Weighted Machine Learning for Spatial-Temporal Data

Mahdi Hashemi and Hassan A. Karimi

***Abstract*—Applying machine learning techniques to spatial-temporal data poses the question that how the recorded location and time for training samples should contribute to the training and testing process. The prior knowledge of how spatial-temporal phenomena are autocorrelated cannot be properly captured by machine learning techniques, which either ignore location and time altogether or consider them as input features. Not to mention that the latter approach leads to slightly increased sparseness of data in the feature space and more free parameters in the predictor; thus, demanding for larger training datasets. We use the prior knowledge about the spatial-temporal autocorrelation to determine how relevant each training sample would be, given its spatial and temporal distances to the irresponsive (unlabeled) sample. Weighted machine learning techniques use this prior knowledge by taking the relevance of training samples with regard to the irresponsive sample into account as training samples' weights. The proposed approach overcomes the aforementioned issues by enriching the training process with the prior knowledge about spatial-temporal autocorrelation. Because the spatial-temporal weight of training samples depends on the irresponsive sample's location and time, the machine needs to be trained separately for each irresponsive sample. However, we show that in practice using only a small subset of training samples with largest spatial-temporal weights not only mitigates the training time but also results in the best accuracy in most cases.**

***Index Terms*—Analytical learning, autocorrelation, inductive learning, machine learning, spatial data, temporal data.**

I. Introduction

DATA from locations near one another in space are more likely to be similar than data from locations remote from one another" [1]–[6]. This observational fact is called spatial autocorrelation [2], [3], [7]–[14] and makes spatial data different from other types of data. The same definition is true in time [3], [15], [16]–[18], referred to as temporal autocorrelation. Temporal data also might have an additional cyclic autocorrelation [19]–[25] termed cyclic temporal autocorrelation. Because of spatial and temporal autocorrelations, spatial-temporal data are not truly random. In other words, phenomena do not vary randomly through space and time.

The spatial-temporal autocorrelation model shows how the autocorrelation (similarity between observations as a function of the space or time lag between them) among observations changes over space and time. The autocorrelation model considers general behaviors of spatial-temporal phenomena: 1) as spatial or temporal distance between observations increases, their systematic similarity decreases; and 2) observations show periodic similarities over time. Instances of these behaviors are abundant in real life, especially in environmental phenomena. For example, 1) temperature is more similar between two locations/times that are close to each other but we do not expect it to be similar between two locations/times that are too far from each other; and 2) temperature has a well-known yearly cycle. These behaviors can be extended to other spatial-temporal phenomena, such as elevation, air or water pollution, soil type, population, landuse, and landslide.

Franklin [26], in her review paper, introduced the spatial dependence/autocorrelation as a source of information that has yet to be exploited in vegetation prediction models. O'Sullivan and Unwin [1] raised the concern with applying machine learning techniques to spatial data by briefly mentioning, in their book on geographic information analysis, that special characteristics of spatial data are ignored in regression and classification models applied by geographers. Shekhar *et al.* [12], [13], [27] and Shekhar and Chawla [3] showed that spatial autocorrelation limits the usefulness of conventional classification and regression techniques for extracting spatial patterns. Santibanez *et al.* [7], [8] also raised this issue by stating that "machine learning algorithms are in general, not designed to deal with spatially autocorrelated data." The assumption of independent and identically distributed random variables is not valid for spatial data because spatial autocorrelation causes the prediction residuals to exhibit clustering over geographic space [3], [7]–[13], [28], [29].

On the other hand, some researchers showed the reversibility (cyclic behavior) of landuse changes [19]–[22] and earthquakes [17], [25] in time. Mertens and Lambin [19] showed that landuse predictions are more reliable in long term when more historic training samples are available. Britto *et al.* [30] investigated the usefulness of a dynamic selection approach to consider the seasonality of data in selecting the neighborhoods. Yet, developing machine learning techniques that capture the cyclic behavior of temporal phenomena and adjust their predictions based on the irresponsive (unlabeled) sample's time has not been fully addressed in the literature.

Current machine learning techniques treat spatial-temporal problems no differently than other types of problems. Current machine learning techniques do not take into account spatial

Manuscript received October 2, 2019; revised February 19, 2020; accepted May 17, 2020. Date of publication May 28, 2020; date of current version June 18, 2020. (*Corresponding author: Mahdi Hashemi*.)

Mahdi Hashemi is with the Department of Information Sciences and Technology, George Mason University, Fairfax, VA 22030 USA (e-mail: mhashem2@gmu.edu).

Hassan A. Karimi is with the School of Computing and Information, University of Pittsburgh, Pittsburgh, PA 15213 USA (e-mail: hkarimi@pitt.edu).

Digital Object Identifier 10.1109/JSTARS.2020.2995834

and/or temporal autocorrelations, neither in training nor in testing the predictor. That results in poor performance of machine learning techniques in the presence of spatial-temporal data [3], [7]–[13], [15], [31]. On the other hand, taking location and time as features in the training process is not the best way to incorporate the result of autocorrelation [3], [12], [13], [27] as it leaves autocorrelated prediction residuals behind [28], [29], [32], not to mention it will increase the sparseness of training samples in the feature space. It also slightly increases the number of free parameters in the predictor and consequently the demand for larger training datasets, referred to as curse of dimensionality [33]–[35].

The contribution of this article is the formulation of spatial-temporal autocorrelations of geographic phenomena and incorporating them as external knowledge in training weighted machine learning techniques. The proposed approach prevents occurrence of the problems associated with considering location and time as features, potentially improves the prediction accuracy by biasing the predictor in favor of more important training samples, and expedites the training process by leaving out training samples with very low spatial-temporal weights. The accuracy and time performance of the proposed approach will be compared against the following approaches:

1) ignoring location and time and using nonweighted machine learning techniques;
2) considering location and time as additional features in nonweighted machine learning techniques;
3) considering location and time as the only features in nonweighted machine learning techniques;
4) estimating the irresponsive sample's response based on the weighted votes (i.e., spatial-temporal weights) of training samples' responses.

The phrase, irresponsive sample, in this article refers to samples (aka observations or points) whose response (aka label, dependent variable, or output value) is not known. In other words, only the feature vector is observed. Conversely, if the response is known for a sample, it is referred to as a responsive sample. This terminology is preferred over training and test samples because responsive and irresponsive samples could both be used as training samples in semi-supervised machine learning. This study is only concerned with supervised machine learning, however.

The rest of this article is structured as follows. Section II provides a review of the related literature. Sections III-A and III-B explain how the training samples' weights are calculated using spatial and temporal semivariograms and discuss the weighted and nonweighted machine learning models applied in this work, respectively. Section IV includes experiments with real spatial-temporal datasets to compare the accuracy and time performance of the proposed approach with traditional ones. Finally, Section V concludes this article by providing insight into the proposed approach and future directions.

II. Related Work

The literature on machine learning for spatial-temporal data deploys two general strategies toward applying location and time during training and testing the machine: 1) ignoring location and time altogether, and 2) considering them as input features. While this section focuses on studies deploying the latter method, examples of the former could be found in [19], [23], and [36]–[42]. It is noteworthy that our proposed methodology is different from existing approaches because location and time contribute in training the machine through weights that are assigned to samples not as features. This is theoretically more in line with spatial-temporal autocorrelation and will indicate to be experimentally more accurate.

This section focuses on machine learning studies for spatial-temporal data. Li *et al.* [43] attempted to predict seabed mud content in the southwest Australian margin, using machine learning techniques, such as support vector machines (SVM), regression tree (RT), and random forest (RF) and spatial interpolation methods, such as inverse distance squared (IDS) and kriging. They showed that combining machine learning with spatial interpolation improves the accuracy over applying either one of them in isolation. In their methodology, machine learning is applied first, next the spatial interpolation is applied to the residuals of the machine learning predictions, and finally the interpolated residual values are added to the predicted values to produce the final predictions. The input features include latitude, longitude, distance-to-coast, bathymetry, seabed-slope, as well as their second and third powers, multiplication of latitude and longitude, multiplication of longitude to the second power of latitude, and multiplication of latitude to the second power of longitude. RF combined with ordinary kriging (RF-OK), RF combined with IDS (RF-IDS), RF, and RT combined with ordinary kriging (RT-OK) achieved the highest accuracies in their experiments, respectively. A combination of SVM (with a linear or Gaussian kernel) with OK or IDS noticeably boosted its prediction accuracy, although it remained less accurate than OK and IDS. RF [44] achieved a higher accuracy than RT and RT achieved a higher accuracy than SVM.

In a similar methodology, Kanevski *et al.* [28] combined geostatistical models and machine learning with the difference that the geographical coordinates of observations formed the only features that were inputted to the machine learning models. They showed that nonlinear regression models including support vector regression [45] with Gaussian kernel and multilayer perceptron (MLP) could capture the nonlinear global spatial trend in the response variable. Both models were trained with geographical coordinates as the only input features. However, prediction residuals were spatially autocorrelated, which means the local spatial autocorrelation was left behind. They applied sequential Gaussian simulation to capture local prediction residuals and factor them into the predictions. The combined approach resulted in a better generalization accuracy than either of geostatistical or machine learning models, applied to predict the radioactive soil contamination. In another effort to predict the radioactive soil contamination, Kanevski *et al.* [46] used spatial coordinates as the only input features to train a general regression neural network (GRNN) and a kNN. GRNN is a nonparametric regression model based on Parzen windows [47]. Two versions of GRNN were considered: one isotropic where the kernel bandwidth is the same in all directions and another anisotropic where the kernel has different bandwidths in different directions.

A matrix is used as the bandwidth instead of a constant value, when bandwidths are different in various directions in a kernel. Directions and bandwidths in their anisotropic GRNN model were optimized using leave-one-out cross-validation. Anisotropic and isotropic GRNN and kNN resulted in a root-mean-squared error (RMSE) of 11.9, 12.4, and, 22.1, respectively. This result indicates that: 1) attributing different weights to neighboring samples based on Parzen windows, which happens in GRNN, improves the accuracy in comparison with equal weights, which is the case in kNN; and 2) considering different bandwidths in different directions for the kernel, which happens in anisotropic GRNN, improves the accuracy in comparison with a single bandwidth, which is the case in isotropic GRNN.

Santibanez *et al.* [7] compared the accuracy of different machine learning techniques in regressing median rent price per zip code of a two-bedroom two-bathroom apartment in the Miami-Fort Lauderdale-West Palm Beach metropolitan area in Florida, USA, based on 23 demographic features. Location and time were not among the features. The best accuracy was achieved by MLP combined with PCA, followed by SVM with Gaussian kernel, RF, cubist, partial least squares (LS), MLP, gradient boosting machine, SVM with linear kernel, and general LS. Santibanez *et al.* [8] compared the accuracy of the same machine learning techniques with the same input features but with simulated data of varying degrees of spatial autocorrelation. SVM with Gaussian kernel resulted in the highest accuracy for weaker spatial autocorrelations, MLP with PCA achieved the same accuracy as SVM with Gaussian kernel as spatial autocorrelation was increased, and finally cubist performed best when the spatial autocorrelation was very strong.

Cracknell and Reading [48] applied five machine learning techniques, Naïve Bayes (NB), kNN, RF, SVM (using the one-against-one scheme), and MLP, in classifying lithology based on airborne geophysics (containing a digital elevation model, total magnetic intensity, and four gamma-ray spectrometry channels comprising potassium, thorium, uranium, and total count channels) and Landsat ETM + images. RF achieved the highest accuracy followed by SVM, kNN, MLP, and NB where kNN ran fastest and SVM slowest. They considered different scenarios for spatial distribution of training samples with/without considering location as an input feature. They observed that regardless of including/excluding location as an input feature, substantial higher accuracies are achieved by all machine learning techniques as training samples become more spatially dispersed across the geographic region. This is not surprising as spatial autocorrelation among responses of training samples limits proper training when training samples are not well scattered in the geographic region. Another observation was that higher generalization accuracies are achieved when location is considered as the only feature compared to the other two scenarios that either exclude location or consider it as an additional feature. Although it is plausible that considering location as an additional feature would improve the generalization accuracy, the better accuracy achieved with using location as the only feature than using it in combination with other features is surprising. This can be true if the spatial distribution of training samples is dense and well-engineered and even in that case the trained machine will not perform as well if an irresponsive sample is beyond the autocorrelation range of all training samples.

Gaussian process regression (GPR) is a nonparametric prediction model. Its functionality is similar to nonparametric Bayesian predictor with a Gaussian kernel. The difference is that it uses the gram matrix, in addition to the weights obtained for training samples using the kernel, to raise the weight of training samples that are lonelier and more isolated and lower the weight of training samples that are densely surrounded by other training samples. In GPR, the output of an irresponsive sample (y_*) is calculated using (1), where K is the gram matrix, k is the kernel function, x_* is the irresponsive sample, x_i is the ith training sample, y is the vector containing the responses, and N is the number of training samples

$$y_* = K_* K^{-1} y \tag{1}$$

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_N) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N, x_1) & k(x_N, x_2) & \cdots & k(x_N, x_N) \end{bmatrix}_{N \times N} \tag{2}$$

$$K_* = \begin{bmatrix} k(x_*, x_1) & k(x_*, x_2) & \cdots & k(x_*, x_N) \end{bmatrix}_{1 \times N}. \tag{3}$$

Flaxman [49] proposed a modified version of GPR for predicting spatial-temporal phenomena. Instead of one kernel, they used two kernels, one spatial (k_s) and one temporal (k_t). They combined the two kernels using the Kronecker product and referred to the combined kernel as the spatial-temporal kernel (k_{st}).

We assign a weight to each training sample based on the spatial and temporal semivariograms. These weights are later used in weighted machine learning techniques. This approach is superior to existing machine learning models, theoretically because it captures the spatial-temporal autocorrelation properly, and practically as our experiments with real data indicate (see Section IV).

III. Methodology

If the spatial-temporal autocorrelation is quantitatively captured, it can be used as external knowledge to enrich the training process. This external knowledge is entered in the training process as spatial-temporal weights assigned to training samples. The higher the spatial-temporal weight, the more effective the training sample is and more biased the training process must be in its favor. Section III-A focuses on developing a quantitative approach to assign a spatial-temporal weight to each training sample.

Sometimes not all training samples are equal in supervised machine learning due to their different accuracy, reliability, source, relevance, or any other reason. Nonweighted machine learning techniques are designed for equally important training samples. On the other hand, the weighted predictor is more concerned about correct prediction of training samples with larger weights than those with smaller weights. As a result, the trained model predicts in favor of training samples with larger weights.

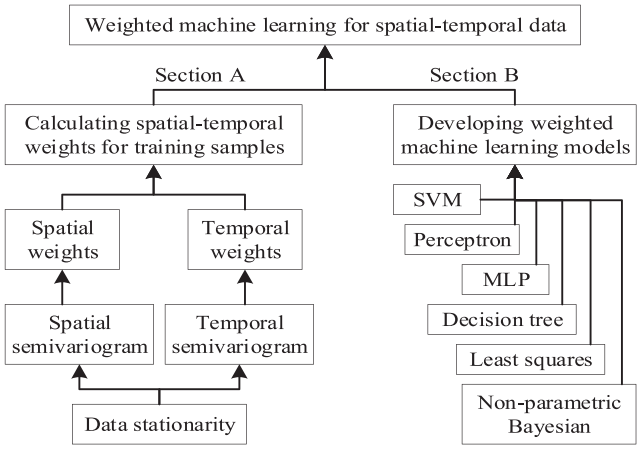


Fig. 1. Proposed methodology's framework.

This makes the weighted predictor different than its nonweighted counterpart. Section III-B discusses weighted machine learning models that have the ability to take the samples' weights into account during both training and testing.

Fig. 1 portrays the general framework of the proposed methodology, where the following sections will provide details on different parts of this framework.

A. Spatial-Temporal Weight for Training Samples

Here we focus on developing a quantitative approach to assign a spatial-temporal weight to each training sample.

Semivariogram is used as the basis in calculating both spatial and temporal semivariances and the spatial-temporal weight is proportional to the inverse of the overall semivariance at specific spatial and temporal distances. To develop the spatial and temporal semivariograms and calculate the spatial-temporal weights, we only need the location, time, and response of training samples. Feature vectors are not needed to calculate the spatial-temporal weights.

1) Spatial Semivariogram: Spatial autocorrelation or self-correlation assesses the similarity of characteristics at geographic locations relative to their spatial distance [1], [2]. In other words, a metric that relates the changes in responses to spatial distance is used. This metric will help us determine the level of similarity between the responses at two geographic locations, knowing their spatial distance.

A measure of spatial autocorrelation among training samples is semivariance (γ). Semivariance for the lag d is calculated through (4) [1]–[3], [12], [13], [27], [29], where Δ is the lag interval, n_d is the number of observation pairs with a distance (d_{ij}) between d–$\Delta/2$ and d+$\Delta/2$, and y_i and y_j are the responses of the observations i and j, respectively. The hat ($\widehat{\ }$) over the semivariance in this equation is to emphasize that the calculated value is the mean over all pairs with a distance of $d\mp\Delta/2$

$$\hat{\gamma}(d) = \frac{1}{2n_d} \sum_{d_{ij}=d-\Delta/2}^{d+\Delta/2} (y_i - y_j)^2. \tag{4}$$

Sill (c_0+c_1) is the semivariance upper bound (see Fig. 2). Partial sill (c_1) is formally defined in (5) as the limit of spatial semivariance as the spatial distance approaches infinity. This can be estimated by the variance of responses (σ^2) [1], [29]

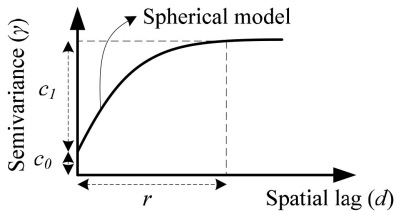


Fig. 2. Spherical semivariogram model.

$$c_1 = \lim_{d_s \to \infty} \hat{\gamma}(d_s) \approx \sigma^2. \tag{5}$$

The range (r) is the lag at which the semivariance reaches the sill and flattens out (see Fig. 2). Beyond the range, there is no particular spatial autocorrelation structure among observations [1], [2], [29]. To find the range in practice, the algorithm sweeps the calculated semivariances at ascending spatial distances, until it reaches a spatial distance where the semivariance stabilizes and shows no more systematic changes. This is formally defined as

$$r = \arg_r \begin{cases} \hat{\gamma}(d) < \hat{\gamma}(r) & \forall d < r \\ \hat{\gamma}(r) \approx \hat{\gamma}(d) & \forall d > r \end{cases} \tag{6}$$

The nugget effect (c_0) presents a discontinuity in the semivariance at the origin (see Fig. 2). In other words, it is the semivariance at zero spatial distance, as shown in (7). Practically, the semivariance calculated at the smallest spatial distance in (4) is used as an estimate of the nugget

$$c_0 \approx \hat{\gamma}(0). \tag{7}$$

Many empirical spatial semivariograms approximate to a spherical model [1], [2], [28], [31] shown in (8) and visualized in Fig. 2. The spherical model is the most frequently used model and is the default in many geographical information systems (GIS) software [1], [2], [28], [31], [50]

$$\widehat{\hat{\gamma}}(d_s) = \begin{cases} c_0 + c_1 \left[\frac{3d_s}{2r} - 0.5\left(\frac{d_s}{r}\right)^3 \right] & d_s \le r \\ c_0 + c_1 & d_s > r. \end{cases} \tag{8}$$

The second hat ($\widehat{\ }$) over the semivariance in (8) is to emphasize that the calculated value is from the fitted semivariogram model and the subscript s in d_s is to emphasize that the distance is in the space domain (not the time domain).

The spatial semivariogram model in (8), after c_0, c_1, and r are replaced with their values obtained from training samples, is used to show how strong is the correlation between each training sample and the irresponsive sample based on their spatial distance (d_s).

2) Temporal Semivariogram: The semivariogram is also used to model the autocorrelation among the responses of observations over time rather than space. Equation (4) can be used to estimate the temporal semivariance, where d refers to the temporal distance between pairs of training samples rather than their spatial distance. The shape of the temporal semivariogram might not necessarily be the same as the spatial semivariogram.

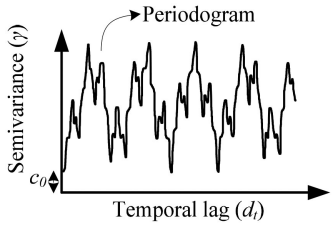


Fig. 3. Semivariance versus temporal distance.

The autocorrelation among the responses of observations is more complicated over time than space because not only temporally closer observations are more likely to have similar responses than temporally farther observations [3] but also responses might exhibit a periodic behavior [19]–[23] over time as shown in Fig. 3. For example, the temperature or weather today is more correlated with the temperature or weather yesterday than a month ago and it is more correlated with the temperature or weather a year ago than four months ago. In other words, the temporal semivariogram, shown in Fig. 3, might never level off but rather show a periodic behavior. Another important point is that the responses might have none or more than one periodic behavior with different frequencies and amplitudes, as exemplified in Fig. 3. For example, there might be weekly, monthly, and yearly cycles with different amplitudes. Therefore, the temporal semivariogram, if stationary, is approximately the result of the random superposition of periodic components oscillating at different frequencies. Correspondingly, if the responses have no cyclic behavior, the semivariogram would be a straight line. It is important to mention that the nuggets (c_0) in Figs. 2 and 3 are the same because they both refer to the semivariance at zero spatial and temporal lags ($d_s = d_t = 0$).

The sinusoid model in (9) captures the periodic behavior of the data at frequency ω where $\beta_1 = A\cos\phi$ and $\beta_2 = -A\sin\phi$, A is the amplitude, and ϕ is the phase shift, determining the start point of the cosine function [51]

$$\widehat{\widehat{\gamma}}\,(d_t) = \beta_1 \cos(2\pi\omega d_t) + \beta_2 \sin(2\pi\omega d_t)\,. \tag{9}$$

The two coefficients (β_1 and β_2) can be estimated through a linear regression via (10) and (11) [51], where *n* is the number of points in the semivariogram cloud and d_1 to d_n are different temporal distances that we calculated the semivariances for via (4)

$$\hat{\beta}_1 = \frac{\sum_{d_t=d_1}^{d_n} \hat{\gamma}\,(d_t)\cos(2\pi\omega d_t)}{\sum_{d_t=d_1}^{d_n} \cos^2(2\pi\omega d_t)} = \frac{2}{n}\sum_{d_t=d_1}^{d_n} \hat{\gamma}\,(d_t)\cos(2\pi\omega d_t) \tag{10}$$

$$\hat{\beta}_2 = \frac{\sum_{d_t=1}^{d_n} \hat{\gamma}\,(d_t)\sin(2\pi\omega d_t)}{\sum_{d_t=d_1}^{d_n} \sin^2(2\pi\omega d_t)} = \frac{2}{n}\sum_{d_t=d_1}^{d_n} \hat{\gamma}\,(d_t)\sin(2\pi\omega d_t)\,. \tag{11}$$

However, (9) captures the periodic behavior at only one frequency (ω), whereas the data might oscillate at different frequencies (ω_i) with different amplitudes ($A = \sqrt{({\beta_1}^2 + {\beta_2}^2)}$). Equation (12) shows different frequencies that need to be considered, where *n* is the number of points in the semivariogram cloud and *i* is the number of cycles through the life time of the data [51]

$$\omega_i = \frac{i}{n}, \quad i = 1,\ 2, \ldots, \left\lceil \frac{n}{2} - 1 \right\rceil. \tag{12}$$

To consider all frequencies, (13) is used to fit a periodic regression on the semivariances based on the temporal lag (d_t)

$$\widehat{\widehat{\gamma}}\,(d_t) = \sum_{i=1}^{\left\lceil \frac{n}{2}-1 \right\rceil} \beta_{i1}\cos(2\pi\omega_i d_t) + \beta_{i2}\sin(2\pi\omega_i d_t)\,,\ \omega_i = \frac{i}{n}. \tag{13}$$

The coefficients (β_{i1} and β_{i2}) in (13) are calculated through (14) and (15) for different frequencies (ω_i). We add frequencies associated with the largest amplitudes to the periodic regression in (13) one by one as long as it produces a closer fit to data points

$$\hat{\beta}_{i1} = \frac{2}{n}\sum_{d_t=d_1}^{d_n} \hat{\gamma}\,(d_t)\cos(2\pi\omega_i d_t)\,,$$
$$\omega_i = \frac{i}{n}\,, \quad i = 1,\ 2, \ldots, \left\lceil \frac{n}{2} - 1 \right\rceil \tag{14}$$
$$\hat{\beta}_{i2} = \frac{2}{n}\sum_{d_t=d_1}^{d_n} \hat{\gamma}\,(d_t)\sin(2\pi\omega_i d_t)\,,$$
$$\omega_i = \frac{i}{n}\,, \quad i = 1,\ 2, \ldots, \left\lceil \frac{n}{2} - 1 \right\rceil. \tag{15}$$

Equation (13) models the semivariance of responses based on the time interval between observations and it can be used to determine how significant the role of each training sample is when we want to predict the response of a new sample.

3) Spatial-Temporal Weight: Equation (16) calculates the spatial-temporal weight of the *i*th training sample (g_i), where $d_s(ip)$ is the spatial distance between the *i*th training sample and the irresponsive sample *p*, $d_t(ip)$ is the temporal distance between the *i*th training sample and the irresponsive sample *p*, $\widehat{\widehat{\gamma}}(d_s(ip))$ is the spatial semivariance at $d_s(ip)$ calculated from (8), $\widehat{\widehat{\gamma}}(d_t(ip))$ is the temporal semivariance at $d_t(ip)$ calculated from (13), and $\widehat{\widehat{\gamma}}(ip)$ is the overall semivariance between the *i*th training sample and the irresponsive sample *p*

$$g_i = \frac{1}{\widehat{\widehat{\gamma}}\,(ip)} \ \text{ where, } \widehat{\widehat{\gamma}}\,(ip) = \frac{\widehat{\widehat{\gamma}}\,(d_s\,(ip)) + \widehat{\widehat{\gamma}}\,(d_t\,(ip))}{2}. \tag{16}$$

The choice of average in (16) is justified as follows. To find out how relevant one training sample is to the irresponsive sample, we need to uncover their anticipated similarity. Their anticipated similarity is obtained by inversing their anticipated dissimilarity. The anticipated dissimilarity is measured through semivariance. The spatial semivariance ($\widehat{\widehat{\gamma}}(d_s)$) tells us how dissimilar these two points are by knowing their spatial distance. The temporal semivariance ($\widehat{\widehat{\gamma}}(d_t)$) tells us how dissimilar these two points are by knowing their temporal distance. Despite the spatial semivariogram uses the spatial distance to produce the dissimilarity value and the temporal semivariogram uses the temporal distance to produce the dissimilarity value, the two values are made of the

same fabric, i.e., they are both semivariances ($\widehat{\gamma}$), which justifies the choice of their average as the spatial-temporal semivariance.

4) Data Constraints:

a) Fixed location or time: When calculating the spatial semivariance, the time must be the same for each pair of y_i and y_j in (4). The same time does not mean the exact same instant and depends on the dataset. For example, if one set of observations are observed in one day (e.g., a series of satellite images taken on January 1, 2001) and another set are observed in another day (e.g., another set of satellite images for the same area taken on February 1, 2002), then the same time means observed on the same day. On the other hand, when calculating the temporal semivariance, the location must be the same for each pair of y_i and y_j in (4). Again, the same location does not necessarily mean the exact same coordinates (*x*,*y*) but depends on the nature of the dataset. For example, if each observation belongs to a separate tree or neighborhood (e.g., the height of the tree or the population of the neighborhood), then the same location means the same tree or the same neighborhood. For a satellite image, the same location means, e.g., the same pixel in the image.

b) Categorical responses: As mentioned before, y_i in (4) is the response of the observation *i*. If the responses are already quantitative (interval- or ratio-scaled), they are used for *y*. Qualitative (or categorical) responses are either ordinal or nominal [52]. If the responses are ordinal such as the agricultural potential of different lands or purity of different water bodies (e.g., good, average, and bad), they can be defined with a quantitative scale (e.g., 1, 2, and 3) somehow that the interval between scales approximates the implicit interval between levels (although ordinal data do not suggest any quantitative interval between levels). Although converting ordinal variables to interval variables in this way is not precise, it is legitimate here as weighted machine learning techniques are not much sensitive to small changes in training samples' weights. If the responses are nominal, where the responses cannot be ordered (e.g., different landuses or building uses: shop, residential, etc.), defining responses with a quantitative scale, incorrectly implies that some categories are closer to each other than others. In this case, we consider y_i–$y_j = 0$ if samples *i* and *j* have the same response and 1 (or any desired constant value) otherwise [53], [54]. An alternative approach is to code responses via dummy variables [34]. In this approach, responses are represented using vectors so as the distance between categories remains constant. The number of elements in the vector is equal to the number of categories. For each category, one element of the vector is one and the rest are zero (e.g., [1,0,0], [0,1,0], [0,0,1]). Hamming distance, L^1, or L^2 norms can then be used to calculate the distance between response vectors. This is a common approach to code nominal variables.

c) Stationarity of data: Fitting the periodogram to the temporal semivariances is only meaningful if the data are temporally stationary [51]. A stochastic process is strictly stationary if the joint statistical distribution of $x_{t_1}, \ldots, x_{t_l}$ is the same as the joint statistical distribution of $x_{t_1+\tau}, \ldots, x_{t_l+\tau}$ for all l and τ [55], where *t* represents the time. This means that statistical properties of all degrees (expectations, variances, third order, and higher) of the process, anywhere are the same. Since, strict stationarity is too unrealistic for real-world processes, weak or second-order stationarity is defined as a process whose mean and variance do not vary with time and the autocovariance between x_t and $x_{t+\tau}$ (shown as $\mathrm{cov}(x_t, x_{t+\tau})$) only depends on the lag τ [55]. We attempt to transform the data closer to a weakly stationary one by first stabilizing the variance and then stabilizing the mean. To stabilize the temporal variance, *y* is replaced by log(*y*) for training samples. If there are negative values among responses, we can add a constant value to make them all positive and then take the log. This constant value will be removed in the next step. To stabilize the temporal mean, after stabilizing the temporal variance, a line, called the trend line, is fitted to all $\log(y_i)$ based on time. Then, the value on the trend line is subtracted from $\log(y_i)$. Temporal semivariances are calculated based on these residuals. Fig. 4 summarizes these steps.

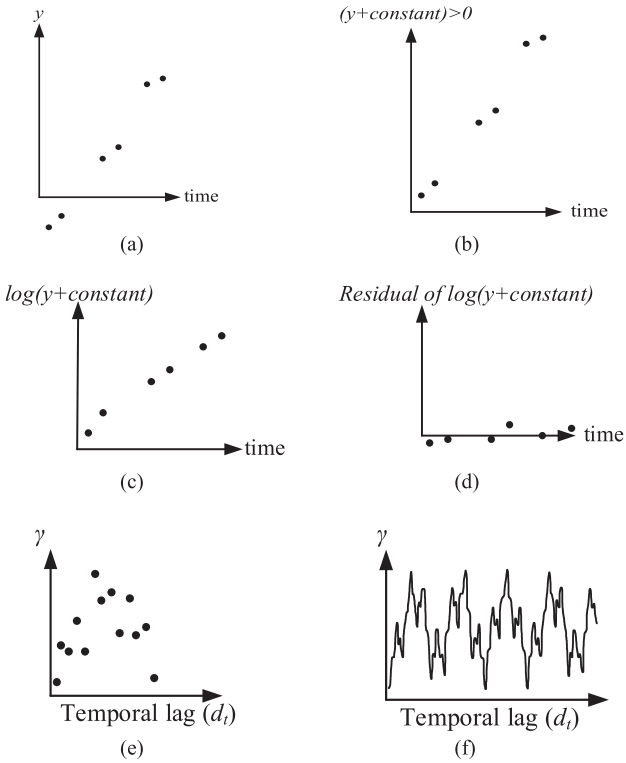


Fig. 4. Stabilizing the temporal variance and mean before developing the temporal semivariogram. (a) Original responses (*y*) over time. (b) Adding a constant value to make all responses positive if necessary. (c) Taking the log to stabilize the variance. (d) Subtracting the trend line. (e) Calculating the semivariances based on residuals. (f) Fitting the periodic semivariogram.

Fitting the spherical semivariogram to the spatial semivariances is only meaningful if the data are spatially stationary [28], [29]. If the spatial mean is not stable or, in other words, if there is a trend among the responses of training samples over space, the spatial semivariances will show an exponential behavior over lags and never flatten out. On the other hand, if the spatial variance is not stable or, in other words, if the range of changes in responses varies dramatically over space, the spatial semivariances will be dramatically scattered around the spherical model. Again, because real-world processes are far from being strictly stationary, we resort to weak stationarity.

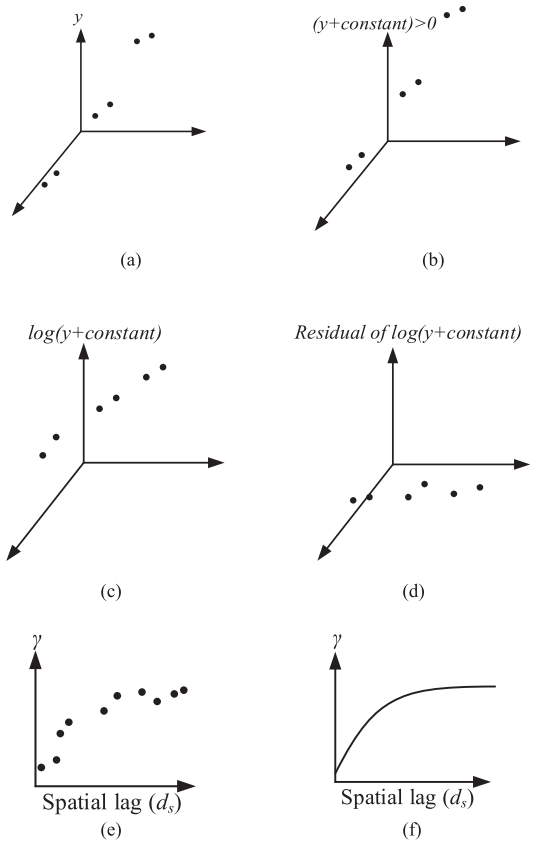


Fig. 5. Stabilizing the spatial variance and mean before developing the spatial semivariogram. (a) Original responses (y) over location. (b) Adding a constant value to make all responses positive if necessary. (c) Taking the log to stabilize the variance. (d) Subtracting the trend plane. (e) Calculating the semivariances based on residuals. (f) Fitting the spherical semivariogram.

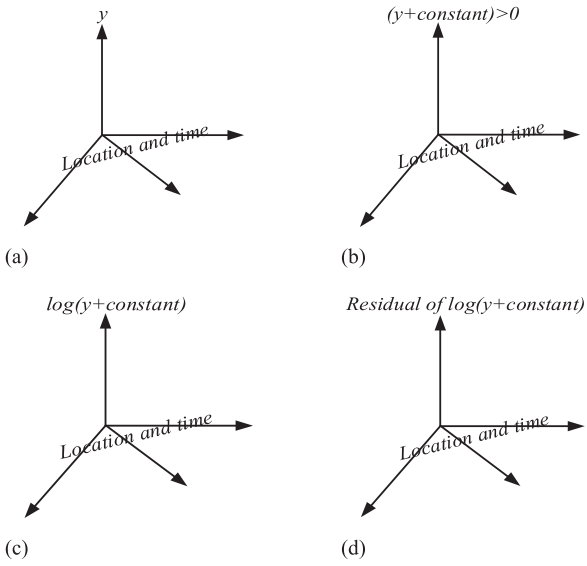


Fig. 6. Stabilizing the variance and mean before developing the spatial and temporal semivariogram. (a) Original responses (y) over location and time. (b) Adding a constant value to make all responses positive if necessary. (c) Taking the log to stabilize the variance. (d) Subtracting the trend plane.

Thus, before calculating the spatial semivariances and fitting the spherical semivariogram model, the spatial variance and mean must be stabilized. The steps are very similar to temporal stabilization and are illustrated in Fig. 5. It is noteworthy that because space has two dimensions, in comparison to time with one dimension, here we have a trend plane instead of a trend line.

If our dataset is only spatial or temporal, we use either of the previous sections to stabilize the dataset. However, if our dataset is spatial and temporal, we do not need to transform the dataset twice, once over location and once over time. Instead, the two processes are combined as shown in Fig. 6. First, we add a constant value to make all responses positive. Then we take the log to stabilize the variance. To stabilize the mean, a 3-D hyperplane is regressed over all responses based on both location and time and subtracted from all responses. These residuals are both spatially and temporally stabilized and we can proceed with developing semivariograms.

B. Weighted Machine Learning

Weighted LS [56] is one of the earliest examples of a weighted machine learning technique. Hashemi and Karimi [57] developed the weighted versions of Bayesian predictor, perceptron, MLP, SVM, and decision tree. Table I lists the weighted and nonweighted versions of the aforementioned machine learning techniques, which are elaborated in [57]. We employ both weighted and nonweighted versions of these machine learning techniques in our experiments. In Table I, M is the number of classes, ω indicates a specific class, N represents the number of samples, l is the number of features, x represents a sample's feature vector, the subscript k in x_k refers to the kth feature in the corresponding feature vector, X is the $N \times l$ input feature matrix, Σ is the covariance matrix of features, y is the training samples' responses, w is a column vector representing the norm of the classifier hyperplane, w_0 is the intercept or threshold of the classifier hyperplane, g represents the training samples' weights, G indicates the $N \times N$ diagonal matrix of samples' weights, $g(\omega)$ is the sum of the weight of training samples belonging to class ω, and K is the kernel function for the Bayesian predictor. C, ξ_i, λ_i, and μ_i are the smoothing parameter, the slack variable, and Lagrangian multipliers for SVM. w_j^r is the weight vector of the jth node in the rth layer in MLP and α is its training rate. N_A is the number of training samples at the ancestor node, N_Y and N_N are the number of training samples in the descendant nodes, ΔI is the impurity decrease, I_A is the impurity of the ancestor node, I_Y and I_N are the descendent nodes' impurities in the decision tree.

IV. Experiments

The MATLAB software on a 64-b platform with 8 GB RAM, a Core i7 CPU, and a 2.00 GHz processor was used for the validation of the proposed techniques. Two applications are presented, regression of air temperature based on meteorological features and classification of land cover based on morphological and remote sensing features.

TABLE I
WEIGHTED AND NONWEIGHTED MACHINE LEARNING TECHNIQUES

	Non-weighted version	Weighted version
Least squares	$w = (X^T X)^{-1} X^T y$	$w = (X^T G X)^{-1} X^T G y$
Non-parametric Bayesian classifier	$p(x\|\omega_j) = \frac{1}{N(\omega_j)} \sum_{\forall i \| x_i \in \omega_j} K(x - x_i, \Sigma_j)$ $K(x - x_i, \Sigma_j) = \begin{cases} \frac{1}{\|\Sigma_j/\sigma\|^{1/2}} & \|x_k - x_{i_k}\| < \frac{1}{2}\|\Sigma_j/\sigma\|^{\frac{1}{2l}} \\ 0 & otherwise \end{cases}$	$p(x\|\omega_j) = \frac{1}{N(\omega_j)} \sum_{\forall i \| x_i \in \omega_j} g_i K(x - x_i, \Sigma_j)$ $K(x - x_i, \Sigma_j) = \begin{cases} \frac{1}{\|\Sigma_j/\sigma\|^{1/2}} & \|x_k - x_{i_k}\| < \frac{1}{2}\|\Sigma_j/\sigma\|^{\frac{1}{2l}} \\ 0 & otherwise \end{cases}$
Non-parametric Bayesian regressor	$y(x) = \frac{\sum_{i=1}^N y_i K(x - x_i, \Sigma)}{\sum_{i=1}^N K(x - x_i, \Sigma)}$	$y(x) = \frac{\sum_{i=1}^N y_i g_i K(x - x_i, \Sigma)}{\sum_{i=1}^N g_i K(x - x_i, \Sigma)}$
SVM	$y_i = \begin{cases} +1 & \forall x_i \in \omega_1 \\ -1 & \forall x_i \in \omega_2 \end{cases}$ $\begin{cases} \max_\lambda \left(\sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j x_i^T x_j \right) \\ subject\ to \sum_{i=1}^N \lambda_i y_i = 0 \\ 0 \le \lambda_i \le C\ (due\ to\ Equation\ 3.20)\ , i = 1,2,\dots,N \end{cases}$ $\begin{cases} \frac{\partial \mathcal{L}(w, w_0, \xi, \lambda, \mu)}{\partial w} = 0 \to w = \sum_{i=1}^N \lambda_i y_i x_i \\ \frac{\partial \mathcal{L}(w, w_0, \xi, \lambda, \mu)}{\partial w_0} = 0 \to \sum_{i=1}^N \lambda_i y_i = 0 \\ \frac{\partial \mathcal{L}(w, w_0, \xi, \lambda, \mu)}{\partial \xi_i} = 0 \to C - \mu_i - \lambda_i = 0 \ , \ i = 1,2,\dots,N \\ \mu_i \xi_i = 0 \ , \ i = 1,2,\dots,N \\ \mu_i \ge 0 \ , \ \lambda_i \ge 0 \ , \ i = 1,2,\dots,N \\ \lambda_i [y_i (w^T x_i + w_0) - 1 + \xi_i] = 0 \ , i = 1,2,\dots,N \\ (complementary\ slackness\ conditions) \end{cases}$	*w and w_0 are initialized for a non-weighted SVM,* $\hat{X}_t = X - \left(1 - \frac{1}{1+g}\right) \cdot \left(\frac{\|X w_{t-1} + w_{0_{t-1}}\|}{\|w_{t-1}\|} \right) \cdot y \frac{w_{t-1}^T}{\|w_{t-1}\|}$ *find $\hat{w}$ and $\hat{w}_0$ for the non-weighted SVM classifier hyperplane based on $\hat{X}_t$*
Perceptron	$J(w) = \sum_{i=1}^N y_i w^T x_i$ $y_i = \begin{cases} +1 & if\ wx_i > 0\ but\ x_i \in \omega_2 \\ -1 & if\ wx_i < 0\ but\ x_i \in \omega_1 \\ 0 & if\ wx_i > 0\ and\ x_i \in \omega_1 \\ 0 & if\ wx_i < 0\ and\ x_i \in \omega_2 \end{cases}$ $w_{t+1} = w_t - \alpha \sum_{i=1}^N y_i x_i$	$J(w) = \sum_{i=1}^N g_i y_i w^T x_i$ $y_i = \begin{cases} +1 & if\ wx_i > 0\ but\ x_i \in \omega_2 \\ -1 & if\ wx_i < 0\ but\ x_i \in \omega_1 \\ 0 & if\ wx_i > 0\ and\ x_i \in \omega_1 \\ 0 & if\ wx_i < 0\ and\ x_i \in \omega_2 \end{cases}$ $w_{t+1} = w_t - \alpha \sum_{i=1}^N g_i y_i x_i$
MLP	$J(w) = \sum_{i=1}^N \varepsilon(i)$ $w_j^r(new) = w_j^r(old) + \Delta w_j^r$ $\Delta w_j^r = -\alpha \sum_{i=1}^N \frac{\partial \varepsilon(i)}{\partial w_j^r}$	$J(w) = \sum_{i=1}^N g_i \varepsilon(i)$ $w_j^r(new) = w_j^r(old) + \Delta w_j^r$ $\Delta w_j^r = -\alpha \sum_{i=1}^N g_i \frac{\partial \varepsilon(i)}{\partial w_j^r}$
Decision tree classifier	$\Delta I = I_A - \frac{N_Y}{N_A} I_Y - \frac{N_N}{N_A} I_N$ $I_{classification} = -\sum_{i=1}^M \frac{N(\omega_i)}{N} log_2 \frac{N(\omega_i)}{N}$ The majority rule is commonly used to determine the response at a leaf	$\Delta I = I_A - \frac{\sum g_Y}{\sum g_A} I_Y - \frac{\sum g_N}{\sum g_A} I_N$ $I_{classification} = -\sum_{i=1}^M \frac{g(\omega_i)}{\sum g} \log_2 \frac{g(\omega_i)}{\sum g}$ The class with the largest total weight ($\underset{\omega_j}{argmax} \sum_{i \in \omega_j} g_i$) is associated with a leaf
Decision tree regressor	$\Delta I = I_A - \frac{N_Y}{N_A} I_Y - \frac{N_N}{N_A} I_N$ $I_{regression} = \frac{\sum_{i=1}^N (y_i - \bar{y})^2}{N}$ The average rule is commonly used to determine the response at a leaf	$\Delta I = I_A - \frac{\sum g_Y}{\sum g_A} I_Y - \frac{\sum g_N}{\sum g_A} I_N$ $I_{regression} = \frac{\sum_{i=1}^N g_i (y_i - \bar{y})^2}{\sum_{i=1}^N g_i}$ The weighted average of the responses ($\sum_{i \in leaf} g_i y_i / \sum_{i \in leaf} g_i$) is associated with a leaf

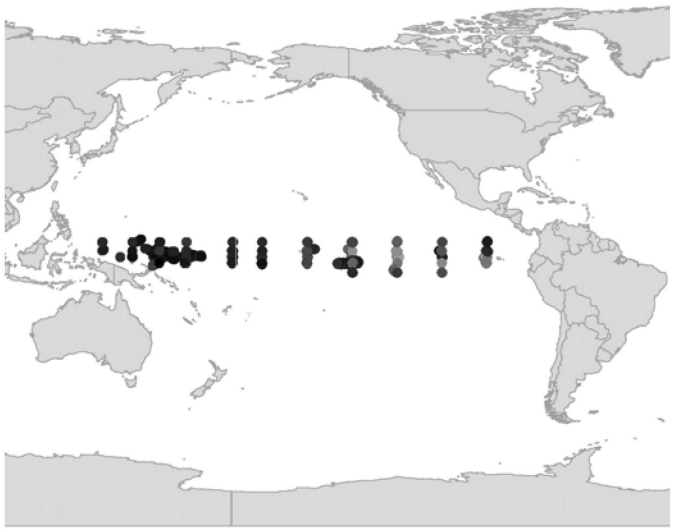


Fig. 7. Geographical location of samples, shaded based on their observed temperature, where the lighter points indicate lower temperatures.

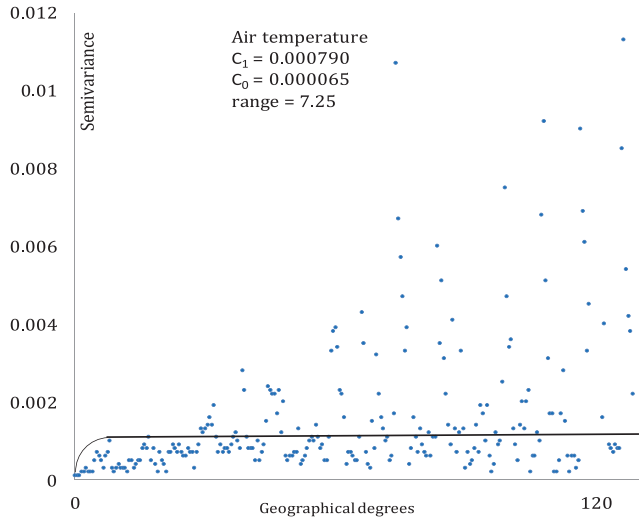


Fig. 8. Spatial semivariogram for temperature.

TABLE II
CORRELATION COEFFICIENT BETWEEN DIFFERENT VARIABLES

	Zonal winds	Meridional winds	Humidity	Temperature
Zonal winds	1.00	0.08	0.06	0.23
Meridional winds	0.08	1.00	0.08	-0.34
Humidity	0.06	0.08	1.00	-0.39
Temperature	0.23	-0.34	-0.39	1.00

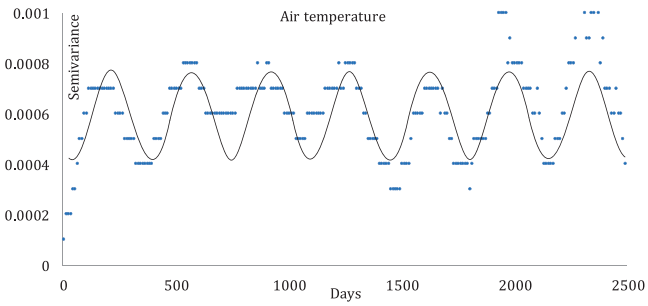


Fig. 9. Temporal semivariogram for temperature with a period of 1 year.

A. *Regression of Air Temperature*

Oceanographic and surface meteorological readings, taken from a series of buoys positioned throughout the equatorial Pacific, from 1980 to 1998, are available to public in [58]. Each reading includes location, date, zonal winds, meridional winds, humidity, and temperature. The dataset has 178 000 records. Removing records with missing features leaves the dataset with 94 000 records. In this experiment, we will predict the air temperature based on zonal winds, meridional winds, and humidity. Fig. 7 visualizes the geographical location of samples, shaded based on their observed temperature, where the lighter points indicate lower temperatures. Table II lists the correlation coefficient between different variables in this dataset.

According to Table II, air temperature is fairly correlated with all other variables, whereas other variables are not much correlated with each other. The spatial and temporal semivariances for the response variable (air temperature) and the spatial and temporal semivariograms fitted to them are shown in Figs. 8 and 9. These figures are produced using the proposed approach in Section III-A. As required in Section III-A-4a, we consider two observations to be at the same time as long as their time difference is less than one day and we consider two observations to be at the same location as long as their distance is less than 10 m.

Spatial and temporal semivariograms in Figs. 8 and 9 can be used to determine the spatial-temporal weight of training samples. Knowing the spatial distance of a training sample to the irresponsive sample, we can use the spatial semivariogram in Fig. 2 [represented by (8)] to calculate its spatial semivariance. Knowing the temporal distance of a training sample to the irresponsive sample, we can use the temporal semivariogram in Fig. 3 [represented by (13)] to calculate its temporal semivariance. Knowing both spatial and temporal semivariances for that training sample, we can use (16) to calculate its spatial-temporal weight.

We use the leave-one-out or N-fold cross-validation to evaluate the performance of weighted regression versus nonweighted regression. RMSE and coefficient of determination (R^2) are reported in Table III for each regressor. The time performance includes the time spent to calculate the spatial-temporal weights in addition to the training and test time. All input features are normalized to have a zero mean and unit variance. Table III lists the accuracy and experimental time performance of different weighted regressors. Estimating the response as the weighted average (i.e., spatial-temporal weights) of training samples' responses, the simplest regression model, is also considered in Table III, for comparison purposes. We also used only the top 30%, 10%, and 1% of training samples with largest spatial-temporal weights for training, to investigate how it will affect

TABLE III
ACCURACY AND TIME PERFORMANCE OF DIFFERENT REGRESSION TECHNIQUES FOR PREDICTING THE AIR TEMPERATURE

Technique	RMSE	R^2	Time (min)	Settings
Weighted average of training samples' responses	1.65	0.03	82	•Spatial-temporal weights of training samples are used as weights
Weighted average of the top 30% of training samples' responses	1.53	0.16	89	
Weighted average of the top 10% of training samples' responses	1.31	0.39	89	
Weighted average of the top 1% of training samples' responses	**1.12**	**0.55**	**89**	
LS with location and time as the only features	1.62	0.07	20	
LS without location and time as additional features	1.37	0.33	19	
LS with location and time as additional features	1.35	0.35	36	
Weighted LS	1.35	0.35	134	
Weighted LS using only the top 30% of training samples	1.26	0.43	123	
Weighted LS using only the top 10% of training samples	1.13	0.54	110	
Weighted LS using only the top 1% of training samples	**0.96**	**0.67**	**105**	
Bayesian regressor with location and time as the only features	1.40	0.30	4150	•Non-parametric Parzen windows with Gaussian kernel (see Table 1) where σ=3
Bayesian regressor without location and time as additional features	1.31	0.39	4180	
Bayesian regressor with location and time as additional features	1.07	0.59	4522	
Weighted Bayesian regressor	1.27	0.42	4956	
Weighted Bayesian regressor using only the top 30% of training samples	1.16	0.52	1149	
Weighted Bayesian regressor using only the top 10% of training samples	1.04	0.62	579	
Weighted Bayesian regressor using only the top 1% of training samples	**0.92**	**0.70**	**154**	
Weighted decision tree using only the top 10% of training samples	1.11	0.56	41395	•A node is considered a leaf if ΔI_{max} is less than 0.2

the accuracy. The resulted accuracy and time performance are reported in Table III.

As discussed in Sections I and II, two previous common approaches in machine learning for spatial-temporal data are ignoring location and time altogether and considering them as additional features. Ignoring location and time means applying nonweighted regressors trained with nonspatial features (zonal winds, meridional winds, and relative humidity). Considering location and time as additional features means applying nonweighted regressors trained with all features including location and time. Table III reports the accuracy and time performance of both these approaches for comparison purposes.

Spatial and spatial-temporal regression are common tools in geographical studies and software, where different machine learning models are deployed whose only inputs are location and time (ignoring all other features). For example, kriging (also known as GPR) in GIS is similar to nonparametric Bayesian regressor with a Gaussian kernel, where location and time (or location alone) are the only input features. For comparison purposes, Table III also reports the results of such regression models, which are equivalent to nonweighted regressors trained with location and time as the only features.

One apparent irregularity in Table III is that for the regressor that simply estimates the irresponsive sample's response as the weighted average of training samples' responses, the performance time does not reduce when training samples with small weights are excluded. It is because if all training samples participate in taking the weighted average, there will be no need to sort the weights of training samples but if one decides to exclude the training samples with very small weights, the weight vector needs to be sorted. The sorting function, which is invoked only if a subset of training samples needs to be deployed, has a greater time complexity than the function that takes the weighted average of training samples' responses.

Nonparametric regressors, in our case the Bayesian regressor with Parzen windows, need to be trained separately for each irresponsive sample, regardless of the regressor being nonweighted or weighted. For parametric regressors, the weighted version needs to be trained separately for each irresponsive sample but the nonweighted version needs to be trained only once for all irresponsive samples. In other words, the weighted regressors need to be trained as many times as the number of irresponsive samples because the spatial-temporal weights for training samples depend upon the location and time of the irresponsive sample. However, the leave-one-out approach for evaluation eliminates this time performance difference between weighted and nonweighted regressors, because in leave-one-out evaluation approach, each time there is only one test sample and consequently both weighted and nonweighted regressors are trained equal number of times.

The weighted and nonweighted nonparametric Bayesian regressors have almost identical performance times with only 10% difference. The weighted LS takes 3.7 times longer than nonweighted LS. Among the same versions of four different regressors (weighted average, LS, nonparametric Bayes, and decision tree) in Table III, the weighted average is the fastest approach, followed by LS, nonparametric Bayesian regressor, and decision tree. Cross-validation of the weighted decision tree, trained with only 10% of training samples, took 29 days. Cross-validation of a decision tree, trained with 30% of training samples (or more), takes months and the results are not reported in Table III.

Despite its simplicity and speed in comparison with other more complicated techniques, the regressor that estimates the irresponsive sample's output as the weighted average of training samples' responses achieves an accuracy close to that of more complicated techniques. Considering that this technique only needs location and time (to calculate the spatial-temporal

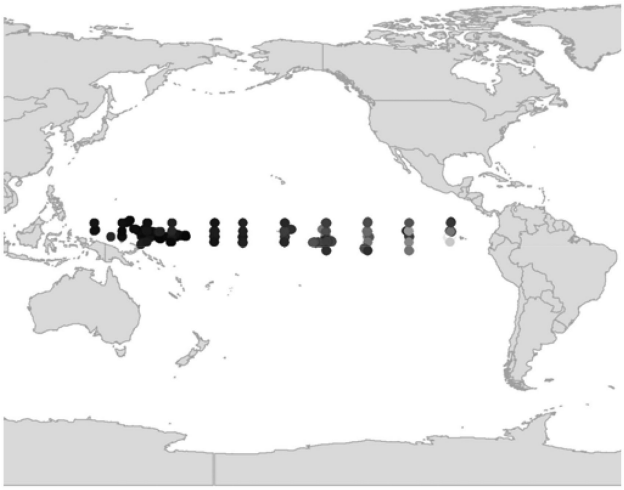


Fig. 10. Geographical location of samples, shaded based on their predicted temperature by weighted Bayesian regressor using only the top 1% of training samples, where the lighter points indicate lower temperatures.

TABLE IV
DIFFERENT LAND COVERS AND THEIR RELATIVE FREQUENCY

Class	dry sclerophyl	E. botryoides	lower slope wet	wet E. maculata	dry E. maculata	rainforest ecotone	rainforest	cleared land
Relative Frequency	0.26	0.04	0.04	0.22	0.16	0.08	0.07	0.13

weights), its high accuracy underscores the efficiency of spatial-temporal weights in depicting the spatial and temporal autocorrelation between training samples and the irresponsive sample.

Among the same versions of four different regressors (weighted average, LS, nonparametric Bayes, and decision tree) in Table III, the nonparametric Bayesian regressor achieves the highest accuracy, followed by LS, decision tree, and weighted average. However, weighted decision tree seems to precede weighted LS, in terms of accuracy, when more training samples are included.

In the following, we compare four general strategies in dealing with location and time in machine learning, where item (a) is the methodology known as spatial-temporal regression in GIS, items (b) and (c) are previously used methodologies in the literature on machine learning for spatial-temporal data, and item (d) is our proposed methodology.

a) Regressors trained using location and time as the only features.
b) Regressors that ignore the location and time altogether.
c) Regressors that take account of location and time as additional input features.
d) Regressors that take account of location and time as weights for training samples.

Rows with a bold font in Table III show the settings leading to the best accuracy for each of the four regression techniques (weighted average, LS, nonparametric Bayes, and decision tree). In all cases, the regressor in group (a) results in the worst accuracy. This highlights the importance of nonspatial/temporal features in proper training. On the other hand, in all cases, the regressor in group (b) results in a lower accuracy than regressors that somehow take account of location and time, i.e., groups

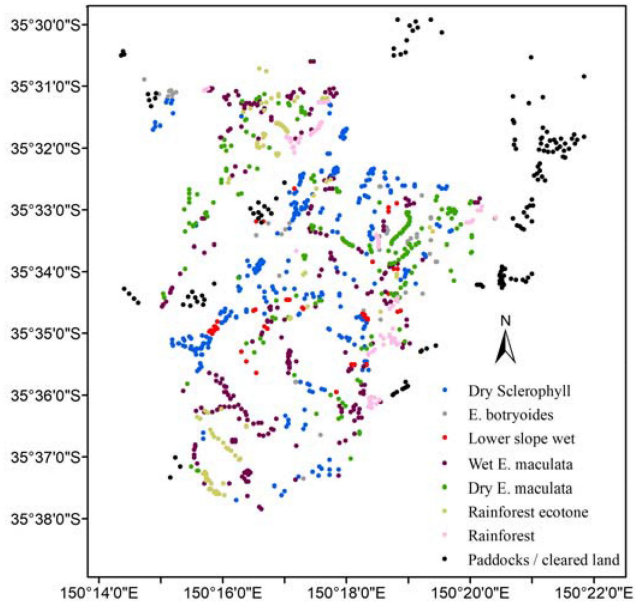


Fig. 11. Geographical location of samples, colored based on their land cover.

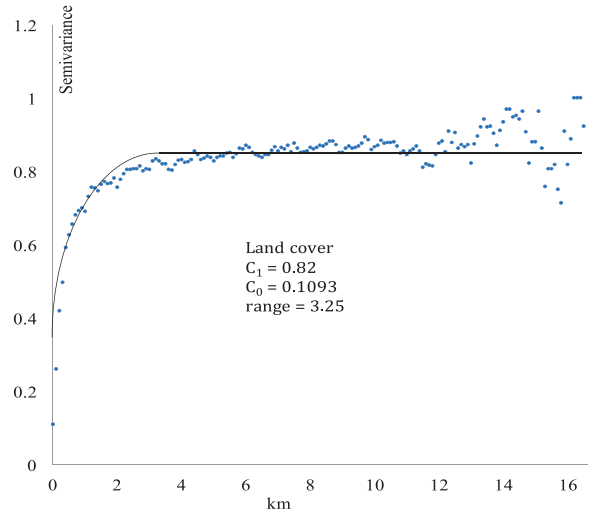


Fig. 12. Spatial semivariogram for land covers.

(c) and (d). This highlights the importance of location and time in proper training. The difference between the accuracy of regressors in groups (c) and (d) is very small in all cases. However, when only a subset of training samples with largest spatial-temporal weights are applied to train the regressors in group (d), the accuracy is considerably improved, revealing the best accuracy and performance time. This also uncovers the fact that training samples with very small spatial-temporal weights are not much constructive in training the regressor. This is in line with our expectations that weighted regression captures the spatial-temporal autocorrelation best and excluding training samples with very small spatial-temporal weights not only reduces the performance time but also improves the accuracy.

Fig. 10 visualizes the geographical location of samples, shaded based on their predicted temperature by weighted

TABLE V
ACCURACY AND TIME PERFORMANCE OF DIFFERENT CLASSIFICATION TECHNIQUES FOR PREDICTING THE LAND COVER

Technique	Over all Acc.	Time (sec)	Settings
Gaussian Process Regression			•Gaussian Kernel
Choosing the class with largest collective weight among training samples	34.61	1	•Spatial-temporal weights of training samples are used as weights
Choosing the class with largest collective weight among the top 30% of training samples	31.85	1	
Choosing the class with largest collective weight among the top 10% of training samples	57.09	1	
Choosing the class with largest collective weight among the top 1% of training samples	**72.61**	**1**	
LS with location as the only feature	36.04	4	•Using the one-against-one scheme since there are more than two classes
LS without location as additional feature	14.99	12	
LS with location as additional feature	16.68	12	
Weighted LS	**52.72**	**33**	
Weighted LS using only the top 30% of training samples	42.64	13	
Weighted LS using only the top 10% of training samples	31.13	8	
Weighted LS using only the top 1% of training samples	22.03	2	
Bayesian with location as the only feature	**65.30**	**45**	•Non-parametric Parzen windows with Gaussian kernel (see Table 1) where σ=85 •Priors are based on class frequencies
Bayesian without location as additional feature	14.09	293	
Bayesian with location as additional feature	11.41	304	
Weighted Bayesian	14.09	293	
Weighted Bayesian using only the top 30% of training samples	16.77	86	
Weighted Bayesian using only the top 10% of training samples	24.44	28	
Weighted Bayesian using only the top 1% of training samples	25.78	4	
Decision tree with location as the only feature	**74.22**	**2955**	•A node is considered a leaf if the maximum impurity decrease (ΔI_{max}) for that node is less than 0.1.
Decision tree without location as additional feature	37.29	4399	
Decision tree with location as additional feature	67.35	15435	
Weighted decision tree	51.47	7317	
Weighted decision tree using only the top 30% of training samples	49.33	2620	
Weighted decision tree using only the top 10% of training samples	32.83	666	
Weighted decision tree using only the top 1% of training samples	22.75	20	
SVM with location as the only feature	19.98	2907	•Smoothing parameter (C)=5 •Using the one-against-one scheme since there are more than two classes
SVM without location as additional feature	28.99	14925	
SVM with location as additional feature	30.95	18174	
Weighted SVM using only the top 30% of training samples	**39.52**	**1184961**	
Weighted SVM using only the top 10% of training samples	29.97	297564	
Weighted SVM using only the top 1% of training samples	23.64	911	
Perceptron with location as the only feature	37.56	1777	•Logistic activation function •Cost function: Sum of squared errors •Maximum number of iterations: 1000 for perceptron and 3000 for MLP •Not updating the weights after those iterations resulting in an increase in the total cost •Multiply all learning rates by 1.1 or 0.8 after each step based on whether the total cost decreases or increases •Adaptive learning rate: multiply the learning rate for a parameter by 1.2 if the partial derivative of the loss, with respect to that parameter, remains the same sign in successive steps and multiply it by 0.7 otherwise [65] •Number of hidden nodes for MLP: 3
Perceptron without location as additional feature	45.85	16392	
Perceptron with location as additional feature	47.90	17291	
Weighted perceptron	43.35	24786	
Weighted perceptron using only the top 30% of training samples	**54.95**	**12472**	
Weighted perceptron using only the top 10% of training samples	36.66	3397	
Weighted perceptron using only the top 1% of training samples	22.57	365	
MLP with location as the only feature	45.41	35219	
MLP without location as additional feature	44.87	37224	
MLP with location as additional feature	48.26	41811	
Weighted MLP	**57.72**	**127349**	
Weighted MLP using only the top 30% of training samples	51.03	34972	
Weighted MLP using only the top 10% of training samples	35.95	13613	
Weighted MLP using only the top 1% of training samples	23.46	1483	

Bayesian regressor using only the top 1% of training samples, where the lighter points indicate lower temperatures.

B. Classification of Land Cover

This is a classification problem with eight input features and eight classes. In this application, we use bands 2, 3, 5, and 7 of Landsat TM images, two categorical features including geology—with six categories: 1) Quaternary Alluvium, 2) Termeil Essexite Permian, 3) Snapper Point Permian, 4) Pebbly Beach Permian, 5) Sedimentary Permian, and 6) Ordovician—and aspect—with five categories: east, north, west, south, and no aspect indicating zero slope—and two quantitative hydrological features including flow accumulation and flow length to predict the land covers shown in Table IV.

Geology and aspect are coded in dummy variables [34]. All features are in 30×30 m grid format and the land covers represent the dominant vegetation cover by field observation for 1121 sites in Kioloa, NSW, Australia. These sites are not contiguous regions but are instead isolated samples, as represented in Fig. 11. The TM images are from November 8, 1994 and other layers

TABLE VI
SAME VERSION OF DIFFERENT CLASSIFIERS RANKED BASED ON THEIR ACCURACY

Classifier with location as the only feature	Classifier without location as additional feature	Classifier with location as additional feature	Weighted classifier	Weighted classifier using only the top 30% of training samples	Weighted classifier using only the top 10% of training samples	Weighted classifier using only the top 1% of training samples
Decision tree	Perceptron	Decision tree	MLP	Perceptron	Collective weight	Collective weight
Bayes	MLP	MLP	LS	MLP	Perceptron	Bayes
MLP	Decision tree	Perceptron	Decision tree	Decision tree	MLP	SVM
Perceptron	SVM	SVM	Perceptron	LS	Decision tree	MLP
LS	LS	LS	Collective weight	SVM	LS	Decision tree
SVM	Bayes	Bayes	Bayes	Collective weight	SVM	Perceptron
				Bayes	Bayes	LS

Accuracy

are from 1992 [36], [59]–[63] and the entire dataset is publicly available in [64].

This dataset is available only for one point in time. Therefore, the spatial weight is used instead of spatial-temporal weight during training. Fig. 12 shows the spatial semivariances for land covers and the spatial semivariogram fitted to them. This figure is produced using the proposed approach in Section III-A.

Table V lists the overall accuracy and experimental time performance of different weighted and nonweighted classifiers based on leave-one-out or *N*-fold cross-validation. The time performance includes the time spent to calculate the spatial weights in addition to the training and test time.

One apparent irregularity in Table V, similar to Table III, is that for the classifier that simply assigns the class with the largest collective spatial weight among training samples to the irresponsive sample, the performance time does not reduce when training samples with small weights are excluded. The reason is the same; there is no need to sort the weights of training samples when they all participate in finding the class with the largest collective weight.

The weighted and nonweighted nonparametric Bayesian classifiers have almost identical performance times with only 4% difference. The weighted SVM takes 65 times longer to be trained than nonweighted SVM in Table V. The reason is that finding the weighted SVM classifier includes an additional step where the original training samples are shifted with respect to the nonweighted SVM classifier proportional to their weight and the nonweighted SVM classifier is recalculated for the shifted training samples. The weighted MLP takes three times longer than nonweighted MLP and the weighted Perceptron takes 1.4 times longer than nonweighted Perceptron. Upon detailed analysis of the results, it became clear that the longer training time for weighted MLP and weighted Perceptron is due to the presence of training samples' weights in computing the synaptic weight correction term. Weighted LS takes 2.8 times longer than nonweighted LS and weighted decision tree halves the performance time in comparison with nonweighted decision tree, which is because the former is developed in a feature space with one less dimension (i.e., location) than the latter and dimensionality plays a crucial role in both training and height of decision trees.

Among the same versions of seven different classifiers in Table V, the collective weight is the fastest approach, followed by LS, nonparametric Bayesian classifier, decision tree, perceptron, SVM, and MLP in most cases. Weighted SVM is the slowest approach. Cross-validation of the weighted SVM, trained with only 30% of training samples, took 14 days. Cross-validation of the weighted SVM, trained with all training samples, takes months and the results are not reported in Table V.

Despite its simplicity and speed, the classifier that assigns the irresponsive sample to the class with the largest collective weight among the top 1% of training samples achieves an accuracy that is surpassed only slightly (1.61%) by decision tree with location as the only feature. Considering that this technique only needs location (to calculate the spatial weights), its high accuracy underscores the efficiency of spatial weights in depicting the spatial autocorrelation between training samples and the irresponsive sample. It is worth noting that this classifier owes its high accuracy, in part, to the almost uniform and dense distribution of samples across space, shown in Fig. 11, which manifests itself in a large value for partial sill ($c_1 = 0.82$) in the spatial semivariogram (see Fig. 12).

The absence of such density and uniformity in distribution of samples across space and lack of a large value for partial sill (c_1) in the spatial semivariogram would make the geographical proximity of much less help in classifying irresponsive samples. This could significantly degrade the accuracy of this classifier. In such circumstances, nonspatial features can help to improve the classification accuracy.

Highly nonlinear classifiers (decision tree, collective weight, and Bayes) with location as the only input feature achieve the highest accuracies, as shown in Table V. This implies two facts in classifying land covers: 1) the distribution of classes in space is very nonlinear and 2) location proximity plays the most important role in identifying the land cover. The first fact explains why the classification accuracy drops for linear and slightly nonlinear classifiers even when location is their only input feature.

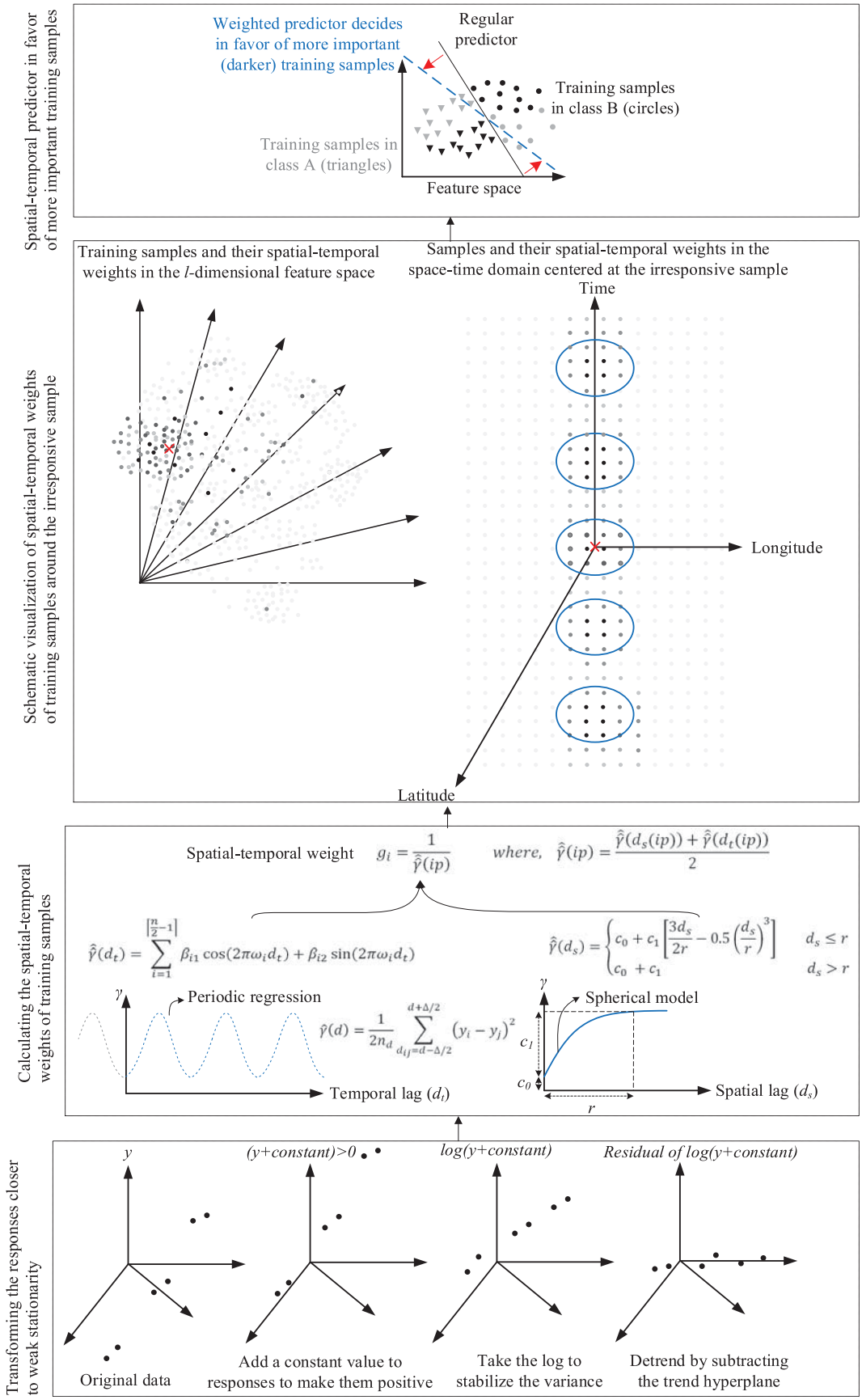


Fig. 13. Weighted machine learning for spatial-temporal data.

The second fact explains the low accuracy of any classifier that ignores the location altogether. None of these facts explain why the weighted version of these highly nonlinear classifiers (decision tree and nonparametric Bayesian) is outperformed by their nonweighted version that uses location as the only input feature. This can be explained by the low distinctive power (noisy behavior) of nonspatial features in identifying the classes, which is proven by the very low accuracy of all nonweighted classifiers with only nonspatial input features (second row in each group in Table V).

The lower sensitivity of perceptron to noisy training samples in comparison with LS comes as a major advantage in such circumstances, creating a considerable difference between their classification accuracies when there are nonspatial features among inputs. The low distinctive power of nonspatial features in identifying the land covers is also accountable for the decreasing accuracy of weighted classifiers as the training samples are shrunk to those with largest spatial weights. However, as the first four rows in Table V indicate, applying a small percentage of training samples with largest spatial weights, by itself, has a positive effect on the prediction accuracy.

Table VI ranks the same version of the seven different classifiers (collective weight, LS, nonparametric Bayes, decision tree, SVM, perceptron, and MLP) based on their accuracy in Table V.

In the following, we compare four general strategies in dealing with location in machine learning, where item (a) is the methodology known as spatial classification in GIS, items (b) and (c) are previously used methodologies in the literature on machine learning for spatial data, and item (d) is our proposed methodology.

a) Classifiers trained using location as the only feature.
b) Classifiers that ignore the location altogether.
c) Classifiers that take account of location as an additional input feature.
d) Classifiers that take account of location as weights for training samples.

Rows with a bold font in Table V show the settings leading to the best accuracy for each of the seven classification techniques (collective weight, LS, nonparametric Bayes, decision tree, SVM, perceptron, and MLP). The accuracy of classifiers in group (d) is always higher than those in group (b), is always higher than the accuracy of their nonweighted counterpart in group (c), with one exception, decision tree, and is higher than those in group (a) in majority of cases.

V. Conclusion and Future Directions

Fig. 13 shows the overall scheme of the proposed weighted machine learning for spatial-temporal data. It starts with calculating the spatial-temporal weight for training samples and ends with using them to bias the predictor in favor of training samples with larger weights. Spatial-temporal weights are visualized using color saturation. Darker samples have larger spatial-temporal weights. The red cross is the irresponsive sample and plays the central role in determining the spatial-temporal weight for training samples. Training samples are weighted based on their spatial and temporal auto-correlation with the irresponsive sample. In the spatial-temporal domain, the irresponsive sample is at the origin of the coordinate system and spheres delineate the training samples with larger spatial-temporal weights. The predictor is trained to be more concerned about the correct prediction of training samples with larger spatial-temporal weights.

The question posed in this study, how the recorded location and time for training samples should contribute to the training and testing process, can be answered more precisely now. We compared the following four approaches:

a) ignoring location and time;
b) considering location and time as the only input features (commonly used in GIS);
c) considering location and time as additional input features;
d) using the spatial-temporal autocorrelation between each training sample and the irresponsive sample as that training sample's weight in weighted machine learning techniques.

The first three are existing approaches, whereas the last one was proposed in this work. While we theoretically showed that the proposed approach captures the spatial-temporal autocorrelation more precisely, this was also confirmed by its higher accuracy in experiments with two real-world datasets. Because the spatial-temporal weight of training samples depends on the irresponsive sample's location and time, the machine needs to be trained separately for each irresponsive sample. We showed that using only a subset of training samples with largest spatial-temporal weights is an effective way to mitigate the training time without compromising the prediction accuracy. The accuracy of predictors in group (d) was followed by predictors in groups (c), (b), and (a), respectively. Nevertheless, this conclusion is based on only two datasets. Further research is required to investigate the generalizability of this conclusion to other datasets. Applying different feature selection and generation methods and investigating their effect on the prediction accuracy is another future research direction.

Our approach of calculating spatial weights for training samples assumes that the underlying phenomenon is isotropic by considering only the distance between pairs of observations and ignoring the direction of the vector connecting them. A future research venue is to increase the accuracy of spatial weights by taking into account the anisotropy or directional effects in the spatial variation of responses. This can be done by developing two (or more) spatial semivariograms, each modeling the spatial similarity in either north–south or east–west direction. To calculate the semivariance in a specific direction, only pairs of samples aligned in that direction (at least approximately) are used. Therefore, the direction of the vector connecting the irresponsive sample to each training sample determines which spatial semivariogram must be used to calculate that training sample's spatial weight. This way training samples aligned in a specific direction with respect to the irresponsive sample might gain a higher spatial weight.

We developed spatial and temporal semivariograms separately because the former is best modeled with a spherical model and the latter with a periodogram. Another future research venue is to find a way to develop a single spatial-temporal semivariogram.

REFERENCES

[1] D. O'Sullivan and D. Unwin, *Geographic Information Analysis*. Hoboken, NJ, USA: Wiley, 2010.

[2] N. Cressie, *Statistics for Spatial Data.*, Revised ed. Hoboken, NJ, USA: Wiley, 1993.

[3] S. Shekhar and S. Chawla, "Introduction to spatial data mining," in *Spatial Databases: A Tour*. Upper Saddle River, NJ, USA: Prentice-Hall, 2003, pp. 182–226.

[4] E. H. Isaaks and R. M. Srivastava, *An Introduction to Applied Geostatistics.*, Oxford, U.K.: Oxford Univ. Press, 1989.

[5] M. Hashemi and A. Alesheikh, "Spatio-temporal analysis of Tehran's historical earthquakes trends," in *Advancing Geoinformation Science for a Changing World.*, New York, NY, USA: Springer, 2011, pp. 3–20.

[6] M. Hashemi and H. A. Karimi, "Seismic source modeling by clustering earthquakes and predicting earthquake magnitudes," in *Smart City 360°*. New York, NY, USA: Springer, 2016, pp. 468–478.

[7] S. Santibanez, M. Kloft, and T. Lakes, "Performance analysis of machine learning algorithms for regression of spatial variables. A case study in the real estate industry," in *Proc. Geocomput. Conf.*, Dallas, TX, USA, 2015, pp. 292–297.

[8] S. Santibanez, T. Lakes, and M. Kloft, "Performance analysis of some machine learning algorithms for regression under varying spatial autocorrelation," in *Proc. AGILE*, Lisbon, Portugal, 2015, pp. 9–12.

[9] L. Anselin, *Spatial Econometrics: Methods and Models*. Dordrecht, The Netherlands: Kluwer, 1988.

[10] A. S. Fotheringham, C. Brunsdon, and M. Charlton, *Geographically Weighted Regression*. Chichester, U.K.: Wiley, 2002.

[11] A. S. Fotheringham, C. Brunsdon, and M. Charlton, *Quantitative Geography: Perspectives on Spatial Data Analysis.*, London, U.K.: Sage, 2000.

[12] S. Shekhar, P. Zhang, and Y. Huang, "Spatial data mining," in *Data Mining and Knowledge Discovery Handbook.*, 2nd ed. New York, NY, USA: Springer, 2010, pp. 837–854.

[13] S. Shekhar, P. Zhang, Y. Huang, and R. R. Vatsavai, "Trends in spatial data mining," in *Data Mining: Next Generation Challenges and Future Directions.*, H. Kargupta and A. Joshi, Eds. Cambridge, MA, USA: AAAI/MIT Press, 2003, pp. 357–380.

[14] W. R. Tobler, "A computer movie simulating urban growth in the Detroit region," *Econ. Geography*, vol. 46, pp. 234–240, 1970.

[15] K. E. Case and R. J. Shiller, "The efficiency of the market for single family homes," *Amer. Econ. Rev.*, vol. 79, no. 1, pp. 125–137, 1989.

[16] R. K. Pace, R. Barry, J. M. Clapp, and M. Rodriquez, "Spatio-temporal autoregressive models of neighborhood effects," *J. Real Estate Finance Econ.*, vol. 17, no. 1, pp. 15–33, 1998.

[17] M. Hashemi, "Reusability of the output of map-matching algorithms across space and time through machine learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3017–3026, Nov. 2017.

[18] M. Hashemi and H. A. Karimi, "A machine learning approach to improve the accuracy of GPS-based map-matching algorithms," in *Proc. IEEE 17th Int. Conf. Inf. Reuse Integr.*, Pittsburgh, PA, USA, 2016, pp. 77–86.

[19] B. Mertens and E. F. Lambin, "Land-cover-change trajectories in southern Cameroon," *Ann. Assoc. Amer. Geographers*, vol. 90, no. 3, pp. 467–494, 2000.

[20] R. M. Lucas, M. Honzak, G. M. Foody, P. J. Curran, and C. Corves, "Characterizing tropical secondary forests using multi-temporal Landsat sensor imagery," *Int. J. Remote Sens.*, vol. 14, no. 16, pp. 3061–3067, 1993.

[21] D. S. Alves and D. L. Skole, "Characterizing land cover dynamics using multi-temporal imagery," *Int. J. Remote Sens.*, vol. 17, no. 4, pp. 835–839, 1996.

[22] C. J. Tucker, H. E. Dregne, and W. W. Newcomb, "Expansion and contraction of the Sahara Desert from 1980 to 1990," *Science.*, vol. 253, pp. 299–301, 1991.

[23] B. Gokaraju, S. S. Durbha, R. L. King, and N. H. Younan, "A machine learning based spatio-temporal data mining approach for detection of harmful algal blooms in the Gulf of Mexico," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 4, no. 3, pp. 710–720, Sep. 2011.

[24] M. Hashemi, A. Alesheikh, and M. R. Zolfaghari, "A GIS-based time-dependent seismic source modeling of Northern Iran," *Earthq. Eng. Eng. Vib.*, vol. 16, no. 1, pp. 33–45, 2017.

[25] M. Hashemi, A. A. Alesheikh, and M. R. Zolfaghari, "A spatio-temporal model for probabilistic seismic hazard zonation of Tehran," *Comput. Geosci.*, vol. 58, pp. 8–18, 2013.

[26] J. Franklin, "Predictive vegetation mapping: Geographic modelling of biospatial patterns in relation to environmental gradients," *Prog. Phys. Geography*, vol. 19, no. 4, pp. 474–499, 1995.

[27] S. Shekhar, M. R. Evans, J. M. Kang, and P. Mohan, "Identifying patterns in spatial information: A survey of methods," *Wiley Interdiscip. Rev., Data Mining Knowl. Discovery*, vol. 1, no. 3, pp. 193–214, 2011.

[28] M. Kanevski *et al.*, "Environmental data mining and modeling based on machine learning algorithms and geostatistics," *Environ. Model. Softw.*, vol. 19, no. 9, pp. 845–855, 2004.

[29] N. Gilardi and S. Bengio, "Comparison of four machine learning algorithms for spatial data analysis," in *Mapping Radioactivity in the Environment: Spatial Interpolation Comparison 97*. Luxembourg City, Luxembourg: Office Official Publ. Eur. Communities, 2003, pp. 222–237.

[30] A. S. Britto Jr., R. Sabourin, and L. E. Oliveira, "Dynamic selection of classifiers—A comprehensive review," *Pattern Recognit.*, vol. 47, no. 11, pp. 3665–3680, 2014.

[31] S. Basu and T. G. Thibodeau, "Analysis of spatial autocorrelation in home prices," *J. Real Estate Finance Econ.*, vol. 16, no. 1, pp. 61–85, 1998.

[32] N. Gilardi and S. Bengio, "Local machine learning models for spatial data analysis," *J. Geographic Inf. Decis. Anal.*, vol. 4, no. 1, pp. 11–28, 2000.

[33] R. E. Bellman, *Adaptive Control Processes*. Princeton, NJ, USA: Princeton Univ. Press, 1961.

[34] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.

[35] P. F. Evangelista, M. J. Embrechts, and B. K. Szymanski, "Taming the curse of dimensionality in kernels and novelty detection," in *Applied Soft Computing Technologies: The Challenge of Complexity.*, Berlin, Germany: Springer, 2006, pp. 425–438.

[36] M. Gahegan, G. German, and G. West, "Improving neural network performance on the classification of complex geographic datasets," *J. Geographical Syst.*, vol. 1, no. 1, pp. 3–22, 1999.

[37] T. He, Y.-J. Sun, J.-D. Xu, X.-J. Wang, and C.-R. Hu, "Enhanced land use/cover classification using support vector machines and fuzzy k-means clustering algorithms," *J. Appl. Remote Sens.*, vol. 8, no. 1, 2014, Art. no. 083636.

[38] R. S. DeFries and J. C.-W. Chan, "Multiple criteria for evaluating machine learning algorithms for land cover classification from satellite data," *Remote Sens. Environ.*, vol. 74, no. 3, pp. 503–515, 2000.

[39] B. Barrett, I. Nitze, S. Green, and F. Cawkwell, "Assessment of multi-temporal, multi-sensor radar and ancillary spatial data for grasslands monitoring in Ireland using machine learning approaches," *Remote Sens. Environ.*, vol. 152, pp. 109–124, 2014.

[40] D. Rizzo, L. Martin, and J. Wohlfahrt, "Miscanthus spatial location as seen by farmers: A machine learning approach to model real criteria," *Biomass Bioenergy*, vol. 66, pp. 348–363, 2014.

[41] C. Ballabio and S. Sterlacchini, "Support vector machines for landslide susceptibility mapping: The Staffora River Basin case study, Italy," *Math. Geosci.*, vol. 44, no. 1, pp. 47–70, 2012.

[42] T. Dube, O. Mutanga, A. Elhadi, and R. Ismail, "Intra-and-inter species biomass prediction in a plantation forest: Testing the utility of high spatial resolution spaceborne multispectral RapidEye sensor and advanced machine learning algorithms," *Sensors*, vol. 14, no. 8, pp. 15348–15370, 2014.

[43] J. Li, A. D. Heap, A. Potter, and J. J. Daniell, "Application of machine learning methods to spatial interpolation of environmental variables," *Environ. Model. Softw.*, vol. 26, no. 12, pp. 1647–1659, 2011.

[44] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[45] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.

[46] M. Kanevski, V. Timonin, and A. Pozdnoukhov, "Automatic mapping and classification of spatial environmental data," in *Geocomputation, Sustainability and Environmental Planning*. New York, NY, USA: Springer, 2011, pp. 205–223.

[47] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, no. 3, pp. 1065–1076, 1962.

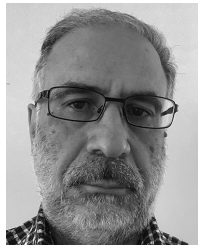
[48] M. J. Cracknell and A. M. Reading, "Geological mapping using remote sensing data: A comparison of five machine learning algorithms, their response to variations in the spatial distribution of training data and the use of explicit spatial information," *Comput. Geosci.*, vol. 63, pp. 22–33, 2014.

[49] S. R. Flaxman, "Machine learning in space and time," Ph.D. dissertation, Mach. Learn. Dept., Carnegie Mellon Univ., Pittsburgh, PA, USA, 2015.

[50] G. Bohling, "Introduction to geostatistics and variogram analysis," Kansas Geological Survey, Lawrence, KS, USA, vol. 2, 2005.
[51] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications: With R Examples.*, 3rd ed. New York, NY, USA: Springer, 2010.
[52] S. S. Stevens, "On the theory of scales of measurements," *Science.*, vol. 103, pp. 677–680, 1946.
[53] A. D. Cliff and J. K. Ord, *Spatial Autocorrelation*. London, U.K.: Pion, 1973.
[54] D. J. Unwin, *Introductory Spatial Analysis.* London, U.K.: Methuen, 1981.
[55] C. Chatfield, *The Analysis of Time Series: An Introduction.*, 6th ed. Boca Raton, FL, USA: CRC Press, 2016.
[56] L. Wasserman, *All of Nonparametric Statistics*, Berlin, Germany: Springer, 2006.
[57] M. Hashemi and H. A. Karimi, "Weighted machine learning," *Statist., Optim. Inf. Comput.*, vol. 6, no. 4, pp. 497–525, 2018.
[58] "Oceanographic and surface meteorological readings taken from a series of buoys positioned throughout the equatorial Pacific," Pacific Marine Environmental Laboratory, National Oceanic and Atmospheric Administration, U.S. Department of Commerce, Seattle, WA, USA, 1998.
[59] R. W. Fitzgerald and B. G. Lees, "Spatial context and scale relationships in raster data for thematic mapping in natural systems," in *Advances in GIS Research*, T. C. Waugh and R. Healey, Eds. Southampton, U.K.: Taylor & Francis, 1995, pp. 462–475.
[60] R. W. Fitzgerald and B. G. Lees, "Temporal context in floristic classification," *Comput. Geosci.*, vol. 22, no. 9, pp. 981–994, 1996.
[61] Z. Huang and B. G. Lees, "Combining non-parametric models for multisource predictive forest mapping," *Photogrammetric Eng. Remote Sens.*, vol. 70, no. 4, pp. 415–425, 2004.
[62] Z. Huang and B. G. Lees, "Representing and reducing error in natural-resource classification using model combination," *Int. J. Geographical Inf. Sci.*, vol. 19, no. 5, pp. 603–621, 2005.
[63] B. Lees, "The spatial analysis of spectral data: Extracting the neglected data," *Appl. GIS*, vol. 2, no. 2, pp. 14.1–14.13, 2006.
[64] B. Lees, "The Kioloa GLCTS Pathfinder Site," 2007. [Online]. Available: http://fennerschool-associated.anu.edu.au/pathfinder/
[65] R. A. Jacobs, "Increased rates of convergence through learning rate adaptation," *Neural Netw.*, vol. 1, no. 4, pp. 295–307, 1988.

Mahdi Hashemi received the Ph.D. degree in computing and information from the University of Pittsburgh, Pittsburgh, PA, USA, in 2017.

He is currently an Assistant Professor with the Department of Information Sciences and Technology, George Mason University, Fairfax, VA, USA. He leads the Machine Learning and Urban Computing Group, where he also specializes in intelligent transportation, spatial-temporal data, and web/social media analytics.



Hassan A. Karimi received the Ph.D. degree in geomatics engineering from the University of Calgary, Calgary, Canada, in 1992.

He is currently a Professor and Director of the Geoinformatics Laboratory with the School of Computing and Information, University of Pittsburgh, Pittsburgh, PA, USA. His research interests include computational geometry, computational topology, machine learning, location-based services, navigation, and distributed/parallel computing.