

where $P_0(\tilde{x}, x_j)$ is the row-normalized output of $K_0(\tilde{x}, x_j) = \exp(-\|\tilde{x} - x_j\|^2/\sigma^2)$. Similarly, the kernels that form the finer resolutions $s_1, \ldots, s_l$ are extended, resulting with

$$f_\ell(\tilde{x}) = f_{\ell-1}(\tilde{x}) + \sum_{x_j \in X} P_\ell(\tilde{x}, x_j) d_\ell(x_j). \tag{6}$$

The iterative train (approximation) algorithm stops once $\text{err}_\ell = \|f - f_\ell\|$ is smaller than a predefined threshold. Because the error of the method decays fast, setting a small threshold may easily result in f_ℓ that almost interpolates f (Kang and Joseph 2016). One way for selecting a preferable scale by applying K-fold cross-validation, in this case, for setting the optimal number of iterations. Here, one may easily incorporate a leave-one-out cross-validation (LOOCV) by slightly modifying the kernels. Modifying the previous kernels to have a zero diagonal, for each scale ℓ by setting $K_\ell(x_i, x_i) = 0$ allows to run the train phase for a predefined number of iterations, here defined by maxIter, and then find the optimal stopping scale L by means of the minimum error. This results in a series of functions $f_0, f_1, \ldots, f_L$ that approximate f in a multiscale manner. The suggested modification makes the method stable and automatic in terms of parameters selection without extra cost.

Algorithms 1 and 2 describe the train and test of the ALP procedures.

Algorithm 1 (ALP Train Model)

Input: $\{x_i, f(x_i)\}_{i=1}^n, \sigma, max_{its}$
Output: Train Model: $f_0(x_i), d_1(x_i), \ldots, d_L(x_i), L$ – stopping scale

Compute $K_0 = \exp(-\|x_i - x_j\|^2/\sigma^2)$, $K_0(x_i, x_i) = 0$.
Normalize K_0 and yield P_0 (see Equation (2)).
Compute $s_0(x_i) = \sum_{x_j \in X} P_0(x_i, x_j) f(x_j)$, set $f_0 = s_0$.
$\sigma = \sigma/2, \ell = 1$.
while $(\ell < max_{its})$ **do**
  Compute $K_\ell = \exp(-\|x_i - x_j\|^2/\sigma^2)$, $K_\ell(x_i, x_i) = 0$.
  Normalize K_ℓ and yield P_ℓ.
  $f_\ell(x_i) = f_{\ell-1}(x_i) + s_\ell(x_i)$, as described in Equation (4).
  $d_\ell(x_i) = f(x_i) - f_\ell(x_i)$.
  $err_\ell = \|d_\ell\|^2$.
  $\sigma = \sigma/2, \quad \ell = \ell + 1$.
$L \leftarrow argmin_\ell(err_\ell)$, stopping scale.

Algorithm 2 (ALP Prediction)

Input: $\{x_i, f(x_i)\}_{i=1}^n$, $\{d_1, d_2, \ldots, d_L\}$, σ, L, test point - $\tilde{x}$
Output: $f_L(\tilde{x})$

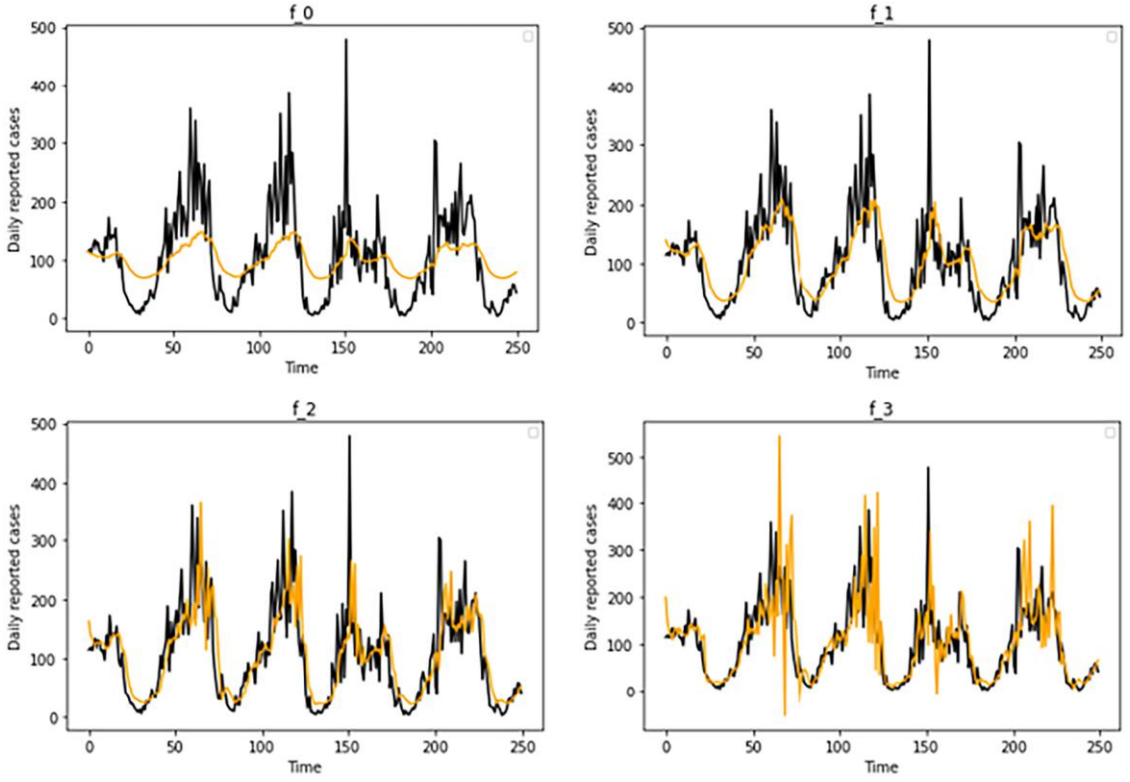
$K_0(\tilde{x}, x_j) = e^{\frac{-\|\tilde{x} - x_j\|^2}{\sigma^2}}$.
$P_0(\tilde{x}, x_j) = \frac{k_0(\tilde{x}, x_j)}{\sum_{x_j \in X} k_0(\tilde{x}, x_j)}$.
$f_0(\tilde{x}) = s_0(\tilde{x}) = \sum_{x_j \in X} P_0(\tilde{x}, x_j) f(x_j)$.
$\sigma = \sigma/2$
for $\ell = 1$ **to** L **do**
  $K_\ell(\tilde{x}, x_j) = e^{\frac{-\|\tilde{x} - x_j\|^2}{\sigma^2}}$.
  $P_\ell(\tilde{x}, x_j) = \frac{k_\ell(\tilde{x}, x_j)}{\sum_{x_j \in X} k_\ell(\tilde{x}, x_j)}$.
  $f_\ell(\tilde{x}) = f_{\ell-1}(\tilde{x}) + \sum_{x_j \in X} P_\ell(\tilde{x}, x_j) d_\ell(x_j)$.
  $\sigma = \sigma/2, \quad \ell = \ell + 1$.

2.2. ALPs for Time Series Forecasting

Because this work focuses on time series forecasting applications, we describe the setting that was used for ALPs in this work. Given a time series data $y(t)$, where $1 \le t \le n$, we wish to make a future prediction, $y(n+1)$ based on short-term trajectories from $y(t)$. Denote the set of overlapping short-term trajectories of length k by $X = \{x(t, :)\}_{t=1}^{n-k+1}$. These are constructed using an overlapping sliding window over $y(t)$. The training set is composed of pairs $\{x(t, :), f(t)\}$, where $f(t) = y(t + k + 1)$ is the target. Algorithm 1 is then applied to $\{x(t, :), f(t)\}$ to yield a multiscale model of $f(t)$. Given a new time-trajectory sample $x(\tilde{t}, :)$, the task is to predict $f(\tilde{t})$. The prediction is done by evoking Algorithm 2 on the constructed train model and the new test point $x(\tilde{t}, :)$. The predicted value is the output $f_L(\tilde{t})$.

For a better understanding of the proposed method and its advantages, we illustrate in this section the classic ALP algorithm on one time series from the Hungarian chickenpox cases data set (described in Section 4.1), a spatio-temporal data set of the weekly chickenpox

Figure 1. (Color online) Training Phase of the ALP Model



Notes. The time series holds the weekly number of reported chickenpox cases in Budapest, while the predicted functions approximate the train data in different scales. The first four steps (of 20) on the Budapest time series example are plotted in the four panels. f_2 was found to be the Lth iteration, where the minimum value of the LOOCV is obtained. It can be seen that f_3 is already too detailed due to its small kernel scale, indeed the iterations stop at f_2.

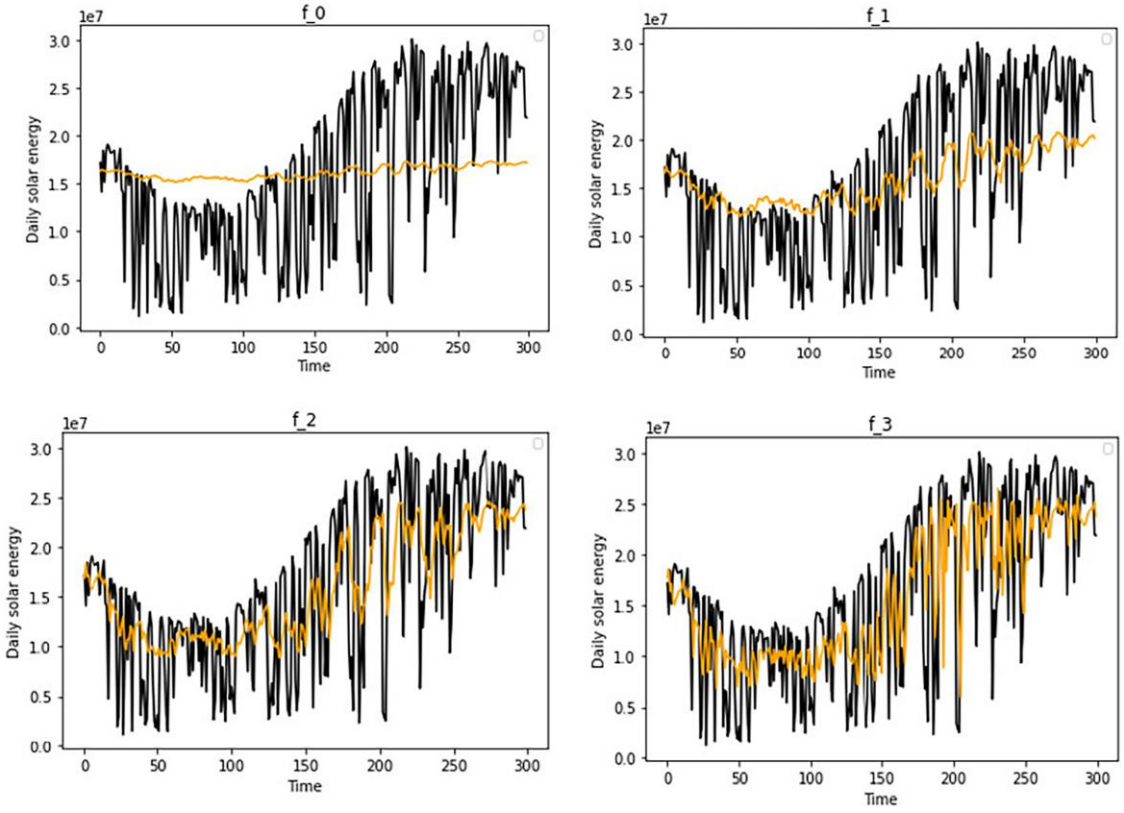
cases from Hungary, taken from the University of California, Irvine repository. Here, we demonstrate the evolution of the ALP predictions for Budapest's time series. Recall that the main advantage of ALP is the approximation of the LOOCV error obtained while we evaluate the training error. Because of this fact, the algorithm iterations stop as the error starts to grow. This effect can be observed in Figure 1, where ALP is applied to Budapest's time series. The time series holds the weekly number of reported chickenpox cases in Budapest, whereas the predicted function approximate the train data in different scales. The ALP model automatically adapts its multiscale behavior to the data, refining the prediction in each iteration by using a more localized kernel, given by a smaller σ. At the beginning, the model approximates the function just by a coarse mean of the target function values (f_0), and in the subsequent iterations when the model starts using sharper kernels and refined residuals, the approximating function captures the different frequencies and amplitudes (f_1 and f_2). In this particular case, the minimum LOOCV value is reached after three iterations, at f_2. It can be seen that the ALP model captures the essential underlying behavior of this time series.

3. Spatio-Temporal Laplacian Pyramids

In the previous section, we introduced the ALP algorithm as a forecasting tool for a single data modality: a single time series. Here, we propose a natural extension of the ALP framework to a spatio-temporal setting. Let $y_1(t),\dots y_\nu(t)$, $1 \le t \le n$, be ν time series that are captured at ν different spatial locations. The task is to forecast the next value of each time series $y_1(n+1)$, $\dots, y_\nu(n+1)$. We formulate the data samples from each spatial location to be short overlapping time series, as described in Section 2.2. Denote these ν data sets by $X^{(1)}, X^{(2)}, \dots, X^{(\nu)}$. The target function for each spatial location is denoted by $f^{(1)}, f^{(2)}, \dots, f^{(\nu)}$, respectively. In what follows, we will focus on the task of forecasting a single station, for example, $\{X^{(1)}, f^{(1)}\}$ based on the spatial time series $\{X^{(j)}, f^{(j)}\}_{j=1}^{\nu}$. In other words, we aim to forecast the values of $f^{(1)}$, belonging to the first spatial location, by the historic time trajectories that are stored in the same location, $X^{(1)}$, and in nearby locations $X^{(2)}, X^{(3)} \dots X^{(\nu)}$.

The model construction begins by forming ν coarse kernels denoted by $K_0^{(1)}, \dots, K_0^{(\nu)}$, based on the data sets $X^{(1)}, \dots, X^{(\nu)}$. The initial corresponding kernel scales are $\sigma^{(1)}, \dots \sigma^{(\nu)}$. Denote the associated row-normalized

Figure 2. (Color online) SALP Train Model



Notes. SALP train model. The time series hold solar energy values in a single location. The multiscale construction goes from coarse to fine (f_0 to f_3). The kernels $\mathcal{P}_\ell$ are formed as a linear combination of three terms: one is the station to be predicted and two other spatial locations.

kernels by $P_0^{(1)}, \ldots P_0^{(\nu)}$. We consider a new series of kernels, which are formed as a convex combination of ν kernels at a given scale. These are defined by

$$\mathcal{P}_\ell = \alpha_1 P_\ell^{(1)} + \cdots + \alpha_\nu P_\ell^{(\nu)}, \quad \text{where} \quad \sum_{i=1}^{\nu} \alpha_i = 1. \tag{7}$$

For the first level, $\ell = 0$, we have

$$\mathcal{P}_0(x_i^*, x_j^*) = \alpha_1 P_0^{(1)}(x_i^1, x_j^1) + \cdots + \alpha_v P_0^{(\nu)}(x_i^\nu, x_j^\nu). \tag{8}$$

We use the notations x_i^* and x_j^* to denote two *generalized* spatial data samples that were recorded at different times, but include information from all of the spatial locations. A first coarse approximation of $f^{(1)}, f_0^{(1)} = s_0^{(1)}$, is then defined by

$$s_0^{(1)}(x_i^1) = \sum_{x_j^1 \in X^{(1)}} \mathcal{P}_0(x_i^*, x_j^*) f^{(1)}(x_j^1), \text{ where } * \in \{1, 2, \ldots, \nu\}. \tag{9}$$

Then, the residual $d_1^{(1)} = f^{(1)} - f_0^{(1)}$ is smoothed by the linear combinations of the kernels at level $\ell = 1$, as defined in Equation (7), denoted by $\mathcal{P}_1$.

The iterations are defined by

$$f_\ell^{(1)}(x_i^1) = f_{\ell-1}^{(1)}(x_i^1) + s_\ell^{(1)}(x_i^1), \tag{10}$$

where

$$s_\ell^{(1)}(x_i^1) = \sum_{x_j^1 \in X^{(1)}} \mathcal{P}_\ell(x_i^*, x_j^*) d_\ell^{(1)}(x_j^1), \text{ where } * \in \{1, 2, \ldots, \nu\}. \tag{11}$$

Here, $d_\ell^{(1)} = f^{(1)} - f_{\ell-1}^{(1)}$.

To optimize the stopping scale, the kernels in the convex combination of Equation (7) are formed with a zero-diagonal, as described in Section 2.1. Extension to a new point $\tilde{x}^1$ (which is a time trajectory) is similar to what is described in Equations (5) and (6), when replacing P_0 and P_ℓ with $\mathcal{P}_0$ and $\mathcal{P}_\ell$ (see Equation (7)).

Figure 2 demonstrates the training stage of the SALP model on the solar energy data set that will be further detailed in Section 4. The time series that is plotted in the four panels is the values in one spatial location. The multiscale approximation use the historic time trajectories of the station to be predicted and two other spatial locations.

3.1. Complexity Analysis

ALP incorporates a modified leave-one-out cross-validation procedure, as it has the attractive property of being an almost unbiased estimator of the true generalization error (Andonova et al. 2002, Cawley and

Talbot 2004). As mentioned in Section 2.1, this procedure is automatic in terms of parameters selection without extra cost. Traditionally, the most obvious drawback of LOOCV is its rather high cost, which in this case would be $N \times O(LN^2)$, where we recall the L is the number of the iterations. The cost of running L steps of ALP is just $O(LN^2)$; thus, we gain the advantage of approximating the exhaustive LOOCV without any additional cost on the overall algorithm (Fernández et al. 2020). For the spatio-temporal setting, the first stage involves selecting a subset of v time series out of n for inclusion in the fusion mechanism. This selection process can be accomplished with a complexity of $O(n)$ and the fusion process in a complexity of $O(1)$. Thus, the cost of running the SALP procedure is just $O(vLN^2)$.

3.2. Error Analysis of the Laplacian Pyramids Model

In this section, we review the analysis of the model (see Fernández et al. (2020) for a detailed version) and extend the results to the spatio-temporal setting. To simplify the analysis, we consider the kernels $P_0, P_1, \ldots, P_\ell$ that were defined in Section 2.1, without the zero-diagonal, assuming that the iterations stop at some fine level ℓ. When working in the continuous kernel setting, the summation becomes an integral. Therefore, we have $k_l(x, x')$ for a Gaussian function.

Furthermore, for all ℓ, writing now $p_\ell(x) = k_\ell(x, 0)$, is an approximation to a delta function satisfying

$$\int p_l(x)dx = 1, \quad \int x p_\ell(x)dx = 0,$$
$$\int \|x\|_2^2 \, p_\ell(x)dx \le 2C, \tag{12}$$

where C is a constant. Assume that f is in L_2, then (Fishelov 1990)

$$\|f_\ell - f\|_{L^2} \le C\sigma^2 \left(\frac{\sigma^2}{\mu^{(\ell+1)}}\right)^\ell \|f\|_{2\ell+2,2}, \tag{13}$$

where $\|f\|_{m,2}$ denotes the Sobolev norm of a function with up to m derivatives in L_2. Therefore, the L_2 norm of the model's error decays at a very fast rate.

Applying the previous result to the kernel $\sum_i \alpha_i k_\ell^{(i)}$, where $\sum_i \alpha_i = 1$, and defining $f - f_\ell = d_{\ell-1}$ (see Equation (11)), we have for the convex combinations of the multiple kernels the same bound for the error as in Equation (13).

4. Experimental Results

This section describes the experimental data sets and the forecasting results.

4.1. Data Sets

Experimental results are demonstrated on three different data sets. The first is the AMS 2013–2014 Solar Energy Prediction Contest.[1] This data set aims to forecast the total daily incoming solar energy. It consists of measurements taken at five-minute intervals of the total daily incoming solar energy collected from 98 Oklahoma Mesonet sites spanning the period from 1994 to 2007. These measurements are then aggregated over the entire day to obtain the total solar energy. The solar irradiance is represented as a time series, where each data point corresponds to the cumulative sum of the solar flux received throughout a day. The nature of the solar energy received at each Mesonet site exhibits periodicity, indicating fluctuations throughout the year. Moreover, the energy received during the summer season tends to be higher compared with the winter season, reflecting the seasonal variability in solar radiation. We plotted the daily incoming solar energy reported from 3 of 98 time series in Figure 3. For this experiment, we selected five batches of size 98×600, which were converted into an overlapping time series of length 7, for each station, as described at the beginning of Section 3. The model was created based on the first 300 time trajectories, and the remaining 300 trajectories were test points.

The second data set is the Hungarian chickenpox[2] data set, which holds the weekly number of reported chickenpox cases in Hungary. The data were collected between 2005 and 2015 from 20 counties and presents various challenges due to the data set's characteristics. First, the number of reported cases is influenced by the population size, with spatial units having higher populations, such as the capital Budapest, typically reporting more cases on average. Second, all of the time series in this data set exhibit strong seasonality, which can be attributed to weather conditions or the cyclic nature of

Figure 3. (Color online) Daily Solar Energy Received at Three Stations

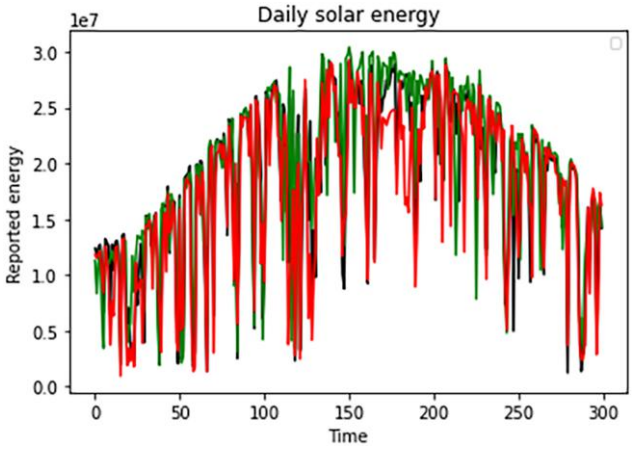
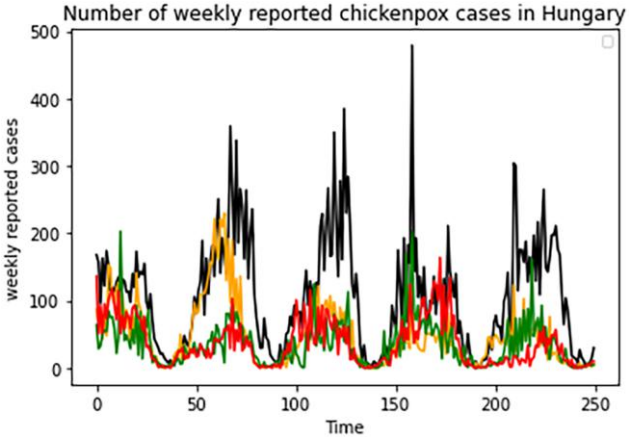


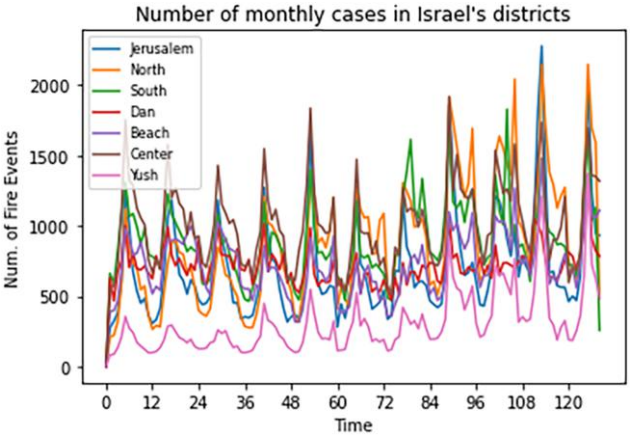
Figure 4. (Color online) The Weekly Number of Chickenpox Cases in Hungarian Counties and the Capital Between 2005 and 2015 in Four Regions



the school year. Finally, during the summer months, a significant number of counties report no new cases, resulting in zero-inflated time series at the county level. We plotted the county-level reported case count from 4 of 20 time series in Figure 4. For this experiment, the model was created based on the first 250 time trajectories, and the remaining 270 trajectories were test points.

The third data set describes the monthly number of reported fire events given by the Israeli Fire and Rescue Services,[3] collected between January 2012 and September 2022 from seven districts. The model was created based on the 2012–2020 time trajectories, and the rest were test points. The representation of the data was transformed to be the percentage of change between the current and last month. This type of normalization made it easier to learn from different districts that hold the same temporal pattern but have different amplitudes. Figure 5 plots the monthly fire events from the seven districts in Israel.

Figure 5. (Color online) Fire and Rescue Data Set from Seven Districts



4.2. Forecasting Results

To construct the SALP for forecasting future values of a given time series $\{X^{(1)}, f^{(1)}\}$, we first identify its most similar spatial locations. One simple approach is to compute the correlation between the train data series in the spatial locations. Based on the correlation results, we identified the spatial locations that would be incorporated in the forecasting model of the given location. Other possible options for encoding similarities between time series may be considered, for example dynamic time wrapping (DTW) (Müller 2007).

In this work, we set a constant number of terms for the linear combination of Equation (7). This number was determined by performing a grid search with $\nu^* \in \{1,2,3,4\}$ for all data sets, where ν^* is the overall optimal number of terms, that is, number of spatial locations that are considered for the predicting for the single station, $f^1(t)$. Setting $\nu^* = 3$ provided the best results. Denote the two most similar time series to $\{X^{(1)}\}$, by $\{X^{(2)}\}$ and $\{X^{(3)}\}$. The weights for the linear combination were then set to $\alpha_1 = 0.9, \alpha_2 = \alpha_3 = 0.05$. Further analysis may be carried out to fine-tune the way ν^* and α_i are chosen; however, even with these fixed weight values, the proposed method yields satisfying results.

The complete train procedure for a given time series that is represented by short trajectories $\{X^{(1)}(t), f^{(1)}(t)\}$ is evoked by computing the kernels $\mathcal{P}_\ell = 0.9P_\ell^{(1)} + 0.05P_\ell^{(2)} + 0.05P_\ell^{(3)}$, and constructing the ALP model as described in Algorithm 1.

In the results tables, we first compare the single-station ALP model to other single-station models. These are kernel ridge regression (KRR) with an Radial basis function (RBF) kernel, support vector regression (SVR), and K Nearest Neighbours (KNN). These were evoked on train and test data from $\{X^{(1)}(t), f^{(1)}(t)\}$. Then, we show how adding spatial information further improves the results of the ALP model, at least for some of the spatial locations. The spatio-temporal results were also compared with LSTM and XGBoost. For these two models, we added the same neighbors that were selected by the correlation similarity. Another model we compared with is denoted by SALP-SS, which stands for SALP single scale. This model contains several stations as inputs (like the other SALP models). However, we only use one single scale ℓ (a single kernel scale $\frac{\sigma}{2^\ell}$) for the kernels. In other words, the model in this case approximates the function f using $\mathcal{P} = \alpha_1 P^{(1)} + \cdots + \alpha_\nu P^{(\nu)}$, where $\sum_{i=1}^{\nu} \alpha_i = 1$, and the approximations are given by $\tilde{f}(x_i^1) = \sum_{x_j^1 \in X^{(1)}} \mathcal{P}(x_i^*, x_j^*) f^{(1)}(x_j^1)$, where $* \in \{1,2,\ldots,\nu\}$. The SALP-SS smoothing kernels $P^{(1)}, \ldots, P^{(\nu)}$ have a single fixed scale, and the residual is not

refined. This yields a spatio-temporal kernel-based regression, but the model does not enjoy the benefits of the multiscale construction.

The hyperparameter settings for each model are crucial for their performance and generalization ability. For XGBoost, a hyperparameter tuning strategy was adopted using GridSearchCV. The learning rate, maximum depth, number of estimators, column subsampling, row subsampling, regularization alpha, and regularization lambda were systematically explored to optimize the model's predictive capability. LSTM, a popular choice for sequential data, was configured with a single layer containing 50 units with a rectified linear unit (ReLU) activation function. The model was trained using the Adam optimizer and mean squared error (MSE) as the loss function. In the case of SVR, the radial basis function (RBF) kernel was selected, with a regularization parameter (C) set to 1.0 and an epsilon value of 0.2. For KRR, an RBF kernel was chosen, and the regularization strength (alpha) was set to one. Finally, KNN's hyperparameters, namely the number of neighbors, algorithm (auto or brute force), and weight function (uniform or distance-based), were optimized through a GridsearchCV.

Traditional metrics like root mean square error (RMSE) and mean absolute error (MAE), although widely used, can be sensitive to the scale of time series data. To address this concern, we additionally calculate robust measures, like the mean absolute scaled error (MASE), ensuring a fair comparison across regions with varying scales in spatio-temporal forecasting scenarios. MASE is defined as the mean of the absolute errors divided by the mean absolute error of a naïve forecast. An MASE value that is smaller than one indicates that the forecasting model performs better than a naive forecast in terms of mean absolute error. A MASE value greater than one implies that the forecasting model is less accurate than a naive forecast.

Table 1 presents the average results for the five batches of the solar energy data set in terms of RMSE and MAE when using a single station. Table 2 presents the RMSE, MAE, and MASE when using spatial information for two additional neighbors. It can be seen the SALP model achieves low errors in all measures. To further illustrate the stability of the predictions, the table also presents the average standard deviation for the five batches, calculated from the errors that were obtained. A smaller standard deviation suggests that the predictions are more consistent and stable across the different batches, indicating the reliability of the approach. This additional information complements the average error values and enhances the overall assessment of the models. When we evoke the XGBoost and LSTM models with all of the available spatial input (98 stations), the results improve. The RMSE, MAE, and MASE is 5, 194, 383, 4, 151, 724, and 0.69, respectively, for the XGBoost and 5, 103, 351, 3, 832, 125, and 0.42 for the LSTM.

The paired t test was used to compare the performance spatio-temporal models, XGBoost, LSTM, and

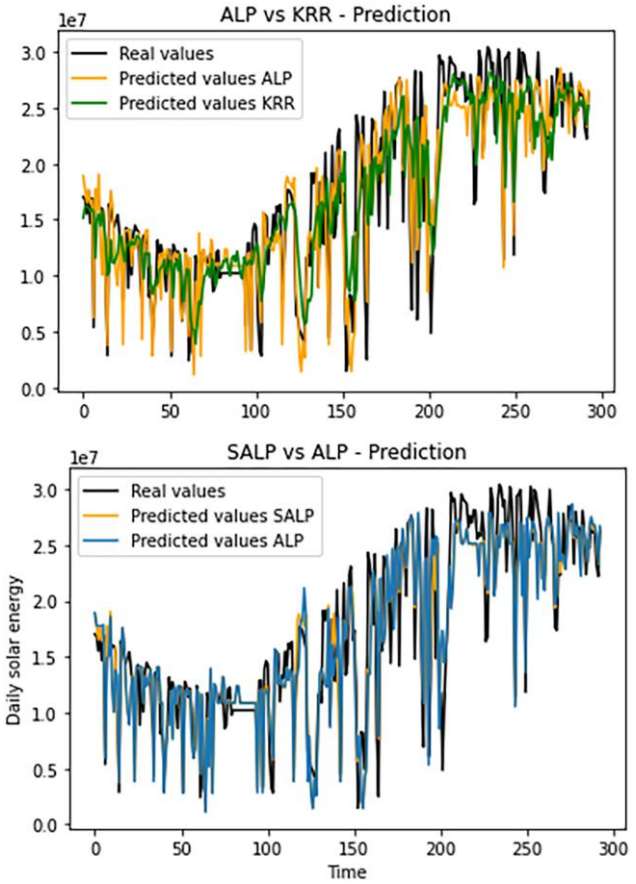
Table 1. Single-Location Prediction Errors for the Solar Data Set

	KNN	KRR	SVR	ALP
RMSE	5,634,483	5,363,311	8,082,705	3,066,830
MAE	4,496,800	4,054,144	6,966,131	2,317,464

Table 2. Spatio-Temporal Prediction Errors for the Solar Data Set

	XGBoost	LSTM	SALP-SS	SALP
RMSE	6,279,577 ±32,107.89	7,747,750 ±653,466.05	3,014,911 ±220,626.84	2,936,759 ±200,238.57
MAE	4,869,450 ±30,194.78	6,112,174 ±517,498	2,315,745.18 ±177,958.55	2,221,837.95 ±135,575.21
MASE	1.09 ±0.0089	1.41 ±0.0861	1.08 ±0.0636	0.55 ±0.0501

Figure 6. (Color online) Prediction Results for the Solar Energy Data Set



Notes. (Top) Single station models, KRR and ALP. (Bottom) Single station ALP vs. spatio-temporal SALP.

Table 3. MASE Values for SALP and ALP by Location: Solar Data Set

Location	MASE (SALP)	MASE (ALP)	Location	MASE (SALP)	MASE (ALP)
1	0.5783	0.5934	26	0.5140	0.5467
2	0.6538	0.6768	27	0.5488	0.5761
3	0.6188	0.6336	28	0.6216	0.5288
4	0.5592	0.5749	29	0.5153	0.5564
5	0.5208	0.5256	30	0.4696	0.6270
6	0.4966	0.5127	31	0.5156	0.5248
7	0.5784	0.6078	32	0.5150	0.4766
8	0.5481	0.5587	33	0.6060	0.5234
9	0.4951	0.4960	34	0.5389	0.5195
10	0.5332	0.5471	35	0.5508	0.6284
11	0.6410	0.6587	36	0.5312	0.5599
12	0.5203	0.5285	37	0.6451	0.5560
13	0.5780	0.5294	38	0.6628	0.5368
14	0.5681	0.5708	39	0.5603	0.6682
15	0.5400	0.5496	40	0.5355	0.6820
16	0.6430	0.5749	41	0.5758	0.5749
17	0.6018	0.6500	42	0.5745	0.5431
18	0.5445	0.6129	43	0.5406	0.5920
19	0.5338	0.5557	44	0.5509	0.5942
20	0.5597	0.5548	45	0.4857	0.5523
21	0.4804	0.4859	46	0.4900	0.5661
22	0.5467	0.5501	47	0.5481	0.6770
23	0.6478	0.6637	48	0.4816	0.4853
24	0.5648	0.5832	49	0.4703	0.4960
25	0.5878	0.6023	50	0.5073	0.5603

SALP in Table 2. When testing XGBoost with SALP and LSTM with SALP, the results demonstrated significant differences in performance with p values of 3.28e-06 and 3.04e-05, respectively.

Figure 6 provides a visual display of the forecasting results of the ALP and SALP models. The top panel displays two single-station models: ALP and KRR with an RBF kernel. The bottom panel displays the ALP versus SALP for the same station. It can be seen that the additional spatial information improves the forecasting, in several test points, as the SALP prediction line is more accurate than the ALP prediction line.

By delving into the performance of individual series across various spatial contexts, we aim to study whether the improvements observed with our method are consistent across diverse regions or whether the model for some locations performs better without the additional spatial information. Table 3 shows the average results for the five batches of the solar energy data set in terms of MASE for the SALP and ALP models for the first 50 of 98 locations. The average MASE for SALP is 0.553 with a standard deviation of 0.0506, whereas the average MASE for ALP is 0.571 with a standard deviation of 0.0522. The paired sample t test conducted on the MASE values for the SALP and ALP models resulted in a p value of 0.0243, which is below the significance level of 0.05. Therefore, we reject the null hypothesis, indicating a statistically significant difference between the MASE values obtained from the two models.

Table 4. Single-Location Prediction Errors for the Chickenpox Data Set

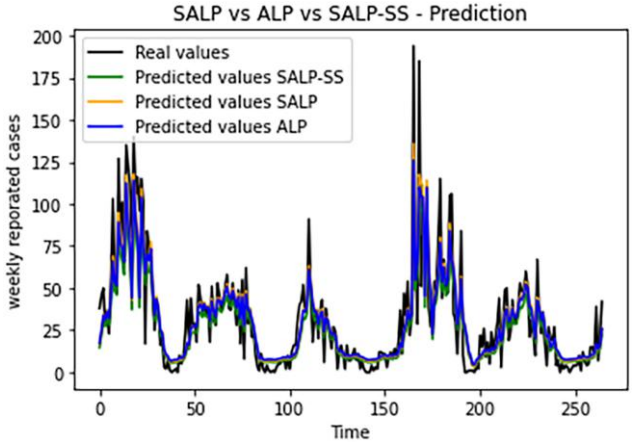
	KNN	KRR	SVR	ALP
RMSE	22.29	23.84	25.35	15.72
MAE	15.12	15.60	17.57	12.18
MASE	1.32	1.34	1.34	0.951

Table 5. Spatio-Temporal Prediction Errors for the Chickenpox Data Set

	XGBoost	LSTM	SALP-SS	SALP
RMSE	24.55	32.09	21.43	15.23
MAE	17.13	22.03	16.30	11.80
MASE	0.97	1.23	1.01	0.932

The results for the chickenpox data set are displayed in Tables 4 and 5. These are the average RMSE, MAE, and MASE values across the 20 counties. It can be seen that the proposed SALP model results with the most accurate predictions. For this example, we plot in Figure 7 the predicted values for Budapest as computed by ALP, SALP, and SALP-SS. It can be seen that the SALP-SS model generates a smoother prediction due to the use of a single-scale kernel and lack of refinements. In addition, SALP generates a better prediction compared with ALP the oscillatory regions. In terms of per-location improvement, Table 6 presents the MASE values for each location. Although SALP improves the MASE value in 12 of the 20 stations, the

Figure 7. (Color online) Prediction Results for Baranya (Hungry Chickenpox Data Set)



Notes. Plot of the real values, predicted values of the single-station ALP model, the single-scale SALP-SS predictions and the SALP model. The single-scale SALP-SS predictions seem to be smoother, capturing less details in the oscillatory areas. It can also be seen how SALP improves ALP in certain points in time.

Table 6. MASE Values for SALP and ALP by Location: Chickenpox Data Set

Location	MASE (SALP)	MASE (ALP)	Correlation score
Budapest	0.75	0.82	0.76
Baranya	0.89	0.91	0.69
Bacs	0.72	0.68	0.63
Bekes	1.04	1.01	0.62
Borsod	0.98	1.05	0.64
Csongrad	1.02	1.01	0.59
Fejer	0.85	0.87	0.72
Gyor	0.93	0.95	0.72
Hajdu	0.78	0.80	0.67
Heves	0.91	0.88	0.58
Jasz	1.05	1.07	0.64
Komarom	0.79	0.75	0.70
Nograd	1.12	1.15	0.61
Pest	0.84	0.80	0.76
Somogy	0.97	0.99	0.64
Szabolcs	1.08	1.10	0.58
Tolna	0.88	0.86	0.59
Vas	0.94	0.92	0.53
Veszprem	0.96	0.98	0.66
Zala	0.89	0.91	0.56

results of the t test are not significant ($p = 0.871$). In addition to the MASE values we added to Table 6, the correlation values between the time series of the given spatial location and its most correlative neighbor. The median value for the 12 locations that showed improvement with SALP is 0.65, whereas the median value for the remaining eight stations is 0.605. It seems reasonable that spatio-temporal models will be less effective for locations that have less in common with other locations.

The influence of the weight in the convex combination of the model were also examined for two locations from the chickenpox data set. Recall, that we chose predefined weight coefficients for the experimental results, and did not use a grid search to determine the optimal coefficients for the spatial combination. However, we included additional results showcasing the impact of varying these parameters in Table 7. Specifically, we examined the effects of different coefficient values for two stations from the chickenpox data set: Bekes and Budapest. It can be seen that for the Bekes location, the addition of spatial information degrades the results, whereas for the Budapest setting, α_1 as 0.9, 0.8, 0.7, or 0.5 yields improved results compared with the ALP model. In the Budapest example, the results for $\alpha_1 = 0.9, 0.8$, or 0.7 are quite similar. It should be mentioned that the ease of transforming between the single station and spatio-temporal setting is our proposed model is an advantage, as both cases may be tested when running a grid search on the coefficients.

Table 7. SALP MASE Values for Different Coefficient Settings

Convex combination coefficients	MASE Bekes	MASE Budapest
$\alpha_1 = 1, \alpha_2 = 0, \alpha_3 = 0$	1.013	0.824
$\alpha_1 = 0.9, \alpha_2 = 0.05, \alpha_3 = 0.05$	1.0467	0.753
$\alpha_1 = 0.8, \alpha_2 = 0.1, \alpha_3 = 0.1$	1.0524	0.761
$\alpha_1 = 0.7, \alpha_2 = 0.15, \alpha_3 = 0.15$	1.0689	0.768
$\alpha_1 = 0.5, \alpha_2 = 0.25, \alpha_3 = 0.25$	1.0727	0.811

Table 8. Single-Location Prediction Errors for the Fire and Rescue Data Set

	KNN	KRR	SVR	ALP
RMSE	0.241	0.275	0.224	0.130
MAE	0.181	0.209	0.168	0.102
MASE	1.197	1.197	1.197	0.883

Last, Tables 8 and 9 display the single-location and spatio-temporal results for the fire and rescue data set. Recall that the predictions were for the percentage change in fire events from the previous month. It can be seen that ALP achieves low errors for the single station prediction and that these are slightly improved when spatial information is added to the model using the RMSE, MAE, and MASE metrics. Although ALP performs better than KNN, KRR, and SVR in the single-station models in Table 8, we see that for this data set, LSTM provided the best result for the spatio-temporal setting in terms of MASE (Table 9). However, for this problem, RMSE and MAE are important measures, as they reflect the total number of fires that may be predicted in advance, and in these measures, SALP provides more accurate results.

The empirical improvement observed in the performance of the proposed model (SALP) compared with the alternative model (ALP) may be attributed to the complex interplay of various factors. Although the SALP model, with its spatially aware layer of kernels, appears to benefit from learning patterns from neighboring locations, the statistical analysis suggests that this improvement might not be statistically significant for the Chickenpox data set. One possible explanation for this discrepancy could be that SALP's reliance on spatial neighbors introduces an additional layer of complexity, forcing the model to adapt to diverse spatial patterns. It is essential to recognize that empirical improvements do not always translate into statistically

Table 9. Spatio-Temporal Prediction Errors for the Fire and Rescue Data set

	XGBoost	LSTM	SALP-SS	SALP
RMSE	0.229	0.269	0.207	0.131
MAE	0.171	0.193	0.158	0.103
MASE	0.898	0.722	0.812	0.880

significant results, particularly when dealing with real-world data sets where inherent variability and noise may play a significant role.

Overall, it is apparent that the multiscale component is important, as small errors are achieved in all of the ALP models. Furthermore, the addition of spatial information further reduces the errors.

5. Discussion

In this paper, we proposed an extension of an iterative, multiscale regression model to a spatio-temporal setting. The proposed model is appealing because it is easy to implement and yields accurate results due to its multiscale construction that capture the lower and higher frequencies of the data. Integrating kernels that are formed as convex combinations of data from similar locations does not change the overall train and test algorithms and is shown to improve the prediction results. We emphasize the importance of both the multiscale and multilocation by comparing the results with the SALP-SS model. This model incorporates spatial information but has only one *layer* of kernels (one convex combination) with a single scale. Our model resembles some characteristics of network models, as information is passed between scales. At the same time, it is simple, has a small number of parameters, and one can understand the relationships between the input and output, as well as analyze the model's convergence rate. The experimental results compared the performance of ALP and SAPL with other well-known regression and machine-learning techniques and highlight the benefits of the proposed kernel-based regression.

6. Limitations and Future Work

The two main ingredients of our proposed method are the convex combination of kernels that hold data from several locations and the multiscale learning mechanism. In this work, we build the convex combination with a limited number of spatial locations. One reason for limiting the spatial dimension in the experimental results section, is to bypass the need to carefully set the coefficients in Equation (7).

Although there is no prevention in using a large number of input kernels, we acknowledge that future work may be carried out to simplify our methods for cases in which the spatial dimension is large. One future direction is to construct an ensemble models, each using a different subset of spatial neighbors, similarly to the construction of random forests, replacing the *features* with *spatial modalities*. Such a construction can take advantage of all the spatial information, bypass the need to carefully tune the coefficients in Equation (7), and provide both the expected predicted value together with a standard deviation.

Acknowledgments

The authors thank the anonymous referees and editors for insightful comments and valuable suggestions that significantly contributed to the improvement of this paper.

Endnotes

[1] See https://www.kaggle.com/.

[2] See https://archive.ics.uci.edu/.

[3] See https://info.data.gov.il/datagov/home/.

References

Andonova S, Elisseeff A, Evgeniou T, Pontil M (2002) A simple algorithm for learning stable machines. *Proc. 15th Eur. Conf. Artificial Intelligence* (IOS Press, Amsterdam), 513–517.

André M, Dabo-Niang S, Soubdhan T, Ould-Baba H (2016) Predictive spatio-temporal model for spatially sparse global solar radiation data. *Energy* 111:599–608.

Beddar-Wiesing S, D'Inverno GA, Graziani C, Lachi V, Moallemy-Oureh A, Scarselli F, Thomas JM (2024) Weisfeiler–Lehman goes dynamic: An analysis of the expressive power of graph neural networks for attributed and dynamic graphs. *Neural Networks* 173:106213.

Cai L, Janowicz K, Mai G, Yan B, Zhu R (2020) Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. *Trans. GIS* 24(3):736–755.

Cawley GC, Talbot NL (2004) Fast exact leave-one-out cross-validation of sparse least-squares support vector machines. *Neural Networks* 17(10):1467–1475.

Comeau D, Giannakis D, Zhao Z, Majda AJ (2019) Predicting regional and pan-arctic sea ice anomalies with kernel analog forecasting. *Climate Dynamics* 52:5507–5525.

Comeau D, Zhao Z, Giannakis D, Majda AJ (2017) Data-driven prediction strategies for low-frequency patterns of north pacific climate variability. *Climate Dynamics* 48:1855–1872.

Cornillon P-A, Hengartner N, Jégou N, Matzner-Løber E (2013) Iterative bias reduction: A comparative study. *Statist. Comput.* 23: 777–791.

Cornillon P-A, Hengartner N, Lefieux V, Matzner-Løber E (2015) Fully nonparametric short term forecasting electricity consumption. *Modeling and Stochastic Learning for Forecasting in High Dimensions* (Springer, Berlin), 79–93.

Dambreville R, Blanc P, Chanussot J, Boldo D (2014) Very short term forecasting of the global horizontal irradiance using a spatio-temporal autoregressive model. *Renewable Energy* 72:291–300.

Fernández Á, Rabin N, Fishelov D, Dorronsoro JR (2020) Auto-adaptive multi-scale Laplacian pyramids for modeling non-uniform data. *Engrg. Appl. Artificial Intelligence* 93:103682.

Fishelov D (1990) A new vortex scheme for viscous flows. *J. Comput. Phys.* 86(1):211–224.

Gao Z, Shi X, Wang H, Zhu Y, Wang YB, Li M, Yeung D-Y (2022) Earthformer: Exploring space-time transformers for earth system forecasting. *Adv. Neural Inform. Processing Systems* 35: 25390–25403.

Guan M, Iyer AP, Kim T (2022) Dynagraph: Dynamic graph neural networks at scale. *Proc. 5th ACM SIGMOD Joint Internat. Workshop Graph Data Management Experiences Systems and Network Data Analytics*, 1–10.

Hamdi A, Shaban K, Erradi A, Mohamed A, Rumi SK, Salim FD (2022) Spatiotemporal data mining: A survey on challenges and open problems. *Artificial Intelligence Rev.* 55:1441–1488.

Hen B, Fernández A, Rabin N (2022) Improving Laplacian pyramids regression with localization in frequency and time. *Proc. Eur. Sympos. Artificial Neural Networks*, Computational Intelligence and Machine Learning (ESANN 2022) (i6dot.com), 363–368.

Hengl T, Nussbaum M, Wright MN, Heuvelink GB, Gräler B (2018) Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables. *PeerJ.* 6:e5518.

Kang L, Joseph VR (2016) Kernel approximation: From regression to interpolation. *SIAM/ASA J. Uncertainty Quantification* 4(1):112–129.

Khaldi R, El Afia A, Chiheb R, Tabik S (2023) What is the best RNN-cell structure to forecast each time series behavior? *Expert Systems Appl.* 215:119140.

Li M, Cohen I, Mousazadeh S (2014) Multisensory speech enhancement in noisy environments using bone-conducted and air-conducted microphones. *Proc. IEEE China Summit Internat. Conf. Signal Inform. Processing* (IEEE, New York), 1–5.

Liang Y, Xia Y, Ke S, Wang Y, Wen Q, Zhang J, Zheng Y, et al. (2023) Airformer: Predicting nationwide air quality in china with transformers. *Proc. Conf. AAAI Artificial Intelligence* 37:14329–14337.

Lin T, Wang Y, Liu X, Qiu X (2022) A survey of transformers. *AI Open* 3:111–132.

Liu R, Ma Z, Liu Y, Shao Y, Zhao W, Bi J (2020) Spatiotemporal distributions of surface ozone levels in China from 2005 to 2017: A machine learning approach. *Environment. Internat.* 142:105823.

Mohamed A, Qian K, Elhoseiny M, Claudel C (2020) Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. *Proc. IEEE/CVF Conf. Computer Vision Pattern Recognition* (IEEE, Piscataway, NJ), 14424–14432.

Müller M (2007) Dynamic time warping. *Information Retrieval for Music and Motion* (Springer, Berlin, Heidelberg), 69–84.

Nadaraya EA (1964) On estimating regression. *Theory Probability Appl.* 9(1):141–142.

Rabin N (2020) Multi-directional Laplacian pyramids for completion of missing data entries. *Proc. Eur. Sympos. Artificial Neural Networks*, Computational Intelligence and Machine Learning (ESANN 2020) (i6dot.com), 709–714.

Rabin N, Coifman RR (2012) Heterogeneous data sets representation and learning using diffusion maps and Laplacian pyramids. *Proc. SIAM Internat. Conf. Data Mining* (SIAM, Philadelphia), 189–199.

Rabin N, Fishelov D (2019) Two directional Laplacian pyramids with application to data imputation. *Adv. Comput. Math.* 45(4): 2123–2146.

Rabin N, Fernández Á, Fishelov D (2023) Multiscale extensions for enhancing coarse grid computations. *J. Comput. Appl. Math.* 427: 115116.

Rao KV, Govardhan A, Rao KC (2012) Spatiotemporal data mining: Issues, tasks and applications. *Internat. J. Computer Sci. Engrg. Survey* 3(1):39.

Rozemberczki B, Scherer P, He Y, Panagopoulos G, Riedel A, Astefanoaei M, Kiss O, et al. (2021) Pytorch geometric temporal: Spatiotemporal signal processing with neural machine learning models. *Proc. 30th ACM Internat. Conf. Inform. Knowledge Management* (Association for Computing Machinery, New York), 4564–4573.

Shekhar S, Jiang Z, Ali RY, Eftelioglu E, Tang X, Gunturi VM, Zhou X (2015) Spatiotemporal data mining: A computational perspective. *ISPRS Internat. J. Geoinform.* 4(4):2306–2338.

Shih S-Y, Sun F-K, Lee H-y (2019) Temporal pattern attention for multivariate time series forecasting. *Machine Learn.* 108:1421–1441.

Souza A, Mesquita D, Kaski S, Garg V (2022) Provably expressive temporal graph networks. *Adv. Neural Inform. Processing Systems* 35:32257–32269.

Wang S, Cao J, Philip SY (2020) Deep learning for spatio-temporal data mining: A survey. *IEEE Trans. Knowledge Data Engrg.* 34(8): 3681–3700.

Wang J, Huyen D, Woznica A, Kalousis A (2011) Metric learning with multiple kernels. Shawe-Taylor J, Zemel R, Bartlett P, Pereira F, Weinberger KQ, eds. *Adv. Neural Inform. Processing Systems*, vol. 24 (Curran Associates, Inc., Red Hook, NY).

Wen Q, Zhou T, Zhang C, Chen W, Ma Z, Yan J, Sun L (2023) Transformers in time series: A survey. *IJCAI '23: Proc. Thirty-Second Internat. Joint Conf. Artificial Intelligence (Macao, S.A.R)*, 6778–6786.

Wood SN (2017) *Generalized Additive Models: An Introduction with R* (CRC Press, Boca Raton, FL).

Xia Z, Stewart K (2023) A counterfactual analysis of opioid-involved deaths during the COVID-19 pandemic using a spatiotemporal random forest modeling approach. *Health Place* 80:102986.

Xu B, Lu X, Bai Y, Xu D, Cui X (2023) A multi-kernel-based spatiotemporal modeling approach for energy transfer of complex thermal processes and its applications. *Internat. J. Heat Mass Transfer* 216:124597.

Xu M, Dai W, Liu C, Gao X, Lin W, Qi G-J, Xiong H (2020) Spatial-temporal transformer networks for traffic flow forecasting. Preprint, submitted January 9, https://arxiv.org/abs/2001.02908.

Yang D, Ye Z, Lim LHI, Dong Z (2015) Very short term irradiance forecasting using the lasso. *Solar Energy* 114:314–326.

Yang D, Dong Z, Reindl T, Jirutitijaroen P, Walsh WM (2014) Solar irradiance forecasting using spatio-temporal empirical kriging and vector autoregressive models with parameter shrinkage. *Solar Energy* 103:550–562.

Zeng A, Chen M, Zhang L, Xu Q (2023) Are transformers effective for time series forecasting? *Proc. Conf. AAAI Artificial Intelligence*, vol. 37 (AAAI Press, Palo Alto, CA), 11121–11128.

Zhang X, Zhong C, Zhang J, Wang T, Ng WW (2023) Robust recurrent neural networks for time series forecasting. *Neurocomputing* 526: 143–157.