

# An Integrated Framework for Simultaneous Classification and Regression of Time-Series Data

Zubin Abraham  
Michigan State University  
Department of Computer Science  
East Lansing, MI-48824, USA  
abraha84@cse.msu.edu

Pang-Ning Tan  
Michigan State University  
Department of Computer Science  
East Lansing, MI-48824, USA  
ptan@cse.msu.edu

## Abstract

Zero-inflated time series data are commonly encountered in many applications, including climate and ecological modeling, disease monitoring, manufacturing defect detection, and traffic monitoring. Such data often leads to poor model fitting using standard regression methods because they tend to underestimate the frequency of zeros and the magnitude of non-zero values. This paper presents an integrated framework that simultaneously performs classification and regression to accurately predict future values of a zero-inflated time series. A regression model is initially applied to predict the value of the time series. The regression output is then fed into a classification model to determine whether the predicted value should be adjusted to zero. Our regression and classification models are trained to optimize a joint objective function that considers both classification errors on the time series and regression errors on data points that have non-zero values. We demonstrate the effectiveness of our framework in the context of its application to a precipitation downscaling problem for climate impact assessment studies.

## 1 Introduction

Predictive models for time series data are commonly employed in the fields of economics, finance, epidemiology, ecology, and meteorology, among others. The prediction accuracy is subject to the choice of model used, which in turn, may be limited by characteristics of the time series observations. For example, studies have shown that the performance of classical regression models is degraded when applied to data sets with excess zero values [2, 4, 17, 7, 28]. Such data are typically encountered in applications such as climate and ecological modeling, disease monitoring, manufacturing defect detection, and traffic monitoring.

Figure 1 shows the histogram of daily precipitation (in log scale) at a weather station in Canada for the period between January 1, 1961 and December 31,

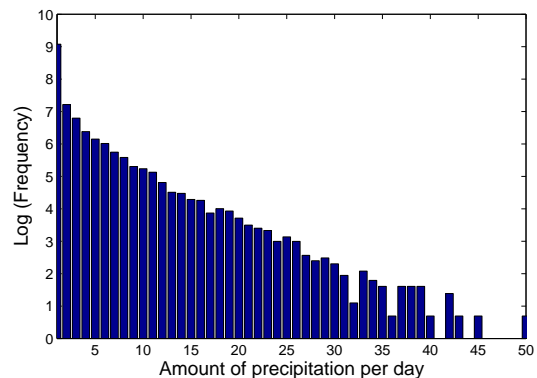


Figure 1: A zero-inflated frequency distribution of daily precipitation at a weather station in Canada

2000. Nearly half of the observations have precipitation values equal to zero. Such zero-inflated data, as they are commonly known, often lead to poor model fitting using standard regression methods as they tend to underestimate the frequency of zeros and the magnitude of non-zero values of the data. A typical strategy for handling such type of data is to first invoke a classification model to predict whether the output value is zero. A regression model, which has been trained on the non-zero data points, is then applied to estimate its magnitude only if the classifier predicts a non-zero output. Such an approach is commonly used for statistical downscaling of precipitation [30], in which the occurrence of rain or wet days is initially predicted prior to applying a regression model to estimate the amount of rainfall for the predicted wet days. The limitation of this approach is that the classification and regressions models are often built independent of each other. As a result, neither models can glean information from the other to potentially improve their prediction accuracy.

The objective of this paper is to develop an in-

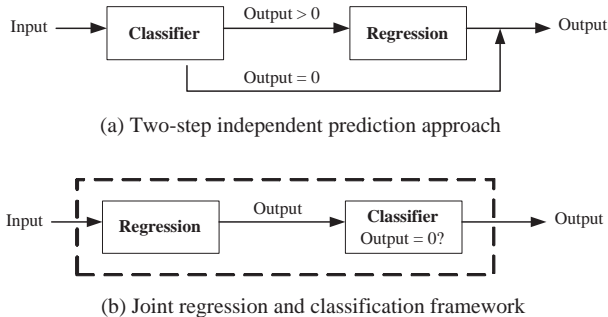


Figure 2: Comparison between independent modeling approach and proposed framework for predicting zero inflated data

egrated framework that accurately estimates the future values of a zero-inflated time series by simultaneously training the classification and regression models. Specifically, the models are trained to optimize a joint objective function that penalizes errors in classifying a data point and errors in predicting the magnitude of non-zero data points. Given a test point, the regression model is applied to estimate the magnitude of the predicted value. The output from the regression model along with the values of other predictor variables of the test point are then fed into a classification model to determine whether the predicted value should be adjusted to zero. The distinction between the traditional two-step independent modeling approach and our proposed framework is illustrated in Figure 2.

We demonstrate the effectiveness of our learning framework in the context of precipitation prediction using climate data from the Canadian Climate Change Scenarios Network Web site [1]. Specifically, we compared the performance of our integrated framework against two baseline methods. The first baseline corresponds to applying standard multiple linear regression (MLR) method on the entire training data, which includes both dry and rain days. The second baseline method (SVM-MLR) uses a combination of support vector machine classifier to predict dry/wet days and multiple linear regression to predict rainfall amount on wet days. Both the models are trained independently. Empirical results showed that our proposed framework outperforms both MLR and SVM-MLR on the majority of the weather stations investigated in this study.

In summary, the main contributions of this paper are as follows:

- We present an integrated framework for simultaneously learning classification and regression models.
- We showed that the proposed framework is more effective at predicting zero-inflated time series than

building a single regression model or building independent classification and regression models to fit the time series data.

- We successfully applied our framework to the real-world problem of downscaling precipitation time series for climate impact assessment studies.

The remainder of this paper is organized as follows. Section 2 presents the related work on time series prediction and zero-inflated regression models. Section 3 introduces the notation and problem formulation. The integrated classification and regression framework proposed in this study is presented in Section 4, followed by a detailed description of our algorithm in Section 5. Experimental results are given in Section 6. Finally, we present our conclusions and suggestions for future work in Section 7.

## 2 Related Work

Time series prediction has long been an active area of research with applications in finance [31], weather forecasting [16][10], network monitoring [8], transportation planning [20][24], etc. There are many time series prediction techniques available, including least square regression [22], recurrent neural networks [19], Hidden Markov Model Regression [18], and support vector regression [25]. In the Earth science domain, there has been extensive research on applying time series regression models for downscaling General Circulation Models (GCM) data [10, 16, 29]. GCMs are computer-generated models for simulating future climate conditions under different greenhouse gas emission scenarios. However, the spatial resolution of GCM outputs are often too coarse to reliably project the future climate scenarios of a local region. Statistical downscaling techniques are therefore used to relate the coarse-scale GCM outputs to the local climate variables such as daily precipitation and temperature [29].

The motivation behind the combined use of classification and regression models for time series prediction is due to the zero-inflated data problem. Previous studies have shown that additional precautions must be taken to ensure that the excess zeros do not lead to poor fits [2, 4, 17, 7, 28] of the regression models. A typical approach to model a zero-inflated data set is to use a mixture distribution of the form  $P(y|\mathbf{x}) = \alpha\pi_0(\mathbf{x}) + (1 - \alpha)\pi(\mathbf{x})$ , where  $\pi_0$  and  $\pi$  are functions of the predictor variables  $\mathbf{x}$  and  $\alpha$  is a mixing coefficient that governs the probability an observation is a zero or non-zero value. This approach assumes that the underlying data is generated from known parametric distributions, for example,  $\pi$  may be Poisson or negative binomial distribution (for discrete data) and log-

normal or Gamma (for continuous data). In contrast, the framework presented in this paper does not require making such a strong assumption about the distribution of the data.

### 3 Preliminaries

Consider a multivariate time series  $\mathbf{L} = (\mathbf{x}_t, \mathbf{c}'_t)$ , where  $t \in \{1, 2, \dots, n\}$  denote the elapsed time,  $\mathbf{x}_t$  is a  $d$ -dimensional vector of predictor variables at time  $t$ , and  $c_t$  is the corresponding value for the response (target) variable. Given an unlabeled sequence of multivariate observations  $\mathbf{x}_\tau$ , where  $\tau \in \{n + 1, \dots, n + m\}$ , our goal is to learn a target function  $f(\mathbf{x}_\tau, \mathbf{w})$  that best estimates the future values of the response variable at each time  $\tau$ . The set of weights  $\mathbf{w} = [w_1, w_2, \dots, w_d]^T$  are the regression coefficients to be estimated from the training data  $\mathbf{L}$ . For applications such as statistical downscaling, the predictor variables  $\mathbf{x}_\tau$  correspond to climate variables at large spatial scales generated from computer-driven general circulation models (GCMs).

For zero-inflated data, the frequency of zero values in the time series is relatively larger than the frequency of each non-zero values, as shown in Figure 1. The response variable  $c'_t$  can be mapped into a binary class  $c_t$ , where

$$(3.1) \quad c_t = \begin{cases} 1, & \text{if } c'_t > 0; \\ 0, & \text{otherwise.} \end{cases}$$

For brevity, we use the notation  $y \equiv f(\mathbf{x}, \mathbf{w})$  as the predicted value of the response variable and  $y$  as its corresponding predicted class.

### 4 Framework for Simultaneous Classification and Regression

This paper considers a framework for predicting future values of a zero-inflated time series using a combination of classification and regression models. The models in our framework are trained to optimize a joint objective function that considers both the classification errors on the time series and regression errors for the non-zero values. A preliminary version of this work appeared in a workshop paper [3], which uses support vector machine (SVM) as the underlying classifier. The method is computationally expensive since it requires the SVM classifier to be re-trained at each iteration. Furthermore, it does not guarantee convergence of the algorithm. The framework presented in this paper trains an SVM classifier only once after the parameters of the regression model have been determined. Proofs of convergence of our algorithm are also presented in this section.

We consider multiple linear regression (MLR) as the underlying regression model in this study, in which

$f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x}$ . Extending the approach to nonlinear models will be a subject for future research.

**4.1 Objective Function** The classification and regression models developed in this study are designed to minimize the following objective function:

$$\begin{aligned} \arg \min_{\mathbf{w}, \mathbf{y}} L(\mathbf{w}, \mathbf{y}) &= \sum_{i=1}^n c_i (c'_i - y_i y'_i)^2 + T_1 \sum_{i=1}^n (y_i - c_i)^2 \\ &+ T_2 \sum_{i,j=1}^n s_{i,j} [c_i y'_i - c_j y'_j]^2 + T_3 \|w\|^2 \end{aligned}$$

where,

$$y'_i = \sum_d w_d x_{i,d}, \quad y_i \in \{0, 1\}$$

and  $s_{ij}$  is the similarity between the values of the predictor variables at  $t_i$  and  $t_j$

The rationale for the design of our objective function is as follows. The first term is somewhat similar to the standard least-square formulation of multiple linear regression, except the estimation of  $\mathbf{w}$  is based on the non-zero values in the time series. The regression model is therefore biased towards estimating the non-zero values more accurately instead of being influenced by the over-abundance of zeros in the time series. The product  $y_i y'_i$  in the first term corresponds to the predicted output of our joint classification and regression models. The second term in the objective function is equivalent to misclassification error in training data. The third term corresponds to a graph regularization constraint to ensure smoothness and consistency in the model predictions. Specifically, for two highly similar data points  $\mathbf{x}_p$  and  $\mathbf{x}_q$ , i.e.,  $s_{pq}$  is large, we penalize the model if the predicted values of the response variables are inconsistent. Finally, the last term in the objective function is equivalent to the  $L_2$  norm used in ridge regression models to shrink the coefficients in  $\mathbf{w}$ .

We consider each data point to be a given elapsed time  $t \in \{1, 2, \dots, n\}$  in the time series. An  $n \times n$  similarity matrix  $\mathbf{S} = [s_{ij}]$  is computed between every pair of data points based on the similarities of their predictor variables. Prior to computing the similarity matrix, each variable is standardized by subtracting its mean value and then dividing by its corresponding standard deviation. The standardization of the variables is needed to account for their varying scales. We use Pearson correlation coefficient to compute the similarity between each pair of data points and then transform the value to a range between 0 and 1 to ensure all the terms in the objective function are non-negative. The choice of Pearson correlation as our similarity measure is due to the popularity of the measure in the Earth

Science domain. We plan to investigate other similarity functions such as the radial basis function (RBF) as part of our future work.

**4.2 Parameter Estimation** The objective function can be further expanded as follows:

$$\begin{aligned} L(\mathbf{w}, \mathbf{y}) &= \sum_{i=1}^n c_i (c'_i - y_i \sum_d w_d x_{i,d})^2 + T_1 \sum_{i=1}^n (y_i - c_i)^2 \\ &+ T_2 \sum_{i,j=1}^n s_{i,j} \left( \sum_d c_i w_d x_{i,d} - \sum_d c_j w_d x_{j,d} \right)^2 \\ &+ T_3 \|w\|^2 \end{aligned}$$

or equivalently,

$$\begin{aligned} L(\mathbf{w}, \mathbf{y}) &= \sum_{i=1}^n c_i (c'_i - y_i \sum_d w_d x_{i,d})^2 \\ &+ T_1 \sum_{i=1}^n (y_i - c_i)^2 + T_3 \|w\|^2 \\ &+ T_2 \sum_{i,j=1}^n s_{i,j} \left( \left( \sum_d c_i w_d x_{i,d} \right)^2 + \left( \sum_d c_j w_d x_{j,d} \right)^2 \right. \\ &\left. - 2 \sum_{d,d'} c_i c_j w_d w_{d'} x_{i,d} x_{j,d'} \right) \end{aligned}$$

To estimate the regression parameter  $\mathbf{w}$  and class labels  $\mathbf{y}$ , we employ the following iterative procedure. First, we compute the partial derivative of  $L(\mathbf{w}, \mathbf{y})$  with respect to each of the  $w$ 's and set it to zero (assuming  $\mathbf{y}$  is fixed):

$$\begin{aligned} \frac{\partial L}{\partial w_k} &= \left[ -2 \sum_{i=1}^n c_i \left( c'_i - y_i \sum_d w_d x_{i,d} \right) \left( y_i x_{i,k} \right) \right. \\ &+ 2T_2 \sum_{i,j=1}^n s_{i,j} \left( \left( \sum_d c_i w_d x_{i,d} \right) \left( c_i x_{i,k} \right) \right) \\ &+ 2T_2 \sum_{i,j=1}^n s_{i,j} \left( \left( \sum_d c_j w_d x_{j,d} \right) \left( c_j x_{j,k} \right) \right) \\ &- 2T_2 \sum_{i,j=1}^n s_{i,j} \left( \sum_d c_i c_j w_d (x_{i,d} x_{j,k} + x_{i,k} x_{j,d}) \right) \\ &\left. + 2T_3 w_k \right] = 0 \end{aligned}$$

This reduces to a system of linear equations of the form  $\mathbf{A}\mathbf{w} = \mathbf{b}$  where

$$b_k = \sum_{i=1}^n c_i y_i c'_i x_{i,k}$$

and  $A$  is a square matrix of dimension  $d \times d$  whose non-diagonal elements is given by,

$$\begin{aligned} \mathbf{A}_{k,l} &= 2T_2 \sum_{i,j=1}^n s_{i,j} c_i x_{i,l} x_{i,k} \\ &- 2T_2 \sum_{i,j=1}^n s_{i,j} c_i c_j x_{i,l} x_{j,k} \\ &+ \sum_{i=1}^n c_i y_i x_{i,l} x_{i,k} \end{aligned}$$

and diagonal elements

$$\begin{aligned} \mathbf{A}_{k,k} &= 2T_2 \sum_{i,j=1}^n s_{i,j} c_i x_{i,k}^2 \\ &- 2T_2 \sum_{i,j=1}^n s_{i,j} c_i c_j x_{i,k} x_{j,k} \\ &+ \sum_{i=1}^n c_i y_i x_{i,k}^2 + T_3 \end{aligned}$$

To estimate  $\mathbf{y}$ , we minimize the following part of the objective function that depends on  $\mathbf{y}$ :

$$L_c(\mathbf{y}) = \sum_{i=1}^n c_i (c'_i - y_i y'_i)^2 + T_1 \sum_{i=1}^n (y_i - c_i)^2$$

subject to the constraint  $y_i \in \{0, 1\}$ . It is straightforward to show that  $L_c$  is minimized according to the following rule:

$$(4.2) \quad y_i = \begin{cases} 1, & \text{if } c_i = 1 \text{ and } (c'_i - y'_i)^2 > c_i'^2 + T_1; \\ 0, & \text{otherwise.} \end{cases}$$

The predicted class labels  $\mathbf{y}$  are then used to re-estimate the regression coefficients  $\mathbf{w}$ . This procedure is repeated until the regression coefficients and class labels converge.

**4.3 Proof of Convergence** This section presents the proof of convergence of our iterative update algorithm. Let  $(w_t, \mathbf{y}_t)$  be the regression coefficients and class labels estimated after the  $t$ -th iteration and  $(w_{t+1}, \mathbf{y}_{t+1})$  be the regression coefficients and class labels estimated after the  $(t+1)$ -th iteration.

**PROPOSITION 1.** *Assuming that the class labels  $\mathbf{y}_t$  are fixed,  $L(w_{t+1}, \mathbf{y}_t) \leq L(w_t, \mathbf{y}_t)$ .*

*Proof.* For a fixed  $\mathbf{y}_t$ , let  $L_r(\mathbf{w})$  be part of the objective function that depends on the regression coefficients  $\mathbf{w}$ :

$$\begin{aligned} L_r(\mathbf{w}) &= \sum_{i=1}^n c_i (c'_i - y_i \sum_d w_d x_{i,d})^2 + T_3 \|w\|^2 \\ &+ T_2 \sum_{i,j=1}^n s_{i,j} \left( \sum_d c_i w_d x_{i,d} - \sum_d c_j w_d x_{j,d} \right)^2 \end{aligned}$$

The Hessian matrix  $\mathbf{H}$  of  $L_r(\mathbf{w})$  is given by:

$$\begin{aligned} \frac{\partial^2 L_r}{\partial w_k \partial w_l} &= 2 \sum_{i=1}^n c_i y_i^2 x_{i,k} x_{i,l} + 2T_3 \delta_{kl} \\ &+ 2T_2 \sum_{i,j=1}^n s_{i,j} (c_i x_{i,k} - c_j x_{j,k})(c_i x_{i,l} - c_j x_{j,l}) \end{aligned}$$

where  $\delta_{kl} = 1$  if  $k = l$  and zero otherwise. Since the parameters  $T_2$  and  $T_3$  are non-negative, it can be shown that, for any non-zero vector  $\mathbf{z}$  with real values,  $\mathbf{z}^T \mathbf{H} \mathbf{z} \geq 0$ , i.e., the Hessian matrix is positive semi-definite. Thus, the stationary point  $\mathbf{w}_{t+1}$  minimizes  $L(\mathbf{w}_{t+1})$  and  $L_r(\mathbf{w}_{t+1}) \leq L_r(\mathbf{w}_t)$ .

**PROPOSITION 2.** *Assuming that the regression coefficients are fixed,  $L(\mathbf{w}_{t+1}, \mathbf{y}_{t+1}) \leq L(\mathbf{w}_{t+1}, \mathbf{y}_t)$ .*

*Proof.* For a fixed  $\mathbf{w}_{t+1}$ , let  $L(\mathbf{w}_{t+1}, \mathbf{y}_t) = L_c(\mathbf{y}_t) + T_2 \sum_{i,j=1}^n s_{i,j} [c_i y'_i - c_j y'_j]^2 + T_3 \|w\|^2$ . Note that last two terms are independent of  $\mathbf{y}_t$ . Since our update formula for  $\mathbf{y}_t$  minimizes  $L_c(\mathbf{y})$ , it follows that  $L(\mathbf{w}_{t+1}, \mathbf{y}_{t+1}) \leq L(\mathbf{w}_{t+1}, \mathbf{y}_t)$ .

**THEOREM 4.1.** *The objective function  $L(w)$  is monotonically non-increasing given the update formula for  $\mathbf{w}$  and  $\mathbf{y}$ .*

*Proof.* The update formula iteratively modifies the objective function as follows:  $L(\mathbf{w}_t, \mathbf{y}_t) \Rightarrow L(\mathbf{w}_{t+1}, \mathbf{y}_t) \Rightarrow L(\mathbf{w}_{t+1}, \mathbf{y}_{t+1})$ . Using the above propositions we have  $L(\mathbf{w}_{t+1}, \mathbf{y}_t) \leq L(\mathbf{w}_t, \mathbf{y}_t)$  and  $L(\mathbf{w}_{t+1}, \mathbf{y}_{t+1}) \leq L(\mathbf{w}_{t+1}, \mathbf{y}_t)$ . Therefore,  $L(\mathbf{w}_{t+1}, \mathbf{y}_{t+1}) \leq L(\mathbf{w}_t, \mathbf{y}_t)$ .

**LEMMA 4.1.** *The objective function will eventually converge, as the value of the loss function is always non-negative and since we know  $L(w)$  is monotonically decreasing.*

**4.4 Classification of Test Data** The update formula presented in the previous subsections compute the regression coefficients  $\mathbf{w}$  and class labels  $\mathbf{y}$  of the

training examples in such a way that minimizes the objective function. For a given test example  $\mathbf{x}_\tau$ , where  $\tau \in \{n+1, \dots, n+m\}$ , the predicted value of the regression model can be computed as follows:  $y'_\tau = \mathbf{w}^T \mathbf{x}_\tau$ . However, the classification output cannot be determined since the update formula for  $\mathbf{y}$  depends on the true class labels  $\mathbf{c}$ , as shown in Equation (4.2). Therefore, to predict the class label  $\mathbf{y}$ , we build an SVM classifier on  $(\mathbf{x}_t, \mathbf{y}'_t)$  as the  $d+1$ -dimensional feature vector and the estimated  $(\mathbf{y}_t)$  as the class labels using only examples from training data. Once the classifier has been constructed, it can be applied to predict the class label of a test example. The final output of our joint classification and regression model is the product  $y_\tau y'_\tau$  (see Figure 2).

Empirically, it was found that SVM may be used as an alternate classifier to predict  $y$  at each iteration, instead of the update formula described above. But since the objective function of the generic classifier does not necessarily minimize both the first and second term of  $L_c(y)$  simultaneously, convergence cannot be guaranteed.

## 5 Algorithm

A summary of our proposed framework is presented in Algorithm 1. In the remainder of this paper, our algorithm will be denoted as ZICR (where ZICR stands for Zero-Inflated Classification-Regression method).

---

**Algorithm 1** Algorithm for Concurrent Regression and Classification

---

**Input:**

$(\mathbf{x}_t, \mathbf{c}'_t)$ : A multivariate time series with  $d$ -dimensional predictor variables  $\mathbf{x}_t$  and response variable  $\mathbf{c}'_t$ .

$(\mathbf{x}_\tau)$ : A sequence of unlabeled observations.

**Output:**

$\mathbf{w}$ : Regression coefficients

$(z_\tau)$ : Predicted values of unlabeled sequence.

**Method:**

Training Phase:

1. Set  $\mathbf{c} = (\mathbf{c}' > 0)$

2. Initialize  $\mathbf{y} = \mathbf{c}$

3. Repeat until convergence

3a. Update  $\mathbf{w}$  by solving  $\mathbf{A}\mathbf{w} = \mathbf{b}$

3b. Update  $\mathbf{y}$  using Equation (4.2)

4. Train an SVM classifier  $g : (\mathbf{x}_t, \mathbf{y}'_t) \rightarrow \mathbf{y}_t$

Testing Phase:

5.  $\forall \tau : y'_\tau = \mathbf{w}^T \mathbf{x}_\tau$

5.  $\forall \tau : y_\tau = g(\mathbf{x}_\tau, y'_\tau)$

6.  $\forall \tau : z_\tau = y_\tau y'_\tau$

---

We assume the time series data has been partitioned

into a training set, a validation set (for model selection), and a test set. Model selection is needed to estimate the parameters  $T_1, T_2, T_3$  of our objective function  $L(\mathbf{w}, \mathbf{y})$ .

The class labels  $\mathbf{c}$  of the training examples are obtained based on the response variable  $\mathbf{c}'$ . The training phase of the algorithm starts by setting  $\mathbf{y} = \mathbf{c}$  for all the  $n$ -training examples. It then iteratively updates the regression coefficients  $\mathbf{w}$  and class labels  $\mathbf{y}$  according to the methodology presented in the previous section. At this stage, the value of the objective function is computed and saved for testing convergence of the objective function. Upon convergence, an SVM classifier  $g$  is constructed to learn the mapping between the input features  $\mathbf{x}, \mathbf{y}'$  and output class  $\mathbf{y}$ .

Once the training phase is completed, the Testing phase begins. Testing is performed by first applying the multiple linear regression model to the predictor variables  $\mathbf{x}_\tau$ . This is followed by invoking the SVM classifier to predict the class label  $y_\tau$  for the  $m$  test examples. The classifier takes  $\mathbf{x}_\tau$  and  $y'_\tau$  as input and returns class labels  $y_\tau$ . Finally, the prediction output is obtained by setting  $z_\tau = y_\tau y'_\tau$ .

The time complexity of the training phase of the algorithm is  $O(k(n^2d + d^3))$ , where  $n$  is the number of training examples,  $d$  the number of predictor variables and  $k$  is the maximum number of iterations required for convergence. The computational complexity of the training phase is composed of two major parts: the first that requires computing the similarity matrix and the second that requires iteratively solving  $\mathbf{w}$  and  $\mathbf{y}$ . The time needed to compute the similarity matrix is  $O(n^2d)$ . The time complexity of each iteration refers to the time needed to compute  $\mathbf{w}$  ( $O(n^2d^2 + d^3)$ ) plus time needed to compute  $\mathbf{y}$  ( $O(n)$ ). Hence, for maximum iterations set to  $k$ , the time complexity for the training phase is  $O(k(n^2d + d^3))$ , where  $d \ll n$ .

## 6 Experimental Evaluation

This section presents the experimental results to demonstrate the effectiveness of our proposed framework.

**6.1 Experimental Setup** The algorithm detailed in the earlier section was applied to climate data from the Canadian Climate Change Scenarios Network Web site [1]. The response variable to be regressed corresponds to daily precipitation values measured at 37 weather stations in Canada. The predictor variables correspond to 26 coarse-scale climate variables derived from the NCEP Reanalysis data set, which include measurements of sea-level pressure, wind direction, vorticity, and humidity, as shown in Table 1. The data are available for a 40-year period, 1961 to 2001. We also truncated the time series for each weather station to exclude the

days in which the precipitation values are missing.

Table 1: List of predictor variables for precipitation prediction.

Predictor Variables	
Mean sea level pressure	Surface zonal velocity
Surface airflow	Surface meridional velocity
Surface vorticity	Surface wind direction
Surface divergence	Mean temp at 2m
500 hPa airflow	850 hPa airflow
500 hPa zonal velocity	850 hPa zonal velocity
500 hPa meridional velocity	850 hPa meridional velocity
500 hPa vorticity	850 hPa vorticity
500 hPa geopotential height	850 hPa geopotential height
500 hPa wind direction	850 hPa wind direction
500 hPa divergence	850 hPa divergence
Relative humidity at 500 hPa	Relative humidity at 850 hPa
Near surface relative humidity	Surface specific humidity

We compared the performance of our algorithm against the multiple linear regression (MLR) model. MLR uses the least square criterion to estimate the weight vector  $\mathbf{w}$  of the model. Our algorithm was also compared with the approach of combining MLR with SVM (SVM-MLR). In SVM-MLR, SVM was used to learn a classifier model to differentiate between **Rain** and **NoRain** days, and MLR was learnt on rain days only. Finally, for the given test set MLR is applied only to those days classified as a **Rain** day. For the choice of SVM during the evaluation phase, a choice of the kernel (Linear or RBF) and its respective parameter are made. The choice of the SVM kernel for ZICR was limited to a linear kernel. Future experiments will include a wider selection during the evaluation phase.

We used the following criteria to evaluate the performance of the models:

- Root Mean Square Error (RMSE), which measures the difference between the actual and predicted values of the response variable, i.e.:  $RMSE = \sqrt{\frac{\sum_1^n (c'_i - y'_i)^2}{n}}$ .
- Accuracy, which measures the number of **Rain** and **NoRain** days predicted correctly by the model.
- F-measure, which is the harmonic mean between recall and precision values for rain days.

**6.2 Experimental Results** The purpose of our experiment is to demonstrate the following:

1. Limitations of classical regression models in terms of handling zero-inflated time series data.
2. Performance comparison between classical regression models and our proposed framework.

**6.2.1 Effect of Zero-Inflated Time Series Data**

The objective of this experiment is to demonstrate the effect of increasing number of zeros in a time series on the performance of a regression model. Specifically, given the precipitation time series of a randomly selected weather station, we classified each day as NoRain or Rain, depending on the amount of precipitation it receives is equal to or greater than zero. We then created several training sets of different sizes and varying percentage of NoRain and Rain days by randomly sampling the original time series. A disjoint test set of size 'ten years' is used for all the experiments in this subsection.

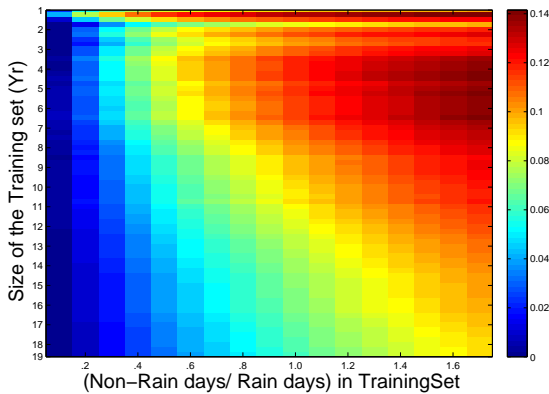


Figure 3: Effect of increasing the number of NoRain days on performance of regression model (best viewed in color).

We evaluated the performance of two multiple linear regression (MLR) models: (1) MLR<sub>1</sub>, which is trained on both Rain and NoRain days and (2) MLR<sub>2</sub>, which is trained on Rain days only. Figure 3 compares the RMSE values of both models for Rain days in the test set. The horizontal axis corresponds to the ratio of NoRain to Rain days in the training set. The larger the ratio, the more inflated the number of zeros in the training data. The vertical axis corresponds to the training set size, where each unit on the scale represents a period of three months. The value of each cell indicates the performance improvement when using

MLR<sub>2</sub> to predict the Rain days:

$$(6.3) \quad \%Improvement = \frac{RMSE(MLR_1) - RMSE(MLR_2)}{RMSE(MLR_1)}$$

Since the % Improvement is greater than or equal to zero, this indicates that MLR<sub>2</sub> consistently outperforms MLR<sub>1</sub> in terms of predicting future Rain days irrespective of the training set size. The amount of improvement becomes even more pronounced when the percentage of NoRain days in the training data increases. A similar improvement pattern is observed for all the weather stations investigated in this study, as shown in Figure 4. In contrast, MLR<sub>1</sub>, which is trained on both Rain and

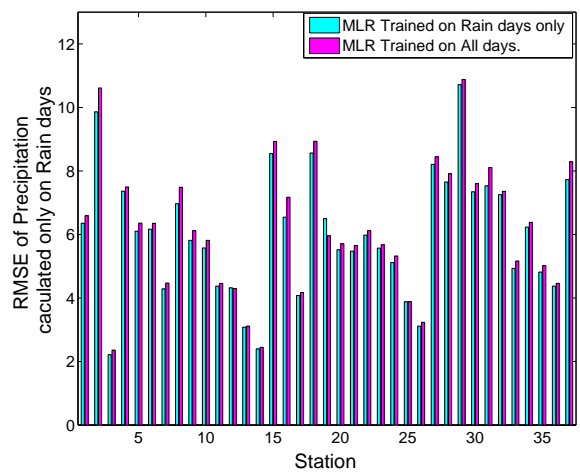


Figure 4: Comparison of RMSE values (Tested on Rain days only) for MLR models trained on all days compared with models trained only on Rain days.

NoRain days, has a lower RMSE compared to MLR<sub>2</sub> when applied to all the days in the test set, as shown in Figure 5. This is because MLR<sub>2</sub> tends to overestimate the amount of precipitation for the NoRain days.

In summary, the experiment given in this section clearly justifies the rationale for applying a combination of classification and regression models to better estimate the precipitation amount of Rain days.

**6.2.2 Impact of Coupling the Classifier and Regression Model Creation**

The objective of this experiment is to demonstrate the advantage of building a classifier and regression model in conjunction with each other, as against building them independent of the other for zero-inflated time-series data. Specifically, empirical results demonstrating improvement in the classification accuracy, F-measure of classification as well as RMSE of the predictors are provided.

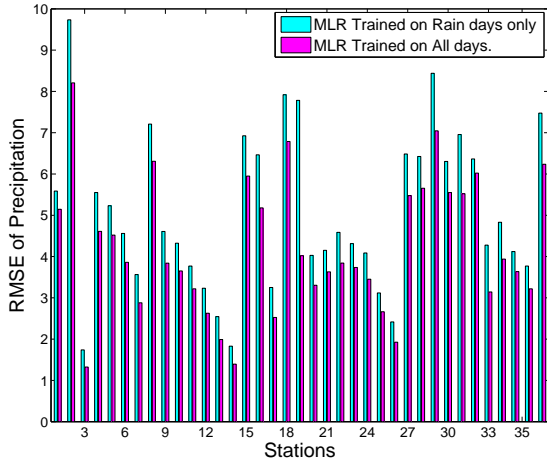


Figure 5: Comparison of RMSE values (Tested on All days) for MLR models trained on all days compared with models trained only on Rain days.

We evaluated and compared the performance of two multiple linear regression models. In the first model, MLR is trained on all days and a quadratic discriminant analysis (QDA) trained on ground truth response variable. In the second model, again MLR is trained on all days but the QDA trained on the predicted response values  $\mathbf{y}' = \mathbf{w}^T \mathbf{x}$ . The results of the experiment show that the model trained on the predicted response values outperformed the model trained on ground truth response variable for all 37 stations, when it came to RMSE, Classification and F-Measure. In particular, the average improvements were 13.4% and 19.3% when it came to classification accuracy.

In summary, these empirical results provide motivation to try and integrate the classifier and regression models to take into consideration the accuracy of the other's prediction for each individual data point.

**6.2.3 Performance Comparison** This section compares the RMSE, accuracy, and F-measure values of the predicted response variable (Precipitation) for our proposed supervised (ICR) framework against that of multiple linear regression (MLR) and SVM-MLR (A model that combines MLR and SVM). All the experiments were performed using a training size ( $n$ ) of 3 years starting from the first observation in the time series. The test set size ( $m$ ) was also fixed at 1 year. After calculating the RMSE on the test set, the training set was shifted by 3 years, such that it now occupied the data set used for testing in the previous iteration.

The experiment is repeated 5 times for each station. The RMSE values reported in this section is the mean value of all 5 iterations. The same approach is used to compute the RMSE values for **Rain** days, accuracy (for all days), F-measure for **Rain** days only and F-measure for **NoRain** days only. We present the results for 37 weather stations when ICR is compared with both MLR and SVM-MLR. Classification accuracy, and F-measures related to classification accuracy of MLR is not plotted on account of MLR not having an explicit classifier.

As shown in Figures 6 and 7, our supervised model, ICR significantly outperformed the MLR model (trained on all days) and the SVM-MLR model in terms of their RMSE values for predicting both **Rain** and **NoRain** days.

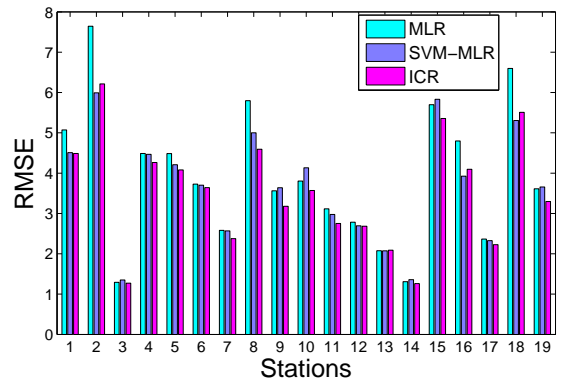


Figure 6: Comparison of RMSE values (for all days) among MLR, SVM-MLR and ICR.

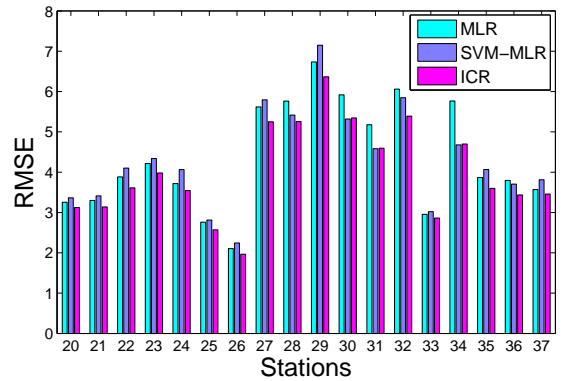


Figure 7: Comparison of RMSE values (for all days) among MLR, SVM-MLR and ICR.

ICR outperformed MLR in 36 out of 37 stations and outperformed SVM-MLR in 30 out of the 37 stations. In terms of percentage improvement in RMSE for all days,



ICR indicated an average 8% improvement over MLR and 5.8% improvement when compared to SVM-MLR.

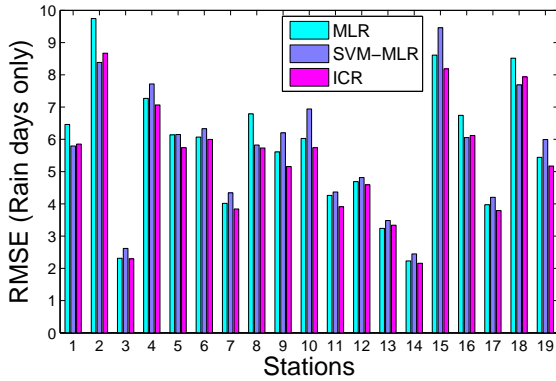


Figure 8: Comparison of RMSE values (for Rain days) among MLR, SVM-MLR and ICR.

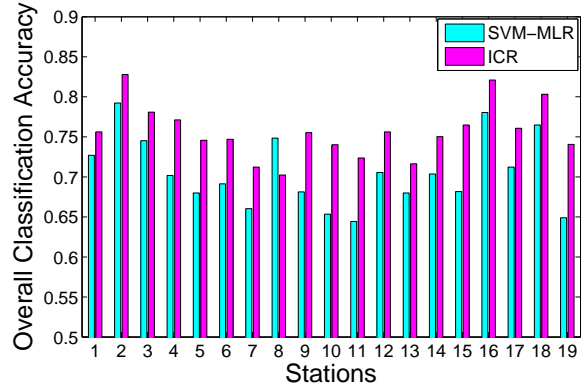


Figure 10: Comparison of classification accuracy (for all days) between SVM-MLR and ICR.

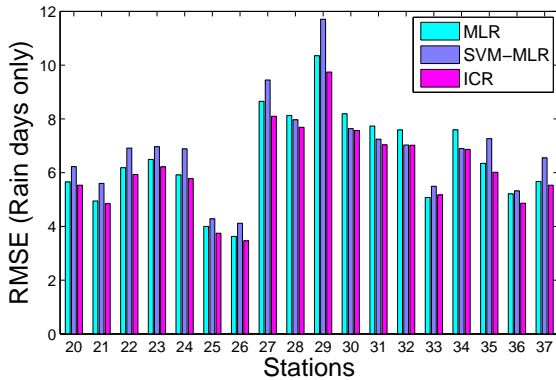


Figure 9: Comparison of RMSE values (for Rain days) among MLR, SVM-MLR and ICR.

In terms of the RMSE values for Rain days only, as shown in Figures 8 and 9, ICR consistently outperformed both the MLR and SVM-MLR model with ICR outperforming MLR in 35 stations and ICR outperforming SVM-MLR in 33 stations. When evaluating average RMSE value for Rain days only, we found that ICR had an improvement of 5.3% over MLR and 8.6% over SVM-MLR.

MLR does not inherently classify any days as Rain or NoRain. Hence, we did not plot a comparison between ICR and MLR with regards to classification accuracy and F-Measure.

As shown in Figures 10 and 11, ICR outperformed SVM-MLR in 36 of the 37 stations and showed a 9.1% improvement in classification accuracy. At the same time, in terms of F-measure for Rain days, our

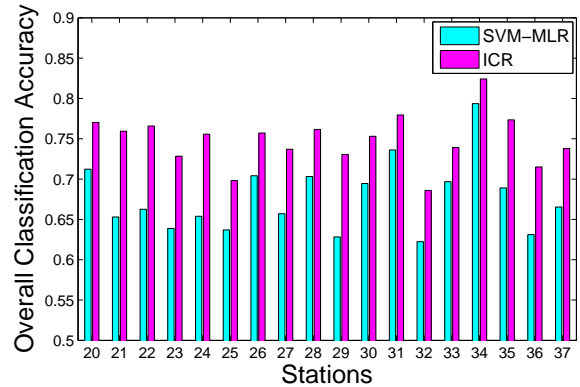


Figure 11: Comparison of classification accuracy (for all days) between SVM-MLR and ICR.

model outperformed SVM-MLR, as shown in Figures 12, 13. ICR outperformed SVM-MLR in 35 out of the 37 stations.

Although, MLR does not inherently classify any days as Rain or NoRain, we trained the Quadratic Discriminant Analysis(QDA) classifier mentioned earlier on the MLR output. ICR witnessed a 21.2% improvement in overall classification accuracy.

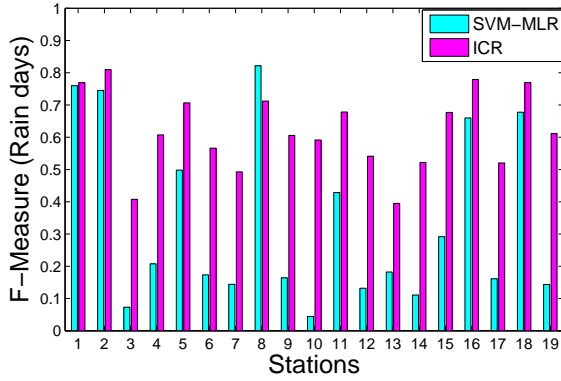


Figure 12: Comparison of F-Measure (for Rain days) between SVM-MLR and ICR.

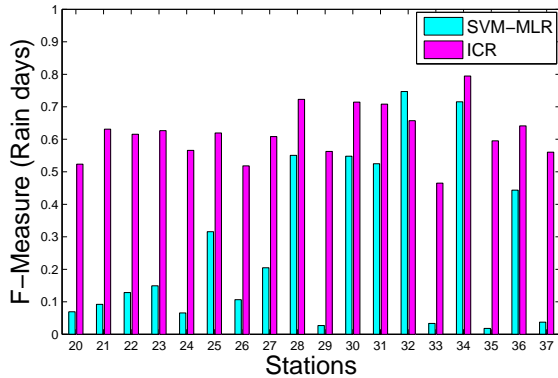


Figure 13: Comparison of F-Measure (for Rain days) between SVM-MLR and ICR.

With regard to F-measure for NoRain days, ICR outperformed SVM-MLR, in 36 stations. As shown in Figures 14,15 that shows the comparison of F-Measure for NoRain days between SVM-MLR and ICR, ICR outperformed SVM-MLR in all but one station and witnessed an 8.1% improvement in F-measure results.

## 7 Conclusions

This paper presents a novel approach for predicting future values of a time series data that are inherently

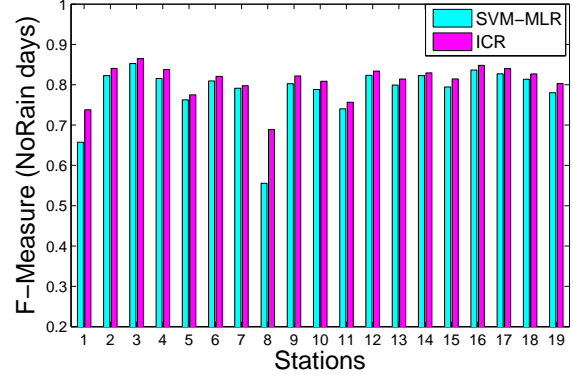


Figure 14: Comparison of F-Measure (for NoRain days) between SVM-MLR and ICR.

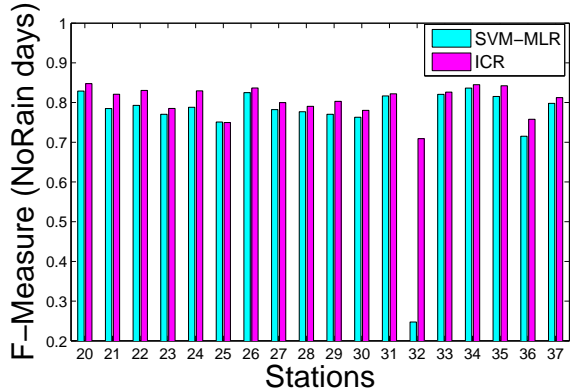


Figure 15: Comparison of F-Measure (for NoRain days) between SVM-MLR and ICR.

zero-inflated. The proposed framework decouples the prediction task into two steps—a classification step to predict whether the value of the time series is zero and a regression step to estimate the magnitude of the non-zero time series value. The effectiveness of the model was demonstrated on climate data to predict the amount of precipitation at a given station.

The framework presented in this paper assumes a linear relationship between the predictor and response variables. For future work, we plan to extend the framework to capture nonlinear relationships via the use of kernel functions and experiment with better parameter selection. The framework can also be extended to a semi-supervised learning setting.

## 8 Acknowledgments

This work is partially supported by NSF grant III-0712987 and subcontract for NASA award NNX09AL60G. The authors would like to thank Dr Julie Winkler and Dr Sharon Zhong for valuable discussion on statistical downscaling for climate change projection.

## References

- [1] Canadian Climate Change Scenarios Network, Environment Canada. <http://www.ccsn.ca/>.
- [2] S. Ancelet, M.-P. Etienne, H. Benot, and E. Parent. Modelling spatial zero-inflated continuous data with an exponentially compound poisson process. *Environmental and Ecological Statistics*, DOI:10.1007/s10651-009-0111-6, April 2009.
- [3] Z. Abraham and P.N. Tan. A Semi-Supervised Framework for Simultaneous Classification and Regression of Zero-Inflated Time Series Data with Application to Precipitation Prediction. In *Proceedings of International Workshop on Spatial and Spatiotemporal Data Mining (SSTD-09)*, Miami, Florida, 2009.
- [4] S. Barry and A. H. Welsh. Generalized additive modelling and zero inflated count data. *Ecological Modelling*, 157(2-3):179–188, November 2002.
- [5] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proc. of the 18th Int'l Conf. on Machine Learning*, pages 19–26, 2001.
- [6] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. of the Workshop on Computational Learning Theory*, pages 92–100, 1998.
- [7] D. Bohning, E. Dierz, and P. Schlattmann. Zero-inflated count models and their applications in public health and social science. In J. Rost and R. Langeheine, editors, *Applications of Latent Trait and Latent Class Models in the Social Sciences*. Waxman Publishing Co, 1997.
- [8] Y.-A. L. Borgne, S. Santini, and G. Bontempi. Adaptive model selection for time series prediction in wireless sensor networks. *Signal Process*, 87(12):3010–3020, 2007.
- [9] U. Brefeld, T. Gärtner, T. Scheffer, and S. Wrobel. Efficient co-regularised least squares regression. In *Proc. of the 23rd Int'l Conf. on Machine Learning*, pages 137–144, 2006.
- [10] S. Charles, B. Bates, I. Smith, and J. Hughes. Statistical downscaling of daily precipitation from observed and modelled atmospheric fields. In *Hydrological Processes*, pages 1373–1394, 2004.
- [11] H. Cheng and P.-N. Tan. Semi-supervised learning with data calibration for long-term time series forecasting. In *Proc of the ACM SIGKDD Int'l Conf on Data Mining*, Las Vegas, NV, 2008.
- [12] I. Cohen, N. Sebe, F. G. Cozman, M. C. Cirelo, and T. S. Huang. Semi-supervised learning of classifiers: Theory and algorithms for bayesian network classifiers and applications to human-computer interaction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(12):1553–1566, Dec 2004.
- [13] C. Cortes and M. Mohri. On transductive regression. In *Advances in Neural Information Processing Systems*, 2006.
- [14] F. Cozman and I. Cohen. Unlabeled data can degrade classification performance of generative classifiers. In *Proc. of the 15th Int'l Florida Artificial Intelligence Society Conference*, pages 327–331, 2002.
- [15] F. Cozman, I. Cohen, and M. Cirelo. Semi-supervised learning of mixture models. In *Proc of the 20th Int'l Conf. on Machine Learning*, 2003.
- [16] W. Enke and A. Spekat. Downscaling climate model outputs into local and regional weather elements by classification and regression. In *Climate Research 8*, pages 195–207, 1997.
- [17] D. Erdman, L. Jackson, and A. Sinko. Zero-inflated poisson and zero-inflated negative binomial models using the countreg procedure. In *SAS Global Forum 2008*, pages 1–11, 2008.
- [18] K. Fujinaga, M. Nakai, H. Shimodaira, and S. Sagayama. Multiple-regression hidden markov model. In *Proc. of IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing*, 2001.
- [19] C. Giles, S. Lawrence, and A. Tsoi. Noisy time series prediction using a recurrent neural network and grammatical inference. *Machine Learning*, 44(1-2), pages 161–183, 2001.
- [20] W. Hong, P. Pai, S. Yang, and R. Theng. Highway traffic forecasting by support vector regression model with tabu search algorithms. In *Proc. of Int'l Joint Conf. on Neural Networks*, pages 1617–1621, 2006.
- [21] T. Joachims. Transductive inference for text classification using support vector machines. In *Proc. of the 16th Int'l Conf. on Machine Learning*, pages 200–209, Bled, SL, 1999.
- [22] B. Kedem and K. Fokianos. Regression models for time series analysis. *Wiley-Interscience ISBN: 0-471-36355*,

2002.

- [23] T. Martin, B. Wintle, J. Rhodes, P. Kuhnert, S. Field, S. Low-Choy, A. Tyre, and H. Possingham. Zero tolerance ecology: improving ecological inference by modelling the source of zero observations. *Ecology Letters*, 8:1235–1246, 2005.
- [24] A. Ober-Sundermeier and H. Zackor. Prediction of congestion due to road works on freeways. In *Proc. of IEEE Intelligent Transportation Systems*, pages 240–244, 2001.
- [25] A. Smola and B. Scholkopf. A tutorial on support vector regression. In *Statistics and Computing*, pages 199–222(24). Springer, 2004.
- [26] Q. Tian, J. Yu, Q. Xue, and N. Sebe. A new analysis of the value of unlabeled data in semi-supervised learning for image retrieval. In *Proc. of IEEE Int'l Conf. on Multimedia and Expo.*, pages 1019– 1022, 2004.
- [27] L. Wei and E. J. Keogh. Semi-supervised time series classification. In *Proc of ACM SIGKDD Int'l Conf on Data Mining*, pages 748–753, Philadelphia, PA, August 2006.
- [28] A. H. Welsh, R. Cunningham, C. Donnelly, and D. B. Lindenmayer. Modelling the abundance of rare species: statistical models for counts with extra zeros. In *Ecological Modelling*. Elsevier, Amsterdam, PAYS-BAS (1975) (Revue), 1996.
- [29] R. Wilby, S. Charles, E. Zorita, B. Timbal, P. Whetton, and L. Mearns. Guidelines for use of climate scenarios developed from statistical downscaling methods. Available from the DDC of IPCC TGCIA, 2004.
- [30] R. L. Wilby. Statistical downscaling of daily precipitation using daily airflow and seasonal teleconnection. In *Climate Research*. 10, pages 163-178, 1998.
- [31] C.-C. Wong, M.-C. Chan, and C.-C. Lam. Financial time series forecasting by neural network using conjugate gradient learning algorithm and multiple linear regression weight initialization. Technical Report 61, Society for Computational Economics, Jul 2000.
- [32] T. Zhang. The value of unlabeled data for classification problems. In *Proc of the Int'l Conf. on Machine Learning*, 2000.
- [33] Z. Zhou and M. Li. Semi-supervised regression with co-training. In *Proc. of Int'l Joint Conf. on Artificial Intelligence*, 2005.
- [34] X. Zhu. Semi-supervised learning literature survey. In *Technical Report, Computer Sciences, University of Wisconsin-Madison*, 2005.
- [35] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. of the 20th Int'l Conf. on Machine Learning*, volume 20, 2003.
- [36] X. Zhu and A. Goldberg. Kernel regression with order preferences. In *Association for the Advancement of Artificial Intelligence*, page 681, 2007.