arXiv:2012.01349v3 [stat.AP] 4 Apr 2022

The temporal overfitting problem with applications in wind power curve modeling

Abhinav Prakash, Rui Tuo, and Yu Ding
Department of Industrial and Systems Engineering
Texas A&M University

Abstract

This paper is concerned with a nonparametric regression problem in which the input variables and the errors are autocorrelated in time. The motivation for the research stems from modeling wind power curves. Using existing model selection methods, like cross validation, results in model overfitting in presence of temporal autocorrelation. This phenomenon is referred to as temporal overfitting, which causes loss of performance while predicting responses for a time domain different from the training time domain. We propose a Gaussian process (GP)-based method to tackle the temporal overfitting problem. Our model is partitioned into two parts—a time-invariant component and a time-varying component, each of which is modeled through a GP. We modify the inference method to a thinning-based strategy, an idea borrowed from Markov chain Monte Carlo sampling, to overcome temporal overfitting and estimate the time-invariant component. We extensively compare our proposed method with both existing power curve models and available ideas for handling temporal overfitting on real wind turbine datasets. Our approach yields significant improvement when predicting response for a time period different from the training time period. Supplementary material and computer code for this article is available online.

Keywords: Autocorrelation, Gaussian process, Nonparametric regression, Time series.

1 Introduction

Wind energy is the forerunner among the renewable energy sources, and by the end of 2020, wind energy accounted for roughly 8.4% of the total electricity used in the United States (EIA, 2021). In various decision making tasks in wind energy, wind power curve plays an important role. A power curve is a function that maps the relationship of wind speed and other environmental variables to the wind power output. A quality estimation of power curve has crucial practical implication for decision-making in many aspects, including wind power prediction and turbine performance evaluation (Ding, 2019).

International Electrotechnical Commission (IEC, 2005) recommends a data-driven approach, known as the *binning method*, to construct the power curve. The binning method

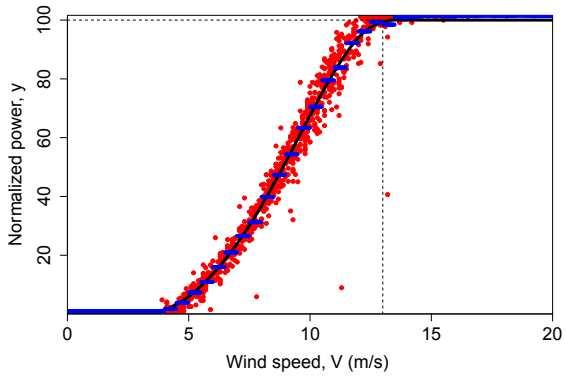

Figure 1: A nominal wind power curve. Dots (red) denote the data; piecewise constant curve (blue) represents binning; smooth curve (black) is from smoothing on binning.

considers the single input of wind speed, partitions wind speed into small bins, say, 0.5 m/s, and uses the sample average of wind power data, whose corresponding wind speed falls into a bin, as the estimate of power response for that bin. The power curve in Figure 1 is generated using the binning method (with some simple post smoothing).

Research reported in Bessa et al. (2012) and Lee et al. (2015) identifies that wind power production is not limited to the effect of wind speed, but also depends on other factors such as wind direction, air density, etc. Bessa et al. (2012) and Lee et al. (2015) both used the kernel regression and kernel density estimation approaches. But Bessa et al. (2012) handle up to three input variables, while Lee et al. (2015) use a new additive-multiplicative model structure, referred to as AMK, that can in principle take in as many inputs as possible. The actual number of inputs used in Lee et al. (2015) is up to seven. In Chapter 5 of Ding (2019) and its companion `DSWE R package` (Kumar et al., 2021), more nonparametric regression methods are provided, including those based on the smoothing splines (SSANOVA) (Gu, 2013, 2014), Bayesian trees (BART) (Chipman et al., 2010), k-nearest neighbors (kNN) (Hastie et al., 2009), and support vector machines (SVM) (Vapnik, 2000). Thus, the nature of power curve modeling falls squarely under the umbrella of nonparametric regression.

During our research, we encountered a problem in wind power curve modeling, which we explain through the following real-life example. We were given (by a wind company) a set of wind/weather inputs—wind speed, its standard deviation, wind direction, ambient temperature—denoted by $\boldsymbol{x}$, and the turbine power output data, denoted by y; all collected

in 2015. We were asked to train a model and then predict y using the $\boldsymbol{x}$ values collected in 2016 (the first six months). The actual y values of 2016 were withheld by the company for evaluation. The binning method is the company's practice and used as the benchmark. The company (in fact any company) would only consider adopting a new method, if the new method outperforms their current benchmark in testing. We chose the AMK method mentioned above, as it was then demonstrated as the most competitive method. We use a five-fold randomized cross validation to train the AMK model, including both variable selection and parameter estimation. Using a forward stepwise variable selection, all four aforementioned inputs are selected as important. The resulting AMK model has a root mean squared error (RMSE) 30% smaller than the binning RMSE, based on the same five-fold randomized cross-validation using the 2015 data. However, when both models (binning and AMK) are applied to the 2016 data, the AMK model produces an RMSE which is 5% higher than binning. This is not a unique problem to AMK. Should we use another nonparametric regression method included in the `DSWE R package`, a similar phenomenon will be observed, that is, all of them outperform binning, with a comfortable margin, when tested using 2015 data through a randomized cross validation, but all of them either fails to outperform binning or see their margin of improvement significantly diminished when tested using the 2016 data.

Exploring the literature (Roberts et al., 2017; Meyer et al., 2018; Sheridan, 2013), we found that we are not alone—the problem encountered is a case of a common issue in nonparametric regression, known as *temporal overfitting* (Meyer et al., 2018). This overfitting is caused due to the temporal autocorrelation in the data. In the wind application, both $\boldsymbol{x}$ and y are autocorrelated. Temporal overfitting differs from the usual notion of overfitting; the latter occurs when a model fits well to the training data, but performs worse on a random holdout data. The usual overfitting can generally be avoided using cross-validation, as cross-validation error tends to estimate the generalization error when the input and error processes are independent and identically distributed (i.i.d.); see Hastie et al. (2009, Chapter 7). For temporal overfitting, however, the model generalizes well for the test datasets originating from the same time domain as the training dataset, that is, there is no temporal extrapolation when moving from training to test inputs and the data points in training and test sets are temporally close. But, the model's performance seriously deteriorates when

the test dataset is from a different time domain (temporal extrapolation), precisely as we observed in the wind power curve modeling.

We can classify test sets and their corresponding test errors into two categories. For convenience, let us call the two test errors as *in-temporal* and *out-of-temporal* test errors, respectively, based on whether the test datasets arise from the same or a different time domain than that of the training dataset. In the earlier example, when a model is trained and tested on the 2015 data through a randomized cross validation, the corresponding test error is the in-temporal error, whereas when a model is trained using the 2015 data and tested using the 2016 data, the corresponding test error is the out-of-temporal error. To be clear, temporal overfitting is *not* concerned with the training error like the usual overfitting problem. Rather, it is concerned with the aforementioned two types of *test errors*.

In this work, we establish a Gaussian process (GP)-based method to deal with temporal overfitting. We split our functional model into two components: a time-invariant component and a time-varying component, each of which is modeled as a GP. The GP model in and of itself does not remove temporal overfitting; we make use of a subsampling scheme from the Bayesian statistics literature, known as thinning, for model inference that reduces the adverse impact of temporal overfitting. Thus, the main contributions of this work is to highlight the temporal overfitting problem in power curve modeling and propose a GP-based inference strategy to overcome the problem. Our numerical studies show that the proposed method performs significantly better for out-of-temporal predictions, as compared to the existing power curve models and other statistical approaches (those described in Section 2) that can be used to deal with the temporal overfitting problem.

The rest of the paper is organized as follows. In Section 2, we describe the problem statement and the major schools of thought in dealing with temporal overfitting. Section 3 presents our proposed method. Section 4 provides the empirical evidence on performance of our method using wind turbine datasets. We conclude our research in Section 5.

2 Problem statement and relevant literature

We consider a nonparametric regression problem, where $Y_i \in \mathbb{R}$, $X_i \in \mathbb{R}^d$, and $u_i \in \mathbb{R}$, and

$$Y_i = f(X_i) + u_i, \tag{1}$$

which has the following features:

1. The form of $f(\cdot)$ is unknown and differs in applications, making nonparametric approaches more appropriate. Furthermore, there are sufficient data pairs, $\{y_i, x_i\}_{i=1}^N$, enabling the nonparametric treatment.

2. The input is multivariate, but the number of input variables that are causal for the response is unknown.

3. The input X and the error u can be considered physically independent with each other.

4. The data are observational, not experimentally designed or controlled. They are sequentially collected from an ongoing physical process over time. The index, i, corresponds to time. Both the input variables in X and the error process, u, are temporally autocorrelated over the time index i.

Our focus here is on #4, because without the temporal autocorrelation, the problem falls under standard nonparametric regression.

A natural question is how the temporal autocorrelation in the data causes the overfitting, even though time is not explicitly considered in the model (only implicitly tagged with X_i) and the function of interest is the relationship between the input variables X and the response Y. To address this question, we consider a closely related problem in the statistics literature—when the error process is correlated with some input variables, namely $u = u(X)$. If one applies the standard statistical learning techniques without accounting for the correlation between the error and the input variable, one may get an overfitted functional estimate as shown in the left panel of Figure 2. The curve is fitted using a kernel regression with a direct plug-in (DPI) bandwidth estimate (Ruppert et al., 1995). One can find similar plots in Opsomer et al. (2001), De Brabanter et al. (2011).

The problem of temporal overfitting can be thought of a case when the errors are correlated with the input variables. We know that when two random variables change slowly over time, it could result in a spurious correlation among these two quantities even if they are independent, as in the case under our consideration; see #3 and #4 in the problem setup. This is to say, when the input variables and errors are autocorrelated in time, it

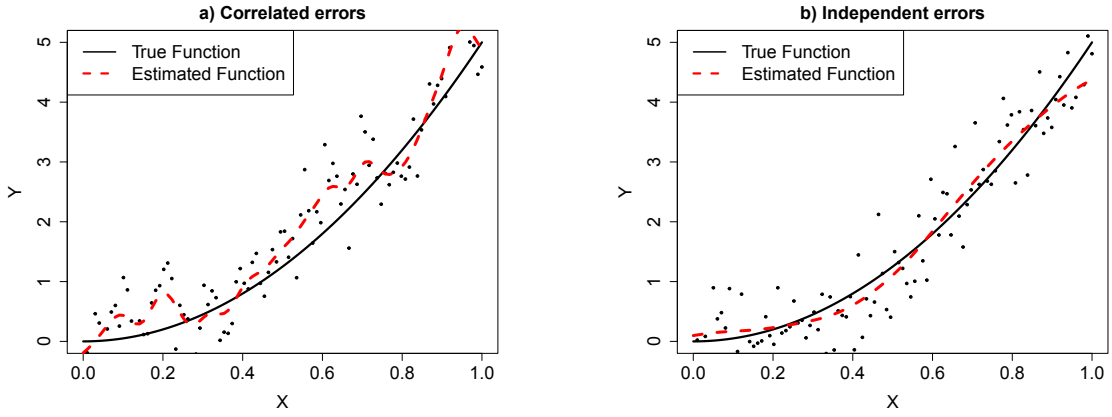

Figure 2: Effect of correlation between input variable and error on functional estimate: a) correlated errors; b) independent errors. We use $f(x) = 5x^2$. The correlated error sequence is generated using a zero mean GP with input x and an exponential kernel with a lengthscale of 0.05.

would create a correlation among them, and may consequently create the overfitting effect as shown in Figure 2. The presence of the nuisance input variables, which are not causal to the response, further aggravates the problem. An input variable could turn out to be 'seemingly important,' owing to the temporal autocorrelation in both the response and the nuisance variable, even when it is not a causal variable. The more input variables considered for modeling, the more likely a non-causal variable to be selected because of the presence of temporal autocorrelation, leading to poor generalization when the test data points are from a different time domain (Roberts et al., 2017).

There are various methods studied in the literature to handle the problem of autocorrelation in error and/or input; Opsomer et al. (2001) provides a survey of the methods up to two decades ago. One class of methods, developed specifically for kernel regression, are based on directly modifying the bandwidth estimation technique; see, for example, Altman (1990) and De Brabanter et al. (2011, 2018). This is generally done by modifying the criterion for computing the optimal bandwidth such as asymptotic squared error (ASE). One of the criticisms for these kernel-based methods, as discussed by Opsomer et al. (2001), is their inability to handle multivariate inputs. This limitation persists even in the recent literature (De Brabanter et al., 2018), which still considers a univariate input while developing their bandwidth estimation. De Brabanter et al. (2018) touch upon extensions to multivariate inputs only in their discussions section. Thus, this class of method is not directly applicable

to our nonparametric regression problem where multiple input variables are dealt with.

There are two other schools of thought applicable to our problem setup. One is known as *pre-whitening* (Xiao et al., 2003; Geller and Neumann, 2018) and the other is through some modification of the cross-validation error (Chu and Marron, 1991; Burman et al., 1994; Racine, 2000; Opsomer et al., 2001; Rabinowicz and Rosset, 2020).

The idea of pre-whitening is to preprocess the response itself such that the resulting data has a white noise (Xiao et al., 2003; Geller and Neumann, 2018). More specifically, pre-whitening is to model u_i using an invertible linear process,

$$u_i = \Sigma_{j=0}^{\infty} c_j \varepsilon_{i-j},$$

where ε's are white noises. Then, one can map u_i's to ε_i's using the inverse process. Practically, one would require to fit a regression model $\hat{m}(X)$ and then compute the residuals $\hat{u}_i = Y_i - \hat{m}(X_i)$. An autoregressive model of a suitable order can be estimated for $\hat{u}_i$. Subtracting the estimated autoregressive part of $\hat{u}_i$ from Y_i, so as to remove the autocorrelation in Y due to u, produces a modified response $\bar{Y}_i$. Theoretically, this modified response $\bar{Y}_i$ would be free from temporal autocorrelation and can be used as the response for the final model. One major challenge in this method is the presence of some nuisance variables in the data. According to Roberts et al. (2017), these nuisance (non-causal) variables can mask the autocorrelation in the residuals, that is, the temporal autocorrelation of the residuals gets modeled through the autocorrelation in the nuisance variables. If that happens, one would underestimate the autocorrelation in the residuals, resulting in minimal or even no changes in the modified response $\bar{Y}_i$. As a result, one gets little or no improvement on the estimate of the regressor.

The idea of modifying cross-validation is arguably a more general framework to deal with correlated errors or temporal overfitting. This branch of methods is also known as h-blocking or $h\nu$-blocking and, more recently, as leave-time-out or time-split cross-validation. The idea is to do cross-validation on temporal blocks of data rather than random samples. Chu and Marron (1991); Burman et al. (1994); Racine (2000) explore this idea under different settings. The time-blocked cross-validation idea is advocated by Roberts et al. (2017) and adopted in Meyer et al. (2018).

Related to the idea of modifying cross-validation but unlike the previous works, Rabinowicz and Rosset (2020) proposed a modification for the cross-validation error by adding

a correction factor to account for the correlation. They motivate the problem using a linear mixed effect model where some of the effects stay the same, whereas other effects change from training to test datasets. The model formulation in Rabinowicz and Rosset (2020) bears certain similarity to ours, as we split the regression function into a time-invariant and time-dependent component (to be presented in the next section). A key difference is that the inputs to the time-invariant component in our model are autocorrelated, while the inputs to Rabinowicz and Rosset (2020)'s fixed effect term are i.i.d. The rest of the treatment in our method also differs substantially from that in Rabinowicz and Rosset (2020). For instance, our inference method does not require cross-validation.

In the case study, we compare our proposed method with the pre-whitening method, the time-split cross-validation, and Rabinowicz and Rosset (2020)'s method. We find that our method consistently outperforms these available approaches when testing on data that are outside the time domain covered by the training data.

3 Proposed method

In this section, we describe our proposed method to mitigate the problem of temporal overfitting while fitting a nonparametric regression model.

3.1 The model

Given a dataset $\mathcal{D} = \{y_i, \boldsymbol{x}_i, t_i\}_{i=1}^N$, we consider the following model:

$$y_i = f(\boldsymbol{x}_i) + g(t_i) + \epsilon_i, \tag{2}$$

where for our target wind application, y is the power output of a wind turbine, $\boldsymbol{x}$ is the d-dimensional vector of environmental input variables, t denotes time, and ϵ is i.i.d. Gaussian noise with zero mean and variance $\sigma_\epsilon^2 < \infty$. We deem that $f(\cdot)$ is a time-invariant function of the input variables $\boldsymbol{x}$ and $g(\cdot)$ is a temporally autocorrelated stationary stochastic process that contains the autocorrelated part of the residual. We stress that while $f(\cdot)$ is time invariant, $\boldsymbol{x}_i$ is time varying and autocorrelated.

Recall that the motivation for this paper is to avoid temporal overfitting and improve the prediction accuracy under temporal extrapolation, that is, for out-of-temporal test sets as defined in Section 1. In Equation (2), $f(\cdot)$ can explain the variance in the data that is

carried over to a different time domain, as we assume that this function does not directly depend on time but only through the input variables. The variance which does not carry over to a different time horizon is modeled through the time-dependent term, $g(\cdot)$. The rest is just i.i.d. noise. Thus, for out-of-temporal predictions, accurately identifying $f(\cdot)$ plays a key role in improving the accuracy.

Before providing further modeling details, we would like to elaborate on this model setup. Among the three main approaches reviewed in Section 2, the pre-whitening approach and the time-split cross-validation do not invoke a model like in Equation (2); rather they work directly with the model in Equation (1). This is especially obvious in the pre-whitening approach, which is to whiten the autocorrelated u and use that as the main apparatus to deal with temporal overfitting. But Rabinowicz and Rosset (2020)'s method does invoke a model of certain similarity to Equation (2). Specifically, Rabinowicz and Rosset (2020) consider a generalized linear mixed model (GLMM) of the form,

$$\boldsymbol{y} = \Phi\boldsymbol{\beta} + Z\boldsymbol{s} + \boldsymbol{\epsilon}, \tag{3}$$

where Φ contains the fixed effect covariates, meaning that the input $\boldsymbol{x}$ would be included through Φ, and Z contains the random effect covariates. The realization of the random effect $\boldsymbol{s}$ can change from training to test cases.

As we pointed out earlier, Rabinowicz and Rosset (2020) assume the input variables in their fixed effect term, Φ, to be i.i.d. samples. In our setting, the input variables in $\boldsymbol{x}_i$ are autocorrelated in time. In other words, the autocorrelation in y_i in our process comes from two sources—the autocorrelation in both $\boldsymbol{x}_i$ and $g(t_i)$. We believe this difference is important and helps explain the difference in the outcome of applying both methods. We revisit this point in Section 4.7 after presenting the numerical results.

Continuing with our modeling process, we model both $f(\cdot)$ and $g(\cdot)$ as realizations of stationary Gaussian processes (GPs) (Rasmussen and Williams, 2006). For $f(\cdot)$, it is assumed to be a sample from a GP with a mean function $\mu(\cdot)$ and a covariance function $k(\cdot,\cdot)$. Specifically, we use a constant mean function and a Matérn covariance function with a smoothness parameter of $\nu = 1.5$ as follows:

$$\mu(\boldsymbol{x}) = \beta, \qquad k(\boldsymbol{x},\boldsymbol{x}') = \sigma_f^2\Big(1+\sqrt{3}r\Big)\exp\Big(-\sqrt{3}r\Big), \quad r = \sqrt{\sum_{\ell=1}^{d}\frac{((\boldsymbol{x})_\ell - (\boldsymbol{x}')_\ell)^2}{(\boldsymbol{\theta})_\ell^2}} \tag{4}$$

where β is an unknown constant, σ_f^2 is the variance of $f(\cdot)$, and $(\cdot)_\ell$ denotes ℓ^{th} component of a vector; for instance, $(\boldsymbol{\theta})_\ell$ is the lengthscale parameter for the ℓ^{th} covariate. We come to this specific choice by experimenting with four covariance functions—squared exponential, two Matérn kernels (smoothness of 1.5 and 2.5), and exponential kernel—and chose the best performing one, which is Matérn with $\nu = 1.5$. While there are some difference in performance, the differences are not that striking; see Supplementary Material S1.

The temporal part $g(\cdot)$ is assumed to be zero mean with a covariance function denoted by $q(\cdot, \cdot)$. We again use a Matérn covariance function with $\nu = 1.5$ for $g(\cdot)$ given as follows:

$$q(t, t') = \sigma_g^2\Big(1 + \sqrt{3}\frac{|t - t'|}{\phi}\Big) \exp\Big(- \sqrt{3}\frac{|t - t'|}{\phi}\Big), \tag{5}$$

where σ_g^2 is the variance of $g(\cdot)$ and ϕ is the lengthscale for time.

3.2 Inference procedure

The key is how to effectively estimate the two components in Equation (2). Recall that our data are observational, not from designed experiments in which the confounding effects can be distinguished through a careful selection of factor settings. Using the observational data, if one conducts the maximum likelihood estimation of the hyperparameters, $(\beta, \sigma_f, \boldsymbol{\theta}, \sigma_g, \phi, \sigma_\epsilon)$, in Equation (2), one would run into an identifiability issue—while attempting to learn the hyperparameters for both $f(\cdot)$ and $g(\cdot)$ together, it is difficult to tell whether the variance in the data is due to some input variables or due to time. We provide numerical evidence on worse performance of joint (direct) estimation in Section 4.6.

Another possible explanation for the inferior performance of direct estimation comes from an intrinsic problem of GP regression. For simplicity, consider the model $y_i = \mu_0 + f_0(\boldsymbol{x}_i) + g_0(t_i) + \epsilon_i$, where μ_0 stands for the global mean value, and f_0 and g_0 are zero-mean GPs. Then there is a known identifiability issue for estimating μ_0; see Tuo and Wu (2016) and Theorem 3 of Wang et al. (2020). This issue in estimation does not affect the prediction properties for applications in which people combine all additive terms for making prediction, because this bias can be compensated by another bias in estimating f_0 and g_0; see Tuo and Wu (2018) and Theorem 2 of Wang et al. (2020). However, in the current context, the estimate of g_0 itself is critical for out-of-temporal predictions, because an inaccurate estimate of g_0 means that $\mu_0 + f_0$ is also inaccurate, resulting in worse out-

of-temporal predictions.

In order to overcome this problem, we decompose the model in Equation (2) as follows:

$$\begin{aligned} y_i &= f(\boldsymbol{x}_i) + u_i, \\ u_i &= g(t_i) + \epsilon_i. \end{aligned}$$

In the first step, we only focus on estimating $f(\cdot)$ such that the variation in y due to temporal correlation of u does not get modeled. The residual left after subtracting the estimate of $f(\cdot)$ would be used to estimate $g(\cdot)$ and σ_ϵ^2.

The first step is equivalent to learning a function with autocorrelated errors. To this end, we adopt an idea frequently used to reduce the autocorrelation in Markov chain Monte Carlo (MCMC) sampling schemes. The method is a subsampling scheme known as *thinning*. The method retains one sample after every T time steps and discards all the samples in between to reduce the autocorrelation; hence the name, thinning, as this results in a thinned dataset. The number T is often referred as the thinning number.

Thinning retains only $1/T$ fraction of the original dataset and discards the rest. We would like to retain all the samples because each sample carries information about the function, which may not be otherwise available in other data points. So, instead of discarding the data, we put the training samples in T number of thinned data bins. Let us denote a data point $(y_i, \boldsymbol{x}_i, t_i)$ as $\mathcal{D}_i$. Then, the first bin will have following data points, $\mathcal{B}_1 := \{\mathcal{D}_1, \mathcal{D}_{T+1}, \mathcal{D}_{2T+1}, \ldots, \mathcal{D}_{\lfloor \frac{N}{T} \rfloor T+1}\}$, where $\lfloor a \rfloor$ rounds a down to its nearest integer. Generally, the j^{th} bin has the following data points:

$$\mathcal{B}_j := \{\mathcal{D}_j, \mathcal{D}_{T+j}, \mathcal{D}_{2T+j}, \ldots, \mathcal{D}_{\lfloor \frac{N}{T} \rfloor T+j}\}.$$

If $\lfloor \frac{N}{T} \rfloor T + j > N$, then $(\lfloor \frac{N}{T} \rfloor - 1)T + j$ would be last element for $\mathcal{B}_j$. We have T bins overall.

Thinning creates a temporal gap between two consecutive data points in a bin, and thus reduces the intra-bin temporal autocorrelation among the training points in any given bin. Hence, for the data points in the same bin, we could assume u to be independent Gaussian noise with some variance $\sigma_u^2 < \infty$. Then, we can proceed to estimate f using a likelihood function of the thinned data.

Let us denote the number of data points in $\mathcal{B}_j$ by n_j and let $\pi_j(\cdot)$ be the function that maps the index of the elements in set $\mathcal{B}_j$ to the index of the original dataset $\mathcal{D}$.

Denote the response vector for $\mathcal{B}_j$ by $\boldsymbol{y}^{(j)} = (y_{\pi_j(1)}, y_{\pi_j(2)}, \ldots, y_{\pi_j(n_j)})^\top$. Let $\mathbf{K}^{(j)}$ be the covariance matrix for $\mathcal{B}_j$ such that the element in r^{th} row and s^{th} column is given by $(\mathbf{K}^{(j)})_{rs} = k(\boldsymbol{x}_{\pi_j(r)}, \boldsymbol{x}_{\pi_j(s)}) \mid r, s = 1 \ldots n_j$, where $k(\cdot, \cdot)$ is the same as defined in Equation (4). Let $\mathbf{I}$ be an identity matrix of a proper size and $\mathbb{1}$ be a vector of ones of a compatible size. Then, the likelihood function for the j^{th} bin $\mathcal{B}_j$ is given as follows:

$$\mathcal{L}_j = \frac{1}{\sqrt{(2\pi)^{n_j}|\mathbf{K}^{(j)} + \sigma_u^2\mathbf{I}|}} \exp\left(-\frac{1}{2}(\boldsymbol{y}^{(j)} - \beta\mathbb{1})^\top[\mathbf{K}^{(j)} + \sigma_u^2\mathbf{I}]^{-1}(\boldsymbol{y}^{(j)} - \beta\mathbb{1})\right) . \tag{6}$$

One could use any individual bin and its likelihood function to estimate the hyperparameters for function f, but doing so makes use of only a fraction of the original data, which is not ideal. Intending to make full use of all the data for estimating f, our approach is to create a pseudo-likelihood function as the product of the likelihood functions of individual bins, namely $\prod_{j=1}^{T} \mathcal{L}_j$. As such, the hyperparameters of function f can be estimated by maximizing this pseudo-likelihood function, as follows:

$$(\hat{\beta}, \hat{\sigma}_f^2, \hat{\boldsymbol{\theta}}, \hat{\sigma}_u^2) = \arg\max \prod_{j=1}^{T} \mathcal{L}_j \ . \tag{7}$$

The temporal correlation between different bins, namely the inter-bin temporal correlation, will still exist, because temporally neighboring data points are now in different bins. But the construction of the pseudo-likelihood function ignores the inter-bin correlations. In other words, from the lens of this pseudo-likelihood function, the bins are not related at all. Optimizing this pseudo-likelihood function forces the estimation of f to the thinned data and will not be affected by the temporal autocorrelation.

The hyperparameters in Equation (7) can be estimated using any optimization routine. In practice, we work with the logarithm of the likelihood function, which changes the finite product structure of the likelihood function to a finite sum. Finite sum functions can also be optimized using parallel processing, that is, each of the summand function can be evaluated independently on a different computing core, which could reduce the computation time.

The hyperparameters therefore estimated will not reflect or minimally reflect the temporal correlation due to u. Once the hyperparameters are estimated, we combine the bins back to create a single dataset and use the single dataset for predictions. We use the following notation: $\mathbf{K}$ is the covariance matrix for all the training points with its element in the i^{th} row and j^{th} column given as $(\mathbf{K})_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) \mid i, j = 1 \ldots N$,

$\boldsymbol{r}(\boldsymbol{x}) = (k(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}, \boldsymbol{x}_N))^\top$ is the correlation vector between any point $\boldsymbol{x}$ and all the training points, and $\boldsymbol{y} = (y_1, \ldots, y_N)^\top$ is the vector of response for all the training points. The value of $\hat{f}(\cdot)$ can be calculated at a given point $\boldsymbol{x}$ as follows:

$$\hat{f}(\boldsymbol{x}) = \hat{\beta} + \boldsymbol{r}^\top(\boldsymbol{x})[\mathbf{K} + \hat{\sigma}_u^2\mathbf{I}]^{-1}(\boldsymbol{y} - \hat{\beta}\mathbb{1}). \tag{8}$$

3.3 Estimating $g(t)$

Let $\hat{\boldsymbol{f}} = (\hat{f}(\boldsymbol{x}_1), \ldots, \hat{f}(\boldsymbol{x}_N))^\top$ be the vector of the estimate of $f(\cdot)$ for all the training points. The vector of residual $\hat{\boldsymbol{e}}$ is calculated as: $\hat{\boldsymbol{e}} = \boldsymbol{y} - \hat{\boldsymbol{f}}$. Each of the residual is associated with a time point. Denote by $\hat{e}_t$ the residual for time point t. We use these residuals as the response for estimating $g(t)$ and σ_ϵ^2.

Here we assume that the autocorrelation in time decays much faster as compared with the overall time span of the training dataset. This is definitely reasonable for our target wind applications, because the autocorrelation in wind speed or other environmental variables only persists in the order of hours (Ding, 2019, Figure 2.3), while our training data spans from a number of months to more than a whole year. For this reason, we do not need all the training points to compute a global estimate for $g(\cdot)$. Instead, we compute a local estimate of $g(t^*)$ at t^*, based only on the training points in the neighborhood of t^*. Doing this substantially reduces the computational burden for estimating $g(\cdot)$.

We use a neighborhood based on the thinning number T, as temporal autocorrelation would be small after a lag of T time units. Thus, we only include the training points that are within $\pm T$ time units from t^*, while estimating $g(t^*)$. Moreover, there is no need to estimate $g(t^*)$ if there happens to be no training points in the T-neighborhood of t^*.

Let us define an index set for training points close to point t^* as $J^* = \{j : |t^* - t_j| \leq T; j = 1 \ldots N\}$. Denote by $\mathbf{Q}^*$ the covariance matrix formed by the time points in J^* such that $(\mathbf{Q}^*)_{ij} = q(t_i, t_j); i, j \in J^*$. Here $q(\cdot, \cdot)$ is the same as defined in Equation (5). Also, denote by $\hat{\boldsymbol{e}}^*$ a vector of residuals with its j^{th} component $(\hat{\boldsymbol{e}}^*)_j = e_{t_j}; j \in J^*$. The hyperparameters for $q(\cdot, \cdot)$ and σ_ϵ^2 is estimated based on the value of t^*, and is given as

$$(\hat{\sigma}_g^2, \hat{\phi}, \hat{\sigma}_\epsilon^2) = \arg\max \frac{1}{\sqrt{(2\pi)^{(|J^*|)}|\mathbf{Q}^* + \sigma_\epsilon^2\mathbf{I}|}} \exp\left(-\frac{1}{2}\hat{\boldsymbol{e}}^{*\top}[\mathbf{Q}^* + \sigma_\epsilon^2\mathbf{I}]^{-1}\hat{\boldsymbol{e}}^*\right). \tag{9}$$

Let $\boldsymbol{s}^*$ denote a covariance vector such that its j^{th} component is $(\boldsymbol{s}^*)_j = q(t^*, t_j); j \in J^*$.

Once the hyperparameters are estimated, $\hat{g}(t^*)$ is given as follows:

$$\hat{g}(t^*) = \boldsymbol{s}^{*\top}[\mathbf{Q}^* + \hat{\sigma}_\epsilon^2 \mathbf{I}]^{-1}\hat{\boldsymbol{e}}^* \tag{10}$$

3.4 Predictions

Once $\hat{f}(\cdot)$ and $\hat{g}(\cdot)$ are estimated, they do not have to be used together in a prediction. If one wants to predict at a time t^*, which is in the far distant future and temporally far away from any training data points, then one only needs $\hat{f}(\boldsymbol{x}^*)$, as $\hat{g}(t^*)$ is going to be zero. The condition to decide whether t^* is temporally enough far away from the training data is to check whether there exists a training data point, t_i, such that $|t^* - t_i| \leq T$. If no, then t^* is temporally far away. We refer to this type of prediction under temporal extrapolation as *out-of-temporal prediction*. By contrast, the *in-temporal prediction* refers to predictions over a time t^* not temporally far away from the training data points.

Of course, a user does not have to check this condition. One can just use $\hat{f}(\boldsymbol{x}^*)+\hat{g}(t^*)$ to predict at any test point $(\boldsymbol{x}^*, t^*)$, regardless of where t^* is. The $\hat{g}(\cdot)$ term takes the temporal distance between t^* and the training data points into consideration and will reduce to zero when t^* is temporally distant from the training data. Simply put, our model can adapt for out-of-temporal versus in-temporal prediction without a user's active involvement.

3.5 Choice of thinning number

The choice of thinning number T is important to reduce the temporal autocorrelation within a data bin. If T is very small, we may still have overfitting problem. On the other hand, if T is very large, the number of data points in each bin may be too low to learn $f(\cdot)$ accurately. The choice of T needs to provide a trade-off between these two aspects. Since, we want to ensure that the temporal autocorrelation is sufficiently reduced, we choose the thinning number as the smallest lag such that the absolute value of the partial autocorrelation function (PACF) for each of the covariates is less than two standard errors of the PACF for a sequence of N i.i.d Gaussian noise, which is considered to be statistically insignificant at the 95% significance level. In other words, the value of T is given as follows:

$$T = \max_{\ell=1,\ldots,d} \min_h (\mid \mathrm{PACF}_{(\boldsymbol{x})_\ell}(h) \mid \leq 2/\sqrt{N}), \tag{11}$$

where $\text{PACF}_{(\boldsymbol{x})_\ell}(h)$ is the PACF for covariate $\ell = 1, \ldots, d$ for lag h. We tested the choice of thinning number in Section 4.5 through a sensitivity analysis on real datasets.

4 Case study: Application to wind turbine datasets

We present two case studies for modeling wind power curves to validate the performance of the proposed method. Both datasets are publicly available. The first case study is based on four datasets. We refer to the first case study as *Case Study I*. The second case study is on a larger number of datasets (thirty turbines) but the data available are for a shorter period of time. We refer to the second case study as *Case Study II*.

4.1 Datasets

Case Study I uses four datasets available on the book website of Ding (2019) (`https://aml.engr.tamu.edu/book-dswe/dswe-datasets/`, Dataset 6). They are associated with four turbines, denoted as WT1 to WT4. The first two datasets are from inland and the remaining are from offshore wind turbines. Each turbine has four years of data collected at a 10-minute frequency. The data for the inland turbines (WT1 and WT2) span from 2008 to 2011, and those for the offshore turbines (WT3 and WT4) extend from 2007 to 2010. Each dataset has five environmental input variables along with the response (y) and time stamp (t) for each data point. The inland turbines have the same input variables: wind speed (V), wind direction (D), air density (ρ), turbulence intensity (I), and wind shear (S). The offshore turbine datasets contain humidity (H) instead of wind shear and have the rest of the four variables same as the inland turbines. One can easily see that all these variables are temporally autocorrelated by plotting their partial autocorrelation function (PACF) plots. One such example for WT1 is provided in the Supplementary Material S2.

Each of the datasets has missing data points. The exact number of data points is given in Table 1. The number of data points are about 50% when compared to the scenario where turbines produce power at all the times (for every 10 minute interval, we have a positive power), which is an ideal condition and not observed in practice. There are two major causes of missing data points: 1) wind speed is either below cut-in speed or above cut-out speed, so there is no power production; 2) wind conditions are favorable but the power

Table 1: Description of the main study datasets.

Dataset	WT1	WT2	WT3	WT4
Type of wind turbine	Inland	Inland	Offshore	Offshore
Time period	2008–2011	2008–2011	2007–2010	2007–2010
Covariates	$\{V, D, \rho, I, S\}$	$\{V, D, \rho, I, S\}$	$\{V, D, \rho, I, H\}$	$\{V, D, \rho, I, H\}$
Number of data points	96,824	89,730	113,378	110,556

output is either curtailed or zero because of grid commitments or operational issues.

Case Study II is based on thirty wind turbines. These datasets are available at `https://github.com/TAMU-AML/Datasets/tree/master/TemporalOverfitting`. The input variables are the same as that of the inland turbines in Case Study I. These thirty datasets can be further classified into groups of ten, as the ten turbines in the same group share the same meteorological tower. The meteorological towers measure wind speed at multiple heights, wind direction, ambient pressure and temperature. The multi-height wind speed measurements are used to calculate the wind shear. Ambient pressure and temperature are used to calculate the air density. Wind direction data are also taken from the meteorological towers. Each of the turbines also measure the wind speed at their nacelle. The data for wind speed and power are collected at individual turbines, and turbine's wind speed data is used to calculate the turbulence intensity. The first meteorological tower has a slightly longer duration of data than the other two towers; see Table 2.

For both case studies, we divide the datasets into three temporally disjoint datasets: $\mathcal{T}_1, \mathcal{T}_2,$ and $\mathcal{T}_3$. In Case Study I, $\mathcal{T}_1$ corresponds to the first two years, and $\mathcal{T}_2$ and $\mathcal{T}_3$ has the data for the third and fourth year, respectively. For example, for the inland turbines (WT1 and WT2), $\mathcal{T}_1$ contains the data for the years 2008 and 2009, and $\mathcal{T}_2$ and $\mathcal{T}_3$ correspond to the year 2010 and 2011, respectively. In Case Study II, the duration corresponding to $\mathcal{T}_1, \mathcal{T}_2,$ and $\mathcal{T}_3$ are listed in Table 2. We select a few methods including our proposed method to learn the power curve for each wind turbine using their corresponding $\mathcal{T}_1$ datasets. The learned model is then used to do out-of-temporal predictions on $\mathcal{T}_2$ and $\mathcal{T}_3$.

Table 2: Duration of the datasets and temporal partitions for Case Study II.

Turbines	Total duration	$\mathcal{T}_1$ duration	$\mathcal{T}_2$ duration	$\mathcal{T}_3$ duration
1 to 10	Apr 29, 2010–Oct 31, 2011	Apr 29, 2010–Nov 30, 2010	Dec 1, 2010–May 31, 2011	Jun 1, 2011–Oct 31, 2011
11 to 20	Jul 30, 2010–Oct 23, 2011	Jul 30, 2010–Dec 31, 2010	Jan 1, 2010–May 31, 2011	Jun 1, 2011–Oct 23, 2011
21 to 30	Jul 30, 2010–Oct 20, 2011	Jul 30, 2010–Jan 31, 2011	Feb 1, 2011–May 31, 2011	Jun 1, 2011–Oct 20, 2011

Table 3: Thinning number, T, for each of the four datasets.

Dataset	WT1	WT2	WT3	WT4
Thinning Number	12	14	22	22

4.2 Implementation and comparison

For implementing our method, we use all the five input variables in the model as the starting point but subset selection is part of the learning task; see Section 4.1 for description of the input variables. Also, since wind direction is a circular variable, we embed it into a two-dimensional Euclidean space by using its *sine* and *cosine* transformations. We standardized all the input variables by subtracting their respective sample mean and dividing them by their sample standard deviation. Standardization of the inputs ensure that the importance of the variables would be clear from their respective lengthscale. A large lengthscale would imply that the input variable is not important in predicting the response.

We proceed by computing the thinning number T for each of the datasets as per Equation (11). The computed value of T for the four datasets in Case Study I is shown in Table 3. Taking the thinning number for WT1 as an example, the value of 12 would be equivalent to 2 hours, as the data points are collected every 10 minutes. This implies that two data points with less than 2 hours of time gap would not be kept in the same bin. Since there are missing data points in the dataset, the time of 2 hours serves as the minimum time gap between two intra-bin points. The actual temporal gap for some of the consecutive data points in a bin would be higher. We bin the datasets using their respective thinning number given in Table 3, and use Equations (6) and (7) to estimate the hyperparameters for the time-invariant function $f(\cdot)$, as described in the previous section. Once we have the hyperparameter estimates for $f(\cdot)$, we compute the out-of-temporal predictions on $\mathcal{T}_2$ and $\mathcal{T}_3$ using just $f(\cdot)$, as given in Equation (8).

We compare our proposed method with two categories of methods: the first category is for the nonparametric power curve methods that do not consider the issues of temporal overfitting, and the second category is for the approaches that address the serial autocorrelation and temporal overfitting issues, which are reviewed in Section 2.

In the first category, we use the following three methods—the IEC binning method, k-nearest neighbors (kNN), and additive multivariate kernel (AMK) by Lee et al. (2015),

but with the focus on out-of-temporal predictions only. We include the binning method because it is the industry baseline. If a proposed method cannot outperform this baseline, the value of that method will be called into question. kNN and AMK are included because the comparison presented in Chapter 5 of Ding (2019) shows that these two methods do a better job than other nonparametric regression methods.

In the second category, we again consider three methods. The first is the time-split cross-validation (Burman et al., 1994; Racine, 2000; Roberts et al., 2017), the second is based on Rabinowicz and Rosset (2020)'s corrected cross-validation error, and third one is based on pre-whitening of the response (Xiao et al., 2003; Geller and Neumann, 2018).

The implementation of the binning method is straightforward; we use a 0.5 m/s bin-width (the IEC standard). For kNN and AMK, one would have to do variable selection. We employ a forward stepwise subset selection using a five-fold cross validation to get the best subset of input variables. The AMK method by Lee et al. (2015), which was specifically developed to model the wind turbine power curves, uses a kernel regression method. In their paper, Lee et al. (2015) consider additive combinations of trivariate kernels, keeping the first two variables common in all the additive terms and varying the third variable. They kept the common variables fixed as wind speed and wind direction. We modify their method by only fixing the first variable as wind speed and let the data decide the second common variable for the additive terms, while still using trivariate kernels. The analyses for both kNN and AMK are done in `R` using the `DSWE package` (Kumar et al., 2021).

In order to do time-split cross-validation, we use kNN as the base method, but modify the cross-validation scheme. We divide the temporally ordered data into small blocks of size T—same as the thinning number used in our proposed method. Instead of randomly sampling training and test datasets, we select training and test samples from the temporal blocks in a way such that if a particular temporal block is in test set, its neighboring blocks must not be in the training set. This splitting ensures that there is low temporal correlation between training and test datasets. A schematic of time-split cross-validation is given in Figure 3. We do a five-fold time-split validation by sampling different test datasets and refer to the resulting method as TS-kNN, namely time-split kNN.

Time-split cross-validation can also be clubbed with any other base method such as AMK; however, given the size of the datasets, doing so is computationally expensive. For

example, it took more than 10 hours to find the best subset of variables and compute the optimal bandwidths for WT1 when AMK method was clubbed with time-split cross-validation whereas TS-kNN took less than 15 minutes to do the same. Also, the result obtained using time-split validation on AMK did not show any significant improvement over standard AMK which uses a direct plug-in (DPI) approach by Ruppert et al. (1995) for estimating the bandwidths. So, we decide to proceed with only kNN as the base method.

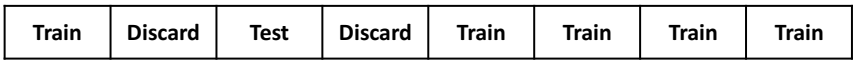

Figure 3: A schematic of time-split cross-validation. Each block represents a group of temporally adjacent data points.

For Rabinowicz and Rosset (2020)'s method, we again use kNN as the base method for the same reasons described for time-split validation; cross-validation with AMK is computationally prohibitive for the data size at hand. We refer to the resulting kNN method as CVc-kNN. The CVc method relies on estimating the covariance matrix of the response, $Cov(\boldsymbol{y}, \boldsymbol{y})$, conditioned on the input variables. Here we run into a problem. For our problem setting, we need to estimate the covariance in $\boldsymbol{y}$ due to $g(t)$ but the covariance in the response data are caused by the autocorrelation in both $\boldsymbol{x}$ and $g(t)$. To apply the idea of CVc, we come up with an *an hoc* procedure, which is to first fit a one-dimensional kNN model to the response data and then use the residual to compute $Cov(\boldsymbol{y}, \boldsymbol{y})$. The thought behind is that the single input of wind speed is the most important variable in power curve models. Subtracting its effect would remove a major portion of the covariance in $\boldsymbol{y}$ due to the temporal autocorrelation in $\boldsymbol{x}$. Once $Cov(\boldsymbol{y}, \boldsymbol{y})$ is estimated, we keep it fixed and do a forward subset selection, based on CVc, to find the best variables subset and corresponding hyperparameter (k). It is apparent that the lack of a quality estimate of $Cov(\boldsymbol{y}, \boldsymbol{y})$ under our problem setting presents a major roadblock to the effective application of CVc (more discussion in Section 4.7).

The last method is based on pre-whitening of the response (Xiao et al., 2003; Geller and Neumann, 2018). We follow the steps in Xiao et al. (2003), but with AMK as our base method. Here, we use AMK not only because AMK as the base method is a better choice than kNN, but also because Xiao et al. (2003) uses a kernel-based local polynomial method,

meaning that AMK is more compatible. We did not use AMK to be clubbed with time-split cross-validation, due to heavy computation that would have otherwise resulted. But for pre-whitening, as it does not require cross-validation to compute the optimal bandwidths, this computational burden is not there. In other words, AMK can be used along with DPI approach once the response is modified.

Specifically, we fit an AMK model to the datasets and estimate the residuals at the training points. We use these residuals to fit an autoregressive (AR) model of order T—the computed thinning number. We then modify the response by subtracting the estimated autoregressive component of residuals from the response. The modified response is then used to build a new AMK model, which would be used for predictions. Since we have already obtained the best input variable subset using forward subset selection for AMK, we use the same subset of input variables for the model and do not carry out subset selection again. We refer to this pre-whitened AMK model as PW-AMK.

4.3 Results for Case Study I

We present the results in two parts, corresponding to the two categories of methods. The first part compares our method with the power curve methods in Category 1, in which we use the binning method as the benchmark and compute performance improvement over binning for each method. The performance criteria is the root mean square error (RMSE) on a test dataset. Our proposed method is referred to as tempGP in the comparison.

Table 4: A comparison table for out-of-temporal RMSE for dataset $\mathcal{T}_2$

Dataset	Binning	kNN	AMK	tempGP ($f(\boldsymbol{x})$)
WT1	4.98	4.96 (0.4%)	4.35 (12.7%)	**3.52** **(29.3%)**
WT2	4.93	4.66 (5.3%)	4.32 (12.2%)	**3.69** **(25.0%)**
WT3	3.95	4.11 (−4.1%)	3.50 (11.4%)	**3.19** **(19.2%)**
WT4	3.73	3.96 (−6.5%)	3.47 (6.7%)	**2.94** **(21.0%)**

Tables 4 and 5 present the performance of binning, kNN, AMK, and tempGP for out-of-temporal predictions on test dataset $\mathcal{T}_2$ and $\mathcal{T}_3$, respectively. Since $\mathcal{T}_2$ and $\mathcal{T}_3$ are temporally disjoint from $\mathcal{T}_1$, we only use the estimate for the time-invariant function $f(\boldsymbol{x})$ for predic-

Table 5: A comparison table for out-of-temporal RMSE for dataset $\mathcal{T}_3$

Dataset	Binning	kNN	AMK	tempGP ($f(\boldsymbol{x})$)
WT1	5.03	4.98 (0.8%)	4.47 (11.0%)	**4.11 (18.1%)**
WT2	5.17	5.04 (2.3%)	4.78 (7.4%)	**4.48 (13.2%)**
WT3	4.23	4.68 (−10.6%)	4.12 (2.6%)	**3.83 (9.5%)**
WT4	3.64	4.05 (−11.3%)	3.14 (13.7%)	**2.84 (22.0%)**

tions. We highlight in boldface font the best prediction performance, i.e., whichever has the lowest RMSE. The values in parentheses denote the percentage improvement over the binning method. A negative percentage implies a worse performance than binning. Evidently, tempGP outperforms both the industry baseline (binning) and the data science competitors (kNN and AMK). In other words, explicitly avoiding the temporal structure in the learned function improves the performance of out-of-temporal predictions.

Next, we present the results for the second category of methods (TS-kNN, CVc-kNN, and PW-AMK) in Tables 6 and 7. We also append the results for kNN and AMK from Tables 4 and 5 for easier comparison. The last column of the tables (% Imp) shows the percentage improvement for tempGP over the second best method; note that the second best method differs for different datasets. For the cross-validation based methods, TS-kNN and CVc-kNN, we notice some improvement in performance as compared to their counterpart kNN, in most of the cases. However, our proposed method still outperforms these methods. The pre-whitening method PW-AMK shows a small but sometimes no improvement over its counterpart AMK. Pre-whitening relies on the autocorrelation in the residuals to modify the response. Lee et al. (2015) show that the autocorrelation in the residuals of the AMK model is still there but nonetheless weakened. As Roberts et al. (2017) explained, the temporal structure in the residual can easily get modeled through some input variables when multiple autocorrelated input variables are present, masking the autocorrelation of the residual. Thus, this weakened autocorrelation in the residual may not be strong enough to modify the response significantly in the pre-whitening step.

Overall, looking at the results of these four datasets, our proposed method is a clear winner. However, the second best method varies from case to case. Interestingly, the

Table 6: A comparison table for out-of-temporal RMSE for dataset $\mathcal{T}_2$ using methods that account for serial autocorrelation.

Dataset	tempGP	TS-kNN	CVc-kNN	PW-AMK	kNN	AMK	% Imp
WT1	**3.52**	4.06	4.10	4.38	4.96	4.35	13.3%
WT2	**3.69**	4.59	4.70	4.38	4.66	4.32	14.6%
WT3	**3.19**	3.98	3.73	3.39	4.11	3.50	5.8%
WT4	**2.94**	3.82	3.55	3.47	3.96	3.47	15.2%

Table 7: A comparison table for out-of-temporal RMSE for dataset $\mathcal{T}_3$ using methods that account for serial autocorrelation.

Dataset	tempGP	TS-kNN	CVc-kNN	PW-AMK	kNN	AMK	% Imp
WT1	**4.11**	4.26	4.25	4.49	4.98	4.47	3.3%
WT2	**4.48**	5.14	5.12	4.82	5.04	4.78	6.3%
WT3	**3.83**	4.32	4.11	3.97	4.68	4.12	3.6%
WT4	**2.84**	3.74	3.51	3.19	4.05	3.14	9.6%

standard version of AMK, which does not model the temporal structure, turn out to be the second best in some of the cases. Rabinowicz and Rosset (2020) explain that if the correlation structure in training and test datasets are the same, there is no need for a special treatment of the correlation structure in the data, and the standard methods would perform well. In practice, we do not know the temporal correlation structure in the training and test datasets, and thus cannot guarantee if the correlation pattern would stay the same. Thus, it is a good idea to assume different correlation structure for training and test sets when one is not certain that the correlation structures are the same. This argument becomes more convincing as we extend our case study to a larger set of datasets. There we notice that not handling the temporal structure results in much worse predictions.

4.4 Results for Case Study II

We extend our case study on another thirty datasets. Since the datasets in Case Study II are of smaller size, we also used a regular version of Gaussian process, that is, without the $g(t)$ term and thinning, and refer it as regGP. In order to present the results concisely, we use plots instead of the tables. Figure 4 (left panels) presents the relative RMSEs of regGP, tempGP, kNN and AMK with respect to binning. To obtain the relative RMSEs, we divide all the RMSEs for different methods by the RMSE of the binning method for each turbine, so that the relative RMSEs in the same scale. Thus, a value larger than one implies performance deterioration over binning; for example, a relative RMSE of 1.1

implies that the method performs 10% worse than binning. A relative RMSE smaller than one implies performance improvement over binning. The dashed horizontal line at one is for the binning method. The actual RMSEs are presented in Supplementary Material S4. We also plot the prediction intervals for select turbines in Supplementary Material S5.

We notice that for the first out-of-temporal dataset $\mathcal{T}_2$ (Figure 4a), the performance of kNN, AMK, and, regGP are much worse than that of the second out-of-temporal dataset $\mathcal{T}_3$ (Figure 4c). As mentioned earlier, these datasets are for 15 to 18 months time periods. Thus, equal division of these datasets into $\mathcal{T}_1$, $\mathcal{T}_2$, and $\mathcal{T}_3$ results in a half year time span for each of $\mathcal{T}_1$, $\mathcal{T}_2$, and $\mathcal{T}_3$. What this means is that if $\mathcal{T}_1$ cover the first half of a year, then $\mathcal{T}_2$ covers the second half and $\mathcal{T}_3$ covers the first half of the next year. Due to the seasonal difference of environmental variables, principally that of wind and temperature, it is commonly understood that using the first half year data to predict the second half of the same year is harder than using the first half year data to predict the same first half year of the next year. In another angle, $\mathcal{T}_1$ and $\mathcal{T}_2$ have rather different temporal structure but $\mathcal{T}_1$ and $\mathcal{T}_3$ share a similar temporal structure. It does not therefore come as a surprise that a model temporally-overfitted on $\mathcal{T}_1$ could perform much worse on $\mathcal{T}_2$ but reasonably well on $\mathcal{T}_3$. We stress that tempGP performs uniformly better for both out-of-temporal test datasets, although the performance gain is admittedly much more pronounced for $\mathcal{T}_2$.

Figure 4 (right panels) presents the relative RMSEs (still relative to binning) for the second category of methods—TS-kNN, CVc-kNN, and PW-AMK—along with tempGP. When the temporal overfitting causes a much worse performance for standard regression methods (kNN, AMK, and regGP), which is the case for $\mathcal{T}_2$, the second category of methods that address temporal overfitting provides a significant improvement over their non-temporal counterpart, except for PW-AMK. When temporal overfitting does not result in a worse performance (for $\mathcal{T}_3$), we do not see much help from these temporal methods. It may be fair to say that these temporal methods are not very sensitive to weak temporal overfitting.

4.5 Further experiments and simulations

The main parameter used in our method is the thinning number, as it regulates the temporal autocorrelation in each of the data bins. Thus, in order to highlight the importance of thinning and validate our choice of thinning number, we did a sensitivity analysis using

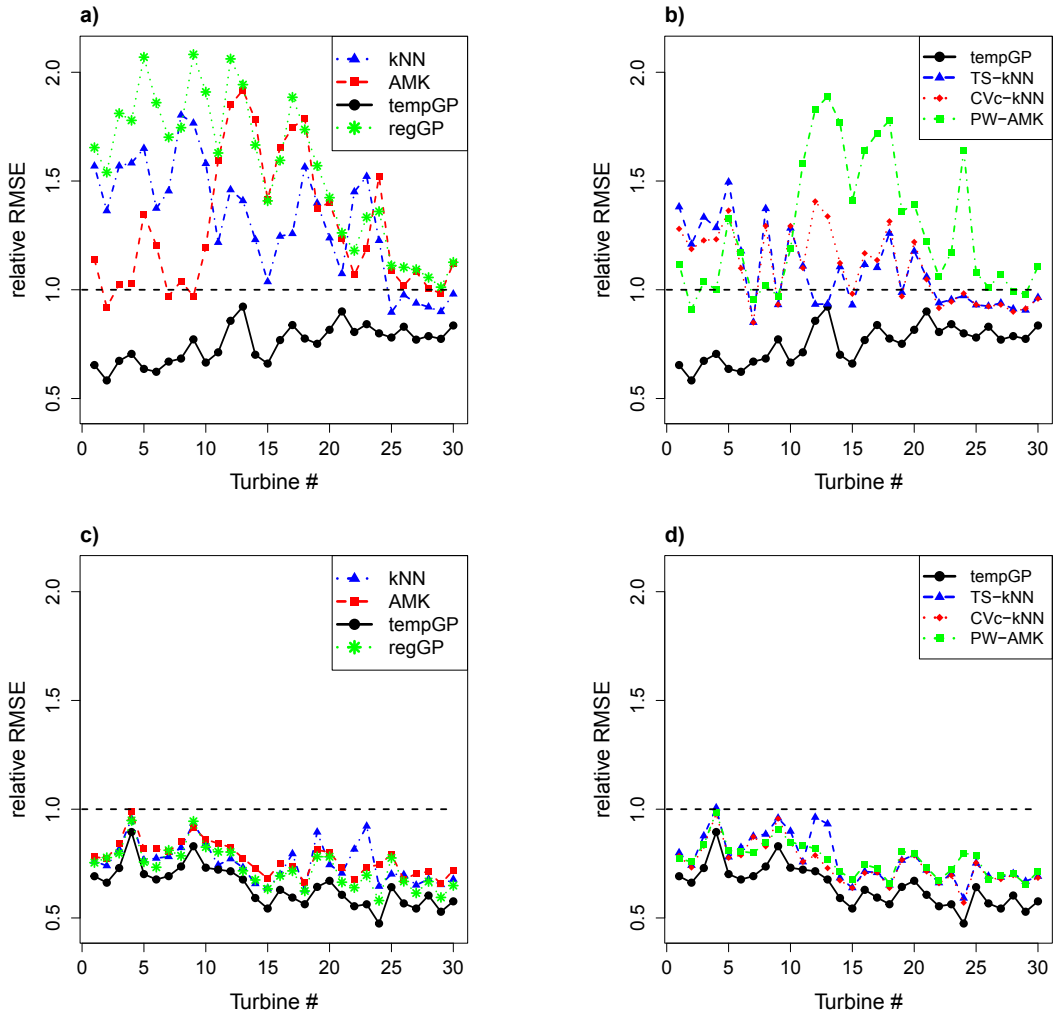

Figure 4: Relative RMSEs as compared to binning RMSE for out-of-temporal datasets. The top two plots are for dataset $\mathcal{T}_2$ with the top-left plot a) for kNN, AMK, tempGP, and regGP and the top-right plot b) for TS-kNN, CVc-kNN, PW-AMK, and tempGP. The bottom two plots are for dataset $\mathcal{T}_3$ with the bottom-left plot c) for kNN, AMK, tempGP, and regGP and the bottom-right plot d) or TS-kNN, CVc-kNN, PW-AMK, and tempGP.

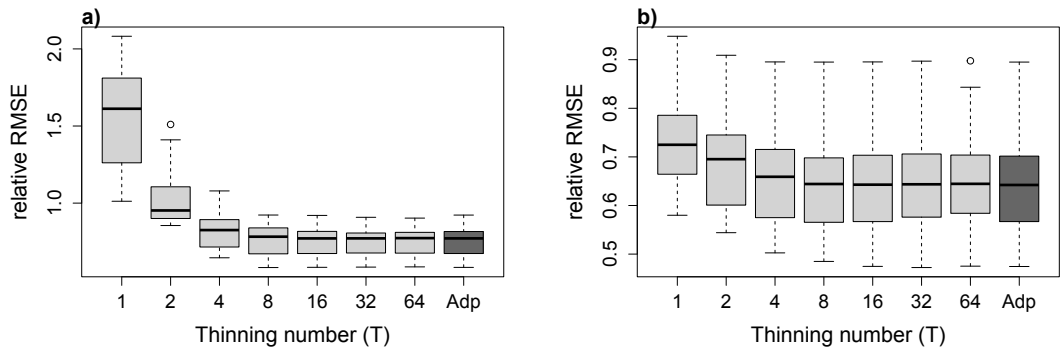

Figure 5: Box plots for relative RMSE using different thinning numbers for all the turbines: a) for test set $\mathcal{T}_2$; b) for test set $\mathcal{T}_3$. "Adp" denotes the adaptive thinning number computed using the proposed approach.

different thinning numbers on Case Study II datasets. We consider the following thinning numbers: $1, 2, 2^2 = 4, \ldots, 2^6 = 64$. The thinning number computed from our proposed approach for these datasets vary between 14 and 17 with a majority of them being 15. Using these thinning numbers, we re-estimate the function f and recompute the test errors for $\mathcal{T}_2$ and $\mathcal{T}_3$. A thinning number of 1 implies no thinning at all, which is essentially a regular version of GP model without the $g(t)$ term, same as regGP in Case Study II.

We present the box plots for relative RMSEs (defined in Section 4.4) for all the turbines in Figure 5. The advantage of thinning is much more pronounced in $\mathcal{T}_2$ than $\mathcal{T}_3$. This is consistent with our previous comments in Section 4.4 about $\mathcal{T}_1$ and $\mathcal{T}_2$ having different temporal structure, and $\mathcal{T}_1$ and $\mathcal{T}_3$ having similar temporal structure because they are approximately the same time period of two consecutive years. We see that when temporal structure between the training and test datasets are different, thinning plays an important role in improving the performance, and our proposed approach for computing the thinning number, referred to as "Adp" in the figure, proves to be quite effective.

We also applied our method on a simulated function where the ground truth is known. The details of the simulation study is available in Supplementary Material S6.

4.6 Direct inference of $f(\cdot)$ and $g(\cdot)$

In the model inference section, we state that estimating the hyperparameters of $f(\cdot)$ and $g(\cdot)$ jointly via a maximum likelihood estimation results in an identifiability problem, which

can result in unreliable hyperparameter estimates and thereby leads to considerable deterioration in prediction performance. We here provide the numerical evidence on the 30-turbine $\mathcal{T}_2$ datasets used in Case Study II. Figure 6 presents the histograms of the ratios of the out-of-temporal RMSEs obtained by using the jointly estimated hyperparameters over that obtained by the thinning-based inference.

We find that under the best case scenario, the direct (joint) estimation results in an error rate that is 6% worse than that of the thinning-based inference, and under the worst case scenario, the direct estimation results in an error rate that is 80% worse, that is the ratio of out-of-temporal RMSE for direct estimation vs thinning-based estimation is approximately 1.8. Out of the 30 turbines, 28 cases are at least 10% worse.

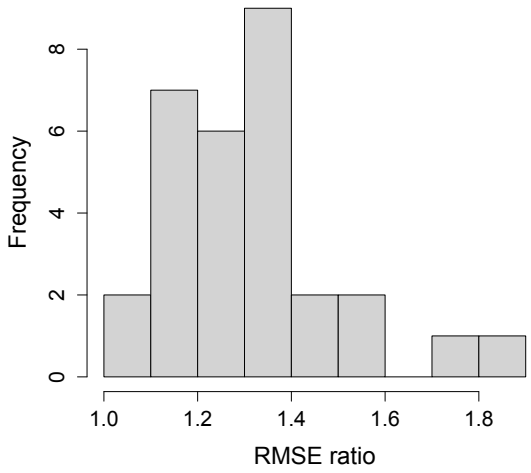

Figure 6: Ratios of the out-of-temporal RMSEs for $\mathcal{T}_2$ obtained from jointly estimated hyperparameters over those using hyperparameters from the thinning-based inference.

Another advantage of thinning-based approach over the direct estimation is the computation time. In general, the time complexity for fitting a GP model is of the order $\mathcal{O}(n^3)$, where n is the number of data points. However, since we bin the data into T bins such that each bin has approximately $n/T = m$ observations and use the pseudo-likelihood defined in Section 3.2, the time complexity for fitting the proposed model is in the order of $\mathcal{O}(Tm^3) = \mathcal{O}(nm^2)$, which is lower than $\mathcal{O}(n^3)$ for any $T > 1$.

4.7 Further discussions on CVc

Looking at all methods that account for correlation in data, CVc looks like the most promising alternative to tempGP. CVc, however, has its own limitations.

The main challenge that arises in our problem setting is the estimation of $Cov(\boldsymbol{y}, \boldsymbol{y})$, which plays a critical role in correcting the standard CV error estimation. As noted in Section 3, Rabinowicz and Rosset (2020) assume the input variables to be i.i.d. Under their assumption, $Cov(\boldsymbol{y}, \boldsymbol{y})$ can be directly estimated using sample covariance matrix of $\boldsymbol{y}$ because no covariance in $\boldsymbol{y}$ is due to the input variables in $\boldsymbol{x}$. For our problem setting, however, $\boldsymbol{x}$ are autocorrelated. We need to estimate the covariance in $\boldsymbol{y}$ due to $g(t)$. This dual autocorrelation makes it harder to estimate $Cov(\boldsymbol{y}, \boldsymbol{y})$ accurately. Strictly speaking, CVc as presented in Rabinowicz and Rosset (2020) is not directly applicable to our model.

Our *ad hoc* procedure in Section 4.2 is an attempt to estimate $Cov(\boldsymbol{y}, \boldsymbol{y})$ in the presence of the dual autocorrelation. We acknowledge that the *ad hoc* procedure may not be the best approach, but it remains unknown how to estimate the covariance in $\boldsymbol{y}$ due to $g(t)$ under our problem setting. While devising the *ad hoc* procedure, we used the residuals of a fixed one-dimensional kNN model to estimate $Cov(\boldsymbol{y}, \boldsymbol{y})$. One may ask if it would be better to increase the number of input variables in the kNN model while estimating $Cov(\boldsymbol{y}, \boldsymbol{y})$? Using a multivariate model weakens the correlation in the residuals, leading to a different estimate of $Cov(\boldsymbol{y}, \boldsymbol{y})$. As such, the issue of variable selection gets entangled with the estimate of CVc. It is unclear to us which multivariate model should be used for estimating $Cov(\boldsymbol{y}, \boldsymbol{y})$. Given these challenges with CVc, tempGP appears better suited for the application at hand. The empirical evidence is rather strong in supporting this claim.

5 Conclusion

We explore a class of regression problems when the input variables and errors are serially correlated over time. Classical regression, which works under the independence assumption, results in overfitted models, known as temporal overfitting. We propose a method to reduce temporal overfitting by explicitly modeling the temporal correlation in the data. We split the variance in response into a time-independent function and a temporally autocorrelated stochastic process. We take advantage of an idea frequently used in Bayesian

statistics—thinning. Using the thinned data, the time-independent function can be separately estimated from the temporally autocorrelated model term.

The thinning-based idea is one of the approaches that can be used to learn the time-independent function. An alternative approach could be to regularize the time-independent function f, or constrain it, following an idea first proposed in Ba and Joseph (2012). Ba and Joseph (2012) also considers an additive model with two GP terms. They separate the effect of the two terms by ensuring that one term is smoother than the other and then constraining the lengthscale of the two kernels accordingly. Unlike in Ba and Joseph (2012) where the two GP terms take the same input, f and g in our model take different inputs, and as a result, it is not immediately clear how the lengthscales of the respective kernels should be constrained, but this could be an interesting future work to pursue.

A final note is that while the paper highlights the problem of temporal overfitting in wind power curves, we believe that the wind energy problem is just one of the many application areas where one could encounter temporal overfitting. Many real datasets in engineering and life sciences are collected over time and could be autocorrelated due to the inertia in the underlying physical processes. We are confident that the resulting methodology is generic and could benefit other nonparametric regressions of the same nature.

Supplementary Material

Supplementary Material: The PDF file contains: (S1) Results for Case Study II with different covariance functions, (S2) PACF plots for WT1, (S3) Hyperparameter estimates, (S4) Actual RMSEs for Case Study II, (S5) Prediction intervals for select turbines, and (S6) Experiments on a simulated function.

Computer Code: The computer code to reproduce all the results in this paper are available on `GitHub` at `https://github.com/TAMU-AML/tempGP-Paper`. A generic `R` function for applying the tempGP algorithm to any dataset is available in `DSWE` package in `R` available through CRAN at `https://CRAN.R-project.org/package=DSWE`.

Funding

Prakash and Ding's research is partially supported by NSF IIS-1741173; Tuo's research by NSF DMS-1914636; and Ding and Tuo's research also by NSF grant CCF-1934904.

References

Altman, N. S. (1990). Kernel smoothing of data with correlated errors. *Journal of the American Statistical Association*, 85(411):749–759.

Ba, S. and Joseph, V. R. (2012). Composite Gaussian process models for emulating expensive functions. *The Annals of Applied Statistics*, 6(4):1838–1860.

Bessa, R. J., Miranda, V., Botterud, A., Wang, J., and Constantinescu, E. M. (2012). Time adaptive conditional kernel density estimation for wind power forecasting. *IEEE Transactions on Sustainable Energy*, 3(4):660–669.

Burman, P., Chow, E., and Nolan, D. (1994). A cross-validatory method for dependent data. *Biometrika*, 81(2):351–358.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298.

Chu, C.-K. and Marron, J. S. (1991). Comparison of two bandwidth selectors with dependent errors. *The Annals of Statistics*, 19(4):1906–1918.

De Brabanter, K., Cao, F., Gijbels, I., and Opsomer, J. (2018). Local polynomial regression with correlated errors in random design and unknown correlation structure. *Biometrika*, 105(3):681–690.

De Brabanter, K., De Brabanter, J., Suykens, J. A., and De Moor, B. (2011). Kernel regression in the presence of correlated errors. *Journal of Machine Learning Research*, 12(55):1955–1976.

Ding, Y. (2019). *Data Science for Wind Energy*. Chapman & Hall, Boca Raton, FL.

EIA (2021). US Energy Information Administration. Webpage: `https://www.eia.gov/energyexplained/wind/electricity-generation-from-wind.php`, Accessed on: 2021-09-17.

Geller, J. and Neumann, M. H. (2018). Improved local polynomial estimation in time series regression. *Journal of Nonparametric Statistics*, 30(1):1–27.

Gu, C. (2013). *Smoothing Spline ANOVA Models*. Springer-Verlag, New York, 2nd edition.

Gu, C. (2014). Smoothing spline ANOVA models: R package gss. *Journal of Statistical Software, Articles*, 58(5):1–25.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, NY, 2nd edition.

IEC (2005). *IEC TS 61400-12-1 Ed. 1, Power Performance Measurements of Electricity Producing Wind Turbines*. Geneva, Switzerland.

Kumar, N., Prakash, A., and Ding, Y. (2021). *DSWE: Data Science for Wind Energy*. R package version 1.5.1, `https://CRAN.R-project.org/package=DSWE`.

Lee, G., Ding, Y., Genton, M. G., and Xie, L. (2015). Power curve estimation with multivariate environmental factors for inland and offshore wind farms. *Journal of the American Statistical Association*, 110(509):56–67.

Meyer, H., Reudenbach, C., Hengl, T., Katurji, M., and Nauss, T. (2018). Improving performance of spatio-temporal machine learning models using forward feature selection and target-oriented validation. *Environmental Modelling & Software*, 101:1–9.

Opsomer, J., Wang, Y., and Yang, Y. (2001). Nonparametric regression with correlated errors. *Statistical Science*, 16(2):134–153.

Rabinowicz, A. and Rosset, S. (2020). Cross-validation for correlated data. *Journal of the American Statistical Association*, in press and online available, DOI: 10.1080/01621459.2020.1801451.

Racine, J. (2000). Consistent cross-validatory model-selection for dependent data: hv-block cross-validation. *Journal of Econometrics*, 99(1):39–61.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, MA.

Roberts, D. R., Bahn, V., Ciuti, S., Boyce, M. S., Elith, J., Guillera-Arroita, G., Hauenstein, S., Lahoz-Monfort, J. J., Schröder, B., Thuiller, W., Warton, D. I., Wintle, B. A., Hartig, F., and Dormann, C. F. (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, 40(8):913–929.

Ruppert, D., Sheather, S. J., and Wand, M. P. (1995). An effective bandwidth selector for local least squares regression. *Journal of the American Statistical Association*, 90(432):1257–1270.

Sheridan, R. P. (2013). Time-split cross-validation as a method for estimating the goodness of prospective prediction. *Journal of chemical information and modeling*, 53(4):783–790.

Tuo, R. and Wu, C. J. (2016). A theoretical framework for calibration in computer models: Parametrization, estimation and convergence properties. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):767–795.

Tuo, R. and Wu, C. J. (2018). Prediction based on the Kennedy-O'Hagan calibration model: Asymptotic consistency and other properties. *Statistica Sinica*, 8(2):743–759.

Vapnik, V. (2000). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 2nd edition.

Wang, W., Tuo, R., and Wu, C. J. (2020). On prediction properties of kriging: Uniform error bounds and robustness. *Journal of the American Statistical Association*, 115(530):920–930.

Xiao, Z., Linton, O. B., Carroll, R. J., and Mammen, E. (2003). More efficient local polynomial estimation in nonparametric regression with autocorrelated errors. *Journal of the American Statistical Association*, 98(464):980–992.