

Article

Change Point Test for the Conditional Mean of Time Series of Counts Based on Support Vector Regression

Sangyeol Lee ^{*}  and Sangjo Lee 

Department of Statistics, Seoul National University, Seoul 08826, Korea; lscho13@snu.ac.kr

^{*} Correspondence: sylee@stats.snu.ac.kr

Abstract: This study considers support vector regression (SVR) and twin SVR (TSVR) for the time series of counts, wherein the hyper parameters are tuned using the particle swarm optimization (PSO) method. For prediction, we employ the framework of integer-valued generalized autoregressive conditional heteroskedasticity (INGARCH) models. As an application, we consider change point problems, using the cumulative sum (CUSUM) test based on the residuals obtained from the PSO-SVR and PSO-TSVR methods. We conduct Monte Carlo simulation experiments to illustrate the methods' validity with various linear and nonlinear INGARCH models. Subsequently, a real data analysis, with the return times of extreme events constructed based on the daily log-returns of Goldman Sachs stock prices, is conducted to exhibit its scope of application.

Keywords: time series of counts; INGARCH model; SVR and TSVR with PSO; change point detection; CUSUM test



Citation: Lee, S.; Lee, S. Change Point Test for the Conditional Mean of Time Series of Counts Based on Support Vector Regression. *Entropy* **2021**, *23*, 433. <https://doi.org/10.3390/e23040433>

Academic Editor: Victor Leiva

Received: 25 March 2021

Accepted: 3 April 2021

Published: 7 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In this study, we developed a forecasting method for the time series of counts based on support vector regression (SVR) with particle swarm optimization (PSO), and used it to detect a change in the conditional mean of the time series based on the cumulative sum (CUSUM) test that is calculated from integer-valued autoregressive conditional heteroscedastic (INGARCH) residuals. Over the past few decades, the time series of counts have gained increased attention from researchers in diverse scientific areas. Considering the research conducted by [1–5], two classes of models, such as integer-valued autoregressive (INAR) and INGARCH models, have been popular for analyzing the time series of counts. See [6] for more details. These models have been harnessed to analyze polio data [7], crime data [8], car accident traffic data [9], and financial data [10].

Although the basic theories and analytical tools for these models are quite well developed in the literature, as seen in [11–15], a restriction on their usage exists because both INAR and INGARCH models are mostly assumed to have a linear structure in their conditional mean. In INGARCH models, Poisson and negative binomial distributions have been widely adopted as the conditional distribution of current observations over past information. This is because assuming these distributions is not impractical, as the correct specification of underlying distributions is not essential when attempting to estimate the conditional mean equation, as demonstrated by [16], who considered the quasi-maximum likelihood estimation (QMLE) method for the time series of counts. However, for the QMLE approach to perform adequately, the conditional mean structure must be correctly specified. As misspecification can potentially lead to false conclusions in real situations, we considered SVR as a nonparametric algorithm for forecasting the time series of counts. To our knowledge, our current study is the first attempt in the literature to use SVR for the prediction of time series of counts based on the INGARCH scheme.

SVR has been one of the most popular nonparametric algorithms for forecasting time series and has been proven to outperform classical time series models, such as autoregressive and moving average (ARMA) and GARCH models, as SVR can approximate

nonlinearity without knowing the underlying dynamic structure of time series [17–25]. SVR has the merit of implementing the “structural risk minimization principle” [26] and seeks a balance between model complexity and empirical risk [27]. Moreover, a smaller number of tuning parameters is required, and determining a global solution is not problematic because it solves a quadratic programming problem (QPP).

SVR has been modified in various manners; for example, smooth SVR [28], least squares (LS)-SVM [29], and twin SVR (TSVR) [30]. TSVR generates two hyperplanes unlike SVR and has a significant advantage over SVR in computational speed. For the relevant references, see [31–33]. Here, we harness the SVR and TSVR methods particularly with the particle swarm optimization (PSO) algorithm, originally proposed by [34], in determining a set of optimal parameters to enhance their efficacy. For an overview of PSO, see [35,36].

As an application of our SVR method, we consider the problem of detecting a significant change in the conditional mean of the INGARCH time series. Since [37], the parameter change detection problem has been a core issue in various research areas. As financial time series often suffer from structural changes, owing to changes in governmental policy and critical social events, and ignoring them leads to a false conclusion, change point tests have been considered as an important research topic in time series analysis. See [38,39] for a general review. The CUSUM test has long been used as a tool for detecting a change point, owing to its practical efficiency [40–44]. As regards the time series of counts, see [7,45–50].

Among the CUSUM tests, we adopted the residual-based CUSUM test, because the residual method can successfully discard the correlations of time series and enhance the performance of the CUSUM test in terms of both stability and power. See [51,52]. The authors of the recent reference [43,53] developed a simplistic residual-based CUSUM test for location-scale time series models, based on which the authors of [21,22] designated a hybridization of the SVR and CUSUM methods for handling the change point problem for AR and GARCH time series and demonstrated its superiority over classical models. However, their approach is not directly applicable and requires a new modification for effective performance, especially on the proxies used for the GARCH prediction, as seen in Section 3.4, as simple or exponential moving average type proxies conventionally used for SVR-GARCH models [22] would not work adequately in our current study. Here, we instead used the proxies obtained through the linear INGARCH fit to time series of counts.

The rest of this paper is organized as follows. Section 2 reviews the principle of the CUSUM test and CUSUM of squares test for the INGARCH models and then briefly describes how to apply the SVR-INGARCH method for constructing the CUSUM tests. Section 3 presents the SVR and TSVR-GARCH models for forecasting the conditional mean and describes the SVR and TSVR methods with PSO. Section 4 discusses the Monte Carlo simulations conducted to evaluate the performance of the proposed method. Section 5 discusses the performance of the real data analysis, using the return times of extreme events constructed based on the daily log-returns of Goldman Sachs (GS) stock prices. Finally, Section 6 provides concluding remarks.

2. INGARCH Model-Based Change Point Test

Let $\{Y_t, t \geq 1\}$ be a time series of counts. In order to make inferences for $\{Y_t\}$, one can consider fitting a parametric model to Y_t , for instance, the INGARCH model with the conditional distribution of the one-parameter exponential family and the link function f_θ , parameterized with $\theta \in \Theta \subset \mathbf{R}^d$, that describes the conditional expectation, namely,

$$Y_t | \mathcal{F}_{t-1} \sim p(y | \eta_t), \quad X_t := E(Y_t | \mathcal{F}_{t-1}) = f_\theta(X_{t-1}, Y_{t-1}), \quad (1)$$

where \mathcal{F}_t denotes the past information up to time t , f_θ is defined on $[0, \infty) \times \mathbf{N}_0$ with $\mathbf{N}_0 = \{0, 1, \dots\}$, and $p(\cdot | \cdot)$ is a probability mass function given by

$$p(y | \eta) = \exp\{\eta y - A(\eta)\} h(y), \quad y \geq 0,$$

where η is the natural parameter, $A(\cdot)$ and $h(\cdot)$ are known real-valued functions, $B = A'$ is strictly increasing, and $\eta_t = B^{-1}(X_t)$. $B(\eta_t)$ and $B'(\eta_t)$ are the conditional mean and variance of Y_t over past observations, respectively. Symbols $X_t(\theta)$ and $\eta_t(\theta)$ are used to emphasize θ .

Conventionally, f_θ is assumed to be bounded below by some real number $c > 0$ and to satisfy

$$\sup_{\theta \in \Theta} |f_\theta(x, y) - f_\theta(x', y')| \leq \nu_1|x - x'| + \nu_2|y - y'| \tag{2}$$

for all $x, x' \geq 0$ and $y, y' \in \mathbf{N}_0$, where $\nu_1, \nu_2 \geq 0$ satisfies $\nu_1 + \nu_2 < 1$, which, according to [12], allows $\{Y_t\}$ to be strictly stationary and ergodic, required for the consistency of the parameter estimates.

In practice, Poisson or negative binomial (NB) linear INGARCH(1,1) models with $X_t = \omega + \alpha X_{t-1} + \beta Y_{t-1}$, $\omega > 0, \alpha \geq 0, \beta \geq 0, \alpha + \beta < 1$, are frequently used. For the former, we assume $Y_t | \mathcal{F}_{t-1} \sim \text{Poisson}(X_t)$, whereas for the latter, we assume $Y_t | \mathcal{F}_{t-1} \sim \text{NB}(r, p_t)$, $X_t = \frac{r(1-p_t)}{p_t} = \omega + \alpha X_{t-1} + \beta Y_{t-1}$, where $r \in \mathbf{N}$ and $Y \sim \text{NB}(r, p)$ denotes the negative binomial distribution with its mass function: $P(Y = k) = \frac{(k+r-1)!}{(r-1)!k!} (1-p)^k p^r$, $k \geq 0$.

Let θ_0 be a true parameter, which is assumed to be an interior point of the compact parameter space Θ . The θ_0 is then estimated using the conditional likelihood function of model (1), based on the observations Y_1, \dots, Y_n :

$$\tilde{\mathcal{L}}_n(\theta) = \prod_{t=1}^n \exp\{\tilde{\eta}_t(\theta)Y_t - A(\tilde{\eta}_t(\theta))\}h(Y_t), \tag{3}$$

where $\tilde{\eta}_t(\theta) = B^{-1}(\tilde{X}_t(\theta))$ is updated through the equations: $\tilde{X}_t(\theta) = f_\theta(\tilde{X}_{t-1}(\theta), Y_{t-1})$ for $t \geq 2$, $\tilde{X}_1(\theta) = \tilde{X}_1$, with an initial value \tilde{X}_1 . The conditional maximum likelihood estimator (CMLE) of θ_0 is then obtained as the maximizer of the likelihood function in Equation (3):

$$\hat{\theta}_n = \operatorname{argmax}_{\theta \in \Theta} \tilde{\mathcal{L}}_n(\theta) = \operatorname{argmax}_{\theta \in \Theta} \sum_{t=1}^n \tilde{\ell}_t(\theta),$$

with $\tilde{\ell}_t(\theta) = \log p(Y_t | \tilde{\eta}_t(\theta)) = \tilde{\eta}_t(\theta)Y_t - A(\tilde{\eta}_t(\theta))$. The authors of the reference [12,50] showed that, under certain conditions, $\hat{\theta}_n$ converges to θ_0 in probability and $\sqrt{n}(\hat{\theta}_n - \theta_0)$ is asymptotically normally distributed as n tends to ∞ . This $\hat{\theta}_n$ is harnessed to make prediction for calculating residuals.

In our current study, we aim to extend Model (1) to the nonparametric model:

$$Y_t | \mathcal{F}_{t-1} \sim p(y | \eta_t), \quad X_t = g(X_{t-1}, Y_{t-1}), \tag{4}$$

where $p(\cdot | \cdot)$ and g are unknown, and g is implicitly assumed to satisfy Equation (2). Provided $g \in \{f_\theta; \theta \in \Theta\}$ and $p(\cdot | \cdot)$ is known a priori, one can estimate g with $\hat{g} = f_{\hat{\theta}}$. Even if $p(\cdot | \cdot)$ is unknown, one can still consider using the Poisson or NB quasi-likelihood estimator (QMLE) method as in [16]. See also [54] for various types of CUSUM tests based on the QMLEs. However, when one has no prior information as to g , the parametric modeling may hamper the inference, and in this case, one can estimate g with the nonparametric SVR method stated below in Section 3.

On Model (1), setting up the null and alternative hypotheses: $H_0 : \theta$ remain the same over $t = 1, \dots, n$. vs. $H_1 : \text{not } H_0$, The authors of the reference [50] considered the problem of detecting a change in θ based on the CUSUM test:

$$\hat{T}_n^{res} = \max_{1 \leq k \leq n} \frac{1}{\sqrt{n} \hat{\tau}_n} \left| \sum_{t=1}^k \hat{\epsilon}_t - \frac{k}{n} \sum_{t=1}^n \hat{\epsilon}_t \right| \tag{5}$$

with the residuals $\hat{\epsilon}_t = Y_t - \tilde{X}_t(\hat{\theta}_n)$ and $\hat{\tau}_n^2 = \frac{1}{n} \sum_{t=1}^n \hat{\epsilon}_t^2 - \left(\frac{1}{n} \sum_{t=1}^n \hat{\epsilon}_t\right)^2$. Furthermore, the authors of the references [55,56] employed the residual-based CUSUM of squares test:

$$\hat{T}_n^{square} = \max_{1 \leq k \leq n} \frac{1}{\sqrt{n\hat{\tau}_n}} \left| \sum_{t=1}^k \hat{\epsilon}_t^2 - \frac{k}{n} \sum_{t=1}^n \hat{\epsilon}_t^2 \right| \tag{6}$$

with $\tilde{\tau}_n^2 = \tilde{\gamma}_n(0) + 2 \sum_{h=1}^{h_n} \tilde{\gamma}_n(h)$, $\tilde{\gamma}_n(h) = \frac{1}{n} \sum_{t=1}^{n-h} (\hat{\epsilon}_t^2 - \bar{\epsilon}^2)(\hat{\epsilon}_{t+h}^2 - \bar{\epsilon}^2)$, $\bar{\epsilon}^2 = \frac{1}{n} \sum_{t=1}^n \hat{\epsilon}_t^2$, and $h_n = \sqrt{2}(\log_{10} n)^2$.

The authors of the reference [50] verified that, under the null H_0 , \hat{T}_n^{res} behaves asymptotically the same as

$$T_n = \max_{1 \leq k \leq n} \frac{1}{n\tau} \left| \sum_{t=1}^k \epsilon_t - \frac{k}{n} \sum_{t=1}^n \epsilon_t \right|,$$

where $\epsilon_t = Y_t - X_t(\theta_0)$ and $\tau^2 = Var(\epsilon_1)$. As $\{\epsilon_t\}$ forms a sequence of martingale differences, we obtain $T_n \approx T := \sup_{0 \leq s \leq 1} |B^\circ(s)|$ in distribution [57], where B° denotes a Brownian bridge, owing to Donsker’s invariance principle, so that, as $\hat{T}_n^{res} \approx T_n$, we have $\hat{T}_n^{res} \approx T$ in distribution. For instance, H_0 is rejected if $\hat{T}_n^{res} \geq 1.3397$ at the level of 0.05, which is obtainable with Monte Carlo simulations. Similarly, the authors of the reference [55] verified that $\hat{T}_n^{square} \approx T$ in distribution, so that the same critical values as for the case of \hat{T}_n^{res} can be harnessed. Provided that a change point exists, the location of change is identified as

$$\hat{k}_n = \operatorname{argmax}_{1 \leq k \leq n} \left| \sum_{t=1}^k \hat{\epsilon}_t^i - \frac{k}{n} \sum_{t=1}^n \hat{\epsilon}_t^i \right|, \quad i = 1, 2.$$

This CUSUM framework for parametric models can be easily adopted for nonparametric models as far as the residuals $\hat{\epsilon}_t$ can be accurately calculated, as seen in [21,22] who deal with the change point problem on SVR-ARMA and SVR-GARCH models. Below, when dealing with Model (4), instead of $\hat{\epsilon}_t$ in Equation (5), we use $\hat{\epsilon}_t = Y_t - \hat{g}(X_{t-1}, Y_{t-1})$ in the construction of the CUSUM tests in Equations (5) and (6).

When estimating g with SVR and TSVR, we train $(y_t, \mathbf{x}_t)^T$ either with $y_t = \tilde{X}_t$ and $\mathbf{x}_t = (\tilde{X}_{t-1}, Y_{t-1})^T$ or $y_t = Y_t$ and $\mathbf{x}_t = (\tilde{X}_{t-1}, Y_{t-1})^T$, with some proper proxy \tilde{X}_{t-1} . The former has been used for the SVR-GARCH model in [21], while the latter is newly considered here, inspired by the fact that $Y_t = g(X_{t-1}, Y_{t-1}) + \nu_t$, where the error process $\{\nu_t\}$ is a sequence of martingale differences, which also holds for Model (1) because we can express $Y_t = X_t + \nu_t$ in this case. See Step 3 in Section 4 below for more details.

3. SVR-INGARCH Model

In this section, we provide an outline of the SVR, TSVR, and PSO methods for a quick reference and describe the change point test based on the SVR-INGARCH model.

3.1. Support Vector Regression

SVR is an extension of the SVM, originally proposed by [58], and merits accurate nonlinear prediction. SVR aims to identify a nonlinear function of the form: $f(x) = w^T \phi(x) + b$, where x denotes a vector of inputs, w and b are vectors of regression parameters, and ϕ is a known kernel function. The optimal w and b are determined from the ϵ -insensitive loss function (Vapnik, 2000):

$$\ell_\epsilon(y, f(x)) = \begin{cases} |y - f(x)| - \epsilon & \text{if } |y - f(x)| \geq \epsilon \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Given input vectors x_i , scalar output y_i , $i = 1, \dots, n$, and a constant $C > 0$, we construct the objective function of the SVR as follows:

$$\text{minimize } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_{1,i} + \xi_{2,i}), \tag{8}$$

$$\text{subject to } \begin{cases} y_i - w^T \phi(x_i) - b \leq \epsilon + \xi_{2,i} \\ w^T \phi(x_i) + b - y_i \leq \epsilon + \xi_{1,i} \\ \xi_{1,i} \geq 0, \xi_{2,i} \geq 0, \end{cases}$$

where $\xi_{1,i}, \xi_{2,i} > 0$ denote slack variables that allow some points to lie outside the ϵ -band with a penalty, and C denotes a trade-off between the function complexity and the training error.

To obtain the optimal w and b , we formulate an unconstrained optimization problem using Lagrange multipliers [27]. The Karush–Kuhn–Tucker (KKT) conditions then lead to the following dual form:

$$\begin{aligned} \text{maximize } & -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_{1,i} - \alpha_{2,i})(\alpha_{1,j} - \alpha_{2,j}) \phi(x_i)^T \phi(x_j) \\ & - \epsilon \sum_{i=1}^n (\alpha_{1,i} + \alpha_{2,i}) + \sum_{i=1}^n (\alpha_{1,i} - \alpha_{2,i}) y_i, \end{aligned} \tag{9}$$

subject to $\sum_{i=1}^n (\alpha_{1,i} - \alpha_{2,i}) = 0, 0 \leq \alpha_{1,i} \leq C, 0 \leq \alpha_{2,i} \leq C$, where $\alpha_{1,i}$ and $\alpha_{2,i}$ denote dual variables [26]. Subsequently, the optimization problem in Equation (9) yields the solutions $\hat{w}, \hat{b}, \hat{f}$ of w, b, f , as follows:

$$\begin{aligned} \hat{w} &= \sum_{i=1}^n (\alpha_{1,i} - \alpha_{2,i}) \phi(x_i), \\ \hat{b} &= \begin{cases} y_i - \hat{w}^T \phi(x_i) - \epsilon, & 0 < \alpha_{1,i} < C \\ y_i - \hat{w}^T \phi(x_i) + \epsilon, & 0 < \alpha_{2,i} < C, \end{cases} \\ \hat{f}(x) &= \sum_{i=1}^n (\alpha_{1,i} - \alpha_{2,i}) K(x_i, x) + \hat{b} \end{aligned} \tag{10}$$

with $K(x, y) = \phi(x)^T \phi(y)$. In particular, we employ the Gaussian kernel for K in Equation (10), $K(x, y) = \exp(-\frac{\|x-y\|^2}{2\gamma^2})$, and determine the tuning parameters γ^2, C in Equation (8), and ϵ in the loss function Equation (7) using the PSO method on the cube of (C, γ^2, ϵ) with $1 \leq C \leq 100, 0.1 \leq \gamma^2 \leq 1$, and $0.1 \leq \epsilon \leq 1$.

3.2. Twin Support Vector Regression

TSVR is a modified version of SVR [30]. Similar to TSVM [59], TSVR derives two nonparallel planes $f_1(x) = w_1^T \phi_1(x) + b_1$ and $f_2(x) = w_2^T \phi_2(x) + b_2$, which respectively determines the ϵ_1 -sensitive downbound and ϵ_2 -sensitive upbound of data. Given input vectors x_i and output $y_i, i = 1, \dots, n$, the linear TSVR can be formulated as the constraint minimization problem as follows:

$$\begin{aligned} \text{minimize } & f(w_1, b_1, \xi_1) = \frac{1}{2} \|Y - Aw_1 - eb_1 - e\epsilon_1\|^2 + C_1 e^T \xi_1, \\ \text{subject to } & Y - Aw_1 - eb_1 \geq e\epsilon_1 - \xi_1; \end{aligned} \tag{11}$$

$$\begin{aligned} \text{minimize } & f(w_2, b_2, \xi_2) = \frac{1}{2} \|Y - Aw_2 - eb_2 + e\epsilon_2\|^2 + C_2 e^T \xi_2, \\ \text{subject to } & Aw_2 + eb_2 - Y \geq e\epsilon_2 - \xi_2, \end{aligned} \tag{12}$$

where $Y = (y_1, \dots, y_n)^T$, $A = (x_1 \dots x_n)^T$, $\epsilon_1, \epsilon_2 \geq 0$, e denotes a vector whose components are all equal to 1, $C_1, C_2 \geq 0$ are hyperparameters, and $\xi_1, \xi_2 \geq 0$ are slack variables. Each QPP has m constraints instead of $2m$ constraints and has an advantage of faster computational speed. To obtain the optimal w_1 and b_1 in Equation (11), we solve the QPP using the Lagrangian function:

$$L(w_1, b_1, \xi_1, \alpha_1, \beta_1) := \frac{1}{2} \|Y - Aw_1 - eb_1 - e\epsilon_1\|^2 + C_1 e^T \xi_1 - \alpha_1^T (Y - Aw_1 - eb_1 - e\epsilon_1 - \xi_1) - \beta_1^T \xi_1, \tag{13}$$

where $\alpha_1 \geq 0$ and β_1 are Lagrangian multiplier vectors. If there is an optimal solution, it must satisfy the following KKT conditions:

$$-A^T(Y - Aw_1 - eb_1 - e\epsilon_1) + A^T \alpha_1 = 0; \tag{14}$$

$$-e^T(Y - Aw_1 - eb_1 - e\epsilon_1) + e^T \alpha_1 = 0; \tag{15}$$

$$C_1 e - \alpha_1 - \beta_1 = 0; \quad \alpha_1^T (Y - Aw_1 - eb_1 - e\epsilon_1 + \xi_1) = 0, \quad \alpha_1 \geq 0;$$

$$\beta_1^T \xi_1 = 0, \quad \beta_1 \geq 0; \quad Y - Aw_1 + eb_1 \geq e\epsilon_1 - \xi_1, \quad \xi_1 \geq 0;$$

$$Aw_2 + eb_2 - Y \geq e\epsilon_2 - \xi_2, \quad \xi_2 \geq 0.$$

We define $G = (A \ e)$, $h_1 = Y - e\epsilon_1$, and $u_1 = (w_1^T \ b_1)^T$. Combining Equations (14) and (15), we have

$$u_1 = (G^T G)^{-1} G^T (h_1 - \alpha_1). \tag{16}$$

However, since $G^T G$ is only semidefinite, we introduce a regularization term σI , where $\sigma > 0$ is very small, to overcome some ill-conditioned setting and use $u_1 = (G^T G + \sigma I)^{-1} G^T (h_1 - \alpha_1)$. Next, substituting Equation (16) and the KKT conditions into Equation (13), we obtain the dual QPP form:

$$\begin{aligned} &\text{maximize} \quad -\frac{1}{2} \alpha_1^T G (G^T G)^{-1} G^T \alpha_1 + h_1^T G (G^T G)^{-1} G^T \alpha_1 - h_1^T \alpha_1, \\ &\text{subject to} \quad 0 \leq \alpha_1 \leq C_1 e, \end{aligned}$$

which yields u_1 . Likewise, Equation (12) can acquire the dual QPP form:

$$\begin{aligned} &\text{minimize} \quad -\frac{1}{2} \alpha_2^T G (G^T G)^{-1} G^T \alpha_2 - h_2^T G (G^T G)^{-1} G^T \alpha_2 + h_2^T \alpha_2, \\ &\text{subject to} \quad 0 \leq \alpha_2 \leq C_2 e, \end{aligned}$$

where α_2 is the Lagrangian multiplier vector and $h_2 = Y + e\epsilon_2$. This yields $u_2 = (w_2^T \ b_2)^T$. Then, the estimated regressor can be formulated as follows:

$$f(x) = \frac{1}{2} (f_1(x) + f_2(x)) = \frac{1}{2} (w_1 + w_2)^T x + \frac{1}{2} (b_1 + b_2).$$

For extending the linear TSVR to a nonlinear one, we use the kernel-generated nonparallel hyperplanes, that is, $f_1(x) = K(x^T, A^T)w_1 + b_1$ and $f_2(x) = K(x^T, A^T)w_2 + b_2$. The optimization problem in this case is similar to the linear TSVR, and the nonlinear TSVR regressor is obtained as follows:

$$f(x) = \frac{1}{2} (f_1(x) + f_2(x)) = \frac{1}{2} (w_1 + w_2)^T K(A, x) + \frac{1}{2} (b_1 + b_2).$$

3.3. Particle Swarm Optimization Method

In a standard PSO algorithm [34], a set of d hyperparameters are considered as a particle of d -dimensional vector in search region $S = \{p = (p_1, \dots, p_d)^T \in R^d; l_k \leq p_k \leq$

u_k with $l_k, u_k \in R$ for $k = 1, \dots, d$. Here, N particles are modeled to move in S , with the position $p_i = (p_{i1}, \dots, p_{id})^T$ and velocity $v_i = (v_{i1}, \dots, v_{id})^T$ for $i = 1, \dots, N$. The previous best position of the i -th particle is represented by $p_i^{best} = (p_{i1}^{best}, \dots, p_{id}^{best})^T$, and the previous best position of all particles is represented by $g^{best} = (g_1^{best}, \dots, g_d^{best})^T$. At each iteration k , where $1 \leq k \leq K_{max}$ with maximum iteration number K_{max} , the velocity and position of the i -th particle are updated as follows:

$$\begin{aligned} v_i^{k+1} &= w_k v_i^k + c_1 r_1 (p_i^{best,k} - p_i^k) + c_2 r_2 (g^{best,k} - p_i^k), \\ p_i^{k+1} &= p_i^k + v_i^{k+1}, \end{aligned}$$

where c_1 and c_2 are two acceleration factors, r_1 and r_2 are two random variables following a uniform distribution over $[0,1]$, and w_k is an inertia factor defined by

$$w_k = (w_{start} - w_{end}) \left(\frac{K_{max} - k}{K_{max}} \right) + w_{end},$$

where w_{start} and w_{end} are initial and final values of inertia. Since the positions of particles are updated, p_i^{best} and g^{best} are also updated as follows:

$$\begin{aligned} p_i^{best,k+1} &= \begin{cases} p_i^{k+1}, & \text{if } f(p_i^{k+1}) < f(p_i^{best,k}) \\ p_i^{best,k}, & \text{otherwise} \end{cases} \\ g^{best,k+1} &= \operatorname{argmin}_{p_i^{best,k+1}} f(p_i^{best,k+1}). \end{aligned}$$

The finally updated g^{best} in the above procedure is used as an optimal hyperparameter in estimating the SVR and TSVR models as seen below.

3.4. PSO-TSVR Model-Based CUSUM Test

In this subsection, we explain how to estimate X_t in Model (4) using the SVR and TSVR methods with PSO as described above and how to construct the CUSUM test based on the residuals obtained from the SVR-INGARCH model. In the following steps, we assume that a time series $\{Y_1, \dots, Y_n, Y_{n+1}, \dots, Y_{n+n'}\}$ has no change points.

- **Step 1.** As in Section 3.3, in order to apply the PSO method, we set a space of hyperparameters and initialize the positions $\{p_1^0, \dots, p_N^0\}$ and velocities $\{v_1^0, \dots, v_N^0\}$ of particles to be evaluated within this space. Subsequently, we divide the given time series into two groups of time series, $\{Y_1, \dots, Y_n\}$ and $\{Y_{n+1}, \dots, Y_{n+n'}\}$. The former is used as a training set while the latter is used as a validation set.
- **Step 2.** Compute the initial estimates of X_t based on the training time series. For this task, we use two different methods. The first method is using moving averages (Niemira, 1994):

$$\tilde{X}_t = \frac{1}{m} \sum_{j=1}^m Y_{t-j+1}, \quad (17)$$

where m is a positive integer. When t is smaller than m , \tilde{X}_t is computed as an average of the first to the t -th squares of observations, i.e. $\tilde{X}_t = \sum_{j=1}^t Y_{t-j+1} / t$ when $t < m$. The second method is using the Poisson QMLE [54] assuming that the time series follows Model (1), for example,

$$\tilde{X}_t = \hat{\omega} + \hat{\alpha} \tilde{X}_{t-1} + \hat{\beta} Y_{t-1}. \quad (18)$$

These estimates play the role of proxy of X_t and replace the true conditional volatility.

- **Step 3.** For particles p_i^k , $k = 1, 2, \dots, K_{max}$, we train $(y_t, \mathbf{x}_t)^T$, either with $y_t = \tilde{X}_t$ and $\mathbf{x}_t = (\tilde{X}_{t-1}, Y_{t-1})^T$ or $y_t = Y_t$ and $\mathbf{x}_t = (\tilde{X}_{t-1}, Y_{t-1})^T$ with proxy \tilde{X}_{t-1} to the SVR and TSVR models to obtain \hat{g} . Subsequently, for the first, we obtain

$$\hat{X}_t = \hat{g}(Y_{t-1}, \tilde{X}_{t-1}), \quad (19)$$

named “ \hat{X}_t -targeting”, and for the second,

$$\hat{Y}_t = \hat{g}(Y_{t-1}, \tilde{X}_{t-1}), \quad (20)$$

named “ \hat{Y}_t -targeting”, where \hat{Y}_t is an estimate of Y_t , which is an estimate X_t as well since \hat{Y}_t itself is the conditional expectation.

- **Step 4.** Applying the estimated SVR and TSVR models and using the same proxy formula as in **Step 2** for the validation time series, the mean absolute error (MAE) is computed as follows:

$$\text{MAE} = \frac{1}{n'} \sum_{t=n+1}^{n+n'} |\hat{X}_t - \tilde{X}_t|$$

for the case of Equation (19), and

$$\text{MAE} = \frac{1}{n'} \sum_{t=n+1}^{n+n'} |\hat{Y}_t - \tilde{X}_t|$$

for the case of Equation (20). The MAE is employed here because it is more robust against outliers in a model fitting than the root mean square error.

- **Step 5.** Update the p_i^k , v_i^k , $p_i^{best,k}$, and $g^{best,k}$ as in Section 3.3 and repeat **Steps 3 and 4** until the MAE in **Step 4** converges within a limit or k reaches the maximum iteration number K_{max} .
- **Step 6.** Apply the estimated SVR and TSVR models with selected parameters in **Step 5** to a testing time series to perform the CUSUM tests in Equations (5) and (6) based on the residuals $\hat{\epsilon}_t = Y_t - \hat{g}(Y_{t-1}, \tilde{X}_{t-1})$.

4. Simulation Results

In this section, we apply the PSO-SVR and -TSVR models to the INAR(1) and INGARCH(1,1) models, and evaluate the performance of the proposed CUSUM tests. For this task, we generate a time series of length 1000 ($n = 500$, $n' = 500$) to evaluate the empirical sizes and powers at the nominal level of 0.05. The size and power are calculated as the rejection number of the null hypothesis of no changes out of 500 repetitions. The simulations were conducted with R version 3.6.3, running on Windows 10. Moreover, we use the following R packages: “pso” for the PSO [60], “kernlab” for the Gaussian kernel [61], and “osqp” [62] for solving the quadratic problem. The procedure for the simulation is as follows.

- **Step 1.** Generate a time series of length $n = 1000$ to train the PSO-SVR and -TSVR models.
- **Step 2.** Apply the estimation scheme described in Section 4. For the proxy of moving averages, we used $m = 5$. In this procedure, we divide the given time series into $n = 500$ and $n' = 500$ in **Step 1** of Section 4.
- **Step 3.** Generate a testing time series of length $n = 1000$ to evaluate the size and power. For computing sizes, we generate a time series of no changes, whereas to examine the power, we generate a time series with a change point in the middle.
- **Step 4.** Apply the estimated model in **Step 2** to the time series of **Step 3** and conduct the residual CUSUM and CUSUM of squares tests.
- **Step 5.** Repeat the above steps N times, e.g., 500, and then compute the empirical sizes and powers.

We consider the INGARCH(1,1) and INAR(1) models, as these are the most acclaimed models in practice:

- **Model 1.** $Y_t | \mathcal{F}_{t-1} \sim \text{Poisson}(X_t)$, $X_t = \omega + \alpha X_{t-1} + \beta Y_{t-1}$,
- **Model 2.** $Y_t = \phi \circ Y_{t-1} + Z_t$, $Z_t \sim \text{Poisson}(\omega)$, where \circ is a binomial thinning operator and $|\phi| < 1$.

Further, upon one referee's suggestion, we also consider the softplus INGARCH(1,1) model in [63]:

- **Model 3.** $Y_t | \mathcal{F}_{t-1} \sim \text{Poisson}(X_t)$, $X_t = s_c(\omega + \alpha Y_{t-1} + \beta X_{t-1})$, where $s_c(x) = c \log(1 + \exp(x/c))$.

Under the null hypothesis, we use the parameter settings as follows.

- Model 1:
 - Case 1: $\omega = 3, \alpha = 0.3, \beta = 0.3$;
 - Case 2: $\omega = 5, \alpha = 0.3, \beta = 0.3$;
 - Case 3: $\omega = 3, \alpha = 0.6, \beta = 0.3$;
 - Case 4: $\omega = 3, \alpha = 0.3, \beta = 0.6$;
- Model 2:
 - Case 1: $\omega = 3, \phi = 0.3$;
 - Case 2: $\omega = 5, \phi = 0.3$;
 - Case 3: $\omega = 3, \phi = 0.7$;
- Model 3:
 - $c = 1, \omega = 3, \alpha = 0.3, \beta = 0.3$.

Under the alternative hypothesis, we only consider the case of one parameter change, while the other parameters remain the same. Tables A1–A4 in Appendix A summarize the results for Model 1. Here, MA and ML denote the proxies obtained from the moving average in Equation (17) and Poisson QMLE in Equation (18), and \hat{Y}_t and \hat{X}_t , respectively, denote the two targeting methods in Equation (20) and Equation (19). The tables show that the difference between the SVR and TSVR methods is marginal. Moreover, in most cases, $\hat{T}_n^{\text{square}}$ appears to be much more stable than \hat{T}_n^{res} ; that is, the latter test suffers from more severe size distortions. In terms of power, \hat{T}_n^{res} with the ML proxy and \hat{X}_t -targeting tends to outperform the others. However, the gap between this test and $\hat{T}_n^{\text{square}}$ is only marginal; therefore, considering the stability of the test, $\hat{T}_n^{\text{square}}$ is highly favored for Model 1. Tables A5–A7 summarize the result for Model 2, showing that \hat{T}_n^{res} exhibits a more stable performance for the INAR models than for the INGARCH models. However, it is still not as stable as $\hat{T}_n^{\text{square}}$ and tends to outperform $\hat{T}_n^{\text{square}}$ in terms of power. Table A8 summarizes the result for Model 3, showing no significant differences from the results of the previous models. This result, to a certain extent, coincides with that of Lee and Lee (2020) who considered parametric INGARCH models for a change point test. Overall, our findings strongly confirm the reliability of using $\hat{T}_n^{\text{square}}$, particularly with the ML proxy and \hat{X}_t -targeting. However, in practice, one can additionally implement \hat{T}_n^{res} because either test can react more sensitively to a specific situation in comparison to the other.

Table A9 lists the computing times (in seconds) of the SVR and TSVR methods when implemented in R on Windows 10, running on a PC with an Intel i7-3770 processor (3.4 GHz) with 8 GB of RAM, wherein the figures denote the averages of training times in simulations, and the values in the parentheses indicate the sample standard deviations. In each model and parameter setting, the values of the two quickest results are written in boldface. As reported by [21,30], the TSVR method is shown to markedly reduce the CPU time. In particular, the results indicate that the computational speed of the TSVR-based method, with the ML proxy and \hat{X}_t -targeting, appears to be the fastest in most cases. The result suggests that using the TSVR-based CUSUM tests is beneficial when computational speed is of significance to the implementation.

5. Real Data Analysis

In this section, we analyze the return times of extreme events constructed based on the daily log-returns of GS stock prices from 1 January 2003, to 28 June 2019, obtained using the R package “quantmod.” We used data from 2 January 2003, to 29 June 2007, as the training set and that from 1 July 2009, to 28 June 2019, as the test set. Figures 1 and 2 exhibit the GS stock prices and 100 times the log-returns, with their ranges denoted by the green and blue vertical lines, respectively. As shown in Figure 2, the time series between the training and test sets has severe volatility, owing to the financial crisis that occurred in 2008; therefore, it is omitted from our data analysis.

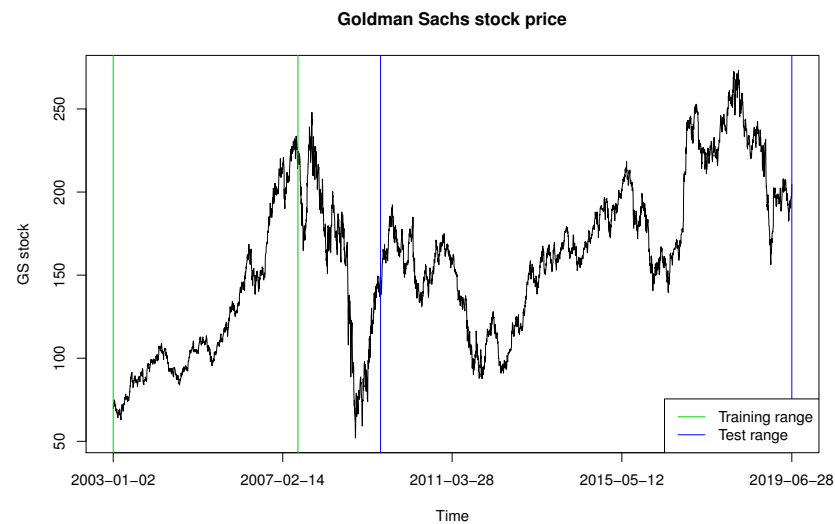


Figure 1. Goldman Sachs stock price.

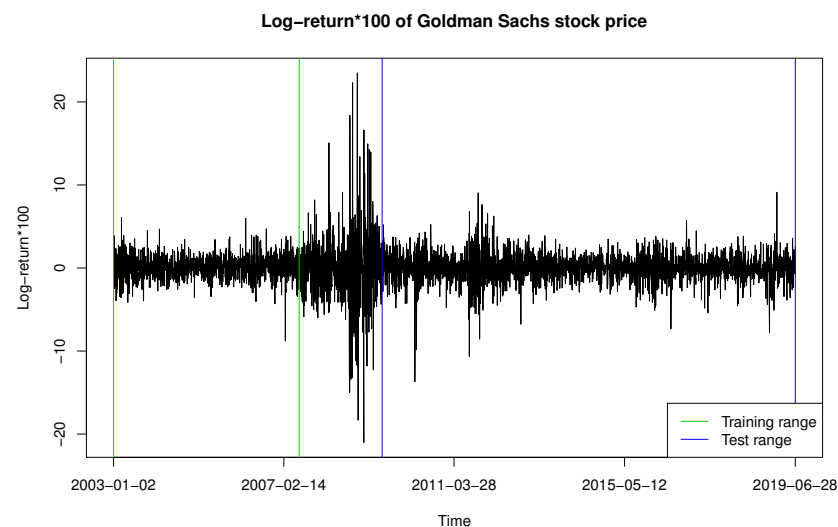


Figure 2. Log-return of Goldman Sachs stock price.

Before applying the PSO-SVR-INGARCH and PSO-TSVR-INGARCH methods, similarly to [12,14], we first transform the given time series into the hitting times τ_1, τ_2, \dots , for which the log-returns of the GS stock fall outside the 0.15 and 0.85 quantiles of the training data, that is, -1.242 and 1.440, respectively. More specifically, $\tau_1 = \inf\{t \geq 1; w_t \notin [-1.242, 1.440]\}$, $\tau_2 = \inf\{t \geq \tau_1; w_t \notin [-1.242, 1.440]\}, \dots$, where w_t denote the 100 times log-returns. We then set $Y_t := \tau_t - \tau_{t-1}$, which forms the return times of these extreme events. Consequently, the training set is transformed into an integer-valued time series of length 341, and the test set is transformed into that of length 844 (see Figure 2), plotting Y_t .

To determine whether the training set exhibits change, we apply the Poisson QMLE method and CUSUM of squares test from [54]. The result shows that the CUSUM statistics \hat{T}_n^{square} has a value of 0.590, which is smaller than the theoretical critical value of 1.358; thus, the null hypothesis of no change is not rejected at the nominal level of 0.05, supporting the adequacy of the training set. The residual-based CUSUM of squares tests based on the SVR and TSVR models with the ML proxy and \hat{X}_t -targeting are then applied; subsequently, both tests detect a change point at the 441st observation of the testing data, corresponding to 16 October 2013. The red vertical line in Figure 3 denotes the detected change point.

To examine how the change affects the dynamic structure of the time series, we fit a Poisson linear INGARCH model to the training and testing time series before and after the change point. For the training time series, the fitted INGARCH model appears to have $\hat{\omega} = 0.334$, $\hat{\alpha} = 0.813$, and $\hat{\beta} = 0.086$. Conversely, for the testing time series before the change, we obtain $\hat{\omega} = 0.135$, $\hat{\alpha} = 0.878$, and $\hat{\beta} = 0.067$, which are not as different as those from the training time series case. However, after the change point, the fitted parameters are shown to be $\hat{\omega} = 1.045$, $\hat{\alpha} = 0.560$, and $\hat{\beta} = 0.147$, thus confirming a significant change in the parameters. For instance, the sum of α and β in the training data is 0.899, which changes from 0.945 to 0.707 in the testing data.

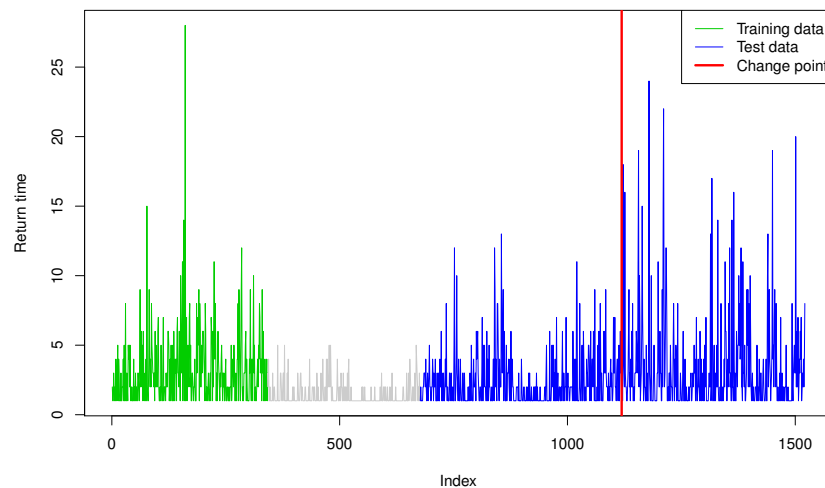


Figure 3. Return time of extreme events of Goldman Sachs stock prices.

6. Concluding Remarks

In this study, we proposed the CUSUM test based on the residuals obtained with the SVR and TSVR-INGARCH models to detect a parameter change in the conditional mean of the time series of counts. To improve accuracy and efficiency, we also employed the PSO method to obtain an optimal set of hyperparameters. Monte Carlo simulations were conducted using the INAR and INGARCH models with various parameter settings. The results showed that the TSVR method using the ML proxy and the conditional mean \hat{X}_t -targeting method is recommendable, as it generally performs well and markedly reduces computational time. Our method was then applied to the analysis of the return times of extreme events constructed based on the daily log-returns of Goldman Sachs stock prices and, subsequently, detected one change. Overall, our findings, based on a simulation study and real data analysis, demonstrated the validity of our method. Although the proposed method performs well in general, it might have a limitation in its performance when the amount of available training data is not large enough or the dataset has features that can violate the stationarity, e.g., high volatilities. The method can also suffer from over-fitting to a specific training sample. Thus, it would be an important task to develop more robust methods, which we leave as our future project.

Author Contributions: Conceptualization, S.L. (Sangyeol Lee); methodology, S.L. (Sangyeol Lee) and S.L. (Sangjo Lee); software, S.L. (Sangjo Lee); formal analysis, S.L. (Sangyeol Lee) and S.L. (Sangjo Lee); data curation, S.L. (Sangjo Lee); writing—original draft preparation, S.L. (Sangyeol Lee) and S.L. (Sangjo Lee); writing—review and editing, S.L. (Sangyeol Lee) and S.L. (Sangjo Lee); funding acquisition, S.L. (Sangyeol Lee). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT), grant no. NRF-2021R1A2C1004009.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: <https://finance.yahoo.com>, (accessed on 10 August 2020).

Acknowledgments: We sincerely thank the Editor and the two anonymous reviewers for their precious time and valuable comments.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SVR	Support vector regression
TSVR	Twin support vector regression
PSO	Particle swarm optimization
INGARCH	Integer-valued generalized autoregressive conditional heteroskedasticity
CUSUM	Cumulative sum
INAR	Integer-valued autoregressive
QMLE	Quasi-maximum likelihood estimation
ARMA	Autoregressive and moving average
GARCH	Generalized autoregressive conditional heteroskedasticity
NB	Negative binomial
CMLE	Conditional maximum likelihood estimator
SVM	Support vector machine
MAE	Mean absolute error

Appendix A

Table A1. Empirical sizes and powers for the INGARCH(1,1) model, Case 1.

$\omega = 3$	Method	SVR				TSVR			
		Target	\hat{Y}_t	\hat{X}_t		\hat{Y}_t		\hat{X}_t	
$\alpha = 0.3$	Proxy	MA	ML	MA	ML	MA	ML	MA	ML
size	\hat{T}_n^{res}	0.072	0.104	0.000	0.096	0.076	0.064	0.000	0.074
	\hat{T}_n^{square}	0.040	0.034	0.032	0.042	0.036	0.038	0.032	0.046
$\omega \rightarrow 5$	\hat{T}_n^{res}	1.000	1.000	0.640	1.000	0.984	0.860	0.274	0.996
	\hat{T}_n^{square}	0.968	0.980	0.776	0.994	0.984	0.964	0.808	0.990
$\omega \rightarrow 10$	\hat{T}_n^{res}	1.000	1.000	1.000	1.000	0.972	0.962	0.948	0.956
	\hat{T}_n^{square}	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
power	\hat{T}_n^{res}	1.000	1.000	0.978	1.000	0.976	0.868	0.620	0.982
	\hat{T}_n^{square}	0.986	0.994	0.936	0.944	0.986	0.992	0.944	1.000
$\alpha \rightarrow 0.5$	\hat{T}_n^{res}	1.000	1.000	0.988	1.000	0.982	0.870	0.664	0.982
	\hat{T}_n^{square}	0.952	0.950	0.826	0.956	0.916	0.924	0.826	0.980

Table A2. Empirical sizes and powers for the INGARCH(1,1) model, Case 2.

$\omega = 5$	Method	SVR				TSVR			
$\alpha = 0.3$	Target	\hat{Y}_t	\hat{X}_t		\hat{Y}_t	\hat{X}_t			
$\beta = 0.3$	Proxy	MA	ML	MA	ML	MA	ML	MA	ML
size	\hat{T}_n^{res}	0.062	0.114	0.000	0.104	0.072	0.068	0.000	0.084
	\hat{T}_n^{square}	0.048	0.044	0.048	0.044	0.038	0.048	0.050	0.046
$\omega \rightarrow 3$	\hat{T}_n^{res}	1.000	1.000	0.878	1.000	1.000	0.986	0.826	1.000
	\hat{T}_n^{square}	0.620	0.558	0.636	0.566	0.556	0.520	0.590	0.504
$\omega \rightarrow 10$	\hat{T}_n^{res}	1.000	1.000	1.000	1.000	0.962	0.920	0.770	1.000
	\hat{T}_n^{square}	0.996	0.996	0.996	1.000	0.992	1.000	0.986	1.000
$\alpha \rightarrow 0.5$	\hat{T}_n^{res}	1.000	1.000	1.000	1.000	0.974	0.918	0.790	1.000
	\hat{T}_n^{square}	0.996	0.994	0.988	1.000	0.986	0.980	0.956	1.000
$\beta \rightarrow 0.5$	\hat{T}_n^{res}	1.000	1.000	1.000	1.000	0.976	0.928	0.824	1.000
	\hat{T}_n^{square}	0.990	0.988	0.902	0.994	0.946	0.948	0.870	1.000

Table A3. Empirical sizes and powers for the INGARCH(1,1) model, Case 3.

$\omega = 3$	Method	SVR				TSVR			
$\alpha = 0.6$	Target	\hat{Y}_t	\hat{X}_t		\hat{Y}_t	\hat{X}_t			
$\beta = 0.3$	Proxy	MA	ML	MA	ML	MA	ML	MA	ML
size	\hat{T}_n^{res}	0.176	0.278	0.002	0.230	0.154	0.180	0.000	0.180
	\hat{T}_n^{square}	0.036	0.036	0.042	0.046	0.032	0.048	0.040	0.042
$\omega \rightarrow 5$	\hat{T}_n^{res}	1.000	1.000	0.990	1.000	0.970	0.928	0.746	1.000
	\hat{T}_n^{square}	0.964	0.948	0.856	0.948	0.956	0.956	0.866	0.972
$\alpha \rightarrow 0.3$	\hat{T}_n^{res}	1.000	1.000	1.000	1.000	0.998	0.996	1.000	1.000
	\hat{T}_n^{square}	0.998	0.998	1.000	0.998	0.994	0.972	0.996	1.000
$\beta \rightarrow 0.1$	\hat{T}_n^{res}	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	\hat{T}_n^{square}	0.992	0.998	0.974	0.998	0.958	0.908	0.860	0.998

Table A4. Empirical sizes and powers for the INGARCH(1,1) model, Case 4.

$\omega = 3$	Method	SVR				TSVR			
$\alpha = 0.3$	Target	\hat{Y}_t	\hat{X}_t		\hat{Y}_t	\hat{X}_t			
$\beta = 0.6$	Proxy	MA	ML	MA	ML	MA	ML	MA	ML
size	\hat{T}_n^{res}	0.258	0.260	0.020	0.174	0.172	0.158	0.018	0.100
	\hat{T}_n^{square}	0.036	0.040	0.032	0.030	0.032	0.026	0.024	0.030
$\omega \rightarrow 5$	\hat{T}_n^{res}	0.992	0.996	0.820	0.998	0.934	0.904	0.422	0.848
	\hat{T}_n^{square}	0.492	0.472	0.326	0.428	0.602	0.586	0.570	0.776
$\alpha \rightarrow 0.1$	\hat{T}_n^{res}	1.000	1.000	0.994	1.000	0.996	0.994	0.994	1.000
	\hat{T}_n^{square}	0.800	0.780	0.436	0.604	0.634	0.674	0.406	0.644
$\beta \rightarrow 0.3$	\hat{T}_n^{res}	1.000	1.000	0.996	1.000	0.998	0.994	1.000	1.000
	\hat{T}_n^{square}	0.926	0.920	0.716	0.862	0.882	0.890	0.712	0.956

Table A5. Empirical sizes and powers for the INAR(1) model, Case 1.

$\omega = 3$ $\phi = 0.3$	Method	SVR				TSVR			
	Target	\hat{Y}_t		\hat{X}_t		\hat{Y}_t		\hat{X}_t	
	Proxy	MA	ML	MA	ML	MA	ML	MA	ML
size	\hat{T}_n^{res}	0.052	0.070	0.000	0.070	0.064	0.060	0.000	0.060
	\hat{T}_n^{square}	0.048	0.060	0.064	0.054	0.066	0.052	0.058	0.062
$\omega \rightarrow 5$	\hat{T}_n^{res}	1.000	1.000	0.466	1.000	0.996	0.990	0.156	1.000
	\hat{T}_n^{square}	0.960	0.994	0.800	1.000	0.982	0.990	0.810	1.000
$\omega \rightarrow 10$	\hat{T}_n^{res}	1.000	1.000	1.000	1.000	0.984	0.966	0.838	0.976
	\hat{T}_n^{square}	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
$\phi \rightarrow 0.1$	\hat{T}_n^{res}	0.906	0.932	0.000	0.940	0.922	0.922	0.000	0.934
	\hat{T}_n^{square}	0.218	0.150	0.166	0.080	0.084	0.078	0.178	0.072
$\phi \rightarrow 0.5$	\hat{T}_n^{res}	1.000	1.000	0.106	1.000	0.994	0.992	0.026	1.000
	\hat{T}_n^{square}	0.600	0.598	0.164	0.584	0.606	0.562	0.186	0.542

Table A6. Empirical sizes and powers for the INAR(1) model, Case 2.

$\omega = 5$ $\phi = 0.3$	Method	SVR				TSVR			
	Target	\hat{Y}_t		\hat{X}_t		\hat{Y}_t		\hat{X}_t	
	Proxy	MA	ML	MA	ML	MA	ML	MA	ML
size	\hat{T}_n^{res}	0.068	0.084	0.000	0.070	0.068	0.068	0.000	0.060
	\hat{T}_n^{square}	0.062	0.044	0.038	0.048	0.042	0.048	0.036	0.040
$\omega \rightarrow 3$	\hat{T}_n^{res}	1.000	1.000	0.630	1.000	1.000	0.998	0.692	1.000
	\hat{T}_n^{square}	0.578	0.566	0.716	0.442	0.492	0.552	0.708	0.452
$\omega \rightarrow 10$	\hat{T}_n^{res}	1.000	1.000	0.992	1.000	0.976	0.962	0.688	1.000
	\hat{T}_n^{square}	0.986	0.998	0.996	1.000	1.000	1.000	0.998	1.000
$\phi \rightarrow 0.1$	\hat{T}_n^{res}	0.990	1.000	0.002	0.996	0.998	0.996	0.000	0.996
	\hat{T}_n^{square}	0.282	0.224	0.162	0.096	0.122	0.148	0.170	0.112
$\phi \rightarrow 0.5$	\hat{T}_n^{res}	1.000	1.000	0.340	1.000	1.000	0.996	0.138	1.000
	\hat{T}_n^{square}	0.800	0.822	0.172	0.830	0.812	0.774	0.204	0.812

Table A7. Empirical sizes and powers for the INAR(1) model, Case 3.

$\omega = 3$ $\phi = 0.7$	Method	SVR				TSVR			
	Target	\hat{Y}_t		\hat{X}_t		\hat{Y}_t		\hat{X}_t	
	Proxy	MA	ML	MA	ML	MA	ML	MA	ML
size	\hat{T}_n^{res}	0.116	0.072	0.002	0.054	0.092	0.056	0.000	0.048
	\hat{T}_n^{square}	0.048	0.058	0.034	0.054	0.054	0.052	0.038	0.058
$\omega \rightarrow 5$	\hat{T}_n^{res}	1.000	1.000	0.762	1.000	0.978	0.916	0.426	0.976
	\hat{T}_n^{square}	0.940	0.960	0.648	0.942	0.940	0.936	0.728	0.990
$\phi \rightarrow 0.3$	\hat{T}_n^{res}	1.000	1.000	0.896	1.000	1.000	1.000	0.946	1.000
	\hat{T}_n^{square}	0.852	0.846	0.284	0.836	0.854	0.906	0.210	0.898

Table A8. Empirical sizes and powers for the softplus INGARCH(1,1) model.

$c = 1, \omega = 3$ $\alpha = 0.3, \beta = 0.3$		Method	SVR				TSVR			
		Target	\hat{Y}_t		\hat{X}_t		\hat{Y}_t		\hat{X}_t	
		Proxy	MA	ML	MA	ML	MA	ML	MA	ML
size	\hat{T}_n^{res}	0.072	0.114	0.000	0.108	0.080	0.062	0.000	0.072	
	\hat{T}_n^{square}	0.034	0.032	0.034	0.038	0.032	0.036	0.030	0.046	
power	$\omega \rightarrow 5$	\hat{T}_n^{res}	1.000	0.998	0.666	1.000	0.978	0.862	0.284	0.996
		\hat{T}_n^{square}	0.966	0.982	0.792	0.992	0.984	0.962	0.828	0.986
	$\alpha \rightarrow 0.5$	\hat{T}_n^{res}	1.000	0.998	0.984	1.000	0.980	0.834	0.658	0.982
		\hat{T}_n^{square}	0.982	0.986	0.932	0.994	0.982	0.988	0.924	1.000
	$\beta \rightarrow 0.5$	\hat{T}_n^{res}	1.000	1.000	0.984	1.000	0.980	0.842	0.704	0.978
		\hat{T}_n^{square}	0.948	0.938	0.810	0.950	0.926	0.936	0.830	0.964

Table A9. Computing times for training the SVR and the TSVR methods.

Method		SVR				TSVR				
		Target	\hat{Y}_t		\hat{X}_t		\hat{Y}_t		\hat{X}_t	
		Proxy	MA	ML	MA	ML	MA	ML	MA	ML
INGARCH	$\omega = 3, \alpha = 0.3, \beta = 0.3$	918.02 (190.61)	863.33 (201.94)	1466.45 (247.97)	825.02 (305.41)	193.70 (19.79)	192.68 (19.11)	185.44 (15.94)	193.09 (21.02)	
	$\omega = 5, \alpha = 0.3, \beta = 0.3$	1633.97 (319.64)	1065.13 (229.89)	1231.81 (191.73)	1029.88 (363.53)	224.41 (22.42)	245.24 (27.33)	155.52 (16.06)	162.26 (17.89)	
	$\omega = 3, \alpha = 0.6, \beta = 0.3$	1979.82 (242.94)	1552.36 (366.01)	2456.07 (395.27)	1766.37 (666.71)	195.97 (20.59)	207.49 (21.05)	191.79 (19.60)	192.45 (19.34)	
	$\omega = 3, \alpha = 0.3, \beta = 0.6$	1877.27 (389.79)	1940.01 (347.92)	2697.95 (356.38)	2219.46 (587.99)	1198.25 (20.95)	196.93 (20.56)	186.23 (18.98)	189.34 (21.95)	
INAR	$\omega = 3, \phi = 0.3$	780.01 (189.10)	1379.59 (358.28)	1340.27 (256.68)	754.08 (255.29)	191.38 (20.17)	189.67 (19.59)	182.64 (16.42)	192.35 (26.07)	
	$\omega = 5, \phi = 0.3$	1000.50 (235.13)	972.17 (240.82)	1128.04 (217.57)	1336.76 (516.75)	240.72 (31.46)	226.47 (23.05)	150.59 (12.85)	105.45 (14.81)	
	$\omega = 10, \phi = 0.3$	828.82 (208.43)	1437.23 (378.40)	1542.48 (239.95)	769.35 (288.29)	194.25 (19.67)	193.05 (17.84)	184.82 (16.14)	194.41 (21.17)	
softplus INGARCH	$c = 1, \omega = 3, \alpha = 0.3, \beta = 0.3$	692.41 (118.22)	711.94 (146.43)	911.11 (157.28)	692.71 (206.60)	193.99 (19.01)	193.54 (18.86)	186.39 (17.34)	196.66 (20.83)	

References

- Al-Osh, M.A.; Aly, E.-E.A.A. First order autoregressive time series with negative binomial and geometric marginals. *Commun. Stat. Theory Methods* **1992**, *21*, 2483–2492. [CrossRef]
- Alzaid, A.A.; Al-Osh, M. An integer-valued pth-order autoregressive structure (INAR(p)) process. *J. Appl. Probab.* **1990**, *27*, 314–324. [CrossRef]
- Ferland, R.; Latour, A.; Oraichi, D. Integer-valued GARCH process. *J. Time Ser. Anal.* **2006**, *27*, 923–942. [CrossRef]
- Fokianos, K.; Rahbek, A.; Tjøstheim, D. Poisson autoregression. *J. Am. Stat. Assoc.* **2009**, *104*, 1430–1439. [CrossRef]
- McKenzie, E. Some simple models for discrete variate time series1. *J. Am. Water Resour. Assoc.* **1985**, *21*, 645–650. [CrossRef]
- Wei, C.H. *An Introduction to Discrete-Valued Time Series*; Wiley: New York, NY, USA, 2018.
- Kang, J.; Lee, S. Parameter change test for random coefficient integer-valued autoregressive processes with application to polio data analysis. *J. Time Ser. Anal.* **2009**, *30*, 239–258. [CrossRef]
- Kim, H.; Lee, S. Improved CUSUM monitoring of Markov counting process with frequent zeros. *Qual. Reliab. Eng. Int.* **2019**, *35*, 2371–2394. [CrossRef]
- Lee, Y.; Lee, S.; Tjøstheim, D. Asymptotic normality and parameter change test for bivariate Poisson INGARCH models. *Test* **2018**, *27*, 52–69. [CrossRef]
- Kim, B.; Lee, S. Robust change point test for general integer-valued time series models based on density power divergence. *Entropy* **2020**, *22*, 493. [CrossRef] [PubMed]
- Christou, V.; Fokianos, K. Quasi-likelihood inference for negative binomial time series models. *J. Time Ser. Anal.* **2014**, *35*, 55–78. [CrossRef]

12. Davis, R.A.; Liu, H. Theory and inference for a class of nonlinear models with application to time series of counts. *Stat. Sin.* **2016**, *26*, 1673–1707. [[CrossRef](#)]
13. Jazi, M.A.; Jones, G.; Lai, C.D. First-order integer valued AR processes with zero inflated Poisson innovations. *J. Time Ser. Anal.* **2012**, *33*, 954–963. [[CrossRef](#)]
14. Kim, B.; Lee, S. Robust estimation for general integer-valued time series models. *Ann. Inst. Stat. Math.* **2019**, *72*, 1371–1396. [[CrossRef](#)]
15. Zhu, F. A negative binomial integer-valued GARCH model. *J. Time Ser. Anal.* **2011**, *32*, 54–67. [[CrossRef](#)]
16. Ahmad, A.; Francq, C. Poisson QMLE of count time series models. *J. Time Ser. Anal.* **2016**, *37*, 291–314. [[CrossRef](#)]
17. Bezerra, P.C.S.; Albuquerque, P.H.M. Volatility forecasting via SVR–GARCH with mixture of Gaussian kernels. *Comput. Manag. Sci.* **2017**, *14*, 179–196. [[CrossRef](#)]
18. Cao, L.; Tay, F.E. Financial forecasting using support vector machines. *Neural. Comput. Appl.* **2001**, *10*, 184–192. [[CrossRef](#)]
19. Chen, S.; Härdle, W.K.; Jeong, K. Forecasting volatility with support vector machine-based GARCH model. *J. Forecast.* **2010**, *29*, 406–433. [[CrossRef](#)]
20. Cherkassky, V.; Ma, Y. Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Netw.* **2004**, *17*, 113–126. [[CrossRef](#)]
21. Lee, S.; Lee, S.; Moon, M. Hybrid change point detection for time series via support vector regression and CUSUM method. *Appl. Soft Comput.* **2020**, *89*, 106101. [[CrossRef](#)]
22. Lee, S.; Kim, C.K.; Lee, S. Hybrid CUSUM change point test for time series with time-varying volatilities based on support vector regression. *Entropy* **2020**, *22*, 578. [[CrossRef](#)]
23. Pérez-Cruz, F.; Afonso-Rodríguez, J.A.; Giner, J. Estimating GARCH models using support vector machines. *Quant. Finance* **2003**, *3*, 163–172. [[CrossRef](#)]
24. Shim, J.; Kim, Y.; Lee, J.; Hwang, C. Estimating value at risk with semiparametric support vector quantile regression. *Comput. Stat.* **2012**, *27*, 685–700. [[CrossRef](#)]
25. Shim, J.; Hwang, C.; Seok, K. Support vector quantile regression with varying coefficients. *Comput. Stat.* **2016**, *31*, 1015–1030. [[CrossRef](#)]
26. Vapnik, V.N. *The Nature Of Statistical Learning Theory*; Springer: New York, NY, USA, 2000. [[CrossRef](#)]
27. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [[CrossRef](#)]
28. Lee, Y.J.; Hsieh, W.F.; Huang, C.M. ϵ -SSVR: A smooth support vector machine for ϵ -insensitive regression. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 678–685. [[CrossRef](#)]
29. Suykens, J.A.K.; Vandewalle, J. Least squares support vector machine classifiers. *Neural Process. Lett.* **1999**, *9*, 293–300. [[CrossRef](#)]
30. Peng, X. TSVR: An efficient twin support vector machine for regression. *Neural Netw.* **2010**, *23*, 365–372. [[CrossRef](#)] [[PubMed](#)]
31. Gupta, D.; Pratama, M.; Ma, Z.; Li, J.; Prasad, M. Financial time series forecasting using twin support vector regression. *PLoS ONE* **2019**, *14*, 1–27. [[CrossRef](#)]
32. Tomar, D.; Agarwal, S. Twin support vector machine: A review from 2007 to 2014. *Egypt. Inform. J.* **2015**, *16*, 55–69. [[CrossRef](#)]
33. Zhong, P.; Xu, Y.; Zhao, Y. Training twin support vector regression via linear programming. *Neural. Comput. Appl.* **2012**, *21*, 399–407. [[CrossRef](#)]
34. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; pp. 1942–1948. [[CrossRef](#)]
35. Wang, D.; Tan, D.; Liu, L. Particle swarm optimization algorithm: An overview. *Soft Comput.* **2018**, *22*, 387–408. [[CrossRef](#)]
36. Zhang, Y.; Wang, S.; Ji, G. A comprehensive survey on particle swarm optimization algorithm and its applications. *Math. Probl. Eng.* **2015**, *2015*, 1–38. [[CrossRef](#)]
37. Page, E.S. A test for a change in a parameter occurring at an unknown point. *Biometrika* **1955**, *42*, 523. [[CrossRef](#)]
38. Chen, J.; Gupta, A.K. *Parametric Statistical Change Point Analysis: With Applications To Genetics, Medicine, and Finance*, 2nd ed.; Birkhäuser: Boston, MA, USA, 2012.
39. Csörgö, M.; Horváth, L. *Limit Theorems In Change-Point Analysis*; John Wiley & Sons Inc.: New York, NY, USA, 2012.
40. Berkes, I.; Gombay, E.; Horváth, L.; Kokoszka, P. Sequential change-point detection in GARCH(p,q) models. *Econ. Theory* **2004**, *20*, 1140–1167. [[CrossRef](#)]
41. Inclán, C.; Tiao, G.C. Use of cumulative sums of squares for retrospective detection of changes of variance. *J. Am. Stat. Assoc.* **1994**, *89*, 913–923. [[CrossRef](#)]
42. Lee, S.; Ha, J.; Na, O.; Na, S. The cusum test for parameter change in time series models. *Scand. Stat. Theory Appl.* **2003**, *30*, 781–796. [[CrossRef](#)]
43. Oh, H.; Lee, S. Modified residual CUSUM test for location-scale time series models with heteroscedasticity. *Ann. Inst. Stat. Math.* **2018**, *71*, 1059–1091. [[CrossRef](#)]
44. Ross, G.J. Modelling financial volatility in the presence of abrupt changes. *Physica A* **2013**, *392*, 350–360. [[CrossRef](#)]
45. Kang, J.; Lee, S. Parameter change test for Poisson autoregressive models. *Scand. Stat. Theory Appl.* **2014**, *41*, 1136–1152. [[CrossRef](#)]
46. Fokianos, K.; Fried, R. Interventions in INGARCH processes. *J. Time Ser. Anal.* **2010**, *31*, 210–225. [[CrossRef](#)]
47. Franke, J.; Kirch, C.; Kamgaing, J.T. Change points in times series of counts. *J. Time Ser. Anal.* **2012**, *33*, 757–770. [[CrossRef](#)]
48. Fokianos, K.; Gombay, E.; Hussein, A. Retrospective change detection for binary time series models. *J. Stat. Plan. Inference* **2014**, *145*, 102–112. [[CrossRef](#)]

49. Rakitzis, A.C.; Castagliola, P.; Maravelakis, P.E. On the modelling and monitoring of general inflated poisson processes. *Qual. Reliab. Eng. Int.* **2016**, *32*, 1837–1851. [[CrossRef](#)]
50. Lee, Y.; Lee, S. CUSUM test for general nonlinear integer-valued GARCH models: Comparison study. *Ann. Inst. Stat. Math.* **2019**, *71*, 1033–1057. [[CrossRef](#)]
51. De Pooter, M.; Van Dijk, D. Testing for Changes in Volatility in Heteroskedastic Time Series—A Further Examination. Available online: <https://repub.eur.nl/pub/1627/> (accessed on 28 July 2020).
52. Lee, S.; Tokutsu, Y.; Maekawa, K. The cusum test for parameter change in regression models with ARCH errors. *J. Jpn. Stat. Soc.* **2004**, *34*, 173–188. [[CrossRef](#)]
53. Lee, S. Location and scale-based CUSUM test with application to autoregressive models. *J. Stat. Comput. Simul.* **2020**, *90*, 2309–2328. [[CrossRef](#)]
54. Lee, S.; Lee, S. Exponential family QMLE-based CUSUM test for integer-valued time series. *Commun. Stat. Simul. Comput.* **2021**, accepted.
55. Lee, S. Residual-based CUSUM of squares test for poisson integer-valued GARCH models. *J. Stat. Comput. Simul.* **2019**, *89*, 3182–3195. [[CrossRef](#)]
56. Lee, S.; Kim, D.; Seok, S. Modelling and inference for counts time series based on zero-inflated exponential family INGARCH models. *J. Stat. Comput. Simul.* **2021**, in press. [[CrossRef](#)]
57. Billingsley, P. *Convergence of Probability Measures*, 2nd ed.; John Wiley & Sons: Hoboken, NJ, USA, 1999.
58. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
59. Khemchandani, R.; Chandra, S. Twin Support Vector Machines for Pattern Classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 905–910. [[CrossRef](#)]
60. Bendtsen, C. Pso: Particle Swarm Optimization. R Package Version 1.0.3. Foundation for Statistical Computing. Available online: <https://cran.r-project.org/web/packages/pso> (accessed on 27 July 2020).
61. Karatzoglou, A.; Smola, A.; Hornik, K.; Zeileis, A. Kernlab—An S4 package for kernel methods in R. *J. Stat. Softw.* **2004**, *11*, 1–20. [[CrossRef](#)]
62. Stellato, B.; Banjac, G.; Goulart, P.; Boyd, S. Osqp: Quadratic Programming Solver Using the ‘osqp’ Library. R Package Version 0.6.0.3; Foundation for Statistical Computing. Available online: <https://cran.r-project.org/web/packages/osqp> (accessed on 27 July 2020).
63. Weiss, C.H.; Zhu, F.; Hoshiyar, A. Softplus INGARCH models. *Stat. Sin.* **2020**, in press.