Data cloning estimator for Stochastic Volatility in Mean model

Romero-Ramos E.[1], Ropero-Moriones E.[2]

(1) Department of Statistics and O.R.
Complutense University, Spain.
email:evarom03@ucm.es
(2) School of Applied Social Sciences and Communication
UNIE University. Spain.
email: eva.ropero@universidadunie.com

Abstract

The Stochastic Volatility in Mean (SVM) model represents an advancement of Stochastic Volatility (SV) models, wherein the latent volatility is incorporated as an explanatory variable in both the mean and variance equations. This integration facilitates the assessment of the relationship between returns and volatility, albeit at the expense of complicating the estimation process.

This study introduces a Bayesian methodology that leverages data cloning algorithms to obtain maximum likelihood estimations for SV and SVM model parameters. By adopting this Bayesian framework, approximate maximum likelihood estimations can be attained without the need to maximize pseudo-likelihood functions. The key contribution of this paper lies in the proposition of an estimator for the SVM Model, which effectively approximates the maximum likelihood estimator through the utilization of Bayesian algorithms. Notably, these estimations yield superior outcomes when compared to those derived from the Markov Chain Monte Carlo (MCMC) method in terms of standard errors, all the while being independent of the selection of prior distributions.

Keywords: Data cloning, Bayesian inference, Stochastic Volatility, Stochastic Volatility in Mean, Bitcoin.

AMS Subject Classification: C11; C15; C22.

1 Introduction

When analyzing time series of financial asset returns it is necessary to consider their specific properties, paying special attention to the behavior of volatility, such as heteroscedasticy, volatility clustering (Mandelbrot, 1963 and Tseng and Li, 2011), and excess leptokurtosis (Koopman and Uspensky, 2002). To account for these properties, models such as GARCH (Bollerslev, 1986 and Katsiampa, 2017) and Stochastic Volatility (SV) (Taylor, 1982 and Taylor, 1994) have been developed.

GARCH models define the conditional variance as a function of past squared innovations and lagged conditional variances (Manera et al., 2016 and Chan and Grant, 2016). In contrast, variance in SV models is characterized as an unobserved component that follows a stochastic process (Koopman and Uspensky, 2002, Trolle and Schwartz, 2009, Brooks and Prokopczuk, 2013).

Moreover, the SV model captures the deviation of returns from the mean using a function of two disturbance terms, whereas the GARCH model relies on a single disturbance term (Koopman and Uspensky, 2002). This added complexity in the SV model allows for more flexibility (Asai et al., 2006, Balcilar and Ozdemir, 2019) and improved accuracy in capturing the volatility clustering of financial series (Kim et al., 1998, Yu, 2002, Carr et al., 2003, Chan and Grant, 2016, Tiwari et al., 2019, Agbeyegbe, 2022). Additionally, SV model is better equipped to handle the negative relationship between volatility and returns. Furthermore, SV models are more robust to misspecification and to radical changes in the data (Tiwari et al., 2019, Balcilar and Ozdemir, 2019), in addition to better estimating the properties of the financial series.

Recently, Stochastic Volatility in Mean (SVM) model has emerged as a further refinement of the SV model. This model allows for the simultaneous modeling of the mean and variance of financial time series data, allowing the simultaneous analysis of the relationship between volatility and returns, which is an important aspect of financial modeling (Koopman and Uspensky, 2002). Other models, such as ARCH-M and GARCH-M, also attempt to estimate this relationship, but they do not provide simultaneous estimation of the ex ante relationship between volatility and returns. Therefore, it is expected that SVM models will provide more accurate estimates of the behavior of financial time series data when analyzing leverage effects (Bouchaud et al., 2001) and the effect of volatility feedback (Koopman and Uspensky, 2002).

While SV models have been shown to be superior to GARCH models in the literature, they are not as widely used due to their complexity in estimation. This is because of the difficulty to directly evaluate the likelihood function and because they require estimating both return and volatility at the same time.

Various techniques have been used to estimate Stochastic Volatility (SV) models, including methods based on the method of moments (Taylor, 1986, Melino and Turnbull, 1990, Renault, 2009) and likelihood-based methods. The estimators of moments have the advantage of not requiring a likelihood assessment to obtain them, but their efficiency is known to be suboptimal compared to likelihood-based inference methods (Sandmann and Koopman, 1998). However, likelihood-based methods have limitations such as being computationally intensive, requiring excessive simulation efforts, and making assumptions that can be difficult to satisfy. Financial markets often require real-time decision making, which requires computationally fast and robust estimators that require less sampling (Yang et al., 2021).

Bayesian methods, such as Monte Carlo Markov Chain (MCMC) (e.g., Shephard, 1993, Jacquier et al., 1994, Kim et al., 1998, Broto and Ruiz, 2004, Andrieu et al., 2010, Beskos et al., 2013, Kastner et al., 2017, Li et al., 2019) and Integrated Nested Laplace Approximations (Martino et al., 2011), are a good solution for estimating the parameters of SV models as they allow for efficient evaluation of the posterior distribution of parameters and volatility. However, these methods also have limitations such as requiring a prior distribution for parameters, a numerical evaluation of the likelihood function, and potential problems with the convergence of simulated chains (Rue et al., 2009).

This paper proposes the use of a different approach called data cloning (Lele et al., 2007), for parameter estimation, utilizing the computational simplicity of MCMC algorithms while enabling frequentist inferences such as maximum likelihood estimates and standard errors. The method involves applying a Bayesian methodology to a data set constructed by cloning the original data set as many times as necessary so that the solution approximates the maximum likelihood solution (Ponciano et al., 2009 and Chaim and Laurini, 2022). The main advantage of using data cloning over other Bayesian methods is that the inferences are invariant to the choice of the prior distributions, and does not require likelihood estimation. Overall, data cloning is a powerful method for estimating and studying complex models, specially when analyzing volatility.

We propose the use of this methodology to estimate parameters of SV and SVM models, as it has been shown to be particularly useful for complex models, as discussed in studies by Lele et al., 2007, Ponciano et al., 2009, Sólymos, 2010 and Chaim and Laurini, 2022. Recently, this method has been succesfully used to estimate parameters of other complex financial models in Marín et al., 2015 and de Zea Bermudez et al., 2020. Although it is beyond the scope of this article, models are recently being developed to estimate volatility in the valuation of financial options, using two volatility components (Pasricha and He, 2023 and Lin and He, 2023). These models are strong candidates for using an algorithm similar to the one we constructed in this paper to estimate their parameters.

This paper makes three important contributions to the literature. First, it provides an algorithm to estimate SV and SVM model parameters based on data cloning method. This is a simpler way of estimating SVM that allows obtaining frequentist inferences without estimating likelihood. Second, we perform an analysis with simulated data using the proposed algorithm and show that its estimates are more accurate than the obtained using MCMC. Third, in order to evaluate the predictive ability of the model over a real financial series, the methodology is applyied to model Bitcoin returns, obtaining new conclusions about the relationship between volatility and profitability in cryptocurrencies, conclusions that can only be obtained with the SVM method.

The structure of the article is as follows. In section 2, we specify the SV and SVM models in order to be able to estimate it later on. In section 3, we explain the data cloning method in general, and then in section 4 we develop the algorithms to apply this method to SV and SVM models. In this section, we also obtain the results and compare them with the MCMC methodology, demonstrating that the data cloning methodology is superior. In section 5 we apply SVM to a real example of financial series (Bitcoin) and analyze the relationship between return and volatility, checking if the hypotheses of leverage effect and volatility feedback are fulfilled. Finally, in section 6 we present the main conclusions of the paper.

2 Definition and specification of SV and SVM models

Definition 1. The Stochastic Volatility model defines the returns of the process Y_t in discrete time t as

$$Y_t = \mu_t + \sigma_t \epsilon_t, \quad \epsilon_t \sim NID(0,1), \tag{1}$$

$$\mu_t = a + \sum_{i=1}^{k} b_i x_{i,t}, \text{ for t = 1,2,... .} \tag{2}$$

Here $x_{i,t}$ can be both, independent variables or lags of the dependent variable. The mean μ_t also depends on a constant a and b_i for $i = 1, \cdots, k$ regression coefficients. The volatility process, σ_t^2 is defined as

$$\sigma_t^2 = \sigma^{*2} e^{h_t}, \tag{3}$$

where σ^{*2} is a positive scaling factor and h_t is a stochastic process defined as

$$h_t = \phi h_{t-1} + \sigma_\eta \eta_t, \quad \eta_t \sim NID(0,1). \tag{4}$$

In (4) ϕ and σ_η are model parameters. Parameter σ_η is the variance of the independent and identically distributed normal variables η_t, while ϕ is the volatility persistence parameter. It is important for ϕ to be positive and smaller than 1 ($\phi \in (0,1)$) to ensure stationarity.

It could be assume that in (3), σ_t^2 is specified in logarithmic form, considering that $h_t = ln\left(\sigma_t^2/\sigma^{*2}\right)$.

SV model has two sources of variability by means of two independent and mutually incorrelated disturbance terms, ϵ_t and η_t. This constitutes the main difference with GARCH models (Bollerslev, 1986 and Koopman and Hol Uspensky, 2002). The unconditional variance implied in SV model is

$$\sigma^{*2} e^{\frac{\sigma_\eta}{2(1-\phi^2)}}.$$

One important characteristic of SV models is that they capture part of the excess of kurtosis that financial series present. The kurtosis of SV series is defined by

$$k_y = \frac{k_\epsilon E(\sigma_t^4)}{(E(\sigma_t^2)^2)} = 3e^{\frac{\sigma_\eta^2}{1-\phi^2}}.$$

Definition 2. The returns of Stochastic Volatility in Mean (SVM) model is defined as (1) and its mean is defined as

$$\mu_t = a + \sum_{i=1}^{k} b_i x_{i,t} + d\sigma_t^2, \tag{5}$$

where parameter d is measuring the effect of volatility in the mean of the process.

The variance of SVM model is defined by equations (3) and (4).

The inclusion of variance in the mean equation allows for a better understanding of the relationship between returns and volatility. It makes possible to perform studies like French et al., 1987 and analyze the returns' partial dependence of volatility, as all financial theory assesses (Koopman and Hol Uspensky, 2002).

3 Data cloning estimation

The estimation of these models, particularly SVM, is not straightforward. Therefore, this paper proposes a technique based on data cloning to obtain approximations of the maximum likelihood estimators through Bayesian algorithms. The main idea is to clone the series k times and assume that each series represents an independent trajectory of the process. We consider all trajectories to be equal because the trajectory with the highest probability is the one obtained. Although the heuristic explanation alludes to the independence of the cloned trajectories, the mathematical proof of the algorithm does not rely on this assumption and in no case does it assume that the k clones are independent.

This method was introduced by Lele et al., 2007 and Lele et al., 2010 as a means to obtain maximum likelihood approaches for parameters of complex models where direct maximization of the likelihood is infeasible.

The data cloning method offers an effective solution for estimating the parameters of SV and SVM models as it avoids the need for direct maximization of the likelihood function. Instead, it utilizes Bayesian algorithms to approximate the likelihood. Moreover, this methodology is not reliant on the specific prior distributions chosen, resulting in improved solutions compared to those provided by MCMC estimators.

Previous studies by Laurini, 2013 and de de Zea Bermudez et al., 2020 have successfully applied this method to estimate the SV model, albeit using a less general model. Their findings demonstrate enhanced accuracy in parameter estimation compared to the standard Bayesian approach. Therefore, we aim to assess the effectiveness of this method in the context of a more general SV model and the SVM model.

The data cloning method begins with an observed data set $y = (y_1, y_2, ..., y_n)$ and the prior distributions for the parameters. It utilizes the posterior distribution of the parameter set θ, denoted as $\pi(\theta|y)$, which is proportional to the likelihood function $L(\theta|y)$ multiplied by the prior distribution $\pi(\theta)$. This posterior distribution is then used to generate samples using a MCMC method. In the data cloning method, samples are drawn from the posterior distribution $\pi^{(k)}(\theta|y)$, which is proportional to the k-th power of the likelihood function $[L(\theta|y)]^{(k)}$ multiplied by the prior distribution $\pi(\theta)$.

The data cloning method is based on the principle that when k is sufficiently large, $\pi^{(k)}(\theta|y)$ converges to a multivariate normal distribution with the maximum likelihood estimator of the model parameters as its mean. Additionally, the covariance of this multivariate normal distribution is equal to $1/k$ times the inverse of the Fisher information matrix for the maximum likelihood estimator (Lele et al., 2007). Based on this, the data cloning algorithm can be summarized in the following steps:

Step 1: Create k-cloned data set $\mathbf{y}^{(k)} = (\mathbf{y}, \mathbf{y}, \ldots, \mathbf{y})$ by cloning the observed data set k times. Each copy of y is treated as an independent sample path of the same process.

Step 2: Use a MCMC method to generate random values from the posterior distribution. Start the algorithm with the prior distribution $\pi(\theta)$ and the cloned data vector $\mathbf{y}^{(k)} = (\mathbf{y}, \mathbf{y}, \ldots, \mathbf{y})$.

Step 3: After running the MCMC method for a total of B iterations, compute the sample mean and variance of the obtained values for the marginal posterior distribution, denoted as $(\theta)_j$, where $j = 1, \ldots, B$. The sample means correspond to the maximum likelihood estimates, while the approximate variances of the maximum likelihood estimates are k times the posterior variances.

4 Data cloning algorithms to estimate SV and SVM models

In order to facilitate the estimation algorithms for both models, the estimation of the constant parameter will be excluded. Although it is possible to include this parameter in the algorithms, its inclusion significantly increases computation time as it requires a higher number of clones. However, after conducting several empirical tests, it has been observed that excluding the constant parameter does not significantly affect the results. Therefore, it has been decided to omit it in the simulations and work with variables in differences.

4.1 Data cloning estimator for SV model

The algorithm based on data cloning method will be able to estimate the model parameters for the SV model described in section two by equations (1), (3) and (4) and simplifying equation (2) to

$$\mu_t = b\tilde{y}_{t-1}, \tag{6}$$

being $\tilde{y} = y_t - \bar{y}$ (the returns in differences).

The model will be described to include just one autorregressive term. More autoregresive terms, or other kind of terms, could be easily included if necessary, but each included term will probably increase the required number of clones to achieve convergence, and consequently the computation time.

This model is characterized by four parameters: ϕ, σ_η, σ^{*2}, b.

To apply data cloning method is required to design a MCMC procedure, which makes necessary to choose prior distributions, even thought it is proved that they do not affect the final results (see Lele et al., 2007). Considering that, the following vaguely informative distributions will be chosen as prior distributions: $\phi \sim U(0,1)$, $\sigma_\eta \sim U(0,10)$, $\sigma^{*2} \sim U(0,10)$ and $b \sim U(-10,10)$.

The joint posterior distribution is obtained assuming that $Y_i \sim N(\mu_t, \sigma_t^2)$ with μ_t defined in (6) and σ_t^2 defined in (3), so the likelihood function of SV model is:

$$L\left(b, \sigma^{*2}, \phi, \sigma_\eta|\tilde{y}\right) = \left(\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^{*2}e^{h_i}}}\right) \exp\left\{-\frac{1}{2\sigma^{*2}}\sum_{i=1}^{n}\frac{\left(\tilde{y}_i - b\tilde{y}_{i-1}\right)^2}{\exp(h_i)}\right\},$$

being h_i defined by (4). With this likelihood function, the joint posterior is

$$\begin{aligned} \pi^{(k)}(\phi, \sigma_\eta, \sigma^{*2}, b) &\propto \left[L(\phi, \sigma_\eta, \sigma^{*2}, b|\tilde{y})\right]^k \pi(\phi)\pi(\sigma_\eta)\pi(\sigma^{*2})\pi(b) \\ &\propto \left(\prod_{i=1}^{n} 2\pi\sigma^{*2}e^{h_i}\right)^{-k/2} \exp\left\{-\tfrac{k}{2\sigma^{*2}}\sum_{i=1}^{n}\tfrac{\left(\tilde{y}_i - b\tilde{y}_{i-1}\right)^2}{\exp(h_i)}\right\} \\ &\quad \cdot I_{(0,1)}(\phi)I_{(0,10)}(\sigma_\eta)I_{(0,10)}(\sigma^{*2})I_{(-10,10)}(b). \end{aligned}$$

And the conditional posterior distributions for the parameters are

$$\begin{aligned} \pi^{(k)}(\phi|\sigma_\eta, \sigma^{*2}, b, \tilde{y}) &\propto \left(\prod_{i=1}^{N}\sigma^{*2}e^{h_i}\right)^{-\frac{k}{2}} \exp\left\{-\frac{k}{2\sigma^{*2}}\sum_{i=1}^{n}\frac{\left(\tilde{y}_i - b\tilde{y}_{i-1}\right)^2}{\exp(h_i)}\right\} I_{(0,1)}(\phi), \\ \pi^{(k)}(\sigma_\eta|, \phi, \sigma^{*2}, b, \tilde{y}) &\propto \left(\prod_{i=1}^{N}\sigma^{*2}e^{h_i}\right)^{-\frac{k}{2}} \exp\left\{-\frac{k}{2\sigma^{*2}}\sum_{i=1}^{n}\frac{\left(\tilde{y}_i - b\tilde{y}_{i-1}\right)^2}{\exp(h_i)}\right\} I_{(0,10)}(\sigma_\eta), \\ \pi^{(k)}(\sigma^{*2}|\phi, \sigma_\eta, b, \tilde{y}) &\propto \left(\prod_{i=1}^{N}\sigma^{*2}e^{h_i}\right)^{-\frac{k}{2}} \exp\left\{-\frac{k}{2\sigma^{*2}}\sum_{i=1}^{n}\frac{\left(\tilde{y}_i - b\tilde{y}_{i-1}\right)^2}{\exp(h_i)}\right\} I_{(0,10)}(\sigma^{*2}), \\ \pi^{(k)}(b|\phi, \sigma_\eta, \sigma^{*2}, \tilde{y}) &\propto \left(\prod_{i=1}^{N}\sigma^{*2}e^{h_i}\right)^{-\frac{k}{2}} \exp\left\{-\frac{k}{2\sigma^{*2}}\sum_{i=1}^{n}\frac{\left(\tilde{y}_i - b\tilde{y}_{i-1}\right)^2}{\exp(h_i)}\right\} I_{(-10,10)}(b). \end{aligned}$$

Data cloning algorithm starts from an initial solution $\phi^{(0)}$, $\sigma_\eta^{(0)}$, $\sigma^{*2(0)}$, $b^{(0)}$ and from the conditional posterior distributions, it generates values for $\phi^{(m)}$, $\sigma_\eta^{(m)}$, $\sigma^{*2(m)}$, $b^{(m)}$ in each iteration m. The initial values will be simulated directly from the prior distributions, since it is not necessary to use specific values to achieve convergence in a reasonable time.

After a large enough number of iterations, a sample will be obtained to constitute the posteriors whose means will be an approach to the maximum likelihood estimations of the model parameters. The steps of this algorithm can be summarized as follows:

Step 1: Set initial solution at $m = 0$ as: $\phi^{(0)}$, $\sigma_\eta^{(0)}$, $\sigma^{*2(0)}$ and $b^{(0)}$.

Step 2: Generate $\phi^{(m+1)}$ from its conditional posterior distribution

$$\phi^{(m+1)} \sim \pi^{(k)}(\phi|\sigma_\eta, \sigma^{*2}, b, \tilde{y}).$$

Step 3: Generate $\sigma_\eta^{(m)}$ from its conditional posterior distribution

$$\sigma_\eta^{(m)} \sim \pi^{(k)}(\sigma_\eta|\phi, \sigma^{*2}, b, \tilde{y}).$$

Step 4: Generate $\sigma^{*2(m)}$ from its conditional posterior distribution

$$\sigma^{*2(m)} \sim \pi^{(k)}(\sigma^{*2}|\phi, \sigma_\eta, b, \tilde{y}).$$

Step 5: Generate $b^{(m)}$ from its conditional posterior distribution

$$b^{(m)} \sim \pi^{(k)}(b|\phi, \sigma_\eta, \sigma^{*2}, \tilde{y}).$$

Step 6: Set $m = m + 1$ and go to Step 2.

This algorithm has been implemented using the package `dclone` (Sólymos, 2010) from the `R` project (R Core Team, 2012).

To test the performance of the algorithm in estimating the parameters of the SV model, a sample path of this model has been simulated. This allows for a comparison between the real parameters and the estimated ones. A simulator for this model has been developed using R to generate the series, which consists of 245 values, approximately representing the number of working days in a year. This is done to assess the performance of the algorithm when considering the annual evolution of the daily returns of a financial asset. The selected parameter values for simulating the model are: $\phi = 0.97$, $\sigma_\eta = 0.12$, $\sigma^{*2} = 0.2$ and $b = 0.2$.

The data cloning algorithm requires determining the optimal number of clones. This is achieved by evaluating the maximum eigenvalue of the posterior variance, the minimum squared error, the R^2 statistic, and the $\hat{R}$ criterion (Lele et al., 2010 and Brooks and Gelman, 1998). All these metrics can be computed using the dclone package. Based on these results, no significant improvements were found by using more than 20 clones, so the optimal number of clones is fixed at 20.

The results obtained by applying the algorithm to a single sample path are presented in Table 1. The table displays the real values for all parameters, the estimated parameters, the standard errors, and the 95% confidence intervals in columns. Additionally, the last two columns include the parameter estimates using an MCMC estimator and the corresponding estimation standard errors. This allows for a comparison with the results obtained using data cloning.

Parameter	Real Value	Data cloning Estimations	S.D.	95% confidence Intervals	MCMC Estimations	S.D. (MCMC)
ϕ	0.97	0.8879	0.03931	$(0.5433, 1.2324)$	0.8335	0.2077
σ_η	0.12	0.1478	0.03758	$(-0.1816, 0.4771)$	0.1910	0.1220
σ^{*2}	0.2	0.2130	0.06577	$(-0.3635, 0.7895)$	0.2036	0.0392
b	0.2	0.2192	0.01462	$(0.0911, 0.3474)$	0.1199	0.0676

Table 1: Estimation for Stochastic Volatility model parameters using data cloning method.

It can be observed that, considering only one simulated sample path, the estimator produces values that closely match the real values used to generate the path. Additionally, the standard errors of the estimation are very small for all cases, indicating that the estimator yields good results based on a single sample path. Moreover, all real values fall within the 95% confidence intervals, as expected.

Comparing these results with those obtained using a traditional MCMC estimator, data cloning demonstrates superior performance in almost all cases. It provides estimates with smaller standard errors that do not depend on the selected priors.

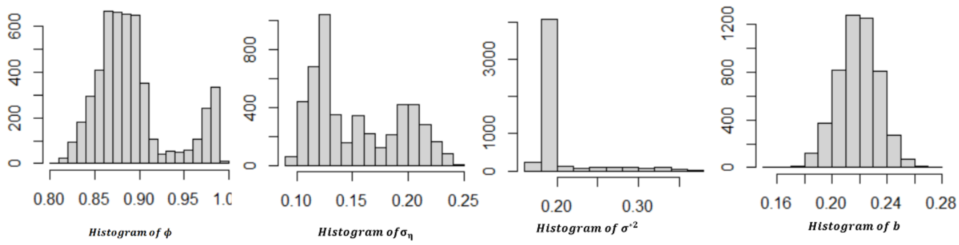

Figure 1: Histograms of the posterior distributions of the Stochastic Volatility model parameters

Figure 1 shows the posterior distributions obtained by the data cloning algorithm, letting understand better the behavior of the estimates. We observe a slight tendency to underestimate the value of ϕ, but for the remaining parameters, the higher probabilities of the posterior density function closely align with the true parameter values.

It should be noted that data cloning estimators are approximations to maximum likelihood estimators, so they will have the same analytical properties.

4.2 Data cloning estimator for SVM model

The estimator for SVM based on data cloning method also requires simplifying the mean equation (5), in order to work with the returns in differences and fix the variables to be used. Thus, the equation of the mean is defined by

$$\mu_t = b\tilde{y}_{t-1} + d\sigma^{*2}e^{h_t}. \tag{7}$$

Again, a unique autorregressive term has been included to simplify the algorithm execution. Thus, the model has five parameters: ϕ, σ_η, σ^{*2}, b and d, one more than SV model, to include h_t in the mean equation.

The prior distributions to be used in the algorithm are: $\phi \sim U(0,1)$, $\sigma_\eta \sim U(0,10)$, $\sigma^{*2} \sim U(0,10)$, $b \sim U(-10,10)$ and $d \sim U(-10,10)$.

The joint posterior distribution will be obtained considering that $Y_i \sim N(\mu_t, \sigma_t^2)$ with μ_t defined in (7) and σ_t^2 defined in (3), so the likelihood function of SVM model is

$$L\left(\phi, \sigma_\eta, \sigma^{*2}, b, d|\tilde{y}\right) = \left(\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^{*2}e^{h_i}}}\right) \exp\left\{-\frac{1}{2\sigma^{*2}} \sum_{i=1}^{n} \frac{\left(\tilde{y}_i - b\tilde{y}_{i-1} - d\sigma^{*2}\exp(h_i)\right)^2}{\exp(h_i)}\right\},$$

being h_i defined by (4).

Based on this likelihood function the joint posterior is

$$\begin{aligned}
\pi^{(k)}(\phi, \sigma_\eta, \sigma^{*2}, b, d) &\propto \left[L(\phi, \sigma_\eta, \sigma^{*2}, b, d|\tilde{y})\right]^k \pi(\phi)\pi(\sigma_\eta)\pi(\sigma^{*2})\pi(b)\pi(d) \\
&\propto \left(\prod_{i=1}^n 2\pi\sigma^{*2}e^{h_i}\right)^{-k/2} \exp\left\{-\frac{k}{2\sigma^{*2}} \sum_{i=1}^n \frac{\left(\tilde{y}_i - b\tilde{y}_{i-1} - d\sigma^{*2}\exp(h_i)\right)^2}{\exp(h_i)}\right\} \\
&\quad \cdot I_{(0,1)}(\phi) I_{(0,10)}(\sigma_\eta) I_{(0,10)}(\sigma^{*2}) I_{(-10,10)}(b) I_{(-10,10)}(d).
\end{aligned}$$

From this likelihood, the conditional posteriors are:

$$\pi^{(k)}(\phi|\sigma_\eta,\sigma^{*2},b,d,\tilde{y}) \propto \left(\prod_{i=1}^{N}\sigma^{*2}e^{h_i}\right)^{-\frac{k}{2}} \exp\left\{-\frac{k}{2\sigma^{*2}}\sum_{i=1}^{n}\frac{\left(\tilde{y}_i-b\tilde{y}_{i-1}-d\sigma^{*2}\exp(h_i)\right)^2}{\exp(h_i)}\right\} I_{(0,1)}(\phi),$$

$$\pi^{(k)}(\sigma_\eta|\phi,\sigma^{*2},b,d,\tilde{y}) \propto \left(\prod_{i=1}^{N}\sigma^{*2}e^{h_i}\right)^{-\frac{k}{2}} \exp\left\{-\frac{k}{2\sigma^{*2}}\sum_{i=1}^{n}\frac{\left(\tilde{y}_i-b\tilde{y}_{i-1}-d\sigma^{*2}\exp(h_i)\right)^2}{\exp(h_i)}\right\} I_{(0,10)}(\sigma_\eta),$$

$$\pi^{(k)}(\sigma^{*2}|\phi,\sigma_\eta,b,d,\tilde{y}) \propto \left(\prod_{i=1}^{N}\sigma^{*2}e^{h_i}\right)^{-\frac{k}{2}} \exp\left\{-\frac{k}{2\sigma^{*2}}\sum_{i=1}^{n}\frac{\left(\tilde{y}_i-b\tilde{y}_{i-1}-d\sigma^{*2}\exp(h_i)\right)^2}{\exp(h_i)}\right\} I_{(0,10)}(\sigma^{*2}),$$

$$\pi^{(k)}(b|\phi,\sigma_\eta,\sigma^{*2},d,\tilde{y}) \propto \left(\prod_{i=1}^{N}\sigma^{*2}e^{h_i}\right)^{-\frac{k}{2}} \exp\left\{-\frac{k}{2\sigma^{*2}}\sum_{i=1}^{n}\frac{\left(\tilde{y}_i-b\tilde{y}_{i-1}-d\sigma^{*2}\exp(h_i)\right)^2}{\exp(h_i)}\right\} I_{(-10,10)}(b),$$

$$\pi^{(k)}(d|\phi,\sigma_\eta,\sigma^{*2},b,\tilde{y}) \propto \left(\prod_{i=1}^{N}\sigma^{*2}e^{h_i}\right)^{-\frac{k}{2}} \exp\left\{-\frac{k}{2\sigma^{*2}}\sum_{i=1}^{n}\frac{\left(\tilde{y}_i-b\tilde{y}_{i-1}-d\sigma^{*2}\exp(h_i)\right)^2}{\exp(h_i)}\right\} I_{(-10,10)}(d).$$

The algorithm starts from an initial solution $\phi^{(0)}$, $\sigma_\eta^{(0)}$, $\sigma^{*2(0)}$, $b^{(0)}$ and $d^{(0)}$ and considering these values generates the new ones ($\phi^{(m)}$, $\sigma_\eta^{(m)}$, $\sigma^{*2(m)}$, $b^{(m)}$ and $d^{(m)}$) in each iteration (m) from the conditional posterior distributions. The posterior sample will be obtained with them and its arithmetic means will constitute the maximum likelihood estimations approach. The algorithm steps can be summarized as follows:

Step 1: Set initial solution at $m = 0$ as: $\phi^{(0)}$, $\sigma_\eta^{(0)}$, $\sigma^{*2(0)}$, $b^{(0)}$ and $d^{(0)}$.

Step 2: Generate $\phi^{(m+1)}$ from its conditional posterior distribution

$$\phi^{(m+1)} \sim \pi^{(k)}(\phi|\sigma_\eta,\sigma^{*2},b,d,\tilde{y}).$$

Step 3: Generate $\sigma_\eta^{(m)}$ from its conditional posterior distribution

$$\sigma_\eta^{(m)} \sim \pi^{(k)}(\sigma_\eta|\phi,\sigma^{*2},b,d,\tilde{y}).$$

Step 4: Generate $\sigma^{*2(m)}$ from its conditional posterior distribution

$$\sigma^{*2(m)} \sim \pi^{(k)}(\sigma^{*2}|\phi,\sigma_\eta,b,d,\tilde{y}).$$

Step 5: Generate $b^{(m)}$ from its conditional posterior distribution

$$b^{(m)} \sim \pi^{(k)}(b|\phi,\sigma_\eta,\sigma^{*2},d,\tilde{y}).$$

Step 6: Generate $d^{(m)}$ from its conditional posterior distribution

$$d^{(m)} \sim \pi^{(k)}(d|\phi,\sigma_\eta,\sigma^{*2},b,\tilde{y}).$$

Step 7: Set $m = m + 1$ and go to Step 2.

The package `dclone` (Sólymos, 2010) from the `R` project (R Core Team, 2012) has been used again to program the algorithm, analogously to the way the data cloning algorithm was programmed to estimate the SV model. Initial values have been simulated directly from the prior distribution.

The same procedure used for the SV model will be followed to analyze the quality of the estimates. Therefore, a series with 245 observations will be simulated using the following parameters for the model: $\phi = 0.97$, $\sigma_\eta = 0.12$, $\sigma^{*2} = 0.2$, $b = 0.2$ and $d = 0.1$. The parameters of the model will then be estimated using the series data, and the proximity of the estimated values to the real values, as well as the standard errors of estimation, will be examined. Confidence intervals will also be obtained for the parameters and it will be checked if they include the true values.

To determine the optimal number of clones, the following criteria from the `dclone` package will be employed: maximum eigenvalue of the posterior variance, minimum squared error, R^2 and $\hat{R}$ (Lele et al., 2010 and Brooks and Gelman, 1998). It can be noted that, as this model have one more parameter, it is necessary to use a considerable higher number of clones to achieve convergence. After trying several estimations it has been possible to conclude that 40 clones are enough to make quality estimates that are not substantially improved by including a larger number of clones. Hence, the optimal number of clones is set at 40.

The table 2 shows in columns the real data used to estimate the series, the estimates obtained, the standard errors of estimation and the confidence intervals for each parameter. The table also includes the estimates obtained using the MCMC method, with their respective standard errors, for the purpose of comparing the estimation quality between the two methodologies. Figure 2 displays the posterior distributions of the parameters obtained by the algorithm.

Parameter	Real Value	Data cloning Estimations	S.D.	Confidence Intervals	MCMC Estimations	S.D. (MCMC)
ϕ	0.97	0.9717	0.0053	$(0.9055, 1.0379)$	0.9368	0.0810
σ_η	0.12	0.1386	0.0171	$(-0.0736, 0.3509)$	0.1878	0.0807
σ^{*2}	0.2	0.1831	0.0671	$(-0.6493, 1.0155)$	0.1717	0.0511
b	0.2	0.2548	0.0103	$(0.1267, 0.3829)$	0.2527	0.0654
d	0.1	0.1386	0.0171	$(-0.1454, 0.4145)$	0.1402	0.1408

Table 2: Estimation for Stochastic Volatility in Mean model parameters using data cloning method.

Only one trajectory has been considered and yet it can be seen how the estimation algorithm provides values very close to the real values of parameters used to simulate it. The standard errors of estimation are also small enough to support the quality of the obtained estimates. Finally, it can be seen that the 95% confidence intervals include the real values of the parameters and the estimations improve the obtained by a MCMC procedure in terms of estimation standard errors. The histograms demonstrate the close correspondence between the estimated SVM parameters and the real values. From the results obtained, it can be observed that in the analyzed case, it overestimates the values of ϕ, σ^{*2} and d, while underestimating the values of σ_η and b.

Although it can be noted that the estimates based on a single trajectory are good enough, different trajectories have also been estimated from the same parameters, obtaining as result the average of all the estimates. As expected, this method provides values which are even closer to the true values of the parameters. We do not include further details of this option as it may not be applicable to real data, where only a single trajectory is available. However, it is worth mentioning that this approach enhances the quality of the estimator by reducing variance and

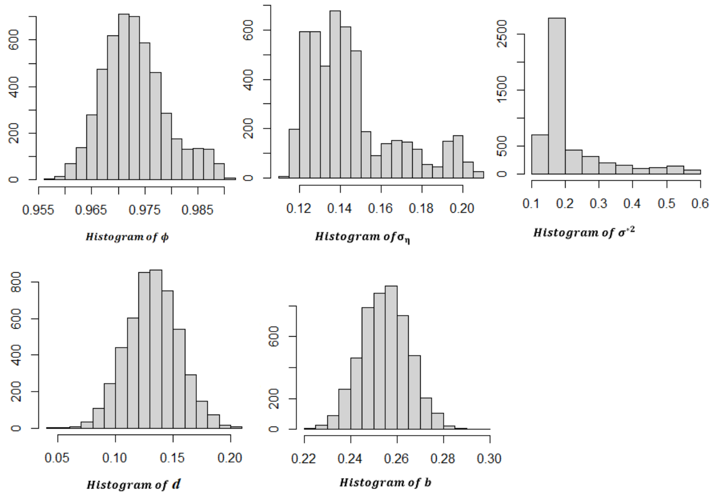

Figure 2: Histograms of the posterior distributions of the Stochastic Volatility in Mean model parameters

improving the accuracy of the mean value.

The algorithm has also been evaluated with different size sample paths, showing good performance in all of them. It is observed that when the sample paths are small in size, the estimation results depend to a greater extent on the path considered. In contrast, convergence is achieved with a number of clones even smaller than the 40 clones proposed to be used. When the size of the sample paths is moderately large, the estimates are more stable and depend less on the considered path, but in some cases, it is necessary to use more than 40 clones to reach convergence. Results are summarized in table 3.

Parameter	Real Value	Estimations (S.D.) n = 100	Estimations (S.D.) n = 245	Estimations (S.D.) n = 500	Estimations (S.D.) n = 1000
ϕ	0.97	0.9635 (0.0054)	0.9717 (0.0053)	0.9893 (0.0039)	0.9631 (0.0024)
σ_η	0.12	0.1143 (0.0143)	0.1386 (0.0171)	0.1401 (0.0086)	0.1465 (0.0047)
σ^{*2}	0.2	0.4505 (0.0261)	0.1831 (0.0671)	0.5086 (0.1575)	0.2571 (0.0049)
b	0.2	0.1335 (0.0190)	0.2548 (0.0103)	0.1341 (0.0072)	0.2042 (0.0051)
d	0.1	-0.061 (0.0252)	0.1386 (0.0171)	0.0656 (0.0133)	0.1512 (0.0092)

Table 3: Estimation for Stochastic Volatility in Mean model parameters using data cloning method in sample paths of different sizes.

5 Applications of estimators to real data: Bitcoin

There is no doubt about the importance of cryptocurrencies in the economy since the introduction of Bitcoin to the markets in 2008 (Urquhart, 2016, Katsiampa, 2017, Akkus and Çelik, 2020). Cryptocurrencies exhibit higher volatility and are more susceptible to bubbles compared to traditional currencies (Cheah and Fry, 2015). In addition, the volatility of Bitcoin returns presents long memory, resulting in their analysis as financial assets rather than traditional currencies. They are increasingly being included in financial portfolios and therefore modeling volatility and its relationship to returns is very important in portfolio optimization, hedging and valuation of derivative securities. Bitcoin remains the most important cryptocurrency in terms of market capitalization (Tiwari et al., 2019, Akkus and Çelik, 2020) and that is why we are going to use it as an example. Tiwari et al. (2019) obtain that in general, SV models consistently outperform the GARCH models when it comes to analyzing cryptocurrencies (particularly in the case of Bitcoin and to a lesser extent in Litecoin). Moreover, they show that in general using t-distributed innovations greatly improves the results of standard GARCH models, but this result is not significant for SV models. Considering that, in this paper we use innovations that follow a normal distribution. Nevertheless, the analysis can be easily extended to incorporate a Student's t distribution.

The data considered are the daily returns of the cryptocurrency from October the 1^{st} of 2020 to March the 1^{st} of 2021. Data set have been obtained from the Spanish financial news website https://es.investing.com/.

5.1 Modeling Bitcoin returns using the SV model estimated by data cloning method

To model real data by a SV model, since the estimation algorithm excludes the intercept term, we will use the deviations from the mean of the data. Furthermore, the most recent 5 data values have been excluded, to be used later to test the predictions. The estimated model parameters, the estimation standard errors and credible intervals are shown in table 4. It also includes the estimates of the model parameters and corresponding standard errors using the MCMC method in order to compare both methodologies. Bayesian confidence intervals are included because they will be used to analyze the significance of the parameters from a Bayesian point of view. However, as shown above, if a frequentist approach to the study is desired, confidence intervals can be readily calculated. This is one of the advantages of the data cloning methodology.

Parameter	Data cloning Estimations	S.D. DC	HPD 0.95	MCMC Estimations	S.D. MCMC
ϕ	0.4722044	0.2779	$(0.0170559, 0.9464303)$	0.4165	0.26151
σ_η	0.1012176	0.06509	$(0.0196413, 0.3175769)$	0.4825	0.3382
σ^{*2}	0.0001425	$7.516e-6$	$(0.0001297, 0.0001603)$	0.0001489	$4.6174e-5$
b	-0.2081462	0.02999	$(-0.2671039, -0.1502172)$	-0.1194	0.1324

Table 4: Estimation for SV model parameters to estimate Bitcoin, using data cloning and MCMC methods.

As expected, the data cloning and MCMC algorithms provide close values for all parameters except for σ_η. This is probably due to a high standard error in the MCMC method. Note that all parameters except ϕ have lower estimation errors in the estimates obtained through Data Cloning.

All the parameters are significant at 5%, according to the credible intervals. The parameter b represents the effect of the lagged return in the expected value of the return and in this case a negative value has been obtained. ϕ is the first order coefficient of the log equation volatility (4) while σ_η is moderating the effect of disturbance in the log-volatility equation (4). Finally, σ^{*2} is the constant coefficient of variance and it takes a small value of the volatility overall, which will be increased by ϕ and σ_η.

The value of ϕ is significant, providing an evidence of volatility clustering. However, its value is relatively low, suggesting that there is not a substantial persistence of volatility across consecutive periods. At the same time, the value of σ_η is quite high and significant, which means that the volatility of a period is strongly affected by the shocks of the same period, increasing the value of the variance. That implies that the volatility process is less easily predictable. Finally, b takes a negative value, indicating that the profitability in differences of one period negatively affects the profitability of the following period. Therefore, we can conclude the following:

- The negative value of b implies that returns from one period have a negative impact on the returns of the subsequent period.
- The variance exhibits a generally high level, showing little dependence on the variance of the previous period but significant sensitivity to shocks occurring in the current period.

These parameters enable the construction of equations for predicting the subsequent values using a one-step prediction method. This approach involves using the actual values from the previous period to generate predictions for returns. For constructing the next values in the series, the true value of the required lag (in this case, one) is used. Similarly, a lag is required for volatility, but since volatility is unobservable, the estimated value is employed in this context.

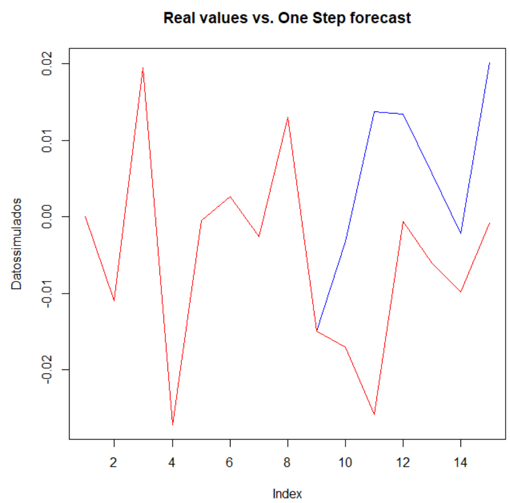

Figure 3: Bitcoin returns vs. estimations of Bitcoin returns by using SV model estimated by data cloning algorithm.

Figure 3 displays the predictions of bitcoin returns obtained through SV modeling compared to the actual bitcoin returns. The figure demonstrates the model's ability to generate accurate one-step predictions for future values in this series.

5.2 Modeling bitcoin returns using the SVM model estimated by data cloning method

The same dataset will also be modeled using the SVM model estimated through the data cloning algorithm introduced earlier. This model is expected to better incorporate the unobservable behavior of volatility by considering its effects on both the return and its mean simultaneously. Table 5 presents the estimated parameter values, their standard estimation errors, and the credible intervals. Additionally, it includes the parameter values estimated through MCMC and their corresponding standard errors.

Parameter	Data cloning Estimations	S.D.	HPD 0.95	MCMC Estimations	S.D. MCMC
ϕ	0.4918337	0.3506	$(0.011611, 0.9855730)$	0.4290	0.2661
σ_η	0.1158941	0.06447	$(0.048043, 0.2844250)$	0.4556	0.3318
σ^{*2}	0.0001422	$7.998e-6$	$(0.000131, 0.0001663)$	$1.49e-4$	$5.28e-5$
b	-0.214432	0.02091	$(-0.25487, -0.173575)$	-0.1212	0.1314
d	7.1425561	1.558	$(3.851365, 9.7414312)$	2.3354	5.0931

Table 5: Estimation for Stochastic Volatility in Mean model parameters to estimate Bitcoin, using data cloning method.

Both estimation methods yield similar parameter values, except for σ_η and d, where the MCMC method exhibits higher standard errors, resulting in less agreement with the data cloning estimations.

All parameters are statistically significant at a 5% significance level, as indicated by the credible intervals. The significance of ϕ once again supports the presence of volatility clustering, although its magnitude is not particularly high. Similarly to the SV model, parameter b takes a negative value, indicating a negative impact of lagged returns on current returns.

In the SVM model, a new parameter d is estimated, which captures the effect of volatility on the mean returns. Its significance suggests that the variance has a substantial influence on the expected returns, and the positive value indicates a feedback effect of volatility on returns, aligning with our expectation when analyzing returns in differences.

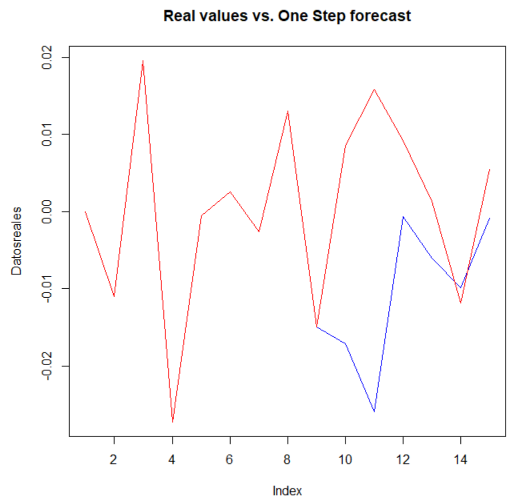

Figure 4: Bitcoin returns vs. estimations of Bitcoin returns by using SVM model estimated by data cloning algorithm

Figure 4 presents the predicted values of the last observations obtained from the SVM model compared to the actual values. It demonstrates the effectiveness of the one-step prediction method in capturing the future behavior of the series. The close alignment between the predicted values and the actual observations highlights the accuracy of the SVM model in forecasting future values.

6 Final conclusions

The main goal of this paper is to introduce an estimator of the SVM model parameters based on the data cloning algorithm, which provides an approximation to the maximum likelihood estimates of the model parameters. The main findings of this study are as follows:

- Data cloning algorithm is a good solution for estimating the parameters of SV and SVM models whose complexity makes it difficult to use other estimation methods.
- Data cloning is especially interesting to estimate SVM model because it let to estimate the return and the volatility at the same time.
- The estimates obtained by the data cloning method to estimate the parameters of the SV and SVM models are shown to be better in terms of standard error than those obtained by the conventional MCMC algorithms in the simulation study.
- The SVM data cloning estimation algorithm demonstrates consistent performance regardless of the sample path size. However, it is observed how the estimates are more stable and less path dependent when we increase its size.
- The hybrid nature of the data cloning methodology proves to be a very suitable solution when estimating parameters by the maximum likelihood method using Bayesian algorithms.
- SV and SVM models are suitable for modeling financial data with volatility jumps and they let to understand these series behavior.
- SV and SVM models empirically show good capabilities to provide one step predictions for cryptocurrencies like Bitcoin. They show that Bitcoin volatility is strongly related to the return in the same period.

Declaration of interest statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Agbeyegbe, T. (2022). Modeling jse stock returns dynamics: Garch versus stochastic volatility. *The Journal of Developing Areas 56*(1), 1–18.

Akkus, H. and Çelik (2020). Modeling, forecasting the cryptocurrency market volatility and value at risk dynamics of bitcoin. *Muhasebe Bilim Dünyası Dergisi 22*(2), 296–312.

Andrieu, C., A. Doucet, and R. Holenstein (2010). Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society Series B 72*, 269–342.

Asai, M., M. McAleer, and J. Yu (2006). Multivariate stochastic volatility: A review. *Econometric Reviews 25*, 145–175.

Balcilar, M. and Z. Ozdemir (2019). The volatility effect on precious metals price returns in a stochastic volatility in mean model with time-varying parameters. *Physica A 534*, 1–14.

Beskos, A., K. Kalogeropoulos, and E. Pazos (2013). Advanced mcmc methods for sampling on diffusion pathspace. *Stochastic Processes and their Applications 123*, 1415–1453.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics 31*(3), 307–327.

Bouchaud, J., A. Matacz, and M. Potters (2001). Leverage effect in financial markets: The retarded volatility model. *Physical Review Letters 87*(22).

Brooks, C. and M. Prokopczuk (2013). The dynamics of commodity prices. *Quantitative Finance 13*, 527–542.

Brooks, S. and A. Gelman (1998). General methods for monitoring convergence of iterative simulations. *Journal of computational and graphical statistics 7*(4), 434–455.

Broto, C. and E. Ruiz (2004). Estimation methods for stochastic volatility models: A survey. *Journal of Economic Surveys 18*, 613–649.

Carr, P., H. Geman, D. Madan, and M. Yor (2003). Stochastic volatility for lévy processes. *Mathematical Finance 13*, 345–382.

Chaim, P. and M. P. Laurini (2022). Data cloning estimation and identification of a medium-scale dsge model. *Stats 6*(1), 17–29.

Chan, J. and A. Grant (2016). Modeling energy price dynamics: Garch versus stochastic volatility. *Energy Economics 54*, 182–189.

Cheah, E. and J. Fry (2015). Speculative bubbles in bitcoin markets? an empirical investigation into the fundamental value of bitcoi. *Economics Letters 130*, 32–36.

de Zea Bermudez, P., J. M. Marín, and H. Veiga (2020). Data cloning estimation for asymmetric stochastic volatility models. *Econometric Reviews 39*(10), 1057–1074.

French, K. R., G. Schwert, and R. F. Stambaugh (1987). Expected stock returns and volatility. *Journal of Financial Economics 19*(1), 3–29.

Jacquier, E., N. Polson, and P. Rossi (1994). Bayesian analysis of stochastic volatility models (with discussions. *Journal of Business and Economic Statistics 12*(4), 371–417.

Kastner, G., S. Frühwirth-Schnatter, and H. Lopes (2017). Efficient bayesian inference for multivariate factor stochastic volatility models. *Journal of Computational and Graphical Statistics 26*, 905–917.

Katsiampa, P. (2017). Volatility estimation for bitcoin: A comparison of garch models. *Economics Letters 158*, 3–6.

Kim, S., N. Shepherd, and S. Chib (1998). Stochastic volatility: Likelihood inference and comparison with arch models. *Review of Economic Studies 65*(3), 361–393.

Koopman, S. and E. Uspensky (2002). The stochastic volatility in mean model: Empirical evidence from international stock markets. *Journal of Applied Econometrics 17*, 667–689.

Koopman, S. J. and E. Hol Uspensky (2002). The stochastic volatility in mean model: empirical evidence from international stock markets. *Journal of applied Econometrics 17*(6), 667–689.

Laurini, M. (2013). A hybrid data cloning maximum likelihood estimator for stochastic volatility models. *Journal of Time Series Econometrics 5*(2), 193–229.

Lele, S., B. Dennis, and F. Lutscher (2007). Data cloning: easy maximum likelihood estimation for complex ecological models using bayesian Markov chain Monte Carlo methods. *Ecology Letters 10*(7), 551–563.

Lele, S., K. Nadeem, and B. Schmuland (2010). Estimability and likelihood inference for generalized linear mixed models using data cloning. *Journal of the American Statistical Association 105*(492), 1617–1625.

Li, H., K. Yang, and D. Wang (2019). A threshold stochastic volatility model with explanatory variables. *Statistica Neerlandica 73*, 118–138.

Lin, S. and X.-J. He (2023). Analytically pricing variance and volatility swaps with stochastic volatility, stochastic equilibrium level and regime switching. *Expert Systems with Applications 217*, 119592.

Mandelbrot, B. (1963). The variation of certain speculative prices. *The journal of business 36*(4), 394–419.

Manera, M., M. Nicolini, and I. Vignati (2016). Modelling futures price volatility in energy markets: Is there a role for financial speculation? *Energy Economics 53*, 220–229.

Marín, J. M., M. T. Rodríguez-Bernal, and E. Romero (2015). Data cloning estimation of GARCH and COGARCH models. *Journal of Statistical Computation and Simulation 85*(9), 1818–1831.

Martino, S., K. Aasb, O. Lindqvist, L. Neef, and H. Rue (2011). Estimating stochastic volatility models using integrated nested laplace approximations. *The European Journal of Finance 17*(7), 487–503.

Melino, A. and S. Turnbull (1990). Pricing foreign currency options with stochastic volatility. *Journal of Econometrics 45*(1-2), 239–265.

Pasricha, P. and X.-J. He (2023). Exchange options with stochastic liquidity risk. *Expert Systems with Applications 223*, 119915.

Ponciano, J., M. Taper, B. Dennis, and S. Lele (2009). Hierarchical models in ecology: confidence intervals, hypothesis testing, and model selection using data cloning. *Ecology 90*(2), 356–362.

R Core Team (2012). *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Renault, E. (2009). Moment-based estimation of stochastic volatility models. In T. Andersen, R. Davis, J. Kreib, and T. Mikosch (Eds.), *Handbook of Financial Time Series*, pp. 269–311. New York: Springer.

Rue, H., S. Martino, and N. Chopin (2009). Approximated bayesian inference for latent gaussian models by using integrated nested laplace approximations (with discussion. *Journal of the Royal Statistical Society seris B 71*(2), 319–392.

Sandmann, G. and S. Koopman (1998). Estimation of stochastic volatility models via monte carlo maximum likelihood. *Journal of Econometrics 87*, 271–301.

Shephard, N. (1993). Fitting non-linear time series models, with applications to stochastic variance models. *Journal of Applied Econometrics 8*, 135–152.

Sólymos, P. (2010). dclone: Data cloning in R. *The R Journal 2*(2), 29–37.

Sólymos, P. (2010). dclone: Data cloning in r. *The R Journal 2*(2), 29–37.

Taylor, S. (1982). Financial returns modelled by the product of two stochastic processes — a study of daily sugar prices 1961-79. In O. Anderson (Ed.), *Time Series Analysis: Theory and Practice*. Amsterdam: North-Holland.

Taylor, S. (1986). *Modelling financial time series*. New York: John Wiley and Sons.

Taylor, S. (1994). Modelling stochastic volatility: A review and comparative study. *Mathematical Finance 4*(2), 183–204.

Tiwari, A., S. Kumar, and R. Pathak (2019). Modelling the dynamics of bitcoin and litecoin: Garch versus stochastic volatility models. *Applied Economics 51*(37), 4073–4082.

Trolle, A. B. and E. S. Schwartz (2009). Unspanned stochastic volatility and the pricing of commodity derivatives. *The Review of Financial Studies 22*(11), 4423–4461.

Tseng, J.-J. and S.-P. Li (2011). Asset returns and volatility clustering in financial time series. *Physica A: Statistical Mechanics and its Applications 390*(7), 1300–1314.

Urquhart, A. (2016). The inefficiency of bitcoin. *Economics Letters(148*, 80–82.

Yang, X., Y. Wu, Z. Zheng, and J.-Q. Hu (2021). Method of moments estimation for lévy-driven ornstein-uhlenbeck stochastic volatility models. *Probability in the Engineering and Informational Sciences 35*, 975–1004.

Yu, J. (2002). Forecasting volatility in the new zealand stock market. *Applied Financial Economics 12*(3), 193–202.