A Time-Dependent Enhanced Support Vector Machine For Time Series Regression

Goce Ristanoski
NICTA Victoria Laboratory
The University of Melbourne
Melbourne, Australia
g.ristanoski@student.unimelb.edu.au

Wei Liu
NICTA ATP Laboratory
The University of Melbourne
Melbourne, Australia
wei.liu@nicta.com.au

James Bailey
NICTA Victoria Laboratory
The University of Melbourne
Melbourne, Australia
baileyj@unimelb.edu.au

ABSTRACT

Support Vector Machines (SVMs) are a leading tool in machine learning and have been used with considerable success for the task of time series forecasting. However, a key challenge when using SVMs for time series is the question of how to deeply integrate time elements into the learning process. To address this challenge, we investigated the distribution of errors in the forecasts delivered by standard SVMs. Once we identified the samples that produced the largest errors, we observed their correlation with distribution shifts that occur in the time series. This motivated us to propose a time-dependent loss function which allows the inclusion of the information about the distribution shifts in the series directly into the SVM learning process. We present experimental results which indicate that using a time-dependent loss function is highly promising, reducing the overall variance of the errors, as well as delivering more accurate predictions.

Categories and Subject Descriptors
H.2.8 [**Information Systems**]: Database Application - Data Mining

Keywords
Time Series, Support Vector Machine, Loss Function.

1. INTRODUCTION

Time series prediction is a classic machine learning task. In our setting, a set of univariate training samples $x_1, \ldots, x_n$ ordered in time are provided and the task is to learn a model for predicting future values. We will consider a regression setting, where all sample values are continuous. A typical approach for learning involves using the most recent n values of the series $x_n, x_{n-1}, x_{n-2}, \ldots x_1$, to learn a model to forecast the future t values of the series: $x_{n+1}, x_{n+2}, \ldots x_{n+t}$. A series can range from very frequent measurements(stock market values taken at a 5 minutes interval) to less frequent and may span a larger period of time(quarterly or yearly reports).

Depending on the value of t and the nature of the time series being learned, we can differentiate between short-term forecasting, which concentrates on forecasting the very next samples following in the time series when they can be confidently described by only using the last samples without any additional knowledge or external variables influence; intermediate-term forecasting which attempts to forecast beyond the very next samples; and long-term forecasting, which besides using a quantitative model requires qualitative analysis and expert opinion. In the time series investigated in this paper, we focus on short term to intermediate-term forecasting.

Many machine learning models have been used for time series prediction, such as Linear Regression, Robust Regression, Gaussian Processes, Neural Networks and Markov models. Support Vector Machines have been used with considerable success for time series forecasting and are often able to outperform other methods. However, they may still learn sub-optimal models, in the presence of challenging aspects such as nonstationarity and volatility, noise, distribution changes and shifts.

Developing a clean and simple way to incorporate the time element into the learning process for SVM regression is the focus of our work. *Our approach is based on the following key insight:* across the samples, there is a correlation between the magnitude of the prediction error and the magnitude of the distribution shift(Figure 1). The samples where high prediction error occurs, tend to be samples where a large amount of shift in the series has occurred (and vice versa). Based on this simple observation, we propose a time sensitive loss function that modifies the learning process to target the samples with large distribution shift, in order to reduce their prediction error. The resulting SVM is not only able to produce more accurate forecasts, but also produce forecasts which are more stable and have lower variance in error. Our main contributions in this paper are as follows:

- We analyse the nature of predicted errors and discover *that the error often arises due to samples which are not outliers, but which contain with important information about the dynamics of the series.*
- We show the correlation between samples with large prediction error and samples with distribution shift, thus identifying an opportunity for an effective methodology of targeting these samples.

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.

- We introduce a time-dependent loss function that incorporates this distribution shift information in the form of a time sensitive loss into the SVM regression learning process.

2. RELATED WORK

We categorise related work into two sub-areas: methods for time series prediction and methods for detecting change in time series.

2.1 Machine learning methods for time series analysis

Extensive research in the area of machine learning models for time series analysis has been conducted in recent years [1, 14]. Hidden Markov Models [13] and Artificial Neural Networks [17, 11] have been adapted to use additional information from the time series in their learning. Linear Regression, and its advanced version Robust Regression offer interpretable models with satisfactory performance [19]. Clustering has proven to be very beneficial as well [9, 12]. Feature selection process [30], Independent Component Analysis [8], time series segmentation [28] and motif discovery [25, 20] are other popular methods for time series analysis.

On the econometric side, models for time series analysis, such as the Autoregressive Integrated Moving Average (ARIMA) models have been proposed and balance complexity with performance. A large variety of more complex extensions also exist, such as the (General) AutoRegressive Conditional Heteroskedasticity ((G)ARCH) [2]. A possible drawback of these approaches is that significant user insight and domain knowledge may be required to achieve good results and their performance may not be as strong when used 'out of the box'.

Support Vector Machines have been widely used in practice due to their generally strong performance, and much research has been conducted to further improve them in several directions, including time series analysis. Use of SVMs has been investigated for non-stationary time series modelling [5] and volatile series analysis [31]. A modification of SVMs that uses dynamic parameters for the purpose of time series datasets analysis and forecasting has also been suggested [3].

2.2 Distribution shift and event detection

The temporal continuity of the samples in time series datasets is an important aspect and a core analysis task is detection of distribution changes or shifts in the series. These changes or shifts that occur over time in the series have been extensively researched (e.g. [22, 4, 18, 15]) in order to discover the effect of all the events whose information is concealed in the series [16], especially ones classified as anomalies [23, 7] or noise, which can complicate the learning process.

3. TIME-DEPENDENT LOSS FUNCTION

We propose an algorithm that has two phases: (1) detection of large-error-producing samples in the series, and (2) targeted minimization of the loss at the large-error-producing samples by using a time-dependent loss function. We explain each phase in turn.

3.1 Large-error-producing sample detection

It is well known that distribution change or shift within a time series can influence and complicate the learning process. We next make a crucial observation about distribution shift and the forecasted error using standard SVM, which is a core motivation in our later development.

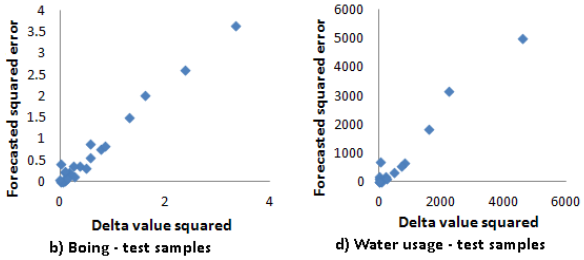

Figure 1: Delta values ($\triangle x_t^2 = (x_t - x_{t-1})^2$) and squared errors for "Boing stock market value" and "Annual water usage in New York" time series test samples, forecasted using standard polynomial SVM. We can observe an almost linear correlation between the delta values and squared errors, but also we notice that most of the errors are clustered around the close to zero values, and a few stand out in error size as the delta values increase.

Consider Figure 1 . For two separate datasets an SVM regression model has been trained and then evaluated using test data (each sample at time t is forecasted based on the past four time series values). For each test sample, the squared error is shown on the y axis and the x-axis shows the amount of "shift" ($\triangle x_t^2 = (x_t - x_{t-1})^2$) undergone by the sample (the difference in water usage compared to the previous sample, or the difference in stock value compared to the previous sample).

We observe that there is strong correlation between magnitude of prediction error and magnitude of shift (Pearson correlation coefficient: Boing stock market value series = 0.99; Annual water usage in New York series = 0.98). In particular, the samples for which the squared prediction error is high, are samples for which high distribution shift has occurred.

We can also observe a Pareto-like principle seems to operate for the prediction error: 80% of the error originates from roughly 20% of the samples. This is highlighted in Figure 2, where for each dataset the test samples are sorted according to prediction error. The trends in Figures 1 and 2 are also similar for samples from the training set.

This suggests the following strategy: if we additionally target the samples with high distribution shift in the learning process, can this produce a model with overall lower prediction error?

In order to attempt this, we need a formal test for whether a sample is a "high distribution shift sample". We choose to do this by analysing the mean and standard deviation of the features describing it, which correspond to the past samples in the time series. For a given sample $\mathbf{x}_t$ at time t and dimensionality d(number of past values used to forecast the sample), let m_d be the calculated mean of the d preceding samples $(\mathbf{x}_{t-p}, \mathbf{x}_{t-d+1}....\mathbf{x}_{t-1})$ and let s_d be the calculated standard deviation. We define the range $m_d \pm k * s_d$ as the range in which we expect to find the value of the sample at

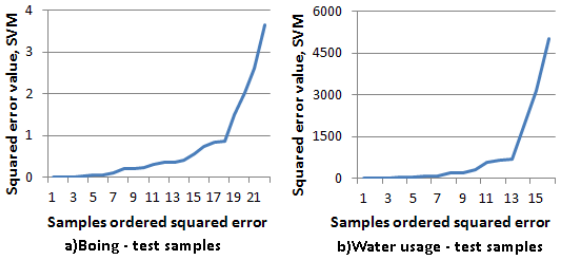

Figure 2: Ordered squared values of the errors for "Boing stock market value" and "Annual water usage in New York" time series test sets forecasted using standard polynomial SVM.

time t - if the sample at time t is not in this range, it can be considered as a (high) distribution shift sample. k here is a parameter that needs to be chosen. Formally, distribution shift for $\mathbf{x}_t$ is calculated as:

$$\mathbf{x}_t \begin{cases} \mathbf{x}_t \in [m_d - k * s_d, m_d + k * s_d], & \text{true} \\ \mathbf{x}_t \notin [m_d - k * s_d, m_d + k * s_d], & \text{false} \end{cases} \quad (1)$$

As mentioned, by applying Equation 1 and using the intuition from Figure 1, we expect the set of samples for which the high distribution shift test is true, to substantially overlap with the set of samples which have high prediction error when (standard) SVM regression is used. Of importance here, is choice of value for the parameter k. We set d equal to the dataset's dimensionality. As $k \to \infty$, no samples will be labelled as distribution shift samples. Our initial experiments indicate a value of k=2 is a good default choice. We will discuss this issue further in the experiments section.

3.2 Outliers vs distribution shift

The set of high prediction error samples might at first be considered as a set of outliers, if we neglect the observation that it is also a set of high distribution shift samples as well. Outlier detection is not an unknown problem in the time series forecasting area [24], and the same can be said for discovering distribution changes and shifts [27, 6]. However, understanding the nature of the large error in high prediction error samples and determining the correlation with high distribution shift samples is crucial to whether or not we should treat these samples as outliers. Looking at the types of outlier that these samples might be classified as, we can comment on the difference between them:

- Additive Outliers (AO) are a type of outliers that affect only single observation [27]. AO is an outlier in which the observation clearly differs from the surrounding samples. In the case of high distribution shift samples, there exists some difference in the distribution with the preceding samples, but not with the following samples.
- Innovation Outliers (IO) are a type of outlier that affect the subsequent samples, starting from the position where the outlier is introduced [27]. We observed the following samples were similar to the high distribution shift sample, and were not affected by the outlier as they follow similar distribution and did not produce large errors, so we cannot classify them as IO.
- Level Shift Outliers (LSO) are the type of outlier when there is a sudden extreme change in the distribution of the process and the change is permanent [27]. This means that until time t_s, the output is explained as $y_t = \mathbf{w}\mathbf{x}_t + b$, and from that point onwards(for $t \geq t_s$) as $y_t = \mathbf{w}\mathbf{x}_t + b + M(t)$, where $M(t)$ is a process(stationary or non-stationary) that caused the level shift. This means the change in the process is significant enough to build a separate model from that point onwards, as these changes occur with a high intensity and less frequently, unlike the distribution shift which is a more continuous process with mild changes that do not require the need from more than one model.

One potential approach might be to remove high distribution shift samples from the time series altogether, for example if we classify the high distribution shift samples as AO or IO. This would not necessarily help, since it would mean that the samples immediately following the AO/IO sample, which have similar distribution, would then likely be labelled as AO/IO samples as well and the problem would remain. Also, removal might result in important information being lost to the training process.

If we consider that case of LSO, we have several methods of dealing with the level shift: if the number of shifts is very small, even an ARIMA model can be adopted, if it satisfies the accuracy criteria of the users. Another approach would be to see if the set of samples is cohesive enough to be learned by using one model or we should keep removing from the samples until we have no more distribution shifts. This will result in removing most of the samples in the series. This conclusion is confirmed the analysis of the occurrence of the distribution shift through the series which we conducted for several datasets, shown in Table 1. In the analysis, each time series dataset has been divided into quarters and the number of distribution shift samples per quarter was counted. The continuous occurrence of distribution shifts, which is rather uniform, further confirms the difference with LSO.

Table 1: Placement of the distribution shift samples detected in the training sets per quarter, in %. We can observe that on average each quarter has around 25% of the detected distribution shift samples of the training set.

Dataset	Q1	Q2	Q3	Q4
Apple	27.4	30.2	22.6	19.8
Coke	35.9	20.3	22.4	21.4
Johnson & Johnson	36.5	21.8	18.9	22.8
Earthquakes	6.7	26.7	13.3	53.3
Employment	25.4	23.6	25.5	25.5
Average	26.4	24.5	20.5	28.6

A more analytical approach would be to model the new process $M(t)$ starting from t_s, for example $y_t = \mathbf{w}\mathbf{x}_t + b + \mathbf{w}_1 J_1(t)\mathbf{x}_t$, where $J_1(t) = 0$, for $t < t_s, J_1(t) = 1$, for $t \geq t_s$. The starting position of the level shift at t_s for J is determined by using conditional probability of t_s being a level shift point, along with the forecasted error at time t_s if we use the all the previous samples in order to build a model. It is apparent that in the case of j level shifts detected, the final model has the form of $y_t = \mathbf{w}\mathbf{x}_t + b + \sum_{i=1}^{j} \mathbf{w}_i J_i(t)\mathbf{x}_t$,

where $J_i(t) = 0$, for $t < t_i, J_i(t) = 1$, for $t \geq t_i$, t_i being the point in time the i-th level shift is detected. Even for low values of j this model is likely to become complex and difficult to interpret.

Instead, the intuition behind our approach will be to change the loss function used in SVM regression, to place special emphasis on distribution shift samples and their immediate successors. The distribution shift is not treated as an outlier, but instead, as useful information that we can incorporate into an enhanced SVM regression algorithm. Details are described next.

3.3 SVM Regression time-dependent empirical loss

Let us consider the class of machine learning methods that address the learning process as finding the minimum of the regularized risk function. Given n training samples $(\mathbf{x}_i,y_i)$ $(i$=1,..., n), $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector of the i–th training sample, d is number of features, and $y_i \in \mathbb{R}$ is the value we are trying to predict, the regularized risk function will have the form of:

$$L(\mathbf{w}^*) = argmin_{\mathbf{w}}\ \phi \mathbf{w}^T \mathbf{w} + R_{emp}(\mathbf{w}) \tag{2}$$

with $\mathbf{w}$ as the weight vector, ϕ is a positive parameter that determines the influence of the structural error in Equation 2, and $R_{emp}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} l(\mathbf{x}_i, y_i, \mathbf{w})$ is the loss function with $l(\mathbf{x}_i,y_i,\mathbf{w})$ as a measure of the distance between a true label y_i and the predicted label from the forecasting done using $\mathbf{w}$. The goal is now to minimize the loss function $L(\mathbf{w}^*)$, and for Support Vector Regression, this has the form of

$$L(\mathbf{w}^*) = \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_{i=1}^{n} (\xi_i^+ + \xi_i^-) \tag{3}$$

subject to

$$\begin{cases} \langle w, x_i \rangle + b + \epsilon + \xi^+ \geq y_i \\ \langle w, x_i \rangle + b - \epsilon - \xi^- \leq y_i \end{cases} \tag{4}$$

with b being the bias term, ξ_i^+ and ξ_i^- as slack variables to tolerate infeasible constraints in the optimization problem(for soft margin SVM), C is a constant that determines the trade-off between the slack variable penalty and the size of the margin, and ϵ being the tolerated level of error.

The Support Vector Regression empirical loss for a sample x_i with output y_i is $l_1(\mathrm{x_i},\mathrm{y_i},\mathbf{w})$=max(0, $|\mathbf{w}^T \mathrm{x_i} - \mathrm{y_i}| - \epsilon)$, as shown in Table 2. Each sample contribution to the loss is independent from the other samples contribution, and all the samples are considered to be of same importance in terms of information they possess.

The learning framework we aim to develop should be capable of reducing the difference in error at selected samples. The samples we focus on are samples where a distribution shift is detected, as these samples are expected to be large-error-producing samples. An example is shown in Figure 3, which displays some large spikes in prediction error (and these coincide with high distribution shift). Instead, we would prefer smoother variation in prediction error across time (shown by the dotted line). Some expected benefits of reducing (smoothing) the difference in error for successive predictions are:

- Reduced impact of the distribution shift reflected as a sudden increase of the error - models that produce volatile errors are not suitable in scenarios where some form of cost might be associated with our prediction, or the presence of an uncertain factor may undermine the model decision.

- Reduced variance of the overall errors - Tighter confidence intervals for our predictions provides more confidence in the use of those predictions.

- Avoiding misclassification of samples as outliers - A sample producing large error can be easily considered an outlier, so in the case of a distribution shift sample, preventing the removal of these samples ensures we have retained useful information.

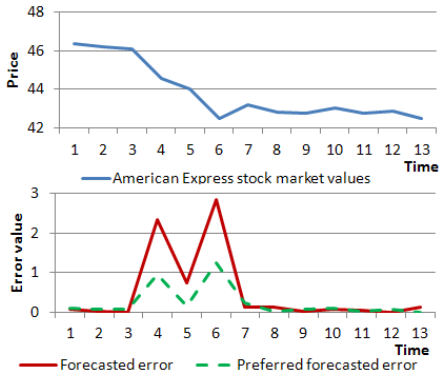

Figure 3: Actual price for American Express stock market shares, with the forecasted error (from SVM) and preferred forecasted error (what we would prefer to achieve). The peaks in error size occur at samples where a distribution shift can be visually observed(samples 4-6) and these errors contribute significantly more to the overall error that the error being produced by the other samples forecasts.

A naive strategy to achieve smoothness in prediction error would be to create a loss function where higher penalty is given to samples with high distribution shift. This would be unlikely to work since a distribution shift is behaviour that is abnormal with respect to the preceding time window. Hence the features (samples from the preceding time window) used to describe the distribution shift sample will likely not contain any information that could be used to predict the distribution shift.

Instead, our strategy is to create a loss function which minimises the difference in prediction error for each distribution shift sample and its immediately following sample. We know from the preceding discussion, that for standard SVM regression the prediction error for a distribution shift sample is likely to be high and the prediction error for its immediately following sample is likely be low (since this following sample is not a distribution shift sample). By reducing the variation in prediction error between these successive samples, we expect a smoother variation in prediction error as time increases. We call this type of loss *time-dependent empirical loss.*

More formally, for a given sample x_i and the previous sample x_{i-1}, the time-dependent loss function

Table 2: Support Vector Regression and Time-Dependent Support Vector Regression empirical loss and its derivative

	Support Vector Regression	Time-Dependent Support Vector Regression
Empirical loss $l_1(\mathrm{x}_i,\mathrm{y}_i,\mathbf{w})$	$\max(0, \lvert\mathbf{w}^T\mathrm{x}_i - \mathrm{y}_i\rvert - \epsilon)$	$\max(0, \lvert\mathbf{w}^T\mathrm{x}_i - \mathrm{y}_i\rvert - \epsilon)$
Derivative $l_1'(\mathrm{x}_i,\mathrm{y}_i,\mathbf{w})$	$\mathrm{sign}(\mathbf{w}^T\mathrm{x}_i - \mathrm{y}_i)$ if $\lvert\mathbf{w}^T\mathrm{x}_i - \mathrm{y}_i\rvert > \epsilon$; otherwise 0	$\mathrm{sign}(\mathbf{w}^T\mathrm{x}_i - \mathrm{y}_i)$ if $\lvert\mathbf{w}^T\mathrm{x}_i - \mathrm{y}_i\rvert > \epsilon$; otherwise 0
Time-Dependent loss $l_2(\mathrm{x}_i,\mathrm{y}_i,\mathbf{w},\mathrm{x}_{i-1},\mathrm{y}_{i-1})$	0 for all samples	$\max(0, \lvert(\mathbf{w}^T\mathrm{x}_i - \mathrm{y}_i) - (\mathbf{w}^T\mathrm{x}_{i-1} - \mathrm{y}_{i-1})\rvert - \epsilon_t)$ for distribution shift samples; otherwise 0
Derivative $l_2'(\mathrm{x}_i,\mathrm{y}_i,\mathbf{w},\mathrm{x}_{i-1},\mathrm{y}_{i-1})$	0 for all samples	$\mathrm{sign}((\mathbf{w}^T\mathrm{x}_i - \mathrm{y}_i) - (\mathbf{w}^T\mathrm{x}_{i-1} - \mathrm{y}_{i-1}))$ if $\lvert(\mathbf{w}^T\mathrm{x}_i - \mathrm{y}_i) - (\mathbf{w}^T\mathrm{x}_{i-1} - \mathrm{y}_{i-1})\rvert > \epsilon_t$ for distribution shift samples, otherwise 0

$l_2(x_i, y_i, \mathbf{w}, x_{i-1}, y_{i-1})$ is

$$\begin{cases} \lvert\,(\mathbf{w}^T x_i - y_i) - (\mathbf{w}^T x_{i-1} - y_{i-1})\,\rvert & x_{i-1}\text{is dist. shift} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where it can be seen that if x_{i-1} is not a distribution shift sample, then no loss is incurred. Otherwise, the loss is equal to the amount of difference between the prediction error for x_i and the prediction error for x_{i-1}. We can further incorporate a tolerable error term ϵ_t to yield a soft time-dependent loss. This is shown in Table 2. By linearly combining our time-dependent loss function with the standard loss function for SVM regression, we formulate a new type of SVM, which we henceforth refer to as TiSe SVM (time sensitive SVM). *Observe that if no samples are classified as distribution shift samples, then a TiSE SVM is exactly the same as SVM.* Using the definitions from Table 2, the modified empirical loss and regularized risk function for TiSE SVM have the form

$$R_{emp}^{TiSe}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(l_1(\mathbf{x}_i, y_i, \mathbf{w}) + \lambda * l_2(\mathbf{x}_i, y_i, \mathbf{w}, \mathbf{x}_{i-1}, y_{i-1})) \tag{6}$$

$$L(\mathbf{w}^*) = argmin_{\mathbf{w}}\ \phi\mathbf{w}^T\mathbf{w} + R_{emp}^{TiSe}(\mathbf{w}) \tag{7}$$

The λ parameter is a regularization parameter that determines the extent to which we want to minimize the time-dependent loss function. Larger values of λ will result in the time-dependent loss function having more influence in the overall error. We investigate the effect of this parameter in the experimental section, in order to determine values suitable for our experimental work.

3.4 SVM Regression time-dependent loss optimization

The new SVM regression which incorporates the time-dependent loss will now have the following form:

$$L^{TiSe}(\mathbf{w}^*) = \frac{1}{2}||\mathbf{w}||^2 + C_1\sum_{i=1}^{n}(\xi_i^+ + \xi_i^-) + C_2\sum_{i=2}^{n}(\zeta_i^+ + \zeta_i^-) \tag{8}$$

subject to

$$\begin{cases} \langle w, x_i\rangle + b + \epsilon + \xi_i^+ \geq y_i \\ \langle w, x_i\rangle + b - \epsilon - \xi_i^- \leq y_i \\ (\langle w, x_i\rangle - y_i) - (\langle w, x_{i-1}\rangle - y_{i-1}) + \epsilon_t + \zeta_i^+ \geq 0 \\ (\langle w, x_i\rangle - y_i) - (\langle w, x_{i-1}\rangle - y_{i-1}) - \epsilon_t - \zeta_i^- \leq 0 \\ \xi_i^+, \xi_i^-, \zeta_i^+, \zeta_i^- \geq 0 \end{cases} \tag{9}$$

where ϵ_t is the allowed value of the differences in the time sensitive error, and ζ^+ and ζ^- are slack variables for the time sensitive loss. In order to solve this, we introduce Lagrange multipliers $\alpha_i^+ \geq 0$, $\alpha_i^- \geq 0$, $\mu_i^+ \geq 0$ and $\mu_i^- \geq 0$ for all i, β_i^+, β_i^-, η_i^+ and η_i^- for $i = 2 \ldots n$:

$$\begin{aligned} L_P =& C_1\sum_{i=1}^{n}(\xi_i^+ + \xi_i^-) + C_2\sum_{i=2}^{n}(\zeta_i^+ + \zeta_i^-) + \frac{1}{2}||\mathbf{w}||^2 \\ &- \sum_{i=1}^{n}(\mu_i^+\xi_i^+ + \mu_i^-\xi_i^-) - \sum_{i=1}^{n}\alpha_i^+(\epsilon + \xi_i^+ - y_i + \langle w, x_i\rangle + b) \\ &- \sum_{i=1}^{n}\alpha_i^-(\epsilon + \xi_i^- + y_i - \langle w, x_i\rangle - b) \\ &- \lambda\sum_{i=2}^{n}(\eta_i^+\zeta_i^+ + \eta_i^-\zeta_i^-) \\ &- \lambda\sum_{i=2}^{n}\beta_i^+((\langle w, x_i\rangle - y_i) - (\langle w, x_{i-1}\rangle - y_{i-1}) + \epsilon_t + \zeta^+) \\ &- \lambda\sum_{i=2}^{n}\beta_i^-(-(\langle w, x_i\rangle - y_i) + (\langle w, x_{i-1}\rangle - y_{i-1}) + \epsilon_t + \zeta^-) \end{aligned} \tag{10}$$

Differentiating with respect to $\mathbf{w}$, b, ξ_i^+, ξ_i^-, ζ_i^+ and ζ_i^- and setting the derivatives to 0, we will get the dual form:

$$\begin{aligned} L_D =& \sum_{i=1}^{n}(\alpha_i^+ - \alpha_i^-)y_i + \lambda\sum_{i=2}^{n}(\beta_i^+ - \beta_i^-)y_i \\ &- \lambda\sum_{i=2}^{n}(\beta_i^+ - \beta_i^-)y_{i-1} - \epsilon\sum_{i=1}^{n}(\alpha_i^+ + \alpha_i^-) - \lambda\epsilon_t\sum_{i=2}^{n}(\beta_i^+ + \beta_i^-) \\ &- \frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-)\mathbf{x}_i\mathbf{x}_j \\ &- \frac{1}{2}\lambda^2\sum_{i,j=2}^{n}(\beta_i^+ - \beta_i^-)(\beta_j^+ - \beta_j^-)(\mathbf{x}_i\mathbf{x}_j - 2\mathbf{x}_i\mathbf{x}_{j-1} + \mathbf{x}_{i-1}\mathbf{x}_{j-1}) \\ &- \lambda\sum_{i=1,j=2}^{n}(\alpha_i^+ - \alpha_i^-)(\beta_j^+ - \beta_j^-)(\mathbf{x}_i\mathbf{x}_j - \mathbf{x}_i\mathbf{x}_{j-1}) \end{aligned} \tag{11}$$

This form allows for a Quadratic Programming to be applied in order to find $\mathbf{w}$ and b. Also, it can be noticed that if we need to move to a higher dimensionality space $\mathbf{x} \to \psi(\mathbf{x})$, such that a kernel function exists $k(\mathbf{x}_i\mathbf{x}_j) = \psi(\mathbf{x}_i)\psi(\mathbf{x}_j)$, we can do so as L_D can be kernelized.

3.5 Quadratic mean for simultaneous loss optimisation

We are proposing simultaneous minimization of the errors for all samples (loss function l_1) and differences of errors for selected samples (loss function l_2). It is because of this reason that we are expecting the trade-off between these two losses to produce a model with smaller variance in the error, but which might incur an increase in the overall error. For this reason we investigate an alternative method using the quadratic mean, for adding our time-dependent loss, aiming to reduce any impact on the overall error.

The quadratic mean has been successfully applied in the case of imbalanced data to simultaneously optimize the loss of two non-overlapping groups of samples [21]. The quadratic mean is a lower bound for the arithmetic mean, and the quadratic mean of two quantities implicitly considers the difference (variance) between their values, as well as their sum. Here we use it to combine two different loss functions calculated over all the samples, with the objective of ensuring that both loss functions are minimized while still minimizing the overall loss as well. The final quadratic empirical loss $R_{emp}^{TiSe-Q}(\mathbf{w})$ and regularized risk function will have the form of

$$R_{emp}^{TiSe-Q}(\mathbf{w}) = \sqrt{\frac{(\sum_{i=1}^{n} l_1(i))^2 + \lambda(\sum_{i=2}^{n} l_2(i))^2}{1+\lambda}} \quad (12)$$

$$L(\mathbf{w}^*) = argmin_{\mathbf{w}}\ \phi \mathbf{w}^T \mathbf{w} + R_{emp}^{TiSe-Q}(\mathbf{w}) \quad (13)$$

with $l_1(i)$=$l_1(\mathrm{x}_i,\mathrm{y}_i,\mathbf{w})$, $l_2(i)$=$l_2(\mathrm{x}_i,\mathrm{y}_i,\mathbf{w},\mathrm{x}_{i-1},\mathrm{y}_{i-1})$ as shown in Table 2. This new time-dependent loss for SVM is a quadratic mean version of our time series SVM, named TiSe-Q SVM. Both l_1 and l_2 are convex functions, and quadratic mean has the feature of producing the resulting function as a convex function.

The quadratic mean time-dependent loss function has been derived in Primal form, so for both linear and quadratic mean time-dependent loss function versions, a linear optimization method, such as the bundle method [26], is an effective way to optimize our new time-dependent loss function: by calculating the subgradients of the empirical loss and time-dependent loss, we can iteratively update w in a direction that minimizes the quadratic mean loss presented at Equation 12.

4. EXPERIMENTS AND RESULTS

Evaluation of the performance in terms of error reduction and error variance reduction of TiSe SVM and TiSe-Q SVM was the main target of the experimental work we conducted. To achieve this goal in the experiments both real datasets and synthetic datasets were used. We tested 35 time series datasets obtained from [29] and [10], consisting of stock market values, chemical and physics phenomenon measurements.

We also created a set of 5 different versions of a synthetic dataset with different levels of added distribution shift: 1 distribution shift free dataset, 2 datasets with distribution shift added to random 10% of all the samples, and 2 more with distribution shift added to 25% of the samples. The sizes of all the datasets are between 100 and 600 samples, divided on training set and test set of around 10-15%, resulting in our forecasting task being classified as short to intermediate-term forecasting.

The effect of the regularization parameters ϕ and λ was investigated by dividing the training set of several datasets into initial training and validation sets. We tested with values of λ in the in the range of 0.001 to 0.01 for TiSe SVM and 0.01 rising to 0.2 for TiSe-Q (Figure 4), and with ϕ in the range of 1E-6 to 1E-3(increasing the range was terminated when no continuous improvements in results were made). A greater range of values for λ was chosen for TiSe-Q as we wished to investigate the effect of the quadratic mean and the limit to which we can minimize the time-dependent loss without affecting the minimization of the empirical loss in a negative way.

Our initial investigation was conducted on several datasets by splitting the training set into initial training set (the first 95% of the samples) and validation set(last 5% of the samples). The results of the ranges of values for ϕ and λ indicated that values of ϕ=5E-6, λ = 0.005 for TiSe SVM and λ = 0.05 for TiSe-Q SVM were good defaults for all of the files, and these values were used in all of the experiments. Using different values for each dataset testing would be a better approach when conducting work on individual datasets, but as with any new method, some form of good default values of the parameters, if such exist, needed to be determined.

We adopted the same approach with initial training and validation set and testing over a range of values for determining the best parameters for the baseline methods as well. The final values which produced best performance on the validation sets are presented later in the description of each method accordingly.

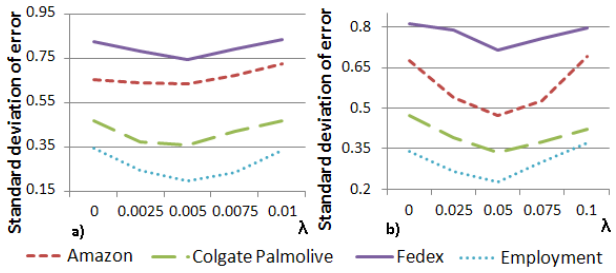

Figure 4: Standard deviation of the forecasted error for different choices of lambda for TiSe and TiSe-Q for validation sets of several datasets. Starting with λ=0, adding the time-dependent loss in the minimization process leads to lower standard deviation, but as λ increases, the minimization of the empirical loss is reduced to the extent that the forecasted errors are becoming large and volatile again.

With regard to the value for the parameter k (which is used to classify whether a sample has undergone distribution shift), we use a value of 2 in all our experiments, testing for high distribution shift in the range of $m_d \pm 2 * s_d$. A specific example of varying k is shown in Figure 5 for the "Walt Disney market values" dataset.

4.1 Testing and results

Comparison of TiSe SVM(ϕ = 1, ϵ=0.001, k=2, λ=0.005, ϵ_t=1E-8) and TiSe-Q SVM(ϕ = 1, ϵ=0.001, k=2, λ=0.05, ϵ_t=1E-8) with 6 methods was conducted in order to evaluate the effect of the TiSe SVM and TiSe-Q SVM methodology: ARIMA(3,0,1), Neural Network (NN, learning rate=0.3,

Table 3: Root Mean Square Error (RMSE), Error Reduction (ER, in %) and Error Standard Deviation Reduction(SDR, in %) of TiSe and TiSe-Q compared to SVM. Both TiSe and TiSe-Q achieved significant SDR reduction, though RMSE increased in several cases for TiSe. In bold is the method with the lowest RMSE.

	RMSE								ER(in %)		SDR(in %)	
Dataset	ARIMA	NN	KNN	SVM	RBF	RR	TiSe	TiSe-Q	TiSe	TiSe-Q	TiSe	TiSe-Q
Imports	24E3	14E4	16E3	130E3	**129E3**	132E3	133E3	130E3	-2.39	-0.18	16.45	13.7
Starbucks	1.627	1.029	1.43	0.705	0.855	0.705	0.738	**0.698**	-4.68	0.99	1.51	34.55
Amazon.com	1.853	1.641	2.66	0.623	0.863	0.625	0.631	**0.614**	-1.28	1.44	2.67	34.67
British Airways	**0.11**	0.129	0.139	0.126	**0.12**	0.129	0.132	0.124	-4.76	1.59	5.26	5.26
Apple	12.25	6.1	16.98	2.169	4.558	2.302	**2.105**	2.133	2.95	1.66	4.95	26.43
Fedex	1.276	1.183	1.745	0.725	0.89	0.73	0.731	**0.711**	-0.83	1.93	12.69	15.38
Johnson & Johnson	0.556	0.369	0.361	0.357	0.404	0.352	**0.341**	0.346	4.48	3.08	11.51	25.03
Chocolate	1986	1842	1852	1701	1880	1697	**1544**	1646	9.24	3.25	35.56	22.69
Ford	0.858	0.324	0.391	0.251	0.27	0.255	0.261	**0.242**	-3.98	3.59	5.19	29.2
IBM	19.25	**6.68**	9.97	7.425	11.66	7.577	7.294	7.145	1.76	3.77	6.52	14.77
Boeing	7.205	3.934	7.14	1.235	2.39	1.274	1.285	**1.185**	-4.05	4.05	18.18	20.59
Auto registrations	106.9	106.4	104.6	104.2	109.2	100.9	102.2	**99.99**	1.91	4.07	6.42	17.96
Earth's rotations	98	16.22	51.31	15.39	46.28	14.74	**14.49**	14.74	5.85	4.22	4.98	13.4
Radioactivity	**10.79**	12.17	11.5	11.499	10.96	12.015	11.698	10.912	-1.73	5.10	13.22	23.92
Simulated sample 5	7.659	8.235	8.21	6.309	7.17	6.449	6.153	**5.978**	2.47	5.25	7.93	15.74
Island Pacific	1.283	1.1	1.39	0.967	1.48	0.997	0.932	**0.916**	3.62	5.27	21.13	17.31
McDonalds	1.854	1.045	3.08	0.662	0.84	0.668	0.649	**0.627**	1.96	5.29	6.73	28.66
Simulated sample 4	13.051	2.549	6.68	1.973	4.1	2.162	1.962	**1.862**	0.56	5.63	14.36	7.87
Water usage	79.55	47.72	39.52	18.6	44.26	19.66	18.28	**17.55**	1.72	5.65	6.92	29.58
American Express	2.253	**0.65**	0.849	0.709	0.956	0.679	0.697	0.668	1.69	5.78	52.2	57.44
Microsoft	0.932	0.828	0.859	0.252	0.375	0.243	0.245	**0.237**	2.78	5.95	9.76	15.4
Walt Disney	3.199	1.161	3.35	0.552	0.87	0.542	**0.518**	**0.518**	6.16	6.16	15.36	35.51
AUD/USD exch.	0.0684	0.018	0.033	0.0175	0.029	0.0167	0.0165	**0.0164**	5.71	6.29	13.71	33.7
Hewlett-Packard	1.865	0.724	0.9	0.664	0.94	0.64	**0.619**	**0.619**	6.78	6.78	5.88	14.02
Colgate Plamolive	1.745	0.569	0.635	0.519	0.696	0.504	0.605	**0.483**	-16.5	6.94	21.23	28.87
Earthquakes	8.8	8.02	8	6.176	7.29	5.976	5.85	**5.738**	5.28	7.09	13.28	14.79
Tree	0.242	0.215	0.203	0.183	0.182	0.187	0.176	**0.17**	3.83	7.10	17.46	34.92
Intel	2.67	0.885	1.35	0.932	1.06	0.912	0.906	**0.856**	2.79	8.15	6.64	25.17
Coke	0.995	0.335	0.529	0.302	0.388	0.295	0.281	**0.277**	6.95	8.28	45.45	42.86
Temp. anomalies	19.628	15.27	18.35	14.834	13.98	14.996	13.952	**13.596**	5.95	8.35	36.08	18.88
Siemens	3.039	2.846	7.04	1.861	2.53	1.831	1.768	**1.704**	5.00	8.44	16.75	38.27
Ebay	1.7	0.583	0.945	0.443	0.516	0.446	0.44	**0.404**	0.68	8.80	7.14	15.04
Employment	3.29	2.29	4.28	0.54	1.56	0.53	0.53	**0.49**	1.85	9.26	43.14	45.71
Rhine	239	194	187	179	192	179	191	**161**	-6.70	10.06	14.84	45.9
Chemical process	0.364	0.234	0.295	0.221	0.28	0.231	0.203	**0.197**	8.14	10.86	10.14	10.64
Robberies	95.17	84.45	91.86	70.92	87	70.06	**61.97**	62.42	12.62	11.99	22.65	45.17
Simulated sample 2	12.2	**1.61**	6.43	1.866	3.74	1.807	1.638	1.631	12.22	12.59	18.03	31.87
Airline passengers	120	64.6	98.99	58.196	105.4	49.529	49.384	**48.081**	15.14	17.38	34.76	40.33
Simulated sample 3	12.008	2.564	4.78	2.054	4.5	1.965	1.581	**1.554**	23.03	24.34	39.05	38.92
Simulated sample 1	11.98	1.607	6.46	1.972	3.9	1.769	1.528	**1.446**	22.52	26.67	43.28	42.21
Wilcoxon matched	9.9E-8	1E-6	5.2E-7	0.003	3.4E-6	0.006	base	0.0002	Average(in %)		Average(in %)	
p.s.ranked p-value	5.4E-8	2E-7	3.7E-8	6.8E-7	7.3E-7	5.5E-8	0.0002	base	3.47	7.07	17.23	26.81

momentum=0.2), K-Nearest Neighbour(KNN, k=4), Polynomial Support Vector Machines (SVM, $\phi = 1$, C=1, ϵ=0.001, $\xi = \xi^*$=0.001), RBF Support Vector Machines(RBF, $\gamma = 0.01$) and Robust (Huber M-estimator) Regression(RR). We choose the Root Mean Square Error(RMSE) as a performance metric to give us an evaluation on how the new time-dependent loss function affected the overall error, and we also calculated the percentage Error Reduction (ER) and Error Standard Deviation Reduction(SDR) achieved by TiSe SVM and TiSe-Q SVM when compared to the Polynomial SVM:

$$ER = (1 - \frac{\text{RMSE of TiSe/TiSe-Q SVM}}{\text{RMSE of Polynomial SVM}}) * 100.$$

$$SDR = (1 - \frac{\text{Standard Dev. of TiSe/TiSe-Q SVM}}{\text{Standard Dev. of Polynomial SVM}}) * 100.$$

4.2 Prediction loss reduction

Presented in Table 3 are the Root Mean Squared Values for all methods, the Error Reduction(ER) and Error Standard Deviation Reduction(SDR), in percentage, of the TiSe SVM and TiSe-Q SVM models compared to SVM. The results are ordered by the Error Reduction of TiSe-Q SVM. As the datasets used were from different scientific areas and with values of different magnitude, we used the Wilcoxon Matched-Pairs Signed-Ranks test, a non-parametric test to determine whether the differences between the methods were statistically significant. We can observe that the TiSe SVM method performed significantly better than the ARIMA, Neural Network and K-Nearest Neighbour models, and still produced statistically significant better performance than Robust Regression and both SVM Regression models, though in few cases a small increase of the RMSE was registered. However, the results show even better performance for the TiSe-Q SVM version of our time-dependent SVM, with only

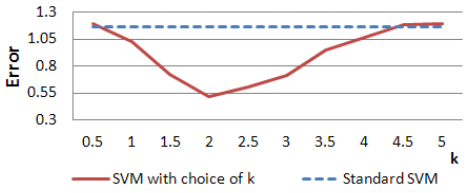

Figure 5: The error achieved for different values of k, "Walt Disney stock market series". For k=0, all the samples are considered as distribution shift samples, we minimize differences of errors already very small, yielding no improvement. As k increases, the minimization of differences in errors at selected samples is visible, and for large enough values of k, no samples are included in the time-dependent loss, equating to standard SVM.

one sample showing an increase in the RMSE value, indicating the quadratic mean was an appropriate choice in the attempt to minimize additional errors without trading of the overall error.

As the purpose of our time-dependent loss function is to additionally minimize the large errors at distribution shift samples, an overview of the errors, particularly the variance or standard deviation, can indicate which model produces a better quality error, with more stability and less volatility. We looked into standard deviations of the errors for SVM, TiSe SVM and TiSe-Q SVM. We found that the desired goal of producing substantially lower variance error was accomplished to a satisfactory level: on average 17.23% reduction in the error standard deviation(Table 3) for TiSe SVM, and TiSe-Q SVM delivered even better performance - 26.81% reduction in the error standard deviation.

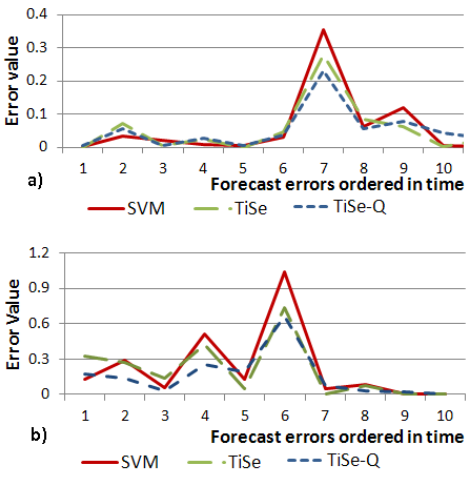

Figure 6: The forecast errors for a) Tree dataset and b) Ebay stock market values dataset. We can observe that both TiSe SVM and TiSe-Q SVM reduced the peaks in the error, with TiSe-Q SVM producing the most significant reduction.

Not only did our models result in keeping the overall error low, but they also successfully targeted the distribution shift sample errors, causing the peaks in forecast errors to reduce significantly, leading to a decrease in the variance of the error. As can be seen from Figure 6, presenting the forecasts for the Tree dataset and Ebay stock market values dataset, the peak regions have been targeted and additional minimization of the error was achieved. Though TiSe managed to achieve sufficient minimization of the targeted errors, TiSe-Q performed better, leading to the conclusion that the quadratic mean was a suitable choice for simultaneous optimization of both overall error and error variance/standard deviation.

5. CONCLUSION AND FUTURE WORK

Time series forecasting is a challenging prediction problem, and SVM regression is a very popular and widely used method. However, it can be susceptible to large prediction errors in time series when distribution shifts occur frequently during the series.

In this paper, we have proposed a novel time-dependent loss function to enhance SVM regression, by minimizing the difference in errors for selected successive pairs of samples, based on consideration of distribution shift characteristics. We combined our time-dependent loss function with the loss function for standard SVM regression, and optimized the two objectives simultaneously. Not only we were able to achieve large reductions in the variance of prediction error, but our method also achieved substantial reductions in root mean squared error as well.

Interesting future work could include extending the time-dependent loss function to consider the difference in error across sequences of three or more samples (rather than only two), as well as deriving the primal form of the quadratic mean, so that Quadratic Programming can be applied, and possibly allow for Kernel functions to be used for the quadratic mean as well.

6. REFERENCES

[1] Z. Abraham and P. N. Tan. An integrated framework for simultaneous classification and regression of time-series data. In *Proceedings of SIAM ICDM*, pages 653–664, Columbus, Ohio, 2010.

[2] B. Awartani and V. Corradi. Predicting the volatility of the s&p-500 stock index via garch models: the role of asymmetries. *International Journal of Forecasting*, 21(1):167–183, 2005.

[3] L. Cao and F. E. H. Tay. Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on Neural Networks*, 14(6):1506–1518, 2003.

[4] V. Chandola and R. R. Vatsavai. A gaussian process based online change detection algorithm for monitoring periodic time series. In *Proceedings of SDM*, 2011.

[5] M. W. Chang, C. J. Lin, and R. Weng. Analysis of nonstationary time series using support vector machines. *Pattern Recognition with Support Vector Machines*, 2388:160–170, 2002.

[6] C. Chen and G. Tiao. Random level-shift time series models, arima approximations, and level-shift detection. *Journal of Business & Economic Statistics*, 8(1):83–97, 1990.

[7] H. Cheng, P. N. Tan, C. Potter, and S. Klooster. Detection and characterization of anomalies in multivariate time series. In *Proceedings of SDM*, 2009.

[8] Y. Cheung and L. Xu. Independent component ordering in ica time series analysis. *Neurocomputing*, 41(1-4):145–154, 2001.

[9] B. R. Dai, J. W. Huang, M. Y. Yeh, and M. S. Chen. Adaptive clustering for multiple evolving streams. *IEEE Transactions on Knowledge & Data Engineering*, 18(9):1166–1180, 2006.

[10] DataMarket. http://datamarket.com/, 2011.

[11] R. Drossu and Z. Obradovic. Rapid design of neural networks for time series prediction. *IEEE Computational Science And Engineering*, 3(2), 1996.

[12] S. Dzeroski, V. Gjorgjioski, I. Slavkov, and J. Struyf. Analysis of time series data with predictive clustering trees. *Knowledge Discovery in Inductive Databases*, 4747, 2007.

[13] C. Freudenthaler, S. Rendle and L. Schmidt-Thieme. Factorizing markov models for categorical time series prediction. In *Proceedings of ICNAAM*, pages 405–409, 2011.

[14] A. W. Fu, E. Keogh, L. Y. Lau, C. A. Ratanamahatana, and R. Wong. Scaling and time warping in time series querying. *VLDB Journal*, 17(4):899–921, 2008.

[15] S. Greco, M. Ruffolo, and A. Tagarelli. Effective and efficient similarity search in time series. In *Proceedings of the 15th ACM CIKM*, pages 808 – 809, 2006.

[16] Q. He, K. Chang, and E. P. Lim. Analyzing feature trajectories for event detection. In *Proceedings of the 30th Annual ACM SIGIR*, pages 207–214, 2007.

[17] K. Huarng and T. H. Yu. The application of neural networks to forecast fuzzy time series. *Physica A: Statistical Mechanics and its Applications*, 363(2):481–491, 2006.

[18] Y. Kawahara and M. Sugiyama. Change-point detection in time-series data by direct density-ratio estimation. In *Proceedings of SDM*, 2009.

[19] N. Khoa and S. Chawla. Robust outlier detection using commute time and eigenspace embedding. *Advances in Knowledge Discovery and Data Mining*, 6119:422–434, 2010.

[20] Y. Li, J. Lin, and T. Oates. Visualizing variable-length time series motifs. In *SIAM Conference on Data Mining (SDM)*, 2012.

[21] W. Liu and S. Chawla. A quadratic mean based supervised learning model for managing data skewness. In *Eleventh SDM*, pages 188–198, 2011.

[22] X. Liu, X. Wu, H. Wang, R. Zhang, J. Bailey, and K. Ramamohanarao. Mining distribution change in stock order streams. In *IEEE ICDE*, pages 105–108, 2010.

[23] Y. Liu, M. T. Bahadori, and H. Li. Sparse-GEV: Sparse latent space model for multivariate extreme value time series modelling. In *Proceedings of ICML*, 2012.

[24] A. D. McQuarrie and C. L. Tsai. Outlier detections in autoregressive models. *Journal of Computational and Graphical Statistics*, 12(12):450–471, 2003.

[25] A. Mueen and E. Keogh. Online discovery and maintenance of time series motifs. In *Proceedings of the 16th ACM SIGKDD*, pages 1089–1098, 2010.

[26] C. H. Teo, S. V. N. Vishwanthan, A. J. Smola, and Q. V. Le. Bundle methods for regularized risk minimization. *Journal of Machine Learning Research*, 11:311–365, 2010.

[27] R. S. Tsay. Outliers, level shifts, and variance changes in time series. *Journal of forecasting*, 7(1):1–20, 1988.

[28] V. S. Tseng, C. H. Chen, P. C. Huang, and T. P. Hong. Cluster-based genetic segmentation of time series with dwt. *Pattern Recognition Letters*, 30(13):1190–1197, 2009.

[29] Wessa. http://www.wessa.net/stocksdata.wasp, 2011.

[30] Z. Xing, J. Pei, P. S. Yu, and K. Wang. Extracting interpretable features for early classification on time series. In *Proceedings of SDM*, 2011.

[31] H. Yang, L. Chan, and I. King. Support vector machine regression for volatile stock market prediction. *Intelligent Data Engineering and Automated Learning - IDEAL*, 2412:391–396, 2002.