University of Economics Prague

Faculty of Finance and Accounting

Department of Banking and Finance

Dissertation Thesis

Modelling and Forecasting of Stochastic Volatility and Jumps

Author: Ing. Milan Fičura

Supervisor: prof. RNDr. Jiří Witzany, Ph.D.

Year: 2018

Abstract

The thesis reviews the most commonly used volatility forecasting models from the ARCH/GARCH, realized volatility and stochastic volatility forecasting frameworks, with the main focus being placed on Stochastic-Volatility Jump-Diffusion (SVJD) models, on the ways of how high-frequency power-variation estimators can be used in SVJD model setting, and on the use of Bayesian methods for the estimation of SVJD model parameters and latent states. SVJD-RV-Z class of models is developed, utilizing the realized variance for better estimation of the stochastic variances, and the non-parametric Z-Estimator for more accurate estimation of price jumps. Several adapted particle filters, specifically designed for latent-state filtering in SVJD models, are derived, and a Sequential Gibbs Particle Filter (SGPF) algorithm is developed for the sequential learning of their parameters. In the empirical study, four SVJD models (with intraday data, self-exciting jumps in prices and volatility, as well as multiple volatility components) are applied for the task of realized volatility forecasting on the time series of 7 foreign exchange rates and 10 ETF/ETN securities in the daily, weekly and monthly forecast horizon. The performance of the SVJD models is compared with 3 GARCH models (GARCH, EGARCH and GJRGARCH), 15 HAR model specifications (HAR, AHAR, SHAR, HARJ and HARQ), and 15 Echo State Neural Network (ESN) based volatility models developed by the author. The SVJD-RV-Z models with jumps in volatility and prices are shown to exhibit the highest out-sample predictive power, comparable to the best HAR and ESN model specifications.

Abstrakt

Disertační práce rozebírá nejpoužívanější modely volatility (modely ARCH/GARCH, modely realizované volatility a modely stochastické volatility), a dále se podrobněji zaměřuje na modely stochastické volatility a skoků (SVJD), na možnosti využití vysoko-frekvenčních odhadů mocninných variací v rámci SVJD modelů, a na Bayesovské metody odhadu SVJD modelů a jejich latentních stavových proměnných. Autor v práci vyvíjí třídu tzv. SVJD-RV-Z modelů, využívajících realizovanou volatilitu pro přesnější odhad stochastické volatility, a neparametrický Z-Estimátor skoků pro přesnější odhad cenových skoků. V práci je vyvinuto několik adaptovaných částicových filtrů, odvozených speciálně pro účely přesnějšího filtrování latentních stavových proměnných v SVJD modelech, a dále Sequential GIbbs Particle Filter (SGPF) algoritmus, umožňující sekvenciální odhad parametrů SVJD modelů. V provedené empirické studii jsou vyvinuty 4 SVJD modely (využívající intradenní data, samo-excitující se skoky v ceně i ve volatilitě, i více komponent pro spojitou volatilitu), které jsou aplikovány pro účely predikce realizované volatility na časových řadách 7 měnových kurzů a 10 ETF/ETN aktiv, v denním, týdenním a měsíčním horizontu. Prediktivní síla SVJD modelů je porovnána s 3 GARCH modely (GARCH, EGARCH a GJR-GARCH), 15 HAR modely (HAR, AHAR, SHAR, HARJ a HARQ, plus jejich logaritmické a odmocninové verze), a 15 modely založených na Echo State Neuronových sítích (ESN). SVJD-RV-Z modely se skoky v ceně a ve volatilitě dosáhly mezi testovanými modely nejvyšší prediktivní síly, srovnatelné s nejlepšími zástupci HAR a ESN modelů.

Declaration

I hereby declare that all sources are properly acknowledged in the text and all errors are my own responsibility.

In Prague, ……………………………

August 26, 2018 Milan Fičura

Acknowledgements

I would like to thank my supervisor prof. RNDr. Jiří Witzany, Ph.D. for his help and discussions thorough the writing of this thesis. I would further like to acknowledge the support from the grant project “Advanced methods of financial asset returns and risks modelling” IGA VŠE/F1/23/2015 from which the data for my research were purchased, and the Czech Science Foundation Grant P402/12/G097 – Dynamical Models in Economics, that supported my research endeavours in the area of volatility and jumps modelling.

Contents

Introduction

Volatility of financial asset returns, commonly represented by their variance or standard deviation, plays a crucial role in many areas of finance, such as asset pricing, portfolio optimization, Value at Risk estimation, derivatives valuation and quantitative trading. Modelling and forecasting of volatility is complicated by the fact that it is an unobservable quantity with relatively complex dynamics that includes volatility clustering, long memory, price and volatility jumps, jump clustering, asymmetries, contagion effects, as well as stochastic variability of the volatility itself (Andersen et al, 2005). The interplay between these effects has a dramatic effect on the shape of the asset return distribution in different horizons, leading to the effects of fat tails (increased kurtosis) and asymmetry (increased skewness) that tend to vary through the time. As the knowledge of the expected return distribution plays an essential role in areas such as option pricing and Value at Risk estimation, the development of volatility modelling and forecasting methods capable of capturing all the empirically observed effects of the volatility process is of high importance.

In the presented study, different approaches of volatility estimation and modelling will be reviewed, including the ARCH/GARCH models, Stochastic volatility models, Realized volatility models, Option-based volatility models, and neural network based approaches. The focus will be placed in particular on the Stochastic-Volatility Jump-Diffusion (SVJD) models, on the ways of how non-parametric high-frequency based estimators can be integrated into them, and on the Bayesian estimation methods such as Markov-Chain Monte-Carlo (MCMC) and Particle Filters that can be used for the estimation of the SVJD models.

While SVJD models represent a theoretically powerful tool for the modelling and forecasting of stochastic volatility and jumps, the difficulties of their estimation – especially regarding the fact that they view volatility and jumps as latent time series that need to be estimated – caused them to often got outperformed by simpler approaches in the empirical studies in the past. Especially in the recent years, Realized Volatility models, utilizing high-frequency estimators such as the realized variance and bi-power variation, computed from intraday returns, exhibited an excellent performance in volatility forecasting studies and outperformed most of the other volatility forecasting methods, including the GARCH models (Andersen and Bollerslev, 1998) and Stochastic Volatility models (Deo et al., 2006), while being on par with the option-based forecasts (Pong et al, 2004, Koopman et al., 2005).

One of the main contributions of the author is thus to introduce the SVJD-RV and SVJD-RV-Z models (proposed in Fičura and Witzany, 2015 and Fičura and Witzany, 2017), which operate under the theoretically appealing SVJD model framework, while using the non-parametric, high-frequency based power-variation estimators of volatility and jumps as additional sources of information. The idea stems from the research in Takahashi, Omori and Watanabe (2009), which will be extended to the case of models with jumps. The SVJD-RV model uses the realized variance (Andersen and Bollerslev, 1998) as an additional source of information for the estimation of the stochastic variance, while the SVJD-RV-Z model uses also the Z-Estimator of jumps (Barndorff-Nielsen and Shephard, 2004) as an additional source of information for the estimation of the times of jump occurrences.

The author presents an MCMC algorithm to estimate SVJD models with self-exciting jumps, that was first proposed in Fičura and Witzany (2016) for the SVJD models and extended in Fičura and Witzany (2017) for the SVJD-RV and SVJD-RV-Z models.

In this thesis, the MCMC estimation approach will be contrasted with an alternative estimation approach based on Particle Filters, stemming from the recently popularized framework of particle learning (Carvalho et al., 2010). The final estimation of the proposed SVJD models will then be performed with the newly developed Sequential Gibbs Particle Filter (SGPF) (Fičura and Witzany, 2018), that builds on the earlier particle learning algorithms, especially the Gibbs Particle Filters developed in Gilks and Berzuini (2001), the Particle MCMC of Andrieu et al. (2010), and the Marginalized Re-Sample Move algorithm developed in Fulop and Li (2013).

Another area explored in the thesis is the notion of adaptation of Particle Filter proposal distributions to the observed returns (and possibly the realized variances and the Z-Statistics as well), with the goal of improving their performance with regards to the estimation of the latent states of SVJD models. The author derives the adapted proposal distributions for the price jump occurrences and sizes, as well as volatility jump occurrences in a SVJD model with self-exciting jumps, and subsequently verifies the supreme performance of the adapted particle filter compared to the un-adapted one in a series of simulation studies.

As an additional contribution, not related to SVJD models, the author extends on his work in Fičura (2017) on the use of Echo State Neural Networks (ESN) for the purpose of realized variance forecasting, and performs an empirical test of different versions of the ESN model with alternative power-variation based predictors, such as realized variance, bi-power variation based jump variance estimates, realized semi-variance and realized quarticity. The proposed ESN models are included as benchmark models in the performed empirical study.

In the empirical study, the author applies four SVJD and SVJD-RV-Z models, with features such as self-exciting jumps in returns and volatility and multiple volatility components, to the time series of 7 major currency exchange rates over the period from 1.11.1999 to 15.6.2015, and to the time series of 10 most commonly traded ETF/ETN funds over the period from 2.1.1998 to 1.6.2018, for the purpose of forecasting the realized variance in the 1-Day, 5-Day and the 22-Day horizon. As benchmark models, 15 different specifications of the ESN model, three most commonly used GARCH model specifications (GARCH, EGARCH and GJR-GARCH), and 18 specifications of the HAR-RV model (HAR, AHAR, SHAR, HARJ, HARQ, plus their logarithmic and square-root versions) are used. The accuracy of the realized variance forecasts of all the models is then assessed with the R-Squared criterion, showing that the proposed SVJD and ESN models exhibit comparable or better predictive power than the best benchmark models from the HAR model family.

The rest of the study is organized as follows. In Section 1 the main volatility modelling frameworks and individual models are discussed. In Section 2, the four SVJD models used in the empirical part of the study are specified. Section 3 explains the Bayesian estimation methods used for SVJD model estimation, parameter learning and latent states filtering. Section 4 contains a simulation study and an empirical study in order to assess the filtering performance and the convergence properties of the proposed particle filters and the SGPF method. In Section 5 is the main empirical study of the predictive power of different volatility forecasting models. Section 6 contains conclusion of the main results and discussion about the topic for future research. Appendix in section 7 contains a derivation of several approximately adapted particle filters, whose performance is tested in the simulation study in Section 5.

1. Volatility models

The unobservable character of volatility of financial time series led the development of several different frameworks of its modelling. These are most commonly categorized into four groups, known as ARCH/GARCH models, Stochastic Volatility models, Realized Volatility models and Option-Based volatility models (Andersen et al, 2005).

1.1. Historical development

The ARCH/GARCH modelling framework models volatility (i.e. the conditional variance of market returns) as a deterministic function of the squared returns in the past, computed on the same time-frequency. Models of this kind include the EWMA model, the ARCH model (Engle, 1982), and its generalized version, the GARCH model (Bollerslev, 1986). Over the years, large number of ARCH/GARCH model extensions have been proposed, striving capture additional features of the volatility dynamics. Among the most commonly used ones are the asymmetric GARCH models, designed to capture the negative correlation between volatility and returns, observable especially on the equity markets (Black, 1976). The models of this kind include the AGARCH model (Engle and Ng 1990), TGARCH model (Zakoian, 1994), QGARCH model (Sentana, 1995), GJR-GARCH model (Glosen, Jagannathan and Runkle, 1993) or the EGARCH model (Nelson, 1991). Another important feature of volatility dynamics is the long-memory, causing the autocorrelation function of squared returns to decay in a slow hyperbolic way instead of the exponential way assumed by the short-memory models. Extensions of the GARCH model striving to capture the long-range dependencies include the FIGARCH model (Bailie, Bollerslev and Mikkelsen, 1996), CGARCH (Engle and Lee, 1999) model or the HARCH model (Muller and Dacorogna, 1997). For an overview of ARCH/GARCH models see Bollerslev (2008), while a large-scale study of predictive power of different ARCH/GARCH model specifications can be found in Hansen and Lunde (2005).

An alternative framework represent Stochastic Volatility models (Shephard, 2004), modelling volatility as a latent stochastic process with its own random component. The fact that volatility is viewed as an unobservable latent state variable significantly complicates the estimation of these models, as it requires the estimation of the whole historical evolution of the stochastic volatility latent states (Andersen et al., 2005). To perform this task, computationally intensive estimation methods have been proposed, such as EMM (Andersen, Benzoni and Lund, 2002, Chernov, 2003), MCMC (Jacquier, Polson and Rossi, 1994, Kim, Shephard and Chib, 1998, Eraker, Johannes and Polson, 2003, Eraker, 2004) or Particle Filters (Danielsson, 1994, Sandmann and Koopman, 1998, Golightly, 2009, Fulop and Li, 2013, or Fulop, Li and Yu, 2015).

Based on the works of Clark (1973), Taylor (1986) formulated the first genuinely stochastic volatility model, commonly known as Log-SV, in which the logarithm of the variance follows an AR(1) process. Correlation between volatility and returns can easily be captured in the SV models by introducing a correlation parameter between the random component of the return and volatility stochastic processes. Extensions capturing the long-memory include the LMSV model (Breidt, Crato and De Lima, 1998, Harvey, 1998), the multiple-component SV model (Shephard, 1996, Barndorff-Nielsen and Shephard, 2002), as well as regime-switching approaches, such as the MSSV model (So, Lam and Li, 1998) or the Markov-Switching Multifractal (MSM) model of Calvet and Fisher (2001).

Stochastic-Volatility Jump-Diffusion (SVJD) models add a discontinuous component (jumps), into the price or the volatility equation, with the goal of better capturing the extreme tails of the asset return distribution. Andersen, Benzoni and Lund (2002) and Chernov et al. (2003) use the EMM method to estimate models with jumps in returns, while Eraker, Johannes and Polson (2003) and Eraker (2004) incorporate jumps in returns as well as volatility and estimate them with the MCMC method. Models with self-exciting jumps have further been proposed in Fulop, Li and Yu (2015) and Fičura and Witzany (2016) and estimated with particle filters and MCMC respectively.

With the increased availability of high-frequency financial data, a third volatility modelling framework emerged, known as Realized Volatility models, striving to estimate the underlying stochastic volatility from the intraday returns (Andersen et al., 2005). The first estimator of this kind was the Realized Variance, proposed by Andersen and Bollerslev (1998), which estimates the quadratic variation of the price process over the given day, by summing up all squared high-frequency returns over the given day. Due to the relatively low levels of noise of these estimates, Andersen et al. (2003) propose to use them as a directly observable proxy of the stochastic variance and use time series models such as ARIMA, or the long-memory ARFIMA for their modelling. Similar approach is used in the HAR model of Corsi (2008), which captures the long-memory of the realized variance in an approximate way, and due to its simple estimation (via OLS linear regression) and high performance in empirical studies has became the industry benchmark.

With further developments in the asymptotic theory of realized power variations, estimators were developed that enable the decomposition of the realized variance into its continuous and discontinuous components (Barndorff-Nielsen and Shephard 2004, Andersen, Bollerslev and Diebold 2007), the most common of which is the bi-power variation estimator of the integrated variance, and the associated Z-Estimator of jumps. As the continuous and the discontinuous components of quadratic variation tend to exhibits different levels of persistence (with the jumps being far less persistent than the stochastic volatility), the decomposition can be used to achieve improved forecasts (Andersen et al., 2007).

Additional estimators have further been proposed such as the realized semi-variance, enabling to capture the asymmetry of the realized variance (Barndorff Nielsen et al., 2010, Patton and Shephard, 2005), or the realized quarticity, associated with the volatility of its noise (Barndorff Nielsen and Shephard, 2002, Bollerslev, Patton and Quadvlieg, 2015).

In the recent years, models that view the realized variance as a noisy estimator of the stochastic variance started to become more common. This includes the GARCH-RV model (Hansen and Lunde, 2012), operating in the GARCH model framework, or the HARQ model (Bollerslev, Patton and Quadvlieg, 2015) and the HARS model (Bekierman and Manner, 2018), which view the noise of the realized variance as heteroskedastic and use the realized quarticity for its estimation.

The SV-RV model, utilizing the realized variance as additional source of information in the stochastic-volatility model setting was first proposed in Takahashi, Omori and Watanabe (2009). The model into a SVJD-RV model, for return process with jumps, in Fičura and Witzany (2015) and Fičura and Witzany (2017), who also proposed to use the Z-Estimator (Barndorff-Nielsen and Shephard 2004) as an additional source of information for the jump estimation, giving rise to the SVJD-RV-Z model. Dobrev and Szerszen (2010) and Koopman and Scharth (2013) have further explored the utilization of realized variance estimators in stochastic volatility models, while Maneesoonthorn, Forbes and Martin (2017) provide an alternative extension to the SVJD model setting.

The fourth volatility modelling framework represent the option-based volatility models, which derive the future volatility expected by the market participants, from the publicly traded option prices. This can be done either with the Black-Scholes formula (Black and Scholes, 1973), or in a more advanced way with the Model-Free volatility (Neuberger and Britten-Jones, 2000 and Jiang and Tian, 2005), which is valid for a wide variety of asset price processes, while the Black-Scholes model assumes the price to follow specifically the Geometric Brownian Motion process.

As the option prices include the volatility risk premium (Bakshi and Kapadia, 2003, Eraker, 2009), the option-based volatility forecasts are valid only under the risk-neutral probability measure and tend to over-estimate the future volatility under the real-world probability measure (Carr and Wu, 2008). To cope with this problem, the option-based forecasts need to be adjusted as shown in Muzzioli (2008). Numerous studies have confirmed that the option-based volatility forecasts tend to perform on a similar level to the best of the realized volatility models (Poon and Granger, 2003, Pong et al, 2004, Koopman et al., 2005). Additionally, their performance could further be increased if the time-varying character of the volatility risk premium was taken into account. The time-variability of the volatility risk premium has been studied in Carr a Wu (2008), Todorov (2009), Chen a Poon (2013) or Fičura (2014).

1.2. ARCH/GARCH models

The traditional volatility modelling framework encompasses the moving average models (EWMA) as well as the family of ARCH and GARCH models. The models operate on a single frequency (usually daily) and model the conditional variance of asset returns as a deterministic function of the past squared returns (or alternatively residuals, if the conditional mean of the return process is modelled as well).

1.2.1. Assumed price process

Let p_t denote the logarithm of the asset price at time t and assume that the daily logarithmic returns $r_t = p_t - p_{t-1}$ are governed by the following generally defined stochastic process:

$$r_t = \mu_{t|t-1} + \varepsilon_t = \mu_{t|t-1} + \sigma_{t|t-1} z_t \tag{1}$$

Where $\mu_{t|t-1}$ denotes the conditional mean, $\sigma_{t|t-1}$ the conditional volatility (square root of the conditional variance) and $z_t \sim i.i.d.$, $E(z_t) = 0$, $Var(z_t) = 1$ is a white noise variable, usually assumed to be Gaussian or Student-t distributed.

The quantity that we will strive to model and predict is the conditional variance $\sigma^2_{t|t-1}$. As the underlying $\sigma^2_{t|t-1}$ is unobservable, the traditional volatility models utilize the fact that squared residuals ε^2_t represent an unbiased estimate of $\sigma^2_{t|t-1}$:

$$E(\mathrm{\varepsilon}^2_{\mathrm{t}}) = E\left(\sigma^2_{t|t-1}\, z^2_t\right) = \sigma^2_{t|t-1} \tag{2}$$

The conditional variance is then modelled as a deterministic function of the past squared residuals, i.e. $\sigma^2_{t|t-1} = f(\mathrm{\varepsilon}^2_{\mathrm{t-1}}, \mathrm{\varepsilon}^2_{\mathrm{t-2}}, \dots,\ \mathrm{\varepsilon}^2_1)$.

As the conditional mean $\mu_{t|t-1}$ is for most financial time series close to zero, it is often assumed that $\mu_{t|t-1} = 0$ and the models are formulated with respect to the squared daily returns as $\sigma^2_{t|t-1} = f(r^2_{t-1}, r^2_{t-2}, \dots,\ r^2_1)$.

i.e. if $\mu_{t|t-1} \approx 0$, then:

$$E(r^2_{\mathrm{t}}) = E\left(\sigma^2_{t|t-1}\, z^2_t\right) = \sigma^2_{t|t-1} \tag{3}$$

1.2.2. Moving Average models

A simple approach of how to model the conditional variance is to apply moving average models to the squared daily residuals (or returns). The most commonly used averages for that purpose are the Simple Moving Average (SMA), Weighted Moving Average (WMA) and the Exponentially Weighted Moving Average (EWMA).

The SMA model estimates the conditional variance as a simple arithmetic average of the squared residuals over the last n days, with n being the parameter of the model:

$$\hat{\sigma}^2_t = n^{-1} \sum_{i=0}^{n-1} (r_{t-i} - \mu)^2 = n^{-1} \sum_{i=0}^{n-1} \varepsilon^2_{t-i} \tag{4}$$

The main drawback of the model is that it assigns the same weights to all of the squared returns in the utilized moving window, which does not correspond to the fact that more recent returns tend to influence the current conditional variance more.

The WMA model solves the problem by assigning linearly decreasing weights to the past returns over the last n periods:

$$\hat{\sigma}_t^2 = \frac{2}{n(n+1)} \sum_{i=0}^{n-1} (n-i)(r_{t-i} - \mu)^2 = \frac{2}{n(n+1)} \sum_{i=0}^{n-1} (n-i)\varepsilon_{t-i}^2 \tag{5}$$

While the EWMA model uses exponentially decreasing weights:

$$\hat{\sigma}_t^2 = \gamma(r_t - \mu)^2 + (1-\gamma)\hat{\sigma}_{t-1}^2 = \gamma \sum_{i=1}^{\infty} (1-\gamma)^{i-1} \varepsilon_{t-i}^2 \tag{6}$$

Where γ is the parameter of the exponential weight decay.

The EWMA model is the most popular model among the moving average models. It is also the recommended volatility model by the RiskMetrics methodology of J.PMorgan (1994), which suggest as recommended parameter values $\gamma = 0{,}06$ and $\mu = 0$.

Alternatively, the parameter γ (as well as the parameter n in the SMA and WMA models) can be found with Maximum Likelihood Estimation (MLE) which will be described in the next section as the estimation method for the GARCH model.

While the EWMA model tends to provide reasonable estimates of the underlying conditional variance, it does not include the important effect of mean-reversion, which plays an important role in the conditional variance dynamics. This issue will be solved by the more advanced class of ARCH and GARCH models.

1.2.3. ARCH model

The ARCH model, introduced by Engle (1982), proposes to model the conditional variance with a regression on the past squared residuals. The ARCH(q) model, utilizing the last q squared residuals, can be defined as follows:

$$\sigma_{t|t-1}^2 = \alpha_0 + \alpha_1\varepsilon_{t-1}^2 + \alpha_2\varepsilon_{t-2}^2 + \cdots + \alpha_q\varepsilon_{t-q}^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i\varepsilon_{t-i}^2 \tag{7}$$

With α_i denoting the parameters of the model which have to be non-negative, i.e. $\alpha_i \geq 0$ and $\alpha_0 > 0$.

As $E(\mathrm{ε}_{\mathrm{t}}^{2}) = \sigma_{t|t-1}^{2}$, the parameters of the model can be estimated with a simple Ordinary Least Squares (OLS) regression performed on the squared residuals:

$$\varepsilon_t^2 = \alpha_0 + \alpha_1\varepsilon_{t-1}^2 + \alpha_2\varepsilon_{t-2}^2 + \cdots + \alpha_q\varepsilon_{t-q}^2 = \alpha_0 + \sum_{i=1}^{q}\alpha_i\varepsilon_{t-i}^2 \tag{8}$$

Or with the Maximum Likelihood Estimation (MLE) as will be case for the generalized model GARCH.

1.2.4. GARCH model

GARCH model, proposed by Bollerslev (1986), represents a generalization of the ARCH model, giving it the structure similar to the ARMA process, used commonly in econometrics, for the modelling of the conditional mean.

The GARCH(p,q) model equation is defined as follows:

$$\sigma_{t|t-1}^2 = \alpha_0 + \sum_{i=1}^{q}\alpha_i\varepsilon_{t-i}^2 + \sum_{i=1}^{p}\beta_i\sigma_{t-i|\mathrm{t-i-1}}^2 \tag{9}$$

Where ε_{t-i}^2 are the past squared residuals, while $\sigma_{t-i|\mathrm{t-i-1}}^2$ are the past conditional variance estimates, with α_0, α_i and β_i being the parameters, which should all be non-negative.

The GARCH(1,1) model, which is most commonly used in application, will then have the following equation:

$$\sigma_{t|t-1}^2 = \omega + \alpha\varepsilon_{t-1}^2 + \beta\sigma_{t-1|\mathrm{t-2}}^2 \tag{10}$$

The parameter ω can further be expressed as $\omega = \gamma\sigma_{LT}^2$, with the parameters $\alpha + \beta + \gamma = 1$. The conditional variance $\sigma_{t|t-1}^2$ is thus modelled as a weighted average between the long-term variance σ_{LT}^2, the yesterday squared return ε_{t-1}^2 and the yesterday conditional variance $\sigma_{t-1|\mathrm{t-2}}^2$.

The GARCH(1,1) can also be transformed into the ARCH(∞) model:

$$\sigma_{t|t-1}^{2} = \omega(1-\beta)^{-1} + \alpha \sum_{i=1}^{\infty} \beta^{i-1} \varepsilon_{t-1}^{2} \tag{11}$$

Or by setting $\omega = 0$, $\alpha = \gamma$ and $\beta = (1-\gamma)$ into the EWMA model (Andersen et al., 2005).

The estimation of the GARCH model parameters can be performed with the Maximum Likelihood Method (MLE) in which we set the parameters to the values that maximize the likelihood of observing the data. The Log-Likelihood function of the GARCH(1,1) model will look as follows:

$$LL(\boldsymbol{\varepsilon}, \omega, \alpha, \beta) = \sum_{t=1}^{T} \ln\left[\frac{1}{\sigma_t\sqrt{2\pi}} \exp\left(-\frac{\varepsilon_t^2}{2\sigma_t^2}\right)\right] \tag{12}$$

1.2.5. Asymmetric GARCH models

ARCH and GARCH models implicitly assume that volatility reacts identically to positive and negative residuals of the asset returns. While this may in the long-run hold on the currency markets (Franses and van Dijk, 2000, p.18), it does not hold on the stock markets, where the volatility typically exhibits strongly negative correlation with market returns (the so called “Leverage effect”, first described by Black, 1976). Positive correlation between volatility and returns can, on the other hand, be observed on some of the commodity markets (Christie, 1982, Kristoufek, 2014).

Multiple GARCH model extensions have been proposed in order the capture the correlation between returns and volatility. In the following description we will review the models AGARCH, GJR-GARCH, TGARCH, QGARCH and EGARCH.

The AGARCH(1,1) (Asymmetric GARCH) model of Engle and Ng (1990) uses the following specification:

$$\sigma_{t|t-1}^{2} = \omega + \alpha(\varepsilon_{t-1} - \gamma)^2 + \beta\sigma_{t-1|\mathrm{t}-2}^{2} \tag{13}$$

Positive value of parameter γ will thus shift the residuals ε_{t-1} towards negative values, causing negative returns to have a more pronounced impact on the conditional variance, than the positive returns.

The GJR-GARCH(1,1) model (Glosen, Jagannathan and Runkle, 1993) attempts to capture the Leverage effect by directly introducing an indicator function for the negative returns:

$$\sigma^2_{t|t-1} = \omega + \alpha\varepsilon^2_{t-1} + \gamma\varepsilon^2_{t-1}I(\varepsilon_{t-1} < 0) + \beta\sigma^2_{t-1|\mathrm{t}-2} \tag{14}$$

Where $I(\varepsilon_{t-1} < 0)$ is an indicator function that attains value 1 if $\varepsilon_{t-1} < 0$, and value 0 otherwise. Parameter γ will then determine the additional impact of negative ε_{t-1} on the conditional variance in the following day $\sigma^2_{t|t-1}$.

Indicator function is also used in the TGARCH model of Zakoian (1994), which is, however, specified for the conditional standard deviation:

$$\sigma_{t|t-1} = \omega + \alpha\varepsilon_{t-1}I(\varepsilon_{t-1} > 0) + \gamma\varepsilon_{t-1}I(\varepsilon_{t-1} < 0) + \beta\sigma_{t-1|t-2} \tag{15}$$

Alternatively, the QGARCH(1,1) model of Sentana (1995) adds the residual ε_{t-1} directly into the variance equation:

$$\sigma^2_{t|t-1} = \omega + \alpha\varepsilon^2_{t-1} + \gamma\varepsilon_{t-1} + \beta\sigma^2_{t-1|\mathrm{t}-2} \tag{16}$$

Which unfortunately has the drawback that the predicted variance can be negative if the residual ε_{t-1} is negative and unexpectedly large.

Finally, the logarithmic EGARCH(1,1) model of Nelson (1991) works with the log-variance $\log(\sigma^2_{t|t-1})$ and uses the normalized residuals $z_t = \sigma^{-1}_{t|\mathrm{t}-1}\varepsilon_t$ to capture the leverage effect. The model equation looks as follows:

$$\log(\sigma^2_{t|t-1}) = \omega + \alpha\big(|z_{t-1}| - E(|z_{t-1}|)\big) + \gamma z_{t-1} + \beta\log(\sigma^2_{t-1|\mathrm{t}-2}) \tag{17}$$

One advantage of the log-variance formulation is that the predicted variance $\sigma_{t|t-1}^2$ can never be negative, and there are thus no restrictions on the parameter values. A drawback of the model is that in order to perform multi-period forecasts, simulations need to be used, while for the models formulated directly for $\sigma_{t|t-1}^2$ it is possible to use recursion on the previous forecasts.

1.2.6. Long-Range dependencies in GARCH models

Another limitation of the GARCH(1,1) model is that it is formally a short-memory model, which means that its autocorrelation function decays exponentially with the increasing lags. This is not consistent with the observation that shocks into the variance process usually decay in a far slower, hyperbolic way (Ding, Granger and Engle, 1993). This may lead to decreased performance of the model, especially when long-horizon volatility forecasts are constructed (Andersen et al, 2005).

To cope with this problem multiple GARCH model modifications have been proposed in the literature. These can generally be classified either as genuine long-memory models with hyperbolic decay, or as model that strive to approximate the long-range dependency in some other way.

The FIGARCH(1,1) model (Fractionally Integrated GARCH) proposed by Bailie, Bollerslev and Mikkelsen (1996) is a genuine long-memory model. The model has the following structure:

$$\sigma_{t|t-1}^2 = \omega + \beta\sigma_{t-1|\mathrm{t}-2}^2 + [1 - \beta L - (1 - \alpha L - \beta L)(1 - L)^d]\varepsilon_{t-1}^2 \tag{18}$$

Where L denotes the lag operator and d is the parameter of the fractional integration. The expression $(1 - L)^d$ is then called the fractional filter and is defined by the following binomial expression:

$$(1 - L)^d = 1 - dL + \frac{d(d-1)}{2!}L^2 - \frac{d(d-1)(d-2)}{3!}L^3 + \cdots \tag{19}$$

For $d = 0$ the FIGARCH(1,1) model becomes GARCH(1,1) model, while for $d = 1$ it becomes the IGARCH(1,1) model (integrated GARCH) in which the variance follows a non-stationary process with no mean-reversion. For $0 < d < 1$ is the model mean-reverting with a hyperbolical decrease of the autocorrelation function, and it can further be shown that for $0 < d < 0.5$ it is mean-reverting as well as stationary, while for $0.5 < d < 1$ it is non-stationary but still mean-reverting (Ishida and Watanabe, 2008).

Another approach of how to tackle the long-range dependencies in the variance process is to approximate the long-memory with multiple short-memory processes. This is done in the CGARCH (Component GARCH) model of Engle and Lee (1999). The standard version of the model decomposes the conditional variance $\sigma^2_{t|t-1}$ into two components. The long-term component $\zeta^2_{t|t-1}$ follows a short-memory mean-reverting process around the long-term unconditional variance:

$$\zeta_t^2 = \omega + \rho\zeta_{t-1}^2 + \varphi(\varepsilon_{t-1}^2 - \sigma_{t-1|\mathrm{t}-2}^2) \tag{20}$$

While the short-term component $(\sigma^2_{t|t-1} - \zeta_t^2)$ follows a mean-reverting process around the long-term component ζ_t^2:

$$\sigma^2_{t|t-1} = \omega + \beta\sigma^2_{t-1|\mathrm{t}-2} + [1 - \beta L - (1 - \alpha L - \beta L)(1 - L)^d]\varepsilon^2_{t-1} \tag{21}$$

This enables the model to approximately capture the long-memory of $\sigma^2_{t|t-1}$.

Another way to tackle the long-range dependencies is to use sums of squared residuals computed over different time-horizons. This is done in the HARCH model (Heterogenous ARCH) proposed by Muller and Dacorogna (1997):

$$\sigma^2_{t|t-1} = \omega + \sum_{i=1}^{n} \gamma_i \left(\sum_{j=1}^{i} \varepsilon_{t-j} \right)^2 \tag{22}$$

The model is usually applied in a restricted version where only few horizons i are chosen, with the common choice being of $i = \{1,5,22\}$, corresponding to 1-day, 1-week (5 days) and 1-month (22 days) horizons. The makes the HARCH an early predecessor of the HAR-RV model of realized variance (Corsi, 2008), which is currently the most popular model from the realized variance modelling framework.

Alternative GARCH model modifications intending to capture the long-memory of the volatility process include the FIEGARCH model of Bollerslev and Mikkelsen (1996) or the model in Ding and Granger (1996), using a superposition of an infinite number of ARCH processes.

1.2.7. Fat tails of the return distribution

Another important effect that may not be perfectly captured by the standard GARCH models are the fat tails of the return distribution. While the time-varying volatility itself induces fat tails into the unconditional return distribution, it is still insufficient to capture some of the rapid movements that occur in the asset prices. To cope with this problem an alternative distribution can be specified for the noise in the price process. Bollerslev (1987) proposes to the use the Student-t distribution, giving rise to the GARCH-t model, while Nelson (1991) proposes to use the Generalized Error Distribution (GED).

1.2.8. Other GARCH modifications

Large number of additional GARCH model modifications have been proposed in the literature. For further discussion and references see Andersen et al (2005). A comprehensive list of GARCH modifications can be found in Bollerslev (2008). Comparison of predictive power of most of the commonly used ARCH/GARCH models can be found in Hansen and Lunde (2005).

1.3. Stochastic volatility models

While the ARCH/GARCH models model the conditional variance as a deterministic function of past squared daily returns, the Stochastic Volatility (SV) models allow it to follow a separate stochastic process with its own random component (Andersen et al., 2005). This is theoretically convenient as it enables wide variety of possible model specifications. At the same time, it complicates the model estimation as the latent state time series of the stochastic variances need to be estimated together with model parameters. For a review of most important studies on stochastic volatility models and their estimation methods, see Shephard (2005).

1.3.1. Log-SV model

One of the first stochastic volatility models that is still widely used today is the log-SV model proposed by Taylor (1986).

Let us assume that the logarithmic returns of the asset follow the following discrete time stochastic process:

$$r(t) = \mu(t) + \sigma(t)\varepsilon(t) \tag{23}$$

Where $r(t) = p(t) - p(t-1)$ denotes the logarithmic return, with $p(t)$ being the logarithm of the asset price at time t, $\sigma(t)$ is the conditional standard deviation for day t, $\mu(t)$ is the conditional mean and $\varepsilon(t) \sim N(0,1)$ is a Gaussian white noise.

The log-SV model assumes that the logarithm of the variance $h(t) = \ln[\sigma^2(t)]$ follows an autoregressive AR(1) process defined as follows:

$$h(t) = \alpha + \beta h(t-1) + \gamma \varepsilon_V(t) \tag{24}$$

Where $\alpha = (1 - \beta)\theta$ is the constant of the model, with θ denoting the long-term log-variance, β is the autoregression parameter, γ is the volatility of the log-variance and $\varepsilon_V(t) \sim N(0,1)$ is the Gaussian white noise variable governing the log-variance process.

The leverage effect (correlation between returns and volatility) can easily be captured by the model by introducing correlation between $\varepsilon(t)$ and $\varepsilon_V(t)$, determined by an additional parameter ρ.

The logarithmic form of the model is very convenient as it assures that the variance will never attain negative values, and there is thus no limitation on the parameter values, apart from the possible requirement for β to be $\beta < 1$ if we want the volatility to be stationary.

As the model contains latent state time series it cannot be easily estimated with standard estimation methods such as Maximum Likelihood Estimation (MLE) or Generalized Method of Moments (GMM). The non-linearity in the relationship between the log-variance and the returns further prevents accurate estimation with the use of Kalman Filter. The model thus usually has to be estimated with computationally intensive simulation methods such as Efficient Method of Moments (EMM), Markov Chain Monte Carlo (MCMC) or Sequential Monte Carlo (SME).

1.3.2. Long memory stochastic volatility models

The main drawback of the log-SV model is that its autocorrelation structure is relatively simple and that it cannot capture the long-memory of the stochastic variance process. Similarly to the ARCH/GARCH modelling framework, the problem can be solved either with a genuine long-memory model or with approximations.

The incorporation of genuine long memory using fractionally integrated processes into the stochastic volatility model setting was first performed by Breidt, Crato and De Lima (1998) and Harvey (1998).

In its most simple version, the LMSV (Long-Memory Stochastic-Volatility) model models the stochastic log-variance $h(t)$ as a fractional white noise:

$$(1-L)^d h(t) = \gamma \varepsilon_V(t) \tag{25}$$

In the more general version of the LMSV model, $h(t)$ is modelled with an ARFIMA(p,d,q) process:

$$(1-L)^d \phi(L) h(t) = \theta(L) \gamma \varepsilon_V(t) \tag{26}$$

Where L denotes the lag operator, and $\phi(L)$ and $\theta(L)$ are the AR and MA operators defined as:

$$\begin{aligned} \phi(L) &= 1 - \phi_1 L - \phi_2 L^2 - \cdots - \phi_p L^p \\ \psi(L) &= 1 + \psi_1 L + \psi_2 L^2 + \cdots + \psi_q L^q \end{aligned} \tag{27}$$

Breidt, Crato and De Lima (1998) propose an estimation of the model based on maximization of the spectral approximation of the Gaussian likelihood. Alternatively, Chronopolous (2017) estimates the model with Bayesian Sequential Monte Carlo (Particle Filters), by using the Particle Learning approach developed in Liu and West (2001).

1.3.3. Multi-Component stochastic volatility models

An alternative approach of how to approximate long-memory in the stochastic volatility model setting is by using models with multiple components. The model can be formulated as follows:

$$r(t) = \mu(t) + \sigma(t)\varepsilon(t)$$
$$\log[\sigma^2(t)] = \theta_0 + \sum_{i=1}^{k} \theta_i(t) \tag{28}$$
$$\theta_i(t) = \beta\theta_i(t-1) + \gamma_i\eta_i(t)$$

Where $\varepsilon(t) \sim N(0,1)$ and $\eta_i(t) \sim N(0,1)$ for all components i.

We can see that the model models the log-variance as an additive superposition of k independent AR(1) processes, which are assumed to exhibit different levels of persistence (values of β), which allows the model to approximately capture the long-memory behaviour of the volatility process.

1.3.4. Markov Switching Stochastic Volatility model

The idea that long-memory of stochastic volatility may in fact be caused by regime switching was first proposed by Lamoureux and Lastrapes (1990). Subsequently, Hamilton and Susmel (1994) proposed the Markov-Switching ARCH (SWARCH) model in which the parameters of an ARCH model change regimes. A similar Markov-Switching ARCH model was also proposed by Cai (1994).

The Markov-Switching Stochastic-Volatility model (MSSV) was developed by So, Lam and Li (1998) and the authors estimate it with an MCMC Gibbs Sampler. The model extends the standard log-SV model by allowing random shifts in the long-term log-variance parameter that are governed by a Markov process.

The MSSV(K) model can be expressed as follows.

Suppose that $s(t)$ is an unobservable discrete K-State Markov process, with domain $\{1,2, \ldots K\}$ and transition probability matrix:

$$\boldsymbol{P} = \begin{pmatrix} p_{1,1} & \cdots & p_{1,K} \\ \vdots & \ddots & \vdots \\ p_{K,1} & \cdots & p_{K,K} \end{pmatrix} \tag{29}$$

Where transition probabilities are given as $p_{ij} = Pr[s(t) = j|s(t-1) = i]$, with $\sum_{j=1}^{K} p_{ij} = 1$ for all $i = 1, \dots, K$.

The log-variance equation is then given as follows:

$$h(t) = \alpha_{s(t)} + \beta h(t-1) + \gamma \varepsilon_V(t) \tag{30}$$

Where:

$$\alpha_{s(t)} = \alpha_1 + \sum_{j=2}^{K} \alpha_j I_j(t) \tag{31}$$

Where $I_j(t)$ denotes an indicator variable that is equal to 1 when $s(t)$ is greater than or equal to j.

The authors further specify the model so that all α_j for $j = 2, \dots K$ are negative. The first regime does thus correspond to the highest log-variance level state, while the last regime corresponds to the lowest state.

1.3.5. Markov Switching Multifractal

An example of a pure-jump Markov-Switching model of stochastic volatility is the Markov-Switching-Multifractal (MSM) model, developed by Calvet and Fisher (2001). The model is able to capture long memory of the volatility process, as well as the fat tails of the short-horizon return distribution. And important advantage of the model is also that it has a tractable likelihood and it can thus be estimated with the Maximum Likelihood Method.

The model solves the problem of the transition matrix estimation by parametrizing it with only 4 parameters. This enables the model to utilize large number of components (10 or more), enabling for large number of volatility latent states, which improves its precision.

Originating from the multifractal Brownian motion process, the MSM model is also able to efficiently capture both, long-memory, as well as the fat tail of the short-horizon return distributions (commonly associated with jumps).

The MSM model assumes the returns to follow:

$$r(t) = \sigma(t)\varepsilon(t)$$

(32)

Where $\varepsilon(t) \sim N(0,1)$.

And the volatility $\sigma(t)$ is given as:

$$\sigma(t) = \sigma_{LT} \left(\prod_{i=1}^{k} M_i(t) \right)^{1/2} \tag{33}$$

Where the $M_i(t)$, for $i = 1, \dots, k$, represent multipliers, which are for each time t drawn randomly from a distribution M, with probability γ_i, given by:

$$\gamma_i = 1 - (1 - \gamma_1)^{(b^{i-1})} \tag{34}$$

Where $\gamma_1 \in (0,1)$ and $b \in (1, \infty)$. The given parametrization was introduced in order to corresponds to the discretization of Poisson arrivals with exponentially increasing intensities. As can be seen from the equation, given the value of b and any single γ_x, all other γ_i will be given. Due to numerical reasons Calvet and Fisher (2001) recommend to set as parameter the largest γ_i, which is the γ_k.

The final part of the model is the distribution M, from which the arrivals are drawn. This can, in principle, be any distribution, as long as $M \geq 0$ and $E(M) = 1$. Due to simplicity of estimation Calvet and Fisher (2001) propose the *Binomial MSM* model, in which distribution M attains only two values, m_0 and m_1, parametrized so that $m_1 = 2 - m_0$.

The Binomial MSM model thus has only 4 parameters: m_0, σ_{LT}, b and γ_k, and one meta-parameter, k, determining the number of multipliers.

A convenient property of the Binomial MSM model is, that it can be estimated with the Maximum Likelihood Method, by using Bayesian updating.

Lets denote $M(t)$ to be the Markov state vector, attaining finite set of values $m^1, \dots, m^d \in R_+^k$, and dynamics given by the transition matrix $A = \left(a_{i,j}\right)_{1 \leq i,j \leq d}$, with components $a_{i,j} = P\left[M(t+1) = m^j | M(t) = m^i\right]$. Conditional on the volatility state m^i, the return has the Gaussian density $N\left[r(t); 0, \sigma^2(m^i)\right]$, determined by the variance $\sigma^2(m^i)$, given as a function of m^i.

While we do not directly observe the Markov state vector $M(t)$, we can compute the conditional probability vector $\Pi(t) = [\Pi^1(t), \dots, \Pi^d(t)] \in R_+^k$, with the conditional probabilities defined as $\Pi^j(t) = P\big[M(t) = m^j | r(1), \dots, r(t)\big]$.

The values of the conditional probability vector $\Pi(t)$ are computed recursively, via Bayesian updating, based on the previous value $\Pi(t-1)$ and the return $r(t)$:

$$\Pi(t) = \frac{\omega[r(t)] * [\Pi(t-1)A]}{\{\omega[r(t)] * [\Pi(t-1)A]\}\mathbf{1'}} \tag{35}$$

Where $\mathbf{1} = (1, \dots, 1) \in R^d$ and $x * y$ denotes the Hadamard product, given as $(x_1 y_1, \dots, x_d y_d)$, for any $x, y \in R^d$. And:

$$\omega[r(t)] = \{N[r(t); 0, \sigma^2(m^1)], \dots, N[r(t); 0, \sigma^2(m^d)]\} \tag{36}$$

With the values of $\Pi^j(0)$ chosen as $\Pi^j(0) = \prod_{l=1}^{k} P\big(M = m^j\big)$ for all j.

The Log-Likelihood function is the equal to:

$$\ln L[r(t), \dots r(T); m_0, \sigma_{LT}, b\ , \gamma_k] = \sum_{t=1}^{T} \ln\{\omega[r(t)]\, .\, [\Pi(t-1)A]\ \} \tag{37}$$

where $x\ .\ y$ denotes the inner product $x_1 y_1 + \cdots + x_d y_d$ for any $x, y \in R^d$.

The Binomial MSM performs well empirical tests, outperforming the GARCH(1,1), as well as the FIGARCH(1,1) model (Calvet and Fisher, 2001).

The model can further be extended into an asymmetric version, which takes into account negative correlation between volatility and returns (Lux, 2008). The asymmetric MSM model can be estimated with Generalized Method of Moments (GMM).

1.3.6. Stochastic-Volatility Jump-Diffusion models

The stochastic volatility modelling framework further enables to decompose the variability of the price into its continuous component (stochastic volatility) and its discontinuous component (jumps). The models of this kind are called Stochastic-Volatility Jump-Diffusion (SVJD) models (first proposed in Bates, 1996).

The most commonly used SVJD model combines the log-SV model of Taylor (1986) with Poisson jumps. The model has 2 equations.

The return equation is:

$$r(t) = \mu + \sigma(t)\varepsilon(t) + J(t)Q(t) \tag{38}$$

Where $r(t)$ is the daily logarithmic return, μ is the constant drift rate, $\sigma(t)$ is the stochastic volatility, $\varepsilon(t)\sim N(0,1)$ is a standard normal white noise, $J(t)\sim N(\mu_J,\sigma_J)$ is a variable determining the jump sizes and $Q(t)\sim\mathrm{Bern}[\lambda]$ is a Bernoulli distributed variable determining the times of jump occurrences that occur with intensity λ.

And the log-variance equation is the given by the log-SV model:

$$h(t) = \alpha + \beta h(t-1) + \gamma\varepsilon_V(t) \tag{39}$$

Where $h(t) = \ln[\sigma^2(t)]$ is the logarithm of the conditional variance, long-term log-variance θ is given by $\alpha = (1-\beta)\theta$, β is the autoregression coefficient, γ is the volatility of the log-variance, and $\varepsilon_V(t)\sim N(0,1)$ is the white noise in the log-variance equation which can be correlated with $\varepsilon(t)$ with correlation ρ.

The model can be extended to account for jump clustering effects by allowing the jumps to be self-exciting (Fičura and Witzany, 2016). This can be achieved by letting the jumps to follow a Hawkes process with jump occurrences given by $Q(t)\sim\mathrm{Bern}[\lambda(t)]$, and the time-varying jump intensity $\lambda(t)$ calculated as follows:

$$\lambda(t) = \alpha_J + \beta_J\lambda(t-1) + \gamma_J Q(t-1) \tag{40}$$

Where $\lambda(t)$ is the jump intensity, long-term jump intensity θ_J can be calculated as $\mathrm{\alpha_J} = \left(1 - \mathrm{\beta_J} - \gamma_J\right)\theta_J$, parameter β_J gives the rate of exponential decay of the jump intensity to its long-term level, while γ_J is the self-exciting parameter telling us how much will the jump intensity increase in the day following a jump.

An additional extension of the model would be to add jumps into the log-variance equation as well (Eraker et al., 2003). The equation for the evolution of log-variances does then become:

$$h_t = \alpha + \beta h_{t-1} + \gamma \varepsilon_{V,t} + G_t U_t \tag{41}$$

Where $G(t) \sim N(\mu_G, \sigma_G)$ determines the log-variance jump sizes and $U(t) \sim \text{Bern}[\lambda_V]$ the log-variance jump occurrences. Correlation can further be imposed between the jump and log-variance jump sizes, $J(t)$ and $G(t)$, with parameter ρ_J, as well as on the binary variables, $Q(t)$ and $U(t)$, determining the jump occurrences, ρ_Q.

SVJD models are most commonly estimated with EMM, MCMC or Particle Filters. The estimation of models with jumps in price via the EMM method was first proposed in Andersen, Benzoni and Lund (2002) and Chernov et al. (2003), and via particle filters in Golightly, 2009. Models with jumps in price and volatility as well were estimated with MCMC in Eraker, Johannes and Polson (2003) and Eraker (2004), and with Particle Filters in Fičura and Witzany (2018). Similarly, a model with self-exciting jumps, estimated with MCMC, is proposed in Fičura and Witzany (2016), while models with infinite activity self-exciting jumps in price and volatility are proposed in Fulop, Li and Yu (2015) and estimated with Particle Filters.

1.3.7. Continuous time stochastic volatility models

Apart from volatility forecasting and Value at Risk estimation, stochastic volatility models have become an important tool used in the area of option pricing. For that purpose, they are usually formulated in continuous time, as the continuous time representation is convenient for the derivation of analytical solutions of the option prices. This allows the models to be estimated with calibration to the currently observed option prices, which should provide us with forward-looking parameter estimates, that can subsequently be used to value more exotic kinds of options on the same underlying asset.

Let us assume that the logarithm of the stock price follows the following continuous time stochastic process:

$$dp(t) = \mu(t)dt + \sigma(t)dW(t)$$

(42)

Where $dp(t)$ denotes the differential of the logarithm of the stock price, $\mu(t)$ is the instantaneous drift rate, dt is the differential of time, $\sigma(t)$ is the instantaneous volatility and $dW(t)$ is the differential of the Wiener process.

The instantaneous volatility $\sigma(t)$ can then be modelled with its own continuous-time stochastic process. A popularly used process for this kind is the Cox-Ingersoll-Ross (CIR) process, which used by the popular Heston model of option pricing (Heston, 1993). The CIR process looks as follows:

$$d\sigma^2(t) = \beta\big(\alpha - \sigma^2(t)\big)dt + \nu\sigma(t)dW_\sigma(t) \tag{43}$$

As the Heston model provides us with semi-analytical solutions for the option price, conditional on the volatility following the CIR process, we can use calibration to set the parameters so that they price all options currently observed on the market correctly. The calibrated model can then be used to price other, more exotic options, that are not quoted on the market.

Another continuous-time stochastic volatility model is the GARCH Diffusion process (Drost and Werker, 1996), which has the property of behaving as the stochastic version of the GARCH model when discretized. In the GARCH Diffusion model, volatility follows the following process:

$$d\sigma^2(t) = \beta\big(\alpha - \sigma^2(t)\big)dt + \nu\sigma(t)dW_\sigma(t) \tag{44}$$

Another popular model is the Hull-White model (Hull and White, 1987), representing a continuous-time version of the Log-SV model:

$$d\log\sigma^2(t) = \beta(\alpha - \log\sigma^2(t))dt + \nu dW_\sigma(t) \tag{45}$$

Similarly to the discrete-time stochastic volatility models, their continuous-time counterparts can also easily be adjusted to include correlation between returns and volatility by setting the correlation between $W(t)$ and $W_{\sigma}(t)$ as equal to ρ.

Similarly to the SV models, discrete time SVJD models can also be formulated in the continuous time as well. The SVJD model with self-exciting jumps in prices, described in the previous chapter, is, for example, just a discretization of the continuous-time model in the following form (Fičura and Witzany, 2015):

$$\begin{gathered} dp(t) = \mu dt + \sigma(t)dW(t) + j(t)dq(t) \\ dh(t) = \kappa[\theta - h(t)]dt + \xi dW_V(t) \\ d\lambda(t) = \kappa_J\big[\theta_J - \lambda(t)\big]dt + \xi_J dq(t) \end{gathered} \tag{46}$$

Where $h(t) = \ln[\sigma^2(t)]$, $\Pr[\mathrm{dq(t)} = 1] = \lambda(\mathrm{t})\mathrm{dt}$, $j(t) \sim N(\mu_J, \sigma_J)$, and the $W(t)$ and $W_V(t)$ are two Wiener processes with correlation ρ.

1.4. Realized volatility estimators

With the increased quality of intraday financial data, a new approach of volatility estimation and modelling emerged, utilizing non-parametric estimators of volatility computed from high-frequency data. The first estimator of this kind is the Realized Variance introduced by Andersen and Bollerslev (1998), converging, with increasing frequency of its estimation, to the quadratic variation of the price process. With advances in the asymptotic theory of power variations, addition estimators were proposed, enabling the estimation of other properties of the price process.

1.4.1. Quadratic variation and integrated variance

Assume that the logarithm of the asset price follows the following generally defined stochastic-volatility jump-diffusion process:

$$dp(t) = \mu(t)dt + \sigma(t)dW(t) + j(t)dq(t) \tag{47}$$

Where $p(t)$ is the logarithm of the asset price, $\mu(t)$ is the instantaneous drift rate, $\sigma(t)$ is the instantaneous volatility, $W(t)$ is a Wiener process, $j(t)$ is a process determining the jump sizes, and $q(t)$ is a counting process determining the times of jump occurrences.

The total variability of the logarithmic price process over a period between $t-1$ and t can be expressed with its *quadratic variation* defined as follows:

$$QV(t) = \int_{t-1}^{t} \sigma^2(s)ds + \sum_{t-1\leq s<t} \kappa^2\,(s), \tag{48}$$

Where the first term on the right-hand side represents continuous component of price variability, called *integrated variance*, while the second term represents the discontinuous component of price variability, called *jump variance*.

The equation can thus be rewritten as follows:

$$QV(t) = IV(t) + \, JV(t) \tag{49}$$

Where $QV(t)$ denotes the quadratic variation, $IV(t)$ the integrated variance and $JV(t)$ the jump variance.

As all of the quantities, $QV(t)$, $IV(t)$ and $JV(t)$, are unobservable on the market, they need to be estimated. This can be done in a parametric way, by the defining the underlying processes of $\sigma(t)$, $j(t)$ and $q(t)$, and estimating their parameters, as well the latent state series of stochastic volatility and jumps with Bayesian methods. Another approach of how to do it is to utilize the asymptotic theory of power variations and estimate the demanded quantities non-parametrically in a model-free way from high-frequency data.

1.4.2. Realized variance

The first and still most commonly used estimator of quadratic variation is the *realized variance*, calculated, one a daily basis, as the some of squared intraday returns, computed on some sufficiently high frequency (5-minut or 15-minute are the most common ones).

Specifically, denoting Δ as some intraday time interval and $r(t,\Delta)$ as the logarithmic return between $t-\Delta$ and t, we can define the realized variance as follows:

$$\mathrm{RV}(\mathrm{t},\Delta) = \sum_{\mathrm{j}=1}^{1/\Delta} \mathrm{r}^2(\mathrm{t}-1+\mathrm{j}\Delta,\Delta), \tag{50}$$

And it holds that $\mathrm{RV}(\mathrm{t},\Delta) \rightarrow \mathrm{QV}(\mathrm{t})$ as $\Delta \rightarrow 0$.

The realized variance should theoretically converge to the underlying quadratic variation when $\Delta \to 0$, providing an unbiased and consistent estimate of it. In practical settings microstructure noise effects (discreetness of the price grid and the bid-ask bounce effect), present at the ultra-high frequencies, create correlation between the high-frequency returns, causing the estimator to be positively biased. The simplest way of how to solve the problem is by using slightly higher frequencies (such as 15-minute) on which only negligible autocorrelation is present. Another option is to use jump-robust realized variance estimators that will be described later.

1.4.3. Realized bipower variation

As already mentioned, the realized variance converges to the quadratic variation of the price process. In many applications it may, however, be useful to decompose the quadratic variation into the integrated variance and the jump variance. The main reason for this is that the jumps tend to often follow different dynamics than the continuous stochastic volatility and it may thus be useful to use a different model for their modelling.

In order to perform the decomposition it is first necessary to estimate the integrated variance, which can be done with the *realized bipower variation* (defined by Barndorff-Nielsen and Shephard, 2004):

$$\mathrm{BV}(t,\Delta) = \frac{\pi}{2}\sum_{j=2}^{1/\Delta} |r(t-1+j\Delta,\Delta)||r(t-1+(j-1)\Delta,\Delta)|, \tag{51}$$

And it holds that $\mathrm{BV}(t,\Delta) \to \mathrm{IV}(t)$ when $\Delta \to 0$.

By taking multiples of subsequent absolute returns, the bipover variation eliminates the effect of jumps on the quadratic variation as with decreasing Δ their impact will converge towards zero.

The contribution of jumps can then roughly be estimated as:

$$\mathrm{RJV}(t,\Delta) = \mathrm{RV}(t,\Delta) - \mathrm{BV}(t,\Delta), \tag{52}$$

And it holds that $\mathrm{RJV}(t,\Delta) \to \mathrm{JV}(t)$ as $\Delta \to 0$.

Furthermore, as the negative values of $\mathrm{RJV(t,\Delta)}$ would lack any interpretation, the estimator is often adjusted as:

$$\mathrm{RJV(t,\Delta) = max[RV(t,\Delta) - BV(t,\Delta)]}, \tag{53}$$

1.4.4. Integrated quarticity

As long as we are not able to sample the asset returns at an infinitely high frequency, the estimates of $\mathrm{BV(t,\Delta)}$ and $\mathrm{RJV(t,\Delta)}$ will inherently be plagued by some estimation noise. This will cause the values of $\mathrm{RJV(t,\Delta)}$ to differ from zero in most of the days. In order to estimate the days on which jumps really occurred it is necessary to normalize the estimator, under the condition of no-jumps, and then take only its statistically significantly positive values. This is performed with the Z-Estimator developed by Barndorff-Nielsen and Shephard (2004) and Andersen, Bollerslev and Diebold (2007).

Before we define the Z-Estimator, it is necessary to define the *integrated quarticity*:

$$\mathrm{IQ}(t) = \int_{t-1}^{t} \sigma^4(s)ds \tag{54}$$

In the case of no jumps in the time series, the realized variance can be shown to follow the distribution $\mathrm{RV(t,\Delta)} \sim N[\mathrm{IV(t)}, 2\Delta\mathrm{IQ}(t)]$, and the integrated quarticity can be estimated with the *realized quarticity*:

$$\mathrm{RQ(t,\Delta)} = \sum_{\mathrm{j=1}}^{1/\Delta} \mathrm{r^4(t-1+j\Delta,\Delta)}, \tag{55}$$

Where $\mathrm{RQ(t,\Delta)} \rightarrow \mathrm{IQ}(t)$ when $\Delta \rightarrow 0$, as long as there are no jumps in the time series.

In the presence of jumps, integrated quarticity can be consistently estimated with the *realized tri-power quarticity*:

$$\mathrm{TQ(t,\Delta)} = \frac{\pi^{3/2}}{4\Delta}\Gamma\left(\frac{7}{6}\right)^{-3} \sum_{\mathrm{j=3}}^{1/\Delta} \mathrm{|r(t-1+j\Delta,\Delta)|^{4/3}|r(t-1+(j-1)\Delta,\Delta)|^{4/3}|r(t-1+(j-2)\Delta,\Delta)|^{4/3}} \tag{56}$$

As it holds that $\mathrm{TQ(t,\Delta)} \rightarrow \mathrm{IQ}(t)$ when $\Delta \rightarrow 0$.

1.4.5. Z-Estimator of jumps

Using the $\mathrm{RV}(\mathrm{t},\Delta)$, $\mathrm{BV}(\mathrm{t},\Delta)$ and $\mathrm{TQ}(\mathrm{t},\Delta)$ it is possible to define the Z-Estimator of jumps, by normalizing the differences between $\mathrm{RV}(\mathrm{t},\Delta)$ and $\mathrm{BV}(\mathrm{t},\Delta)$, to get a $\mathrm{Z}(\mathrm{t},\Delta)$ variable, which should asymptotically follow the standard normal distribution as long as they are no jumps in the price process:

$$Z(\mathrm{t},\Delta) = \frac{[RV(t,\Delta) - B\mathrm{V}(\mathrm{t},\Delta)]RV(t,\Delta)^{-1}}{\sqrt{[(\pi/2)^2 + \pi - 5]\max\{1, TV(\mathrm{t},\Delta)B\mathrm{V}(\mathrm{t},\Delta)^{-2}\}\Delta}} \tag{57}$$

The jumps in the time series can then be identified as corresponding to the days in which the values of $Z(\mathrm{t},\Delta)$ exceed a sufficiently high quantile α of the standard normal distribution. The jump variance can thus be estimated as:

$$EJV_{\alpha}(\mathrm{t},\Delta) = I\{Z(\mathrm{t},\Delta) > \Phi(\alpha)^{-1}\}[\mathrm{RV}(\mathrm{t},\Delta) - \mathrm{BV}(\mathrm{t},\Delta)] \tag{58}$$

Where $\mathrm{EJV}_{\alpha}(\mathrm{t},\Delta)$ is the estimator of the jump variance, $I\{.\}$ is the indicator function and $\Phi(\alpha)^{-1}$ is the quantile function of the standard normal distribution. The quantile α is then typically set to the values of 0.95, 0.99 or 0.999.

In order to assure that the sum of the integrated variance estimate and the jump variance estimate equals the quadratic variance estimate, it is common to re-estimate the integrated variance as follows:

$$EIV_{\alpha}(\mathrm{t},\Delta) = \mathrm{RV}(\mathrm{t},\Delta) - I\{Z(\mathrm{t},\Delta) > \Phi(\alpha)^{-1}\}[\mathrm{RV}(\mathrm{t},\Delta) - \mathrm{BV}(\mathrm{t},\Delta)] \tag{59}$$

1.4.6. Other jump estimators

Several alternative jump estimators have been proposed, see for example Corsi, Pirino and Reno (2010), Lee and Mykland (2008), or a comparison in Dumitru and Urga (2012).

1.4.7. Noise-robust realized estimators

Due to the microstructure noise effects (bid-ask bounce, discreetness of the price grid), inducing an artificial positive autocorrelation of returns on the ultra-high frequencies (typically 5-minute or less), the realized variance computed from these returns often exhibit a significant positive bias (Andersen et al., 2005). Multiple noise-robust versions of the realized variance estimator have been proposed in the past, such as the autocorrelation-robust realized variance (Hansen and Lunde, 2004), the sub-sampling based realized variance estimator (Zhang, Mykland and Ait-Sahalia, 2005), the realized kernel estimator (Barndorff-Nielsen et al, 2008) the duration-based realized variance estimator (Andersen, Dobrev and Schaumbur, 2009), or the small-sample robust estimator of Donovon et al. (2014). Nevertheless, in a large-scale empirical study Liu, Patton and Sheppard (2015) conclude that most of the alternative realized quarticity estimators do not perform significantly better than the 5-Minute realized variance.

Among additional theoretically interesting estimators based on power-variations, we would further like to mention the multi-power estimators of Ysusi (2006) and Shi and Peng (2009), and the nearest-neighbour truncation estimators of Andersen, Dobrev and Schaumburg (2015).

1.4.8. Realized semi-variance

In order to capture asymmetries in the realized variance, Barndorff-Nielsen et al. (2010) propose realized semi-variance estimators which compute the realized variance alternatively either only from positive, or only from negative values. This can then be used in volatility models in order to better capture the asymmetric reactions of volatility on positive vs. negative returns in the past.

Positive realized semi-variance can be defined as:

$$RV^{+}(\mathrm{t},\Delta) = \sum_{\mathrm{j}=1}^{1/\Delta} \mathrm{r}^2(\mathrm{t}-1+\mathrm{j}\Delta,\Delta) I[\mathrm{r}(\mathrm{t}-1+\mathrm{j}\Delta,\Delta) > 0], \tag{60}$$

While negative realized semi-variance is:

$$RV^{-}(\mathrm{t},\Delta) = \sum_{\mathrm{j}=1}^{1/\Delta} \mathrm{r}^2(\mathrm{t}-1+\mathrm{j}\Delta,\Delta) I[\mathrm{r}(\mathrm{t}-1+\mathrm{j}\Delta,\Delta) < 0], \tag{61}$$

Where $I[.]$ denotes the indicator function.

1.5. Realized volatility models

Once realized variance (or an analogical estimator of the quadratic variation or the integrated variance) is computed, it can be viewed as a proxy for the underlying stochastic variance of the underlying time series. The early approaches usually ignored the estimation noise and modelled directly the time series of the realized variance, by using time series models of the conditional mean. The ARIMA-RV and ARFIMA-RV models (Andersen et al., 2003), as well as the HAR-RV model (Corsi, 2008), and their modifications, are the major models of this group. The application of these models showed that realized variance is a powerful tool for modelling of the stochastic volatility, with the realized variance based models typically outperforming the ARCH/GARCH models as well as the stochastic volatility models (Pong et al, 2004, Koopman et al. 2005, Deo et al., 2006).

Subsequently, models that take into account the estimation noise of the realized variance started to be developed. Hansen and Lunde (2012) proposed the GARCH-RV model, modelling the conditional variance in a similar way to the standard GARCH model, but using the realized variance as an additional source of information. Similarly, papers of Takahashi et al. (2009), Dobrev and Szerszen (2010), Koopman and Scharth (2013), Maneesoonthorn (2016), or Fičura and Witzany (2017) propose to utilize the realized variance as an additional source of information in the estimation of stochastic volatility models.

In the recent years, models have also been proposed that view the estimation noise of the realized variance as heteroskedastic, with variance given by the integrated quarticity. This is the case of the popular HARQ model proposed by Bollerslev et al. (2016), as well as the HARS model proposed by Bekierman and Manner (2018).

1.5.1. ARIMA-RV and ARFIMA-RV

The use of ARIMA/ARFIMA models for the modelling realized variance was first proposed by Andersen et al. (2003).

The ARIMA methodology (Box and Jenkins, 1970) represents the standard approach used for the modelling of the conditional mean of time series with linear dependencies. The model combines AR (autoregression) terms and MA (moving average) terms to account for the relationships in the autocorrelation and the partial autocorrelation function of the stochastic process, while allowing for the underlying process to be stationary (integrated of order $d = 0$) as well as non-stationary (integrated of order $d \geq 1$).

The model can be further extended into the ARFIMA model (Granger and Joyeux, 1981), by allowing the order of integration parameter d to attain fractional values. This model will then contain long-memory, with slow, hyperbolically decaying autocorrelation function, as is commonly observed for the realized variance time series.

The ARIMA(p,d,q) as well as the ARFIMA(p,d,q) models can be expressed as:

$$\phi(L)(1-L)^d(RV(t,\Delta)-\mu_{RV})=\psi(L)z_t \tag{62}$$

Where $\phi(L)$ and $\psi(L)$ represent the polynomial AR and MA operators.

The AR operator $\phi(L)$ is defined as:

$$\phi(L)=1-\phi_1 L-\phi_2 L^2-\cdots-\phi_p L^p \tag{63}$$

The MA operator $\psi(L)$ is defined as:

$$\psi(L)=1+\psi_1 L+\psi_2 L^2+\cdots+\psi_q L^q \tag{64}$$

Where L corresponds to the lag operator, d to the (fractional) difference operator, μ_{RV} is the unconditional mean of the realized variance and $z_t \sim i.i.d.$, $E(z_t)=0$, $Var(z_t)=1$ is the white noise variable, usually assumed to be Gaussian.

The difference parameter d plays a crucial role with respect to the memory of the modelled process. In the ARIMA model framework, the value of $d=0$ corresponds to a stationary, short-memory process, while a value of $d=1$ corresponds to a non-stationary process with infinite memory.

In the ARFIMA modelling framework, the value of d can attain fractional values. The model has a long memory if $d>0$. For $d\geq 1$ it is nonstationary, with infinite memory and without mean-reversion. For $1/2\leq d<1$ it is non-stationary, with long memory, but with a mean-reversion. For $0<d<1/2$ it is stationary, with long memory and with mean reversion. For $d=0$ it is a short-memory process. And for $-1<d<0$ it will have medium memory, while being stationary and mean-reverting.

When applied to realized variance forecasting, the parameter d usually attains values around 0.4-0.5, corresponding to a stationary, mean-reverting process, with a very long memory.

1.5.2. HAR-RV models

HAR-RV (Heterogenous Autoregression) model of the realized variance (Corsi, 2008) utilizes realized variance aggregated over different horizons (daily, weekly and monthly) in order to approximate the long-memory of the volatility process. The simplicity of the model, its ability to be estimated with simple OLS regression, and its good predictive power (comparable to the ARFIMA-RV) model, have made the HAR-RV model into one of the most popular realized volatility models used today.

HAR-RV model performs regression of the daily realized variance on the realized variance aggregated over the last 1-day, 1-week (5 days) and 1-month (22 days):

$$RV_d(t) = \beta_0 + \beta_d RV_d(t-1) + \beta_w RV_w(t-1) + \beta_m RV_m(t-1) + \epsilon(t) \tag{65}$$

Where RV_d denotes the daily realized variance, RV_w the weekly realized variance and RV_m the monthly realized variance.

The model can alternatively be used in a square root version or a logarithmic version. Their equations would look as follows:

$$RV_d^{1/2}(t) = \beta_0 + \beta_d RV_d^{1/2}(t-1) + \beta_w RV_w^{1/2}(t-1) + \beta_m RV_m^{1/2}(t-1) + \epsilon(t) \tag{66}$$

And

$$\begin{aligned} \log[RV_d(t)] = \beta_0 + \beta_d \log[RV_d(t-1)] + \beta_w \log[RV_w(t-1)] \\ + \beta_m \log[RV_m(t-1)] + \epsilon(t) \end{aligned} \tag{67}$$

1.5.3. Asymmetric HAR models

Multiple extensions of the HAR model have been proposed in the literature in order to capture the correlation between volatility and returns.

A simple Asymmetric HAR (AHAR) model (Corsi and Reno, 2009) can be constructed as follows:

$$RV_d(t) = \beta_0 + \beta_d RV_d(t-1) + \beta_w RV_w(t-1) + \beta_m RV_m(t-1) + \gamma_d Ret_d(t-1) + \gamma_w Ret_w(t-1) + \gamma_m Ret_m(t-1) + \epsilon(t) \quad (68)$$

Where RV_d, RV_w and RV_m correspond to the daily, weekly and monthly realized variances, while Ret_d, Ret_w and Ret_m are the daily, weekly and monthly realized returns.

Alternatively, Patton and Shephard (2015), utilize the semi-variance measures developed by Barndorff-Nielsen et al. (2010) to construct a semi-variance HAR model (SHAR). The model can be defined as follows:

$$RV_d(t) = \beta_0 + \beta_d^+ RV_d^+(t-1) + \beta_w^+ RV_w^+(t-1) + \beta_m^+ RV_m^+(t-1) + \beta_d^- RV_d^-(t-1) + \beta_w^- RV_w^-(t-1) + \beta_m^- RV_m^-(t-1) + \epsilon(t) \quad (69)$$

Where RV_d^+, RV_w^+ and RV_m^+ correspond to the daily, weekly and monthly positive semi-variance, while RV_d^-, RV_w^- and RV_m^- correspond to the daily, weekly and monthly negative semi-variance.

1.5.4. HAR model with jumps

As the jump component of the quadratic variation tends to have different dynamics than the continuous component, Andersen et al. (2007) proposed extensions of the HAR model that decompose the realized variance into its components and estimate the parameters for each of them separately.

The HAR-J model uses the simple decomposition based on realized variance and bi-power variation. Specifically, lets define the realized jump variance as follows:

$$\mathrm{RJV}(\mathrm{t}, \Delta) = \max[\mathrm{RV}(\mathrm{t}, \Delta) - \mathrm{BV}(\mathrm{t}, \Delta)], \quad (70)$$

Where $\mathrm{RV}(\mathrm{t}, \Delta)$ is the realized variance and $\mathrm{BV}(\mathrm{t}, \Delta)$ the realized bi-power variation.

The HAR-J model, used in Andersen et al. (2007), is then defined as:

$$RV_d(t) = \beta_0 + \beta_d RV_d(t-1) + \beta_w RV_w(t-1) + \beta_m RV_m(t-1) + \beta_{J,d} RJV_d(t-1) + \epsilon(t) \quad (71)$$

Where RJV_d corresponds to the jump component estimated over the last one day.

The model can further be extended into a specification utilizing the jump component estimated over the last week and month as well. The equation will then look as follows:

$$RV_d(t) = \beta_0 + \beta_d RV_d(t-1) + \beta_w RV_w(t-1) + \beta_m RV_m(t-1) + \\ \beta_{J,d} RJV_d(t-1) + \beta_{J,w} RJV_w(t-1) + \beta_{J,d} RJV_w(t-1) + \epsilon(t) \tag{72}$$

Where $RJV_w(t-1) = \sum_{i=1}^{5} \mathrm{RJV}(\mathrm{i}, \Delta)$ and $RJV_m(t-1) = \sum_{i=1}^{22} \mathrm{RJV}(\mathrm{i}, \Delta)$.

Similarly to the HAR model, the HAR-J model can also be specified in its logarithmic or a square root version. The equations are analogical to the HAR model case, with the only difference being that the RJV terms need to be set as $\log(1+RJV)$ instead of $\log(RJV)$ to avoid computing logarithms from zero numbers.

Additionally, Andersen et al. (2007) propose the HAR-JC model, utilizing the Z-Estimator to decompose the realized variance into its continuous and discontinuous part.

Lets define as EJV_α the jump component of the realized variance and as EIV_α the continuous component of the realized variance, both estimated by using the Z-Estimator on a probability level α. Skipping the α in the notation, we can defined the HAR-JC model as:

$$RV_d(t) = \beta_0 + \beta_{C,d} EIV_d(t-1) + \beta_{C,w} EIV_w(t-1) + \beta_{C,m} EIV_m(t-1) + \\ \beta_{J,d} EJV_d(t-1) + \beta_{J,w} EJV_w(t-1) + \beta_{J,d} EJV_w(t-1) + \epsilon(t) \tag{73}$$

1.5.5. HAR-GARCH and ARFIMA-GARCH

As the volatility of the stochastic variance, and subsequently the realized variance, is itself time-varying, models have been proposed that view the realized variance as heteroskedastic. The first of these models was HAR-GARCH model, proposed by Corsi et al. (2008). The combines the HAR model, modelling the conditional mean of the realized variance, with the GARCH model, modelling the conditional volatility of the realized variance. With a similar logic, an ARFIMA-GARCH model was proposed by Ishida and Watanabe (2008).

1.5.6. HARQ and HARQ-F models

A powerful extension of the HAR model is the HARQ model of Bollerslev, Patton and Quaedvlieg (2015), which utilizes the realized quarticity in order to make the slope parameters of the HAR model time-varying, depending on the noise of the realized variance estimator. The underlying idea is that while the noise in the realized variance is high, the slope parameter should be lower, while in the cases of low noise, it should be high in order to capture the realized volatility persistence.

As the authors argue, the noise has the most adverse effects on the estimation of the β_d parameter of the HAR model. The thus propose to make only the β_d parameter time-varying. The specification of the HARQ model does then look as follows:

$$\begin{aligned} RV_d(t) = \beta_0 + \left[\beta_d + \beta_{Q,d} RQ_d^{1/2}(t-1)\right] RV_d(t-1) + \beta_w RV_w(t-1) \\ + \beta_m RV_m(t-1) + \epsilon(t) \end{aligned} \tag{74}$$

Where $RQ_d^{1/2}(t-1)$ denotes the square root of the yesterday realized quarticity.

We can see that by multiplying the equation in front of $RV_d(t-1)$ the model becomes equal to the standard HAR model, with an additional term, given by $\beta_{Q,d} RQ_d^{1/2}(t-1) RV_d(t-1)$. The can thus by estimated with simple OLS linear regression, which is an additional benefit of its construction.

Time-variability of the parameters, depending on the realized quarticity, can of course be applied to the other regressors as well. Bollerslev, Patton and Quaedvlieg denote this specification as HARQ-Full (HARQ-F) model. Its equation would be as follows:

$$\begin{aligned} RV_d(t) = \beta_0 + \left[\beta_d + \beta_{Q,d} RQ_d^{1/2}(t-1)\right] RV_d(t-1) \\ + \left[\beta_w + \beta_{Q,w} RQ_w^{1/2}(t-1)\right] RV_w(t-1) \\ + \left[\beta_m + \beta_{Q,m} RQ_m^{1/2}(t-1)\right] RV_m(t-1) + \epsilon(t) \end{aligned} \tag{75}$$

Where $RQ_d^{1/2}$, $RQ_w^{1/2}$ and $RQ_m^{1/2}$ denote the square roots of the realized quarticity, computed over the last 1-day, 1-week (5 days) and 1-month (22-days), respectively.

In the empirical tests, performed by Bollerslev, Patton and Quaedvlieg (2015), the HARQ and HARQ-F models outperformed all of the other tested HAR specifications. At the same time there was no significant difference in performance between HARQ and HARQ-F.

1.5.7. State Space HAR

A more advanced approach of how to cope with the heteroskedasticity of the realized variance noise was proposed by Bekierman and Manner (2018), who propose to make the autoregression parameter in the HAR model time-varying via a state-space model. They call their model HARS (or State Space HAR) and it can be written as follows:

$$\begin{aligned} &RV_d(t) = \beta_0 + [\beta_d + \lambda(t)]RV_d(t-1) + \beta_w RV_w(t-1) + \beta_m RV_m(t-1) + \epsilon(t) \\ &\lambda(t) = \phi\lambda(t-1) + \eta(t) \end{aligned} \tag{76}$$

Where $\eta(t) \sim N(0, \sigma_\eta^2)$.

The model thus contains a latent component. Nevertheless, as the relationships between the observable and the latent time series are linear, and the error components are Gaussian, the estimation can be performed with Maximum Likelihood, computed via the Kalman Filter (the model thus differs from the stochastic volatility models in which the underlying volatility itself is a latent variable, which results in non-linear relationships between the latent variable and the observable time series).

Bekierman and Manner (2018) further propose several extensions of the model, utilizing the realized quarticity $RQ(t)$ as an additional factor in the estimation of $\lambda(t)$, leading to a class of HARSQ models. The simplest way of how to achieve this would be to set:

$$\lambda(t) = \phi\lambda(t-1) + \gamma RQ^{1/2}(t) + \eta(t) \tag{77}$$

In the study performed by the authors, the state space HAR models outperformed the traditional HAR as well as the HARQ model of Bollerslev, Patton and Quaedvlieg (2015) on a dataset of 40 stocks. The results also indicate that estimating the model in its logarithmic version (i.e. with all predictors logarithmized) further increases its performance.

1.6. Realized Variance in SV models

The realized variance and other power-variation estimators can be incorporated into Stochastic Volatility models as well. The main benefit of this is that while these estimators provide relatively accurate estimates of the underlying volatility, they are still plagued by some estimation noise, which has an effect on the realized volatility model parameters and may negatively influence their performance. In the stochastic volatility model setting it is possible to address this noise, and as the modelled quantity is the unobservable stochastic variance, the parameter estimates should theoretically be unbiased by the estimation noise.

1.6.1. SV-RV

The SV-RV model, proposed by Takahashi, Omori and Watanabe (2009) uses the realized variance estimator as additional source of information in the estimation of the Log-SV model. They are thus using two observable time series for its estimation. The series of logarithmic returns $r(t)$ and the series of realized variances $RV(t)$. The model can be written as follows:

$$\begin{aligned} r(t) &= \mu + \sigma(t)\varepsilon(t) \\ h(t) &= \alpha + \beta h(t-1) + \gamma\varepsilon_V(t) \\ \log[RV(t)] &= h(t) + \sigma_{RV}\varepsilon_{\mathrm{RV}}(t) \end{aligned} \tag{78}$$

Where $h(t) = \log[\sigma^2(t)]$, $\varepsilon(t) \sim N(0,1)$, $\varepsilon_V(t) \sim N(0,1)$, and $\varepsilon_{RV}(t) \sim N(0,1)$. The noise terms $\varepsilon(t)$ and $\varepsilon_V(t)$ can potentially be mutually correlated, while they are both assumed to be uncorrelated with $\varepsilon_{\mathrm{RV}}(t)$.

We can further see that the model implicitly assumes that the logarithm of the realized variance provides an unbiased estimate of the log-variance $h(t)$, with constant variance of the noise equal to σ_{RV}. This unbiasedness property is usually fulfilled only approximately, due to the microstructure noise effects which may cause the $RV(t)$ to slightly overestimate the underlying quadratic variation. In the cases where this problem appears to be severe, an additional parameter μ_{RV} can be included into the third equation of the model, in order to adjust for this bias. As for the second important assumption, the heteroskedasticity of the noise $\sigma_{RV}\varepsilon_{\mathrm{RV}}(t)$, it can be alleviated by making the variance of the noise time-varying which will be examined later.

1.6.2. SVJD-RV

The SV-RV model can be extended to include jumps in the returns equation. The resulting SVJD-RV model, proposed by Fičura and Witzany (2015) and Fičura and Witzany (2017), looks as follows:

$$\begin{aligned} r(t) &= \mu + \sigma(t)\varepsilon(t) + J(t)Q(t) \\ h(t) &= \alpha + \beta h(t-1) + \gamma\varepsilon_V(t) \\ \log[RV(t) - J^2(t)Q(t)] &= h(t) + \sigma_{RV}\varepsilon_{\mathrm{RV}}(t) \end{aligned} \tag{79}$$

The third equation does in this case stem from the definition of the integrated quarticity as the sum of the integrated variance and the jump variance, equal to the sum of the squared jumps during the given day. As we want to use the $RV(t)$ to estimate the $h(t)$, it is necessary to adjust it for the impact of estimated jumps first, before is logaritmized and the noise term applied to it.

1.6.3. SVJD-RV-Z

Fičura and Witzany (2017) further propose the SVJD-RV-Z model, utilizing the realized variance as well as the Z-Estimator to improve the estimation of jumps in the stochastic volatility model setting.

A problematic aspect of using the Z-Estimator is that it tends to identify large number of intraday jumps on almost every day. Most of these jumps are relatively small and do not have a significant impact on the distribution of return on the daily frequency. The idea of the SVJD-RV-Z model therefore is to make a distinction between small intra-day jumps and the large jumps that have an impact on the daily returns, and to utilize the fact that large jumps tend to influence the Z-Estimator more.

The SVJD-RV-Z model used the same equations as the SVJD-RV model, with an additional fourth equation, providing a link between the value of the Z-Estimator and the identified jumps.

$$Z(t) = \mu_Z + \xi_Z Q(t) + \sigma_Z \varepsilon_Z(t) \tag{80}$$

Where $Z(t)$ denotes the Z-Estimator and $Q(t)$ the identified jump occurrences, while the μ_Z correspond to the mean value of the Z-Estimator in the days of no-jumps or only intraday jumps, while the ξ_Z represents a shift in the mean of $Z(t)$ in the days when the large jumps occur, σ_Z is the volatility of $Z(t)$ around the estimated mean and $\varepsilon_Z(t) \sim N(0,1)$ is a Gaussian white noise variable.

1.6.4. Other approaches

Dobrev and Szerszen (2010) and Koopman and Scharth (2013) have further explored the utilization of realized variance estimators in stochastic volatility models, including multi-component ones and the possibility to use multiple realized variance at once. Maneesoonthorn, Forbes and Martin (2017), on the other hand, provide an alternative extension to the SVJD model setting.

1.7. Machine learning methods

Multiple studies have tried to apply machine learning based methods to the problem of realized volatility forecasting, with generally conflicting results as to whether the machine learning methods are able to outperform traditional approaches (McAller and Medeiros, 2011, Vortelinos, 2015, Fičura, 2017). The main benefit of machine learning methods, such as neural networks, is that they are able to learn non-linear dependencies in the analysed time series, without the need to specify their exact functional form by the researcher (i.e. they are universal approximators).

1.7.1. Echo State Neural Networks

Echo State Neural Networks (ESN) represent simple but powerful recurrent neural networks, whose main advantage is that they can be trained quickly with penalized Ridge Regression (Jaeger and Haas, 2008), while most other types of recurrent neural networks have to be trained with Backpropagation Through Time algorithm, which is computationally much more demanding and tends to suffer from the vanishing gradient problem, preventing it to learn long-range dependencies. Echo State Neural Networks were thus able to significantly outperform many earlier recurrent neural networks, in task such as chaotic time series prediction, wind speed forecasting or financial time series prediction (Lukoševičius and Jaeger, 2009). Nowadays, more complex neural networks, such as LSTM (Long Short-Term Memory neural networks), managed to outperform ESN, especially on task where large datasets are available. Nevertheless, ESN still represent a powerful and easy to use approach, especially on small datasets with nonlinear dependencies. Echo State Neural Networks were first used for realized variance forecasting in Fičura (2017), where they achieved performance comparable to the AHAR model on 19 stock market index realized variance time series.

The Echo State Neural Network model has the following three equations:

$$\begin{aligned} Rez_t^* &= f(W_{IN}[1;X_t] + W_{Rez}Rez_{t-1}) \\ Rez_t &= (1-\alpha) * Rez_{t-1} + \alpha * Rez_t^* \\ y_t &= W_{Out}[1;Rez_t] \end{aligned} \tag{81}$$

Where X_t denotes a m x 1 vector of predictors, y_t is the target variable, Rez_t is a n x 1 vector representing an output of the hidden layer of the neural network which is in the ESN literature called *reservoir*, α is a n x 1 vector of smoothing parameters (with the $*$ denoting a Hadamard product). W_{IN} is a $\mathrm{n\,x\,m} + 1$ matrix of parameters of the *input layer*, W_{Rez} is a n x n matrix of parameters corresponding to the *recurrent layer*, and W_{Out} is a $1 \text{ x } \mathrm{n}+1$ vector of parameters corresponding to the *output layer*. The function $f(.)$ is a sigmoid function (most commonly the logistic function or the hyperbolic tangent function), representing the *activation function* of the neural network, which is applied to each element of the hidden layer separately.

Echo State Neural Networks differ from traditional recurrent neural networks (Elman, 1990) in the fact that the W_{IN} and W_{Rez} matrices are generated randomly (with only the scaling of W_{IN} and the spectral radius of W_{Rez} possibly being optimized with meta-optimization). The output vector W_{Out} is then the only part of the neural network being estimated, which can be done with penalized Ridge Regression (linear regression with the penalization on the L_2 norm of the parameter vector, with the penalization parameter either optimized with cross-validation or set to a default value of 1 for normalized time series). For best practices regarding the implementation of ESN, see Lukoševičius (2012).

Echo State Neural networks were first used for realized volatility forecasting in Fičura (2017). The model proposed by the author used $y_t = RV_t$ as the target variable, and either $X_t = RV_{t-1}$ or $X_t = [RV_{t-1}, r_{t-1}]$ as the explanatory variables. In the current study, we will slightly modify the model proposed in Fičura (2017) by setting $y_t = RV_t - RV_{t-1}$ and $X_t = RV_{t-1} - RV_{t-2}$, predicting thus the differences of the realized variance instead of its level. The motivation for this change is the fact that ESN trained on the levels of the realized variance on a period not including the financial crisis tended to significantly underestimate the realized variance during the crisis. This is caused by the fact that the logistic activation function in the hidden layer is bounded, and the neural network is thus not able to extrapolate well towards values that lie outside of the range of values that it has seen during training.

In addition the model using only RV_{t-1} as the explanatory variable, further models will be tested in the empirical part of this thesis, utilizing additional power variation estimators as predictors, such as the realized quarticity, the realized semi-variance, and the realized bi-power variation based estimates of jumps.

An additional change will be that instead of setting α as a scalar equal to $\alpha = 1$, we will set it as a vector of equally spaced values ranging from 0.05 to 1. Specifically, if n is the number of elements (neurons) in the hidden layer, then $\alpha = 0.05 + 0.95[\frac{0}{n}, \frac{1}{n}, \frac{2}{n}, \dots, \frac{n}{n}]$, and each neuron will thus use different level of smoothing. This should improve the performance of the model to learn dependencies occurring at different frequencies (Lukoševičius, 2012), by enabling the neural network to pick the information from the neurons that use smoothing corresponding to the frequencies that are most informative with respect to the future realized variance.

In addition to the simple version of the ESN model, containing only single predictor, equal to $X_t = RV_{t-1} - RV_{t-2}$, extended versions of the model will be tested, with X_t containing the lagged differences of the realized variance, bi-power variation based jump estimates, realized semi-variance and the realized quarticity (mimicking the variables used in different versions of the HAR model, but with one lag 1 value used as predictors, as the neural network should be able to learn the long-memory by itself, due to its recurrent structure). All models will additionally be estimated in a standard, logarithmic and a square-root form (i.e. with transformed inputs and outputs), in the same way as it is commonly done in the HAR models.

1.8. Option-based volatility models

Alternative approach to forecast volatility of financial instruments is to utilize the information embedded in the option prices quoted on the market. Due to the non-linearity of the option payoff function, option prices strongly depend on the expected volatility of the underlying asset price until maturity. If we assume all other determinants of the option price (asset price, strike price, risk free interest rate and time until maturity) as given, it is possible to use the option prices to calculate the implied volatility, expected by the market participants until the maturity of the option.

One drawback of this approach is that the option prices tend to include the volatility risk premium, which causes the implied volatility forecasts to systematically overestimate the future realized volatility. In spite of this bias, option based volatility forecasts possess the significant advantage with respect to the other models, in the fact that they are forward-looking, while the traditional volatility models, such as GARCH or HAR are backwards-looking. This causes the option based volatility models to often outperform the traditional approaches (Poon and Granger, 2003, Pong et al, 2004, Koopman et al., 2005).

1.8.1. Black-Scholes implied volatility

The simplest way of how to compute the implied volatility from option prices is to utilize the Black-Scholes option pricing model (Black and Scholes, 1973):

$$C(S,t) = Se^{-q(T-t)}N(d_1) - Ke^{-r(T-t)}N(d_2) \tag{82}$$

Where

$$d_1 = \frac{\ln\left(\frac{S}{K}\right) + (r-q)(T-t)}{\sigma\sqrt{T-t}} + \frac{\sigma\sqrt{T-t}}{2} \tag{83}$$

And:

$$d_2 = \frac{\ln\left(\frac{S}{K}\right) + (r-q)(T-t)}{\sigma\sqrt{T-t}} - \frac{\sigma\sqrt{T-t}}{2} = d_1 - \sigma\sqrt{T-t} \tag{84}$$

Where $C(S,t)$ denotes the European option price at time t, S is the price of the underlying asset, K is the strike price of the option, σ is the expected volatility until maturity of the option, $T-t$ is the time until maturity, r is the risk-free rate, q is the asset yield, and $N(.)$ is the standard normal cumulative distribution function.

In order to calculate the implied volatility from the option price, all other inputs should be put into the formula (except for σ) and it should solved for σ with numerical methods.

The main drawback of the Black-Scholes implied volatility is that the Black-Scholes model assumes a simple Geometric Brownian Motion process, with constant volatility for the behavior of the asset price. This process is highly unrealistic and implies the future asset price to follow the lognormal distributed, while the real empirical distribution has usually far greater tails, and it may also be skewed with respect to the lognormal distribution in the cases when correlation between asset price and volatility is present. As a result of this, the B-S Implied volatility is typically depends on the strike price of the options from which it is computed (Jackwerth and Rubinstein, 1996), leading to the famous effect of volatility smile (options far from at-the-money have higher implied volatilities, to compensate for the excess kurtosis of the price process) on the forex markets, and the volatility skew on the stock markets (options with strike prices below the current stock price have the highest implied volatilities in order to compensate between the correlation between stock returns and volatility). Similarly, for assets with mean-reverting processes (such as interest rates), a volatility frown (i.e. inverted smile) can sometimes be observed, in order to compensate for the mean-reversion of the price process that violates the Geometric Brownian Motion assumption of the B-S model.

1.8.2. Model-Free implied volatility

Neuberger and Britten-Jones (2000) introduced an alternative approach of implied volatility computation called Model-Free volatility. The main advantages of the approach are that it is theoretically valid for wide variety of underlying asset price processes, including processes with jumps, which was proved by Jiang and Tian (2005). The approach also uses all options quoted on the market with given maturity, so it works with a broader set of information than the B-S implied volatility in which a single option has to be chose.

The *Model-Free volatility* is defined as follows:

$$E_0^F\left[\int_0^T\left(\frac{dF_t}{F_t}\right)^2\right] = 2\int_0^{\infty}\frac{C^F(T,K)-\max(0,F_0-K)}{K^2}dK \tag{85}$$

Where $C^F(T,K)$ denotes the forward option price with maturity at T, so that $C^F(T,K)=C(T,K)/B(T,t)$, where $B(T,t)$ is the price of a bond at time t that will pay 1 USD at time T. F_t denotes the forward price of the asset at time t, with maturity in T, and E_0^F denotes an expectation under the T-forward risk-neutral probability measure.

In practical settings, the integral in the equation needs to be approximated with numerical integration over the options with maturity in T, over the whole range of available strike prices (as long as the quoted options are reasonably liquid). Call options as well as Put options can be used in the computation, with the praxis being to use a combination of all Call and Put options that are out-of-the money (as they tend to be more liquid than in-the-money options).

1.8.3. Volatility risk premium

Option based volatility forecasts typically achieve comparable or better predictive results than the of the econometric time series models (Poon and Granger, 2003, Pong et al, 2004, Koopman et al., 2005).

A problematic aspect of using option based volatility forecast is the existence of volatility risk premium, which cause the option implied volatility to significantly overestimate the future realized volatility (Bakshi and Kapadia, 2003, Eraker, 2009). The volatility risk premium is a premium return, charged by the option sellers, for the inconvenience of being short in volatility, which is highly negatively correlated with market returns, and also highly asymmetric, with the potential of increase being usually significantly higher than the potential of future decrease (Carr and Wu, 2007).

The standard way of how to cope with the problem of volatility risk premium, when forecasting volatility, is to compute a regression between the realized volatility and the option implied volatility on historical data. The parameters of the regression can then be used to adjust the implied volatility forecasts and eliminate their bias (Muzzioli, 2008).

The computed regression would be:

$$\mathrm{RV}(\mathrm{t}, \mathrm{t}+\mathrm{h}) = \alpha + \beta\sigma_{IV}^2(t, t+h) + \varepsilon(t) \tag{86}$$

Where $\mathrm{RV}(\mathrm{t}, \mathrm{t}+\mathrm{h})$ denotes the historically observed realized variance for the period from t to $\mathrm{t}+\mathrm{h}$, $\sigma_{IV}^2(t, t+h)$ denotes the implied variance from the option prices at time t, for the horizon from t to $t+h$, $\varepsilon(t)$ is a white noise variable and α and β are parameters estimated with OLS linear regression. Once the parameters are estimated, they can subsequently be used to adjust the implied volatility forecasts into the future. Alternatively, the equation can be estimated for the implied standard deviation, or the implied log-variance.

A drawback of the aforementioned approach is that it does not take into account the time-varying character of the volatility risk premium, which has been empirically observed in practice, see Carr a Wu (2008), Todorov (2009), Chen a Poon (2013) or Fičura (2014).

2. SVJD models analysed in the thesis

The main focus of the thesis is placed on SVJD models and their extensions utilizing high-frequency power-variation estimators as additional sources of information (i.e. the SVJD-RV-Z class of models), following the research in Fičura and Witzany (2015), Fičura and Witzany (2016) and Fičura and Witzany (2017). In the following sections, Bayesian methods used for SVJD model estimation are discussed, and the Sequential Gibbs Particle Filter algorithm (developed in Fičura and Witzany, 2018) is explained, as an efficient tool for parameter learning and latent state filtering in SVJD models. In the empirical part of the study, the predictive performance of the proposed SVJD models is compared with a series of benchmark models from the GARCH and HAR model frameworks, as well as with models based on Echo State Neural Networks, proposed for realized volatility forecasting in Fičura (2017). Due to the large number of possible SVJD model architectures and high computational demand for their estimation, we will focus on four SVJD model architectures that are described below.

2.1. SVJD model with self-exciting jumps in prices

The SVJD model with self-exciting jumps in prices (proposed in Fičura and Witzany, 2016 and estimated with MCMC) has the following three equations:

$$\begin{aligned} r(t) &= \mu + \sigma(t)\varepsilon(t) + J(t)Q(t) \\ h(t) &= \alpha + \beta h(t-1) + \gamma\varepsilon_V(t) \\ \lambda(t) &= \alpha_J + \beta_J\lambda(t-1) + \gamma_J Q(t-1) \end{aligned} \tag{87}$$

Where $r(t)$ is the daily logarithmic return, μ is the constant drift rate, $\sigma(t)$ is the stochastic volatility, $\varepsilon(t)\sim N(0,1)$ is a standard normal white noise, $J(t)\sim N(\mu_J,\sigma_J)$ is a variable determining the jump sizes and $Q(t)\sim\text{Bern}[\lambda(t)]$ is a variable determining the times of jump occurrences. $h(t) = \ln[\sigma^2(t)]$ is the logarithm of the conditional variance, long-term log-variance θ is given by $\alpha = (1-\beta)\theta$, β is the autoregression coefficient, γ is the volatility of the log-variance, and $\varepsilon_V(t)\sim N(0,1)$ is the white noise in the log-variance equation, which can be correlated with $\varepsilon(t)$ with correlation ρ. $\lambda(t)$ is the jump intensity, long-term jump intensity θ_J is given by $\mathrm{\alpha_J} = \left(1-\mathrm{\beta_J}-\gamma_J\right)\theta_J$, parameter β_J gives the rate of exponential decay of the jump intensity to its long-term level, and γ_J is the self-exciting parameter telling us how much will the jump intensity increase in the day following a jump.

2.2. SVJD-RV-Z model with self-exciting jumps in prices

The SVJD-RV-Z model (proposed in Fičura and Witzany 2015, and Fičura and Witzany, 2017) represents an extension of the aforementioned SVJD model with self-exciting jumps in prices, that uses the realized variance, $RV(t)$, and the Z-Estimator of jumps, $Z(t)$, as additional sources of information. The model has the following 5 equations:

$$\begin{aligned}
&r(t) = \mu + \sigma(t)\varepsilon(t) + J(t)Q(t)\\
&h(t) = \alpha + \beta h(t-1) + \gamma\varepsilon_V(t)\\
&\lambda(t) = \alpha_J + \beta_J\lambda(t-1) + \gamma_J Q(t-1)\\
&\log[RV(t) - J^2(t)Q(t)] = h(t) + \sigma_{RV}\varepsilon_{\mathrm{RV}}(t)\\
&Z(t) = \mu_Z + \xi_Z Q(t) + \sigma_Z\varepsilon_Z(t)
\end{aligned} \tag{88}$$

Where $\varepsilon_{RV}(t) \sim N(0,1)$ and $\varepsilon_Z(t) \sim N(0,1)$. We can see that the realized variance adjusted for the influence of jumps, $\log[RV(t) - J^2(t)Q(t)]$, is assumed to provide unbiased estimates of the stochastic variance $h(t)$, that are plagued by a noise with volatility σ_{RV}. If we wanted to include bias into the model, an additional parameter μ_{RV} could be added to the right-hand side of the realized variance equation. The parameter μ_Z represents the mean value of $Z(t)$ in the days when no jumps or only small intraday jumps occur, while ξ_Z represents an increase of the mean in the case of a large jump, i.e. when $Q(t) = 1$. Finally, parameter σ_Z represents the volatility of the noise of the $Z(t)$ estimator around its mean.

2.3. SVJD-RV-Z model with self-exciting jumps in prices and volatility

A further extension of the SVJD-RV-Z model presented above is to add self-exciting jumps in volatility into the model. The SVJD-RV-Z model with self-exciting jumps in prices and volatility has the following 6 equations:

$$\begin{aligned}
&r(t) = \mu + \sigma(t)\varepsilon(t) + J(t)Q(t)\\
&h(t) = \alpha + \beta h(t-1) + \gamma\varepsilon_V(t) + J_V(t)Q_V(t)\\
&\lambda(t) = \alpha_J + \beta_J\lambda(t-1) + \gamma_J Q(t-1)\\
&\lambda_V(t) = \alpha_{JV} + \beta_{JV}\lambda_V(t-1) + \gamma_{JV} Q_V(t-1)\\
&\log[RV(t) - J^2(t)Q(t)] = h(t) + \sigma_{RV}\varepsilon_{\mathrm{RV}}(t)\\
&Z(t) = \mu_Z + \xi_Z Q(t) + \sigma_Z\varepsilon_Z(t)
\end{aligned} \tag{89}$$

Where $J_V(t) \sim N(\mu_{JV}, \sigma_{JV})$ determines the log-variance jump sizes and $Q_V(t) \sim \mathrm{Bern}[\lambda_V(t)]$ the log-variance jump occurrences. In the models that will be tested we will make an important simplifying assumption by assuming that the jumps in volatility and prices are mutually uncorrelated. The volatility jumps will again follow a discretized Hawkes process with intensity $\lambda_V(t)$ and parameters α_{JV}, β_{JV} and γ_{JV}.

2.4. 2-Component SVJD-RV-Z model with self-exciting jumps in prices and volatility

The most complex model analysed in this thesis will be the two-component SVJD-RV-Z model with self-exciting jumps in returns and volatility, in which the log-variance $h(t)$ is composed of a sum of two independent AR(1) processes with different levels of persistence. The model has the following 8 equations:

$$\begin{aligned}
&r(t) = \mu + \sigma(t)\varepsilon(t) + J(t)Q(t)\\
&\sigma(t) = \exp[h_{ST}(\mathrm{t}) + h_{LT}(\mathrm{t})]^{\frac{1}{2}}\\
&h_{ST}(t) = \beta h_{ST}(t-1) + \gamma\varepsilon_V(t) + J_V(t)Q_V(t)\\
&h_{LT}(t) = \phi_0 + \phi_1 h_{LT}(t-1) + \phi_2\varepsilon_{VL}(t)\\
&\lambda(t) = \alpha_J + \beta_J\lambda(t-1) + \gamma_J Q(t-1)\\
&\lambda_V(t) = \alpha_{JV} + \beta_{JV}\lambda_V(t-1) + \gamma_{JV}Q_V(t-1)\\
&\log[RV(t) - J^2(t)Q(t)] = h(t) + \sigma_{RV}\varepsilon_{\mathrm{RV}}(t)\\
&Z(t) = \mu_Z + \xi_Z Q(t) + \sigma_Z\varepsilon_Z(t)
\end{aligned} \tag{90}$$

Where $h_{ST}(\mathrm{t})$ and $h_{LT}(\mathrm{t})$ correspond to the two log-variance components, with the short-term component, $h_{ST}(\mathrm{t})$, containing self-exciting volatility jumps $J_V(t)$ and $Q(t)$, occurring with intensity $\lambda_V(t)$. Parameters β and γ represent the persistence and the volatility of the short-term log-variance component, while the parameters ϕ_0, ϕ_1 and ϕ_2 govern the autoregression behaviour of the long-term log-variance component.

3. Bayesian estimation of SVJD models

Bayesian estimation methods are especially suitable for the estimation of non-linear, non-Gaussian state-space models with latent state variables, such as the Stochastic-Volatility Jump-Diffusion models that will be used in the empirical part of this study. In the following section the Markov-Chain Monte-Carlo method and the Sequential Monte Carlo method (Particle Filters) will be explained and applied for SVJD model estimation.

3.1. Markov-Chain Monte-Carlo

The Markov-Chain Monte-Carlo (MCMC) method enables us to sample from the high-dimensional joint posterior density $p(\Theta|\text{data})$, where $\Theta = (\theta_1, \dots, \theta_k)$ denotes the vector of all model parameters and latent states, by constructing a Markov Chain, converging to the density $p(\Theta|\text{data})$, while using only the information about the conditional densities $p(\theta_j|\theta_i, i \neq j, \text{data})$. The idea of the method is then to construct a long-enough chain, so that its later iterations represent independent sample from the density $p(\Theta|\text{data})$, from which we can then estimate the parameters of the model based on posterior sample means (or modes), and the standard errors of the estimate as posterior standard deviations.

Multiple MCMC methods exist, differing for the cases when we are able to sample directly from the conditional densities $p(\theta_j|\theta_i, i \neq j, \text{data})$, and for the cases when even these densities are intractable and need to be approximated.

3.1.1. Gibbs Sampler

In the case when we can sample directly from the densities $p(\theta_j|\theta_i, i \neq j, \text{data})$, we can construct a Markov Chain, converging to the joint posterior $p(\Theta|\text{data})$, by using the *Gibbs Sampler* algorithm.

The Gibbs Sampler proceeds as follows:

0. Assign vector of initial values to $\Theta^0 = (\theta_1^0, \dots, \theta_k^0)$ and set $j = 0$
1. Set $j = j + 1$
2. Sample $\theta_1^j \sim p(\theta_1|\theta_2^{j-1}, \dots, \theta_k^{j-1}, \text{data})$
3. Sample $\theta_2^j \sim p\left(\theta_2\middle|\theta_1^j, \theta_3^{j-1}, \dots, \theta_k^{j-1}, \text{data}\right)$

   …
4. Sample $\theta_k^j \sim p(\theta_k|\theta_1^j, \theta_2^j, \dots, \theta_{k-1}^j, \text{data})$ and return to step 1.

As the univariate conditional densities $p(\theta_j|\theta_i, i \neq j, \text{data})$ fully characterize the joint posterior density $p(\Theta|\text{data})$, it can be proved, according to the Clifford-Hammersley theorem (Johannes and Polson, 2009), that the constructed Markov Chain converges to the joint posterior density $p(\Theta|\text{data})$ as its equilibrium density.

The conditional densities $p(\theta_j|\theta_i, i \neq j, \text{data})$ necessary for the Gibbs Sampler construction are typically derived by applying the Bayes theorem to the likelihood function and the prior density as follows:

$$p\left(\theta_1\middle|\theta_2^{j-1}, \dots, \theta_k^{j-1}, \text{data}\right) \propto L\left(\text{data}\middle|\theta_1, \theta_2^{j-1}, \dots, \theta_k^{j-1}\right) * \text{prior}\left(\theta_1\middle|\theta_2^{j-1}, \dots, \theta_k^{j-1}\right) \qquad (91)$$

With $L(.)$ denoting the likelihood function, $\text{prior}(.)$ the Bayesian prior density of the given parameter and $\propto$ the proportional relationship. If no prior information is available, the uninformative prior densities, $\text{prior}(\theta_i) \propto 1$, can be used for the prior.

In order to replace the proportional relationship in the equation with equality and to derive $p(\theta_j|\theta_i, i \neq j, \text{data})$, it is necessary to divide the right-hand side of the equation with the integral of the right-hand side over θ_1, corresponding to $p\left(\text{data}\middle|\theta_2^{j-1}, \dots, \theta_k^{j-1}\right)$.

Unfortunately, the integration of the right-hand side over θ_1 may often be infeasible, in which case we are left only with the shape of the density $p(\theta_j|\theta_i, i \neq j, \text{data})$, but do not know the normalizing constant, and thus cannot sample from the density directly.

In these cases, alternative sampling schemes have to be used, such as the Accept-Reject Gibbs Sampler or the Metropolis-Hastings algorithm.

3.1.2. Metropolis-Hastings algorithm

Metropolis-Hastings is a rejection sampling algorithm that can be used to sample from $p(\theta_j|\theta_i, i \neq j, \text{data})$ in the case when we know the shape of it, but cannot compute the normalizing constant. The algorithm works by sampling from a proposal density $q(\theta_1|\theta_2^{j-1}, \dots, \theta_k^{j-1}, \text{data})$ instead and then accepting or rejecting the proposal based on a given probability α.

Specifically, to utilize the Metropolis-Hastings algorithm, Step 2 in the Gibbs Sampler is replaced by the following two-step procedure:

A. Sample θ_1^j from the proposal density $q(\theta_1|\theta_2^{j-1},\dots,\theta_k^{j-1},\text{data})$

B. Sample θ_1^j with probability $\alpha = \min(R,1)$, where R denotes the so-called acceptance ratio:

$$R = \frac{p(\theta_1^j|\theta_2^{j-1},\dots,\theta_k^{j-1},\text{data})q(\theta_1^{j-1}|\theta_1^j,\theta_2^{j-1},\dots,\theta_k^{j-1},\text{data})}{p(\theta_1^{j-1}|\theta_2^{j-1},\dots,\theta_k^{j-1},\text{data})q(\theta_1^j|\theta_1^{j-1},\theta_2^{j-1},\dots,\theta_k^{j-1},\text{data})} \tag{92}$$

Which may in practice be evaluated by sampling $u{\sim}U(0,1)$ from the uniform distribution and then accepting the proposal θ_1^j only if $u < R$, while otherwise the value of the parameter θ_1^{j-1} is kept instead.

As in the case of the Gibbs Sampler algorithm, it can be shown that the so constructed Markov Chain converges to the joint posterior density $p(\Theta|\text{data})$ as its equilibrium density (Johannes and Polson, 2009).

3.1.3. Random-Walk Metropolis Hastings

Multiple versions of the Metropolis-Hastings algorithm exist, differing in the way of how the proposal density q is constructed. The simplest and most universal version is the *Random-Walk Metropolis-Hastings* algorithm, in which the proposal density follows a Random Walk through the parameter space. The proposal q is thus defined as:

$$\theta_1^j \sim \theta_1^{j-1} + N(0,c) \tag{93}$$

With c being the step-size meta-parameter which may significantly influence the computational efficiency of the algorithm and the practice is to set it so that approximately 50% of the proposals get accepted and 50% rejected.

A convenient property of the Random-Walk Metropolis-Hastings algorithm is that its proposal density is symmetric, so that the probability of moving from θ_1^{j-1} to θ_1^j is the same as the probability of sampling from θ_1^j to θ_1^{j-1}. This causes the terms corresponding to q in the acceptance ratio to cancel out, causing the acceptance ratio to reduce to the likelihood ratio:

$$R = \frac{L(\text{data}|\theta_1^j,\theta_2^{j-1},\dots,\theta_k^{j-1})}{L(\text{data}|\theta_1^{j-1},\theta_2^{j-1},\dots,\theta_k^{j-1})}$$

(94)

As long as we are able to calculate the likelihood of the model for all θ, we can then use the Random-Walk Metropolis-Hastings algorithm to sample from the joint posterior density of the parameters and the latent states.

3.2. MCMC estimation of SVJD models

In the MCMC estimation algorithm for the SVJD-RV-Z model with self-exciting price jumps we have to estimate 13 model parameters $(\mu, \mu_J, \sigma_J, \alpha, \beta, \gamma, \alpha_J, \beta_J, \gamma_J, \sigma_{RV}, \mu_Z, \xi_Z, \sigma_Z)$ and 3 vectors of latent state variables $(\boldsymbol{V}, \boldsymbol{J}, \boldsymbol{Q})$. The algorithm was developed in Fičura and Witzany (2015) and is based on earlier results from Witzany (2013) and is based on the methodology developed in Jacquier et al. (2007) and Johannes and Polson (2009). The MCMC algorithms for SVJD and SVJD-RV proceed in the same fashion, with slight modifications and some of the steps missing.

The algorithm for SVJD-RV-Z model proceeds as follows:

1. Sample initial values of the model latent state variables $\boldsymbol{V}^{(0)}, \boldsymbol{J}^{(0)}, \boldsymbol{Q}^{(0)}$ and parameters $\mu^{(0)}, \mu_J^{(0)}, \sigma_J^{(0)}, \alpha^{(0)}, \beta^{(0)}, \gamma^{(0)}, \alpha_J^{(0)}, \beta_J^{(0)}, \gamma_J^{(0)}, \sigma_{RV}^{(0)}, \mu_Z^{(0)}, \xi_Z^{(0)}, \sigma_Z^{(0)}$.
2. For $i = 1, \dots, T$ sample the jump sizes $J_i^{(g)} \propto \varphi\left(J; \mu_J^{(g-1)}, \sigma_J^{(g-1)}\right)$ if $Q_i^{(g)} = 0$ using the Gibbs Sampler, or if $Q_i^{(g-1)} = 1$, use the Random-Walk Metropolis-Hastings to sample from:
$$J_i^{(g)} \propto \varphi\left(r_i; \mu^{(g-1)} + J, \sqrt{V_i^{(g-1)}}\right) \varphi\left(log(RV_i - J^2); h_i^{(g-1)}, \sigma_{RV}^{(g-1)}\right) \varphi\left(J; \mu_J^{(g-1)}, \sigma_J^{(g-1)}\right)$$
3. For $i = 1, \dots, T$ sample the jump occurrences $Q_i^{(g)} \in \{0,1\}$, using the expression $\Pr[Q = 1] = p_1/(p_0 + p_1)$, where:
$$p_0 = \varphi\left(r_i; \mu^{(g-1)}, \sqrt{V_i^{(g-1)}}\right) \varphi\left(\log(RV_i); h_i^{(g-1)}, \sigma_{RV}^{(g-1)}\right) \varphi\left(Z_i; \mu_Z^{(g-1)}, \sigma_Z^{(g-1)}\right)\left(1 - \lambda_i^{(g-1)}\right)$$
$$p_1 = \varphi\left(r_i; \mu^{(g-1)} + J_i^{(g)}, \sqrt{V_i^{(g-1)}}\right) \varphi\left(\log\left(RV_i - \left(J_i^{(g)}\right)^2\right); h_i^{(g-1)}, \sigma_{RV}^{(g-1)}\right) \varphi\left(Z_i; \mu_Z^{(g-1)} + \xi_Z^{(g-1)}, \sigma_Z^{(g-1)}\right) \lambda_i^{(g-1)}$$
4. Sample new stochastic log-variances $h_i^{(g)} = \log\left(V_i^{(g)}\right)$ for $i = 1, \dots, T$ using the Gibbs Sampler with accept-reject procedure developed by Kim, Shephard and Chib (1998), i.e. first calculate the series $y_i = r_i - \mu^{(g-1)} - J_i^{(g)} Q_i^{(g)}$ and then sample $h_i^{(g)}$ from the proposal distribution $\varphi(h_i; \mu_i, \sigma)$, where:
$$\mu_i = \phi_i + \frac{\sigma^2}{2}[y_i^2 \exp(-\phi_i) - 1],$$
$$\phi_i = \frac{\gamma^2 \log(RV_i - J_i^2 Q_i) + \sigma_{RV}^2[\alpha(1-\beta) + \beta(\log V_{i+1} + \log V_{i-1})]}{\gamma^2 + (1+\beta^2)\sigma_{RV}^2},$$

$$\sigma = \frac{\gamma\sigma_{RV}}{\sqrt{\gamma^2 + (1+\beta^2)\sigma_{RV}^2}}$$

The proposal is accepted with probability f^*/g^*, where:

$$\log(f^*) = -\frac{h_i}{2} - \frac{{y_i}^2}{2}[\exp(-h_i)]$$

$$\log(g^*) = -\frac{h_i}{2} - \frac{{y_i}^2}{2}[\exp(-\phi_i)\,(1+\phi_i) - h_i \exp(-\phi_i)]$$

If no accepted, then another proposal is drawn until acceptance occurs.

5. Sample new stochastic log-variance autoregression coefficients $\alpha^{(g)}, \beta^{(g)}, \gamma^{(g)}$, denoting $h_i = \log\left(V_i^{(g)}\right)$ for $i = 1, \dots, T$, using the Bayesian linear regression model (Lynch, 2007), i.e. define $\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}\boldsymbol{y}$ and $\widehat{\boldsymbol{e}} = \boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}$, where $\boldsymbol{X} = \begin{pmatrix} 1 & \dots & 1 \\ h_1 & \dots & h_{T-1} \end{pmatrix}'$ and $\boldsymbol{y} = (h_2 \dots h_T)'$, and sample:

$$\left(\gamma^{(g)}\right)^2 \propto IG\left(\frac{n-2}{2}, \frac{\widehat{\boldsymbol{e}}'\widehat{\boldsymbol{e}}}{2}\right),$$

$$\left(\alpha^{(g)}, \beta^{(g)}\right)' \propto \varphi\left[(\alpha, \beta)'; \widehat{\boldsymbol{\beta}}, \left(\gamma^{(g)}\right)^2 (\boldsymbol{X}'\boldsymbol{X})^{-1}\right]$$

6. Sample $\mu^{(g)}$ based on the normally distributed time series $r_i - J_i^{(g)} Q_i^{(g)}$ with variances $V_i^{(g)}$ as follows:

$$p\left(\mu^{(g)} | \boldsymbol{r}, \boldsymbol{J}^{(g)}, \boldsymbol{Q}^{(g)}, \boldsymbol{V}^{(g)}\right) \propto \varphi\left(\mu; \sum_{i=1}^{T} \frac{r_i - J_i^{(g)} Q_i^{(g)}}{V_i^{(g)}} \Bigg/ \sum_{i=1}^{T} \frac{1}{V_i^{(g)}}, \sum_{i=1}^{T} \frac{1}{V_i^{(g)}}\right)$$

7. Sample the Hawkes process parameters $\theta_J, \beta_J, \gamma_J$, using the Random-Walk Metropolis-Hastings algorithm with the proposal densities given as:

$$\theta_J^{(g)} = \theta_J^{(g-1)} + N(0, c),$$

$$\beta_J^{(g)} = \beta_J^{(g-1)} + N(0, c),$$

$$\gamma_J^{(g)} = \gamma_J^{(g-1)} + N(0, c),$$

and the likelihood function equal to $L\left(\boldsymbol{Q}^{(g)} | \theta_J, \beta_J, \gamma_J\right) = \prod_{i=1}^{T} \lambda_i^{Q_i} (1-\lambda_i)^{1-Q_i}$.

8. Sample $\mu_J^{(g)}, \sigma_J^{(g)}$ based on the normally distributed time series $\boldsymbol{J}^{(g)}$ and uninformative priors $p(\mu) \propto 1$ and $p(\log\sigma^2) \propto 1$, equivalent to $p(\sigma^2) \propto 1/\sigma^2$:

$$p\left(\mu_J^{(g)} | \boldsymbol{J}^{(g)}, \sigma_J^{(g-1)}\right) \propto \varphi\left(\mu_J^{(g)}; \frac{\sum_{i=1}^{T} J_i^{(g)}}{T}, \frac{\sigma_J^{(g-1)}}{\sqrt{T}}\right)$$

$$p\left[\left(\sigma_J^{(g)}\right)^2 | \boldsymbol{J}^{(g)}, \mu_J^{(g)}\right] \propto IG\left[\left(\sigma_J^{(g)}\right)^2; \frac{T}{2}, \frac{\sum_{i=1}^{T}\left(J_i^{(g)} - \mu_J^{(g)}\right)^2}{2}\right]$$

9. Sample $\sigma_{RV}^{(g)}$ using the Inverse Gamma density:

$$p\left[\left(\sigma_{RV}^{(g)}\right)^2 | \boldsymbol{RV}, \boldsymbol{J}^{(g)}, \boldsymbol{Q}^{(g)}, \boldsymbol{V}^{(g)}\right] \propto IG\left[\left(\sigma_{RV}^{(g)}\right)^2; \frac{T}{2}, \frac{\sum_{i=1}^{T}\left(\log\left(RV_i - \left(J_i^{(g)}\right)^2 Q_i^{(g)}\right) - h_i^{(g)}\right)^2}{2}\right]$$

10. Sample $\mu_Z^{(g)}, \xi_Z^{(g)}, \sigma_Z^{(g)}$ using the normally distributed series $Z_i - Q_i^{(g)}\xi_Z^{(g-1)}$, with variance $\sigma_Z^{(g-1)}$ to sample $\mu_Z^{(g)}$, series $Q_i^{(g)}\left(Z_i - \mu_Z^{(g)}\right)$, at points where $Q_i^{(g)} = 1$, with variance $\sigma_Z^{(g-1)}$, to sample $\xi_Z^{(g)}$, and the centralized time series $Z_i - \mu_Z^{(g)} - Q_i^{(g)}\xi_Z^{(g)}$ to sample $\sigma_Z^{(g)}$ using the Inverse Gamma distribution. Specifically, the sampling densities will be:

$$p\left[\mu_Z^{(g)}|\boldsymbol{Z},\boldsymbol{Q}^{(g)},\xi_Z^{(g-1)},\sigma_Z^{(g-1)}\right] \propto \varphi\left(\mu_Z^{(g)};\frac{\sum_{i=1}^{T}\left(Z_i - Q_i^{(g)}\xi_Z^{(g-1)}\right)}{T},\frac{\sigma_Z^{(g-1)}}{\sqrt{T}}\right)$$

$$p\left[\xi_Z^{(g)}|\boldsymbol{Z},\boldsymbol{Q}^{(g)},\mu_Z^{(g)},\sigma_Z^{(g-1)}\right] \propto \varphi\left(\xi_Z^{(g)};\frac{\sum_{i=1}^{T}Q_i^{(g)}\left(Z_i - \mu_Z^{(g)}\right)}{\sum_{i=1}^{T}Q_i^{(g)}},\frac{\sigma_Z^{(g-1)}}{\sqrt{\sum_{i=1}^{T}Q_i^{(g)}}}\right)$$

$$p\left[\left(\sigma_Z^{(g)}\right)^2|\boldsymbol{Z},\boldsymbol{Q}^{(g)},\mu_Z^{(g)},\xi_Z^{(g)}\right] \propto IG\left[\left(\sigma_Z^{(g)}\right)^2;\frac{T}{2},\frac{\sum_{i=1}^{T}\left(Z_i - \mu_Z^{(g)} - Q_i^{(g)}\xi_Z^{(g)}\right)^2}{2}\right]$$

3.3. Particle Filters

While the MCMC method represents a powerful tool for the estimation of the parameters and latent states of SVJD models, it is problematic to use it for predictions, and especially for back-testing. The problem stems from the fact that MCMC estimates the so called smoothing distribution of the latent states, $p(x_t|\mathcal{F}_T,\Theta)$, contitional on the whole information set $\mathcal{F}_T$ up to time T. The algorithm thus looks into the future when estimating the densities of individual latent states x_t for $t < T$, and in order to get the filtering densities $p(x_t|\mathcal{F}_t,\Theta)$, conditional only on the available information $\mathcal{F}_t$, we would need to re-run the MCMC for each time point in the time series. As the re-running of the MCMC algorithm would be very time-consuming, the most commonly used approach is to combine MCMC with Particle Filters.

Particle Filters (first proposed by Gordon et al., 1993) represent a method that sequentially estimates the filtering densities $p(x_t|\mathcal{F}_t,\Theta)$, conditional on known parameters Θ and the information set $\mathcal{F}_t$ for each t. For this purpose, the distribution $p(x_t|\mathcal{F}_t,\Theta)$ is represented with a weighted set of particles (i.e. discrete points), that are sequentially updated with Bayesian update equations as new information arrives (see Doucet and Johannses, 2009, or Speekenbrink, 2016).

As the basic particle filters usually assume a known parameter vector Θ, the common approach in SVJD literature is to first use the MCMC algorithm to estimate the model parameters and latent states over the in-sample period, while the Particle Filter is then used to compute the filtering distributions of the latent states $p(x_t|\mathcal{F}_t, \Theta)$ for all t over the out-sample period. These can subsequently be used to perform forecasts of $p(x_{t+h}|\mathcal{F}_t, \Theta)$, by sampling the latent states x_t from $p(x_t|\mathcal{F}_t, \Theta)$ and simulating their evolution into the future.

In recent years, methods have been proposed that enable the Particle Filters to sequentially estimate both, the model parameters Θ, as well as the latent states x_t, for each time-point in the time series, conditional only on the available information $\mathcal{F}_t$. The methods of this kind have become known as *particle learning*. For major developments in the field, see Liu and West (2001), Gilks and Berzuini (2001), Storvik (2002), Fearnhead (2002), Carvalho et al. (2010), Fulop and Li (2013), Nemeth et al. (2013) and Fičura and Witzany (2018).

For application of particle learning methods on SVJD models see Fulop and Li (2014), Chronoupolou (2018) and Fičura and Witzany (2018).

In the following sections, the use of particle filters for latent state filtering and parameter learning will be explained, focused especially on the SIR Particle Filter of Gordon (1993), the Marginalized Re-Sample Move approach of Fulop and Li (2013), and the Sequential Gibbs Particle Filter of Fičura and Witzany (2018). The developed methods will subsequently be used for SVJD model estimation.

3.3.1. Filtering vs. Smoothing problem

Assume we have an observable time series y_t, for $t = 1, \ldots, T$, whose dynamics depends on an unobservable time series x_t, via set of equations with known parameters θ.

MCMC solves the *smoothing problem* of estimating the posterior densities $p(x_t|y_1, \ldots, y_T, \theta)$, for all t. Take note that the conditional distribution is expressed with respect to all y_t, up to the $t = T$, and the model is thus looking into the future when estimating the densities of all x_t, except for the last x_T.

Particle Filters, on the other hand, solve the *filtering problem* of estimating the posterior densities $p(x_t|y_1, \ldots, y_t, \theta)$ for all t. We thus express the posterior density of each x_t, conditional only on the observations y_t, available at the given time t.

This then enables us to perform forecasts of $p(x_{t+h}|y_1, \ldots, y_t, \theta)$, by simulating the future evolution of $x_{t+1}, \ldots x_{t+h}$ based on the inferred distribution of x_t and the known parameters of the model θ.

As a result, Particle Filters are far more convenient for out-of-sample volatility forecasting than the MCMC method, which would need to be re-estimated for each time t in order to be able to generate out-of-sample forecasts of $p(x_{t+h}|y_1, \dots, y_t, \theta)$ via simulations.

3.3.2. General State-Space model

A general state-space model contains the following two equations.

Observation equation:

$$y_t = H(x_t, w_t, \theta) \tag{95}$$

And transition equation:

$$x_t = F(x_{t-1}, v_t, \theta) \tag{96}$$

Where $H(.)$ and $F(.)$ are arbitrary functions, θ are the parameters, and w_t and v_t are the noise terms, assumed to be mutually independent.

The observations y_t are assumed to be conditionally independent given the hidden states x_t, with the observation density $p(y_t|x_t, \theta)$. The hidden state x_t is commonly assumed to follow a Markov proces, with initial density $p(x_0|\theta)$ and a transition density $p(y_t|x_t, \theta)$.

The task of the state filtering and parameter learning is to find the joint posterior distribution of the latent states and the parameters conditional on the observations until t:

$$p(x_t, \theta|y_{1:t}) = p(x_t|y_{1:t}, \theta)p(\theta|y_{1:t}) \tag{97}$$

Where $p(x_t|y_{1:t}, \theta)$ solves the state filtering problem and $p(\theta|y_{1:t})$ the parameter learning problem.

In the cases when the functions $H(.)$ and $F(.)$ are linear and the distributions of x_0, w_t and v_t are Gaussian, the problem can be solved analytically with a Kalman Filter (Kalman, 1960).

In the general, non-linear and non-Gaussian case, particle filters can be used, approximating the joint posterior density $p(x_t, \theta|y_{1:t})$ with a weighted set of particles.

3.3.3. SIR Particle Filter

In the first step, we will assume that the parameters θ are known, with the goal being to solve the filtering problem of estimating $p(x_t|y_{1:t},\theta)$. This can be tackled with the Sequential Importance Re-Sampling (SIR) Particle Filter developed by Gordon et al. (1993).

Given a set of M particles $\left\{x_{t-1}^{(i)}; i = 1,2,\dots,M\right\}$, with weights $\widetilde{w}_{t-1}^{(i)}$ representing the density $p(x_{t-1}|y_{1:t-1})$ at time $t-1$, we can use the recursion:

$$p(x_t|y_{1:t}) \propto \int p(y_t|x_t)\, p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \tag{98}$$

To approximate $p(x_t|y_{1:t})$, we draw from a known proposal density $\mathrm{g}(x_t|x_{t-1},y_t)$ and assign importance weights to the sample as follows:

$$w_t^{(i)} = \frac{p(y_t|x_t^i)p(x_t^i|x_{t-1}^i\ )}{g(x_t^i|x_{t-1}^i,y_t)}\widetilde{w}_{t-1}^{(i)} \tag{99}$$

For the case of a simple unadapted filter, the proposal density equals the transition density, i.e. $g(x_t^i|x_{t-1}^i,y_t) = p(x_t^i|x_{t-1}^i\ )$, and the weight adjustment thus reduces to:

$$w_t^{(i)} = p(y_t|x_t^i)\widetilde{w}_{t-1}^{(i)} \tag{100}$$

The weights are then normalized according to:

$$\widetilde{w}_t^{(i)} = w_t^{(i)} \Big/ \sum_{j=1}^{M} w_t^{(i)} \tag{101}$$

And the population is re-sampled, if $ESS = 1/\sum_{j=1}^{M}\left(\widetilde{w}_t^{(i)}\right)^2 < ESS_{Thr}$, where ESS_{Thr} is the re-sampling treshold to prevent sample degeneration. If re-sampling is performed, all weights are subsequently set to $\widetilde{w}_t^{(i)} = 1/M$.

We thus arrive at a sample of M particles $\left\{x_t^{(i)}; i = 1,2,\dots,M\right\}$, with weights $\widetilde{w}_t^{(i)}$, representing the density $p(x_t|y_{1:t})$ at time t, which can be used in the next iteration step.

The weighted sample can further be used to estimate the expectation of any function $f(x_t)$, by applying the relationship:

$$\int_{-\infty}^{\infty} f(x_t)p(x_t|y_{1:t})dx_t \approx \sum_{i=1}^{M} f\left(x_t^{(i)}\right)\widetilde{w}_t^{(i)} \tag{102}$$

Additionally, likelihood of the observations conditional on the parameter vector θ can be estimated as:

$$\hat{p}(y_{1:t}|\theta) = \prod_{l=2}^{t} \hat{p}(y_l|y_{1:l-1},\theta)\hat{p}(y_1|\theta) \tag{103}$$

Where:

$$\hat{p}(y_l|y_{1:l-1},\theta) = \sum_{i=1}^{M} w_l^{(i)} \tag{104}$$

According to Del Moral (2004), the estimate is unbiased, so that $E[\hat{p}(y_{1:t}|\theta)] = p(y_{1:t}|\theta)$, which will be crucial in the parameter learning phase of estimating $p(\theta|y_{1:t})$ in the MSM approach of Fulop and Li (2013).

3.3.4. Parameter estimation vs. Parameter learning

Analogically to the smoothing vs. filtering problem, regarding the latent state x_t, we can define the parameter estimation vs. learning problem with regards to the estimation of the parameter vector θ. The MCMC method performs *parameter estimation* of the joint posterior density $p(\boldsymbol{x},\theta|y_1,\ldots,y_T)$, conditional on all observations $y_t,\ldots,y_T$. The goal of the *parameter learning* problem, on the other hand, is to sequentially estimate $p(x_t,\theta|y_1,\ldots,y_t)$ for all t. For discussion see Doucet and Johannses, 2009, or Speekenbrink, 2016.

The parameter learning problem can, similarly to the latent state filtering, be tackled with particle filters (thus it is also known as *particle learning*, see Carvalho et al. 2010). The problem is that when the time-constant parameters are viewed in a same way as the latent states, their distribution will quickly degenerate to a situation where only one parameter combination has a non-zero weight, while all others are very close to zero. This degeneracy is far worse than in the case of the latent states, which change through the time, and thus even if the particle filter degenerates to a single particle with non-zero weight, the latent states in the next period (after particle re-sampling) will be sampled at different values (through the model transition equations) (see Doucet and Johannses, 2009, or Speekenbrink, 2016).

To tackle the problem, early approaches proposed to perturb the parameter values corresponding to the particles with artificial noise (Gordon et al., 1993). While this may be efficient to prevent parameter degeneration, the artificial noise leads to overly diffuse parameter distributions. Liu and West (2001) therefore propose to use kernel density estimation of the parameter distribution, together with shrinkage to alleviate the problem. Alternatively, particle degeneracy can be tackled by adding an MCMC step to the particle filter (Gilks and Berzuini, 2001, Storvik 2002, Fearnhead 2002, Lopes et al., 2010). Unfortunately, as shown by Chopin et al. (2010), the parameter distribution will still degenerate, regardless of the previous proposed methods, if the time-series is sufficiently long and the latent state particles filter is not re-run on the whole history.

As a result, Fulop and Li (2013) propose the Marginalized Re-Sample Move (MSM) approach, which is able to estimate the posterior parameter distributions consistently, but unfortunately contains a step during which the latent state particle filter needs to be re-run (often repeatedly) over the whole history of observations (i.e. the algorithm is not completely online). Especially for complex models with large number of parameters, the latent states particle filter also needs to be re-run multiple times during each parameter re-sampling step, which makes the algorithm very time consuming. As a result, Ficura and Witzany (2018) propose a new Sequential Gibbs Particle Filter, combining the results from Fulop and Li (2013) and Gilks and Berzuini (2001), which seems to work better for complex SVJD models and does not require the re-running of the latent states filter to approximate the parameter posterior distributions. The Marginalized Re-Sample Move (MSM) approach (Fulop and Li, 2013), and the Sequential Gibbs Particle Filter (SGPF) (Fičura and Witzany, 2018), are explained in the following sections. As most of the steps are identical for the two methods, we start by explaining the MSM first, with the SGPF being presented subsequently as a replacement of the final, *Move* step, of the algorithm.

3.3.5. Marginalized Re-Sample Move approach

The goal of the Marginalized Re-Sample Move (MSM) approach (Fulop and Li, 2013) is to extend the particle filter to the parameter learning problem of estimating $p(\theta|y_{1:t})$.

The logic of the approach is to first marginalize out the latent states from $p(\theta, x_t|y_{1:t})$, followed by the application of the resample-move algorithm of Gilks and Berzuini (2001) and

Chopin (2002), while using the results from Andrieu and Roberts (2009) and Andrieu et al. (2010), and using the relationship that:

$$\hat{p}(\theta|y_{1:t}) \propto \prod_{l=2}^{t} \hat{p}(y_l|y_{1:l-1},\theta)\hat{p}(y_1|\theta)\, p(\theta) \tag{105}$$

We start by defining an auxiliary state space including all random quantities produced by the particle filter $u_l = \left\{x_l^{(i)}, a_l^{(i)}; i = 1, \dots, M\right\}$ for step l, where $a_l^{(i)}$ denotes the indices sampled during the re-sampling.

At time t the filter depends only on the population of the state particles from step $t-1$, so we can write:

$$\psi(u_{1:t}|y_{1:t},\theta) = \prod_{l=2}^{t} \psi(u_l|u_{l-1},y_l,\theta)\, \psi(u_1|y_1,\theta) \tag{106}$$

Where $\psi(u_{1:t}|y_{1:t},\theta)$ denotes the joint density of all random variabls produced by the particle filter up to the time t.

Furthermore, the predictive likelihood can be written as

$$\hat{p}(y_t|y_{1:t-1},\theta) \equiv \hat{p}(y_t|u_t,u_{t-1},\theta) \tag{107}$$

We can then construct an auxiliary density:

$$\tilde{p}(\theta,u_{1:t}|y_{1:t}) \propto p(\theta) \prod_{l=2}^{t} \hat{p}(y_l|u_l,u_{l-1},\theta)\psi(u_l|u_{l-1},y_l,\theta)\hat{p}(y_1|u_1,\theta)\psi(u_1|y_1,\theta) \tag{108}$$

With the target density $p(\theta|y_{1:t})$ being the marginal of the auxiliary density that also has the same normalizing constant. Therefore, if we draw samples from the auxiliary density, we automatically obtain samples from the original target density $p(\theta|y_{1:t})$.

Assume we have a set of weighted particles $\left\{\left(\theta^{(n)}, u_{t-1}^{(n)}, \hat{p}(y_{1:t-1}|\theta)^n\right), s_{t-1}^{(n)}; n = 1, \dots, N\right\}$, representing the target distribution $\tilde{p}(\theta, u_{1:t-1}|y_{1:t-1})$ at time $t-1$. Where $s_{t-1}^{(n)}$ denotes the sample weights.

For each n, the relevant part of $u_{t-1}^{(n)}$ are the M particles representing the hidden states $\left\{x_t^{(i,n)}; i = 1, \dots, M\right\}$. In total we thus have to maintain MxN particles of the hidden states.

To approximate the target density $\tilde{p}(\theta, u_{1:t}|y_{1:t})$ at time t, we can use a recursive relationship between the target distributions at time $t-1$ and time t:

$$\tilde{p}(\theta, u_{1:t}|y_{1:t}) \propto \hat{p}(y_t|u_t, u_{t-1}, \theta)\psi(u_t|u_{t-1}, y_t, \theta)\tilde{p}(\theta, u_{1:t-1}|y_{1:t-1}) \tag{109}$$

The Marginalized Re-Sample Move (MSM) algorithm will run as follows:

1. *Augmentation step*: For each $\theta^{(n)}$, run the SIR particle filtering algorithm on the new observation y_t. This is equivalent to sampling from $\psi(u_t|u_{t-1}, y_t, \theta)$.
2. *Reweighting step*: The incremental weights are equal to $\hat{p}(y_t|u_t, u_{t-1}, \theta)$, leading to new weights, given by:

$$s_t^{(n)} = s_{t-1}^{(n)}\hat{p}\left(y_t|u_t^{(n)}, u_{t-1}^{(n)}, \theta^{(n)}\right) \tag{110}$$

And the estimated likelihood of the fixed parameters is updated by:

$$\hat{p}(y_{1:t}|\theta)^{(n)} = \hat{p}(y_{1:t-1}|\theta)^{(n)}\hat{p}\left(y_t|u_t^{(n)}, u_{t-1}^{(n)}, \theta^{(n)}\right) \tag{111}$$

The weighted sample $\left\{\left(\theta^{(n)}, u_t^{(n)}, \hat{p}(y_{1:t}|\theta)^n\right), s_{t-1}^{(n)}; n = 1, \dots, N\right\}$ is then distributed according to the target $\tilde{p}(\theta, u_{1:t}|y_{1:t})$.

The normalized weights are then computed as $\pi_t^{(n)} = s_t^{(n)}/\sum_{k=1}^N s_t^{(k)}$ and the effective sample size as $ESS_t = 1/\sum_{k=1}^N\left(\pi_t^{(k)}\right)^2$.

The marginal likelihood of y_t can be approximated as:

$$p(y_t|y_{1:t-1}) \approx \sum_{k=1}^{N} \pi_{t-1}^{(k)}\hat{p}\left(y_t|u_t^{(k)}, u_{t-1}^{(k)}, \theta^{(k)}\right) \tag{112}$$

3. *Resample-Move step*: In order to avoid degeneration of the parameter particle set, we perform the resample-move step if $ESS_t < B_1$. During this step, the particles are:
   (1) *Resampled* proportional to $\pi_t^{(n)}$, to get an equally weighted sample
   (2) *Moved* through a Markov kernel with stationary distribution $\tilde{p}(\theta, u_{1:t}|y_{1:t})$, while the number of unique particles is $<B_2$

   For the *Move* step, the marginal particle MCMC kernel with the following proposal can be used:

$$h(\theta, u_{1:t}|\theta') = h_t(\theta|\theta')\psi(u_{1:t}|\theta) \tag{113}$$

Where $h_t(\theta|\theta')$ is the proposal, adapted to the past of the algorithm, for example the multivariate normal distribution with mean and covariance fitted to the sample posterior covariance of θ.

The acceptance probability of the proposal $(\theta^*, u_{1:t}^*, \hat{p}(y_{1:t}|\theta)^*)$ is then computed as:

$$min\left\{1; \frac{p(\theta^*)\hat{p}(y_{1:t}|\theta)^* h_t\left(\theta^{(n)}|\theta^*\right)}{p(\theta^{(n)})\hat{p}(y_{1:t}|\theta)^{(n)} h_t(\theta^*|\theta^{(n)})}\right\} \tag{114}$$

A joint sample from $p(\theta, x_t|y_{1:t})$ can be obtained from the algorithm by drawing one particle of the hidden states for each randomly drawn parameter particle $\theta^{(n)}$ at any time t.

Alternatively, the full particle population can be used to approximate any expectation $E[f(\theta, x_t)|y_{1:t}]$ as:

$$E[f(\theta, x_t)|y_{1:t}] \approx \sum_{n=1}^{N}\sum_{i=1}^{M} \pi_t^{(n)} \widetilde{w}_t^{(i)} f\left(\theta^{(n)}, x_t^{(i,n)}\right) \tag{115}$$

A sequential Bayes factor can also be constructed for sequential model comparison. For any two models M_1 and M_2, the Bayes factor at time t can be computed with the following recursive formula:

$$BF_t \equiv \frac{p(y_{1:t}|M_1)}{p(y_{1:t}|M_2)} = \frac{p(y_t|y_{1:t-1}, M_1)}{p(y_t|y_{1:t-1}, M_2)} BF_{t-1}$$

(116)

3.3.6. Sequential Gibbs Particle Filter

While the use of multivariate Gaussian distribution, fitted to the mean and covariance of θ, as proposed by Fulop and Li (2013), provides a natural proposal distribution to sample the values of the parameters from, in the more complex SVJD models, with strong dependencies between the parameters, it can often happen that the vast majority of proposals end up being rejected. The result is, that the computationally highly demanding re-sampling of the parameter particles (which involves the re-running of the latent state particle filters in order to be able to compute the acceptance ratio), needs to be repeated many times, to achieve reasonable acceptance rates and to replenish the parameter particle set.

In order avoid this problem, Ficura and Witzany (2018) propose to rather use Gibbs Sampler (or alternatively the Random-Walk Metropolis-Hastings, in the cases when Gibbs Sampler cannot be used), to re-sample the parameter particles. The difference in comparison to the Fulop and Li (2013) algorithm is only in the *Move* phase, which is performed with the Gibbs Sampler on a randomly selected particle path. This can be re-constructed, even in the presence of re-sampling of the latent states, as long as the particle values and the indices of the particles used in each re-sampling step are saved. Andrieu et al. (2010), who developed a related method called Particle MCMC, call it the ancestral lineage. The idea is to know for each particle j at time $s = 1, \dots, t$, the index $a(j, s)$ of the particle from time $s - 1$ from which particle j originated (i.e. the particle that was used in the transition density during the sampling of particle j at time s), before re-sampling at was applied.

3.3.7. SGPF sampling in a SVJD model with volatility jumps

The SGPF algorithm of Fičura and Witzany (2018) proceeds in the same way as the MSM algorithm of Fulop and Li (2013), up until the *Move* step, which is in the case of the SGPF algorithm performed with the Gibbs sampler (or a Random-Walk Metropolis-Hastings for the parameters where the conditional density is intractable). Take note that in order for the algorithm to be consistent, the re-sampling of the latent state particles needs to be performed daily, i.e. the re-sampling threshold $ESS_{Thr} = M$, where M is the number of latent state particles used in the particle filter.

In the case of a single component SVJD model with self-exciting jumps in returns and volatility, the SGPF sampling in the *Move* step proceeds as follows:

1. For each parameter particle $g = 1, \dots, N$ sample a single latent state particle from $j = 1, \dots, M$ and reconstruct the indices of its ancestral lineage $a(j, s)$ for $s = 1, \dots, t$. To avoid overly complex index notation, we will assume in the rest of the algorithm that the hidden states denoted with index (g) for parameter particle g, correspond to the ancestral lineage $a(j, s)$ of the sampled latent state particle j for $s = 1, \dots, t$.
2. Sample new stochastic log-variance autoregression coefficients $\alpha^{(g)}, \beta^{(g)}, \gamma^{(g)}$, by using the Bayesian linear regression model (Lynch, 2007), i.e. define as $\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}\boldsymbol{y}$ and $\hat{\boldsymbol{e}} = \boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}$, where $\boldsymbol{X} = \begin{pmatrix} 1 & \dots & 1 \\ h_1^{(g)} & \dots & h_{t-1}^{(g)} \end{pmatrix}'$ and $\boldsymbol{y} = \left(h_2^{(g)} \dots h_t^{(g)}\right)'$, and sample:
$$\left(\gamma^{(g)}\right)^2 \propto IG\left(\frac{n-2}{2}, \frac{\hat{\boldsymbol{e}}'\hat{\boldsymbol{e}}}{2}\right),$$
$$\left(\alpha^{(g)}, \beta^{(g)}\right)' \propto \varphi\left[(\alpha, \beta)'; \widehat{\boldsymbol{\beta}}, \left(\gamma^{(g)}\right)^2 (\boldsymbol{X}'\boldsymbol{X})^{-1}\right]$$
3. Sample new drift parameter $\mu^{(g)}$ based on the normally distributed time series $r_i - J_i^{(g)} Q_i^{(g)}$ with variances $V_i^{(g)}$ as follows:
$$p\left(\mu^{(g)} | \boldsymbol{r}, \boldsymbol{J}^{(g)}, \boldsymbol{Q}^{(g)}, \boldsymbol{V}^{(g)}\right) \propto \varphi\left(\mu; \sum_{i=1}^{t} \frac{r_i - J_i^{(g)} Q_i^{(g)}}{V_i^{(g)}} \Big/ \sum_{i=1}^{t} \frac{1}{V_i^{(g)}}, \sum_{i=1}^{t} \frac{1}{V_i^{(g)}}\right)$$
4. Sample $\sigma_{RV}^{(g)}$ using the Inverse Gamma density:
$$p\left[\left(\sigma_{RV}^{(g)}\right)^2 | \boldsymbol{RV}, \boldsymbol{J}^{(g)}, \boldsymbol{Q}^{(g)}, \boldsymbol{V}^{(g)}\right] \propto IG\left[\left(\sigma_{RV}^{(g)}\right)^2; \frac{t}{2}, \frac{\sum_{i=1}^{t}\left(\log\left(RV_i - \left(J_i^{(g)}\right)^2 Q_i^{(g)}\right) - h_i^{(g)}\right)^2}{2}\right]$$
5. Sample $\mu_Z^{(g)}, \xi_Z^{(g)}, \sigma_Z^{(g)}$ by using the densities:
$$p\left[\mu_Z^{(g)} | \boldsymbol{Z}, \boldsymbol{Q}^{(g)}, \xi_Z^{(g)}, \sigma_Z^{(g)}\right] \propto \varphi\left(\mu_Z^{(g)}; \frac{\sum_{i=1}^{t}\left(Z_i - Q_i^{(g)} \xi_Z^{(g)}\right)}{t}, \frac{\sigma_Z^{(g)}}{\sqrt{t}}\right)$$
$$p\left[\xi_Z^{(g)} | \boldsymbol{Z}, \boldsymbol{Q}^{(g)}, \mu_Z^{(g)}, \sigma_Z^{(g)}\right] \propto \varphi\left(\xi_Z^{(g)}; \frac{\sum_{i=1}^{t} Q_i^{(g)}\left(Z_i - \mu_Z^{(g)}\right)}{\sum_{i=1}^{t} Q_i^{(g)}}, \frac{\sigma_Z^{(g)}}{\sqrt{\sum_{i=1}^{t} Q_i^{(g)}}}\right)$$
$$p\left[\left(\sigma_Z^{(g)}\right)^2 | \boldsymbol{Z}, \boldsymbol{Q}^{(g)}, \mu_Z^{(g)}, \xi_Z^{(g)}\right] \propto IG\left[\left(\sigma_Z^{(g)}\right)^2; \frac{t}{2}, \frac{\sum_{i=1}^{t}\left(Z_i - \mu_Z^{(g)} - Q_i^{(g)} \xi_Z^{(g)}\right)^2}{2}\right]$$
6. Sample $\mu_J^{(g)}, \sigma_J^{(g)}$ based on the normally distributed time series $\boldsymbol{J}^{(g)}$:
$$p\left(\mu_J^{(g)} | \boldsymbol{J}^{(g)}, \sigma_J^{(g-1)}\right) \propto \varphi\left(\mu_J^{(g)}; \frac{\sum_{i=1}^{t} J_i^{(g)}}{t}, \frac{\sigma_J^{(g)}}{\sqrt{t}}\right)$$
$$p\left[\left(\sigma_J^{(g)}\right)^2 | \boldsymbol{J}^{(g)}, \mu_J^{(g)}\right] \propto IG\left[\left(\sigma_J^{(g)}\right)^2; \frac{t}{2}, \frac{\sum_{i=1}^{t}\left(J_i^{(g)} - \mu_J^{(g)}\right)^2}{2}\right]$$

Alternatively, if the number of estimated jumps in $\boldsymbol{Q}^{(g)}$ is large enough (in our case we will use the condition that $\sum_{i=1}^{t} Q_i^{(g)} > 5$), it is better to use densities:

$$p\left(\mu_J^{(g)} | \boldsymbol{J}^{(g)}, \boldsymbol{Q}^{(g)}, \sigma_J^{(g-1)}\right) \propto \varphi\left(\mu_J^{(g)}; \frac{\sum_{i=1}^{t} J_i^{(g)} Q_i^{(g)}}{\sum_{i=1}^{t} Q_i^{(g)}}, \frac{\sigma_J^{(g)}}{\sqrt{\sum_{i=1}^{t} Q_i^{(g)}}}\right)$$

$$p\left[\left(\sigma_J^{(g)}\right)^2 | \boldsymbol{J}^{(g)}, \boldsymbol{Q}^{(g)}, \mu_J^{(g)}\right] \propto IG\left[\left(\sigma_J^{(g)}\right)^2; \frac{\sum_{i=1}^{t} Q_i^{(g)}}{2}, \frac{\sum_{i=1}^{t}\left(J_i^{(g)} - \mu_J^{(g)}\right)^2 Q_i^{(g)}}{2}\right]$$

7. Sample $\mu_{JV}^{(g)}, \sigma_{JV}^{(g)}$ based on the normally distributed time series $\boldsymbol{J}_V^{(g)}$:

$$p\left(\mu_{JV}^{(g)} | \boldsymbol{J}_V^{(g)}, \sigma_{JV}^{(g-1)}\right) \propto \varphi\left(\mu_{JV}^{(g)}; \frac{\sum_{i=1}^{t} J_{V,i}^{(g)}}{t}, \frac{\sigma_{JV}^{(g)}}{\sqrt{t}}\right)$$

$$p\left[\left(\sigma_{JV}^{(g)}\right)^2 | \boldsymbol{J}_V^{(g)}, \mu_{JV}^{(g)}\right] \propto IG\left[\left(\sigma_{JV}^{(g)}\right)^2; \frac{t}{2}, \frac{\sum_{i=1}^{t}\left(J_i^{(g)} - \mu_J^{(g)}\right)^2}{2}\right]$$

As in the case of price jumps, if $\sum_{i=1}^{t} Q_{V,i}^{(g)} > 5$, we will use the densities:

$$p\left(\mu_{JV}^{(g)} | \boldsymbol{J}_V^{(g)}, \boldsymbol{Q}_V^{(g)}, \sigma_{JV}^{(g-1)}\right) \propto \varphi\left(\mu_{JV}^{(g)}; \frac{\sum_{i=1}^{t} J_{V,i}^{(g)} Q_{V,i}^{(g)}}{\sum_{i=1}^{t} Q_{V,i}^{(g)}}, \frac{\sigma_{JV}^{(g)}}{\sqrt{\sum_{i=1}^{t} Q_{V,i}^{(g)}}}\right)$$

$$p\left[\left(\sigma_{JV}^{(g)}\right)^2 | \boldsymbol{J}_V^{(g)}, \boldsymbol{Q}_V^{(g)}, \mu_{JV}^{(g)}\right] \propto IG\left[\left(\sigma_{JV}^{(g)}\right)^2; \frac{\sum_{i=1}^{t} Q_{V,i}^{(g)}}{2}, \frac{\sum_{i=1}^{t}\left(J_i^{(g)} - \mu_J^{(g)}\right)^2 Q_{V,i}^{(g)}}{2}\right]$$

8. Sample the Hawkes process parameters $\theta_J, \beta_J, \gamma_J$ by using the Random-Walk Metropolis-Hastings algorithm, and the likelihood function

$$L\left(\boldsymbol{Q}^{(g)} | \theta_J, \beta_J, \gamma_J\right) = \prod_{i=1}^{t} \lambda_i^{Q_i} (1 - \lambda_i)^{1-Q_i}$$

In the applied algorithm we will use step sizes $c_{\theta_J} = 0.003$, $c_{\beta_J} = 0.08$ and $c_{\gamma_J} = 0.008$

9. Sample volatility jump Hawkes process parameters $\theta_{JV}, \beta_{JV}, \gamma_{JV}$ by using the Random-Walk Metropolis-Hastings algorithm, and the likelihood function

$$L\left(\boldsymbol{Q}_V^{(g)} | \theta_{JV}, \beta_{JV}, \gamma_{JV}\right) = \prod_{i=1}^{t} \lambda_{V,i}^{Q_{V,i}} (1 - \lambda_{V,i})^{1-Q_{V,i}}$$

In the applied algorithm we will use step sizes $c_{\theta_{JV}} = 0.003$, $c_{\beta_{JV}} = 0.08$ and $c_{\gamma_{JV}} = 0.008$

In the case of the two-component SVJD model, the step 2 needs to be replaced with the following two steps:

A. Sample new short-term log-variance component autoregression coefficients $\beta^{(g)}, \gamma^{(g)}$, by using the Bayesian linear regression model where $\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}\boldsymbol{y}$, $\hat{\boldsymbol{e}} = \boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}$, and:
$\boldsymbol{X} = \left(h_{ST,1}^{(g)} \dots h_{ST,t-1}^{(g)}\right)'$ and $\boldsymbol{y} = \left(h_{ST,2}^{(g)} \dots h_{ST,t}^{(g)}\right)'$, and sample:

$$\left(\gamma^{(g)}\right)^2 \propto IG\left(\frac{n-1}{2}, \frac{\hat{\boldsymbol{e}}'\hat{\boldsymbol{e}}}{2}\right),$$

$$\beta^{(g)} \propto \varphi\left[\beta; \widehat{\boldsymbol{\beta}}, \left(\gamma^{(g)}\right)^2 (\boldsymbol{X}'\boldsymbol{X})^{-1}\right]$$

B. Sample new long-term log-variance component autoregression coefficients $\phi_0^{(g)}, \phi_1^{(g)}, \phi_2^{(g)}$ by using the Bayesian linear regression model where $\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}\boldsymbol{y}$, $\hat{\boldsymbol{e}} = \boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}$, and
$\boldsymbol{X} = \begin{pmatrix} 1 & \dots & 1 \\ h_{LT,1}^{(g)} & \dots & h_{LT,t-1}^{(g)} \end{pmatrix}'$ and $\boldsymbol{y} = \left(h_{LT,2}^{(g)} \dots h_{LT,t}^{(g)}\right)'$, and sample:

$$\left(\phi_2^{(g)}\right)^2 \propto IG\left(\frac{n-2}{2}, \frac{\hat{\boldsymbol{e}}'\hat{\boldsymbol{e}}}{2}\right),$$

$$\left(\phi_0^{(g)}, \phi_1^{(g)}\right)' \propto \varphi\left[(\phi_0, \phi_1)'; \widehat{\boldsymbol{\beta}}, \left(\phi_2^{(g)}\right)^2 (\boldsymbol{X}'\boldsymbol{X})^{-1}\right]$$

When a correlation between volatility and returns is assumed, the value of the correlation parameter ρ needs to be sampled as well. This can be done by saving the random terms in the price and log-variance processes. We can then proceed as follows:

- Sample new correlation parameter ρ by using the Bayesian linear regression model where $\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}\boldsymbol{y}$, and $\hat{\boldsymbol{e}} = \boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}$, and:
$\boldsymbol{X} = \left(\varepsilon_{V,1}^{(g)} \dots \varepsilon_{V,t}^{(g)}\right)'$ and $\boldsymbol{y} = \left(\varepsilon_1^{(g)} \dots \varepsilon_t^{(g)}\right)'$, and sample:

$$(\sigma_\varepsilon)^2 \propto IG\left(\frac{n-1}{2}, \frac{\hat{\boldsymbol{e}}'\hat{\boldsymbol{e}}}{2}\right),$$

$$\rho^{(g)} \propto \varphi\left[\rho; \widehat{\boldsymbol{\beta}}, (\sigma_\varepsilon)^2 (\boldsymbol{X}'\boldsymbol{X})^{-1}\right]$$

Especially in more complex SVJD models, we have found that the use of priors may be necessary to improve the convergence of the algorithm to the global optimum and avoid convergence to inferior local optima. In our applications the priors will be put on all of the variance parameters in the models. In the two-component SVJD model with jumps in volatility and returns these will be the parameters ϕ_2, γ, σ_{RV}, γ_J and γ_{JV}.

3.4. Particle filtering of the latent states in SVJD models

We will now return to the issue of sampling the latent states and introduce the notion of adapted vs. un-adapted particle filters in the framework of SVJD models.

3.4.1. Adapted vs. un-adapted filters

The latent states in a particle filter need to be sampled from a proposal distribution $g(x_t|x_{t-1}, y_t)$, with the weights of the particles subsequently re-weighted with:

$$w_t^{(i)} = \frac{p(y_t|x_t^i)p(x_t^i|x_{t-1}^i)}{g(x_t^i|x_{t-1}^i, y_t)} \widetilde{w}_{t-1}^{(i)} \tag{117}$$

In the ideal case of a fully-adapted particle filter, the proposal distribution should be set equal to $g(x_t^i|x_{t-1}^i, y_t) = p(x_t^i|x_{t-1}^i, y_t)$. Unfortunately, the true conditional distribution $p(x_t^i|x_{t-1}^i, y_t)$ is often intractable and a different proposal distribution needs to be used.

The simplest way is to construct the particle filter as un-adapted, in which case the proposal distribution is set equal to the transition distribution, $g(x_t^i|x_{t-1}^i, y_t) = p(x_t^i|x_{t-1}^i)$, with the weight update reducing $w_t^{(i)} = p(y_t|x_t^i)\widetilde{w}_{t-1}^{(i)}$.

Unfortunately, this approach tends be very inefficient, especially when multiple latent states need to be samples (i.e. x_t^i is a vector), which often leads to a situation where majority of the particles get immediately rejected (i.e. their weight drops towards values close to zero) during the re-weighting step, when the information about the information y_t is introduced.

To improve the efficiency of the particle filter it is therefore advisable to adapt the proposal distribution $g(x_t^i|x_{t-1}^i, y_t)$ at least partially to the observation y_t.

The methods of how to do it in the framework of the presented SVJD model will be discussed in the following sections. For additional discussion of this topic, see the Auxiliary Particle Filter of Pitt and Shephard (1999), and the Approximate Rao-Blackwellization method of Johansen et al. (2012).

3.4.2. Un-adapted sampling for a SVJD model with price jumps

Lets consider the SVJD model with self-exciting price jumps with the following 3 equations:

$$\begin{aligned} r_t &= \mu + \sigma_t \varepsilon_t + J_t Q_t \\ h_t &= \alpha + \beta h_{t-1} + \gamma \varepsilon_{V,t} \\ \lambda_t &= \alpha_J + \beta_J \lambda_{t-1} + \gamma_J Q_{t-1} \end{aligned} \tag{118}$$

And $\varepsilon_t \sim N(0,1)$, $\varepsilon_{V,t} \sim N(0,1)$, $h_t = \log(V_t), V_t = \sigma_t^2$, $\alpha = (1-\beta)h_{LT}$, $\alpha = (1-\beta)h_{LT}$, $Q_t \sim Bern[\lambda_t]$, $\Pr[Q_t = 1] = \lambda_t$, $J_t \sim N(\mu_J, \sigma_J)$.

In the un-adapted version of the particle filter, we set the proposal density equal to the transition density:

$$p\left(h_t^i, J_t^i, Q_t^i | h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i\right) \tag{119}$$

The proposal values of h_t^i, J_t^i and Q_t^i can thus be sampled from the model equations as:

$$\begin{aligned} &p\left(h_t^i | h_{t-1}^i\ \right) \sim N(\alpha + \beta h_{t-1}, \gamma) \\ &p\left(J_t^i\right) \sim N\left(\mu_J, \sigma_J\right) \\ &p\left(Q_t^i | \lambda_t^i\right) \sim Bern\left[\lambda_t^i\right] \end{aligned} \tag{120}$$

And as the likelihood of the model is given by:

$$p\left(r_t | h_t^i, J_t^i, Q_t^i\right) \sim N\left(r_t; \mu + J_t^i Q_t^i, \sigma_t^i\right) \tag{121}$$

The weight update would be:

$$w_t^{(i)} = \ p\left(r_t | h_t^i, J_t^i, Q_t^i\right) \widetilde{w}_{t-1}^{(i)} \tag{122}$$

3.4.3. Adaptation to the SVJD model

We can see that the particle filter proposed in the previous section does not utilize the information about the observation r_t when sampling the proposal values of the latent states h_t^i, J_t^i and Q_t^i, although the information is already available at time t.

The result of this is that the sampled values of the latent states may be inefficient and the majority of weights will drop to negligible values when the information about r_t is introduced during the re-weighting.

Consider a situation when the jump intensity λ_t^i is low, but a jump still occurred at t. As only very few particles will be sampled with $Q_t^i = 1$, due to the low value of λ_t^i, the vast majority of the particles will get rejected, when the information about the large jump-induced return r_t is used in the re-weighting step. A very large number of particles will thus have to be used in order to get reasonably precise estimates of the other latent states J_t^i and h_t^i.

To cope with the problem, we can use a proposal density that already uses the information about r_t for the sampling of the latent states:

$$g\left(h_t^i, J_t^i, Q_t^i | h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t\right) \tag{123}$$

With the re-weighting step performed according to:

$$w_t^{(i)} = \frac{p\left(r_t | h_t^i, J_t^i, Q_t^i\right) p\left(h_t^i, J_t^i, Q_t^i | h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i\right)}{g\left(h_t^i, J_t^i, Q_t^i | h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t\right)} \widetilde{w}_{t-1}^{(i)} \tag{124}$$

Unfortunately, the fully adapted density $p\left(h_t^i, J_t^i, Q_t^i | h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t\right)$ is in this case intractable, due to the non-linearity of the relationship between h_t^i and r_t.

The approach that will be used is therefore to adapt the particle filter at least to the jump occurrences and the jump sizes, while sampling the h_t^i from its transition density.

3.4.4. Adaptation to price jump sizes and occurrences

To adapt the particle filter to price jump sizes and occurrences, we rewrite the proposal density as:

$$\begin{aligned} &g\left(h_t^i, J_t^i, Q_t^i | h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t\right) \\ &= p\left(J_t^i, Q_t^i | h_t^i, h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t\right) p\left(h_t^i | h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i\right) \end{aligned} \tag{125}$$

And then decompose the bi-variate density of J_t^i and Q_t^i into:

$$\begin{aligned} &p\left(J_t^i, Q_t^i | h_t^i, h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t\right) \\ &= p\left(J_t^i | h_t^i, Q_t^i, h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t\right) p\left(Q_t^i | h_t^i, h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t\right) \end{aligned} \tag{126}$$

Which is equivalent to:

$$p\left(J_t^i, Q_t^i | h_t^i, h_{t-1}^i, J_{t-1}^i, Q_{t-1}^i, \lambda_{t-1}^i, r_t\right) = p\left(J_t^i | h_t^i, Q_t^i, r_t\right) p\left(Q_t^i | h_t^i, \lambda_t^i, r_t\right) \quad (127)$$

For $Q_t^i = 0$ it then hold that:

$$p\left(J_t^i | h_t^i, r_t, Q_t^i = 0\right) = p\left(J_t^i\right) \quad (128)$$

With $p\left(J_t^i\right) \sim N\left(\mu_J, \sigma_J\right)$.

For $Q_t^i = 1$, we can use the Bayes theorem to express $p\left(J_t^i | h_t^i, r_t, Q_t^i = 1\right)$ as:

$$p\left(J_t^i | h_t^i, r_t, Q_t^i = 1\right) \propto p\left(r_t | h_t^i, J_t^i, Q_t^i\right) p\left(J_t^i\right) \quad (129)$$

Which is a multiple of two Gaussian densities:

$$p\left(J_t^i | h_t^i, r_t, Q_t^i = 1\right) \propto N\left(r_t; \mu + J_t^i, \sigma_t^i\right) N\left(J_t^i; \mu_J, \sigma_J\right) \quad (130)$$

And the adapted density can be derived as:

$$p\left(J_t^i | h_t^i, r_t, Q_t^i = 1\right) \sim N\left(\mu_J^*, \sigma_J^*\right) \quad (131)$$

Where:

$$\mu_J^* = \frac{(r_t - \mu)\sigma_J^2 + \mu_J V_t^i}{\sigma_J^2 + V_t^i} \quad (132)$$

$$\sigma_J^* = \frac{\sigma_J \sigma_t^i}{\sqrt{\sigma_J^2 + V_t^i}} \quad (133)$$

The next task is to derive $p\left(Q_t^i | h_t^i, \lambda_t^i, r_t\right)$. Using the Bayes theorem:

$$p\left(Q_t^i | h_t^i, \lambda_t^i, r_t\right) \propto p\left(r_t | h_t^i, Q_t^i\right) p\left(Q_t^i | \lambda_t^i\right) \quad (134)$$

The density $p\left(r_t|h_t^i, Q_t^i\right)$ is the marginal of $p\left(r_t, J_t|h_t^i, Q_t^i\right)$ computed by integrating it over J_t.

When $Q_t^i = 0$, the J_t and r_t are independent, and thus:

$$p\left(r_t|h_t^i, Q_t^i = 0\right) = N\left(r_t; \mu, \sigma_t^i\right) \tag{135}$$

When $Q_t^i = 1$, then we decompose the density $p\left(r_t, J_t|h_t^i, Q_t^i = 1\right)$ into:

$$p\left(r_t, J_t|h_t^i, Q_t^i = 1\right) = p\left(r_t|J_t, h_t^i, Q_t^i\right)p\left(J_t|h_t^i, Q_t^i\right) \tag{136}$$

Which is a multiple of two Gaussian densities:

$$p\left(r_t, J_t|h_t^i, Q_t^i = 1\right) = N\left(r_t; \mu + J_t, \sigma_t^i\right)N\left(J_t; \mu_J, \sigma_J\right) \tag{137}$$

Or equivalently:

$$p\left(r_t, J_t|h_t^i, Q_t^i = 1\right) = N\left(r_t - \mu; J_t, \sigma_t^i\right)N\left(J_t; \mu_J, \sigma_J\right) \tag{138}$$

The density $r_t - \mu$ is a Compound Gaussian density, with mean distributed according to the Gaussian density $N\left(J_t; \mu_J, \sigma_J\right)$. We can thus integrate over J_t to get:

$$p\left(r_t|h_t^i, Q_t^i = 1\right) = N\left(r_t; \mu_J + \mu, \sqrt{\sigma_J^2 + V_t^i}\right) \tag{139}$$

Returning to the relationship $p\left(Q_t^i|h_t^i, \lambda_t^i, r_t\right) \propto p\left(r_t|h_t^i, Q_t^i\right)p\left(Q_t^i|\lambda_t^i\right)$, we can see that we can easily compute the normalizing constant, as the Q_t^i is only binary.

Thus, we get for $p\left(Q_t^i|h_t^i, \lambda_t^i, r_t\right)$:

$$p\left(Q_t^i|h_t^i, \lambda_t^i, r_t\right) \sim Bern\left[\lambda_t^{i*}\right] \tag{140}$$

Where:

$$\lambda_t^{i*} = \frac{N\left(r_t; \mu_J + \mu, \sqrt{\sigma_J^2 + V_t^i}\right)\lambda_t^i}{N\left(r_t; \mu_J + \mu, \sqrt{\sigma_J^2 + V_t^i}\right)\lambda_t^i + N(r_t; \mu, \sigma_t^i)(1 - \lambda_t^i)} \tag{141}$$

In order to sample from the proposal density, we will thus first sample from the transition density $p(h_t^i | h_{t-1}^i)$ the log-variance, as:

$$p(h_t^i | h_{t-1}^i\ ) \sim N(\alpha + \beta h_{t-1}, \gamma) \tag{142}$$

Then we sample from the adapted jump occurrence density $p(Q_t^i | h_t^i, \lambda_t^i, r_t)$:

$$p(Q_t^i | h_t^i, \lambda_t^i, r_t) \sim Bern[\lambda_t^{i*}] \tag{143}$$

And finally the adapted jump sizes from $p(J_t^i | h_t^i, Q_t^i, r_t)$, as already described.

The importance weight update is then given by:

$$w_t^{(i)} = \frac{p(r_t | h_t^i, J_t^i, Q_t^i) p(J_t^i) (\lambda_t^i)^{Q_t^i} (1 - \lambda_t^i)^{1 - Q_t^i}}{p(J_t^i | h_t^i, r_t, Q_t^i) (\lambda_t^{i*})^{Q_t^i} (1 - \lambda_t^{i*})^{1 - Q_t^i}} \widetilde{w}_{t-1}^{(i)} \tag{144}$$

3.4.5. Adaptation of price and volatility jumps

The adaptation become slightly more complicated in the case when the model contains jumps in price, as well as in the volatility. The model equations would in that case be:

$$\begin{aligned} r_t &= \mu + \sigma_t \varepsilon_t + J_t Q_t \\ h_t &= \alpha + \beta h_{t-1} + \gamma \varepsilon_{V,t} + J_{V,t} Q_{V,t} \\ \lambda_t &= \alpha_J + \beta_J \lambda_{t-1} + \gamma_J Q_{t-1} \\ \lambda_{V,t} &= \alpha_{JV} + \beta_{JV} \lambda_{V,t-1} + \gamma_{JV} Q_{V,t-1} \end{aligned} \tag{145}$$

As was the case for the log-variance itself, the distribution of the log-variance jump sizes unfortunately cannot be combined with the other distributions, so we cannot adapt the particle filter perfectly. Nevertheless, it turns out that the filter can at least be adapted to the price jump sizes, as well as the price jump and the volatility jump occurrences.

The adapted sampling will proceed as follows.

We first sample the diffusion log-variances and log-variance jump sizes:

$$\begin{aligned} &p\left(h_{t,Diff}^{i}|h_{t-1}^{i}\right)\sim N(\alpha+\beta h_{t-1},\gamma) \\ &p\left(J_{V,t}^{i}\right)\sim N\left(\mu_{JV},\sigma_{JV}\right) \end{aligned} \tag{146}$$

Where $h_{t,Diff}^{i}$ denotes the diffusion log-variance, while the jump-diffusion log-variance can be computed as $h_{t,Jump}^{i}=h_{t,Diff}^{i}+J_{V,t}^{i}$.

In the next step, we define the following likelihood densities, corresponding to all possible combinations of price and volatility jumps:

$$\begin{aligned} p_{0,0} &= p\left(r_t|h_{t,Diff}^{i},\lambda_t^i,\lambda_{V,t}^i,Q_t^i=0,Q_{V,t}^i=0\right) \\ p_{0,1} &= \ p\left(r_t|h_{t,Jump}^{i},\lambda_t^i,\lambda_{V,t}^i,Q_t^i=0,Q_{V,t}^i=1\right) \\ p_{1,0} &= \ p\left(r_t|h_{t,Diff}^{i},\lambda_t^i,\lambda_{V,t}^i,Q_t^i=1,Q_{V,t}^i=0\right) \\ p_{1,1} &= \ p\left(r_t|h_{t,Jump}^{i},\lambda_t^i,\lambda_{V,t}^i,Q_t^i=1,Q_{V,t}^i=1\right) \end{aligned} \tag{147}$$

Which can be calculated as follows:

$$\begin{aligned} &p_{0,0}\sim N\left(r_t;\mu,\sigma_{t,Diff}^{i}\right)\left(1-\lambda_t^i\right)\left(1-\lambda_{V,t}^i\right) \\ &p_{0,1}\sim N\left(r_t;\mu,\sigma_{t,Jump}^{i}\right)\left(1-\lambda_t^i\right)\lambda_{V,t}^i \\ &p_{1,0}\sim N\left(r_t;\mu+\mu_J,\sqrt{\sigma_J^2+V_{t,Diff}^{i}}\right)\lambda_t^i\left(1-\lambda_{V,t}^i\right) \\ &p_{1,1}\sim N\left(r_t;\mu+\mu_J,\sqrt{\sigma_J^2+V_{t,Jump}^{i}}\right)\lambda_t^i\lambda_{V,t}^i \end{aligned} \tag{148}$$

Where $V_{t,Diff}^{i}=\exp(h_{t,Diff}^{i})$, $V_{t,Jump}^{i}=\exp(h_{t,Jump}^{i})$, $\sigma_{t,Diff}^{i}={V_{t,Diff}^{i}}^{1/2}$ and $\sigma_{t,Jump}^{i}=\sigma_J^2+{V_{t,Jump}^{i}}^{1/2}$.

The volatility jump occurrence $Q_{V,t}^{i}$ can then be sampled from the adapted density:

$$p\left(Q_{V,t}^{i}|h_{t,Diff}^{i},J_{V,t}^{i},\lambda_t^i,\lambda_{V,t}^i,r_t\right)\sim Bern\left[\lambda_{V,t}^{i*}\right] \tag{149}$$

Where:

$$\lambda_{J,t}^{i*}=\frac{p_{0,1}+p_{1,1}}{p_{0,0}+p_{0,1}+p_{1,0}+p_{1,1}} \tag{150}$$

In the next step, we use the sampled $h^i_{t,Diff}$, $J^i_{V,t}$ and $Q^i_{V,t}$ to computed the final log-variances h^i_t:

$$h^i_t = h^i_{t,Diff} + J^i_{V,t} Q^i_{V,t} \tag{151}$$

We can then proceed in the same way as in the previous section and sample the price jump occurrences from $p\left(Q^i_t | h^i_t, \lambda^i_t, r_t\right)$ and finally the price jump sizes from $p\left(J^i_t | h^i_t, Q^i_t, r_t\right)$.

The weight update will then be given as:

$$w^{(i)}_t = \frac{p\left(r_t | h^i_t, J^i_t, Q^i_t\right) p\left(J^i_t\right) \left(\lambda^i_t\right)^{Q^i_t} \left(1 - \lambda^i_t\right)^{1-Q^i_t} \left(\lambda^i_{V,t}\right)^{Q^i_{V,t}} \left(1 - \lambda^i_{V,t}\right)^{1-Q^i_{V,t}}}{p\left(J^i_t | h^i_t, r_t, Q^i_t\right) \left(\lambda^{i*}_t\right)^{Q^i_t} \left(1 - \lambda^{i*}_t\right)^{1-Q^i_t} \left(\lambda^{i*}_{V,t}\right)^{Q^i_{V,t}} \left(1 - \lambda^{i*}_{V,t}\right)^{1-Q^i_{V,t}}} \widetilde{w}^{(i)}_{t-1} \tag{152}$$

3.4.6. Adaptation for SVJD-RV and SVJD-RV-Z models

Realized variance can be added to the previously presented SVJD models (with price jumps and volatility jumps alike) by including an additional equation that links the estimated stochastic variances to the observed realized variances:

$$\log[RV(t) - J^2(t) Q(t)] = h(t) + \sigma_{RV} \varepsilon_{\mathrm{RV}}(t) \tag{153}$$

Similarly, the Z-Estimator of jumps can be included via equation that links its values to the days in which price jumps were estimated by the SVJD model:

$$Z(t) = \mu_Z + \xi_Z Q(t) + \sigma_Z \varepsilon_Z(t) \tag{154}$$

With the likelihood of the model changing from $p\left(r_t | h^i_t, J^i_t, Q^i_t\right)$ to $p\left(r_t, RV_t | h^i_t, J^i_t, Q^i_t\right)$ in the case of the SVJD-RV model, and to $p\left(r_t, RV_t, Z_t | h^i_t, J^i_t, Q^i_t\right)$ in the case of the SVJD-RV-Z model.

Although contested by some authors (Koopman and Scharth, 2013), the common approach is to assume independence between the r_t and RV_t noise, in which case, the likelihood for the SVJD-RV model is:

$$p\left(r_t, RV_t|h_t^i, J_t^i, Q_t^i\right) = \left(r_t|h_t^i, J_t^i, Q_t^i\right)\left(RV_t|h_t^i, J_t^i, Q_t^i\right) \tag{155}$$

And analogously for the SVJD-RV-Z model, as long as we assume r_t, RV_t and Z_t independence, the likelihood is:

$$p\left(r_t, RV_t, Z_t|h_t^i, J_t^i, Q_t^i\right) = \left(r_t|h_t^i, J_t^i, Q_t^i\right)\left(RV_t|h_t^i, J_t^i, Q_t^i\right)\left(Z_t|J_t^i, Q_t^i\right) \tag{156}$$

The latent states can then theoretically be sampled from the proposal distributions adapted to the returns, which were derived in the previous sections, with only the likelihood in the final, weight-update step, changing either to $p\left(r_t, RV_t|h_t^i, J_t^i, Q_t^i\right)$ or $p\left(r_t, RV_t, Z_t|h_t^i, J_t^i, Q_t^i\right)$.

Nevertheless, as the proposal distributions were adapted only to the returns, and not the RV_t and Z_t, which are already observable at t, the can be viewed as sub-optimal, with large number of particles getting sampled at values inconsistent with the observed RV_t and Z_t, causing them to get rejected during the re-weighting and re-sampling steps.

A problem arising when adapting the proposal distribution to RV_t is that the relationship between RV_t and J_t^i is non-linear, giving rise to an intractable proposal distribution. We therefore decided to adapt the particle filter only with respect to the jump occurrences Q_t^i and $Q_{V,t}^i$.

In the first step, we will thus sample the $h_{t,Diff}^i$, $J_{V,t}^i$ and J_t^i:

$$\begin{aligned} &p\left(h_{t,Diff}^i|h_{t-1}^i\right) \sim N(\alpha + \beta h_{t-1}, \gamma) \\ &p\left(J_{V,t}^i\right) \sim N\left(\mu_{JV}, \sigma_{JV}\right) \\ &p\left(J_t^i\right) \sim N\left(\mu_J, \sigma_J\right) \end{aligned} \tag{157}$$

We then calculate the adapted $p_{0,0}$, $p_{0,1}$, $p_{1,0}$ and $p_{1,1}$, defined analogously as in the previous section, but this time including also the information about RV_t and Z_t, as being conditional on the sampled value of J_t^i (which was in the previous section marginalized out, which unfortunately cannot be done now, as long as we want to include RV_t).

For the SVJD-RV model, the densities will be calculated as

$$\begin{aligned}
&p_{0,0} \sim N\left(r_t; \mu, \sigma^i_{t,Diff}\right) N(\log(RV_t); \mu_{RV}, \sigma_{RV})\left(1-\lambda^i_t\right)\left(1-\lambda^i_{V,t}\right)\\
&p_{0,1} \sim N\left(r_t; \mu, \sigma^i_{t,Jump}\right) N(\log(RV_t); \mu_{RV}, \sigma_{RV})\left(1-\lambda^i_t\right)\lambda^i_{V,t}\\
&p_{1,0} \sim N\left(r_t; \mu + J^i_t, \sigma^i_{t,Diff}\right)\left(\log\left(RV_t - \left(J^i_t\right)^2\right); \mu_{RV}, \sigma_{RV}\right)\lambda^i_t\left(1-\lambda^i_{V,t}\right)\\
&p_{1,1} \sim N\left(r_t; \mu + J^i_t, \sigma^i_{t,Jump}\right)\left(\log\left(RV_t - \left(J^i_t\right)^2\right); \mu_{RV}, \sigma_{RV}\right)\lambda^i_t\lambda^i_{V,t}
\end{aligned} \tag{158}$$

While for the SVJD-RV-Z model, the densities $p_{0,0}$ and $p_{0,1}$ will further need to be multiplied with $N(Z_t; \mu_Z, \sigma_Z)$, and the densities $p_{1,0}$ and $p_{1,1}$ with $N(Z_t; \mu_Z + \xi_Z, \sigma_Z)$

The volatility jumps $Q^i_{V,t}$ can then be sampled from:

$$p\left(Q^i_{V,t} | h^i_{t,Diff}, J^i_{V,t}, \lambda^i_t, \lambda^i_{V,t}, r_t, RV_t, Z_t\right) \sim Bern\left[\lambda^{i*}_{V,t}\right] \tag{159}$$

Where:

$$\lambda^{i*}_{J,t} = \frac{p_{0,1} + p_{1,1}}{p_{0,0} + p_{0,1} + p_{1,0} + p_{1,1}} \tag{160}$$

From the sampled $Q^i_{V,t}$ we can then compute the log-variances $h^i_t = h^i_{t,Diff} + J^i_{V,t}Q^i_{V,t}$, and use them to define the jump likelihood densities p_1 and p_0. These are for the SVJD-RV model equal to:

$$\begin{aligned}
&p_0 \sim N\left(r_t; \mu, \sigma^i_t\right) N(\log(RV_t); \mu_{RV}, \sigma_{RV})\left(1-\lambda^i_t\right)\\
&p_1 \sim N\left(r_t; \mu + J^i_t, \sigma^i_t\right)\left(\log\left(RV_t - \left(J^i_t\right)^2\right); \mu_{RV}, \sigma_{RV}\right)\lambda^i_t
\end{aligned} \tag{161}$$

While for the SVJD-RV-Z model p_0 is further multiplied with $N(Z_t; \mu_Z, \sigma_Z)$ and p_1 with $N(Z_t; \mu_Z + \xi_Z, \sigma_Z)$.

Price jump occurrences can then be sampled from:

$$p\left(Q^i_t | h^i_t, \lambda^i_t, r_t\right) \sim Bern\left[\lambda^{i*}_t\right] \tag{162}$$

Where

$$\lambda^{i*}_t = \frac{p_1}{p_0 + p_1}$$

(163)

The weight update for is then equal to:

$$w_t^{(i)} = \frac{p\left(r_t | h_t^i, J_t^i, Q_t^i\right)\left(\lambda_t^i\right)^{Q_t^i}\left(1-\lambda_t^i\right)^{1-Q_t^i}\left(\lambda_{V,t}^i\right)^{Q_{V,t}^i}\left(1-\lambda_{V,t}^i\right)^{1-Q_{V,t}^i}}{\left(\lambda_t^{i*}\right)^{Q_t^i}\left(1-\lambda_t^{i*}\right)^{1-Q_t^i}\left(\lambda_{V,t}^{i*}\right)^{Q_{V,t}^i}\left(1-\lambda_{V,t}^{i*}\right)^{1-Q_{V,t}^i}} \widetilde{w}_{t-1}^{(i)} \tag{164}$$

With the difference between SVJD-RV and SVJD-RV-Z models being only in the values of λ_t^{i*} and $\lambda_{V,t}^{i*}$, computed as described in the previous steps.

3.4.7. Approximate adaptations

As long as we have multiple observable time series in our model (in the SVJD-RV-Z model these would be: r_t, RV_t and Z_t), there is a natural possibility of adapting the particle filter only to the ones of them where the proposal densities are tractable, while leaving the filter un-adapted with respect to the other ones. Lets take for example the adaptation of h_t^i in the one-component SVJD-RV-Z models. The sampling of h_t^i involves two major hurdles. The first one is that the distribution $p\left(h_t^i | r_t\right)$ is intractable due to the non-linearity of the relationship between h_t^i and r_t. The second hurdle is related to the distribution $p\left(h_t^i | RV_t\right)$. As the assumed relationship between h_t^i and RV_t in the analysed model is given by $\log[RV_t - J_t^2 Q_t] = h_t + \sigma_{RV}\varepsilon_{\mathrm{RV},t}$, it is apparent that as long as we know the values of J_t^i and Q_t^i, we can easily express the density $p\left(h_t^i | RV_t, J_t^i, Q_t^i\right)$. Unfortunately, as long as we want to sample the values of J_t^i and Q_t^i in an adapted way, we need to know the value of h_t^i in advance. The values of J_t^i and Q_t^i are thus unavailable during the sampling of h_t^i, and the marginalized density $p\left(h_t^i | RV_t\right)$ is intractable. A possible approximate solution of how to utilize the fact that at least in the no-jump case, h_t^i can be accurately adapted to RV_t, is discussed in the appendix. Based on similar logic, approximate proposal densities are then derived also for the size of price and volatility jumps, and for the two log-variance components of the two-component SVJD-RV-Z model.

4. Application of Particle Filters for SVJD model estimation

In order to assess the ability of the proposed Particle Filtering methods (adapted SIR particle filters and the Sequential Gibbs Particle Filter) to filter the latent states and learn the model parameters of complex SVJD models, the following tests will be performed:

1. Simulation tests of the un-adapted and adapted particle filters, performed with known model parameters, with the goal of evaluating the ability of the particle filters to filter the latent states of SVJD and SVJD-RV-Z models
2. Simulation tests of the Sequential Gibbs Particle Filter (SGPF), performed to assess the ability of the method to learn SVJD model parameters and filter the latent states in the case when parameters are unknown in advance
3. Tests of convergence of the SGPF particle filter parameter estimates, when applied to real-word data of the EUR/USD exchange rate, with the goal of evaluating whether multiple runs of the algorithm converge to the same values

4.1. Simulation tests of adapted and un-adapted particle filters

The first part of the simulation study focuses solely on the filtering performance of the adapted and un-adapted particle filters. The filters will thus be applied to simulated time series generated by SVJD models with known parameters. Three tests are performed:

1. Filtering of the latent states of a SVJD model with self-exciting jumps in prices
2. Filtering of the latent states of SVJD and SVJD-RV-Z models with self-exciting jumps in prices and volatility
3. Filtering of the latent states of a 2-component SVJD-RV-Z model with self-exciting jumps in prices and volatility

4.1.1. Filtering of latent states in a SVJD model with self-exciting jumps

Several adapted and un-adapted particle filters were applied for the filtering of the latent states of a SVJD model with self-exciting price jumps. The parameters of the SVJD model used in the simulations are $\mu = \frac{0.05}{252}$, $\mu_J = -0.01$, $\sigma_J = 0.04$, $V_{LT} = e^{\frac{\alpha}{1-\beta}} = 0.01^2$, $\beta = 0.98$, $\gamma = 0.2$, $\lambda_{LT} = \frac{\alpha_J}{1-\beta_J-\gamma_J} = 0.02$, $\beta_J = 0.95$, and $\gamma_J = 0.04$. The values were chosen so that they correspond to the parameters commonly observed on foreign exchange time series.

500 time series were simulated, each with 4000 observations, and a SIR particle filter with 100 particles and 50 particle re-sampling threshold was run on each of them. The filter was run in 4 different versions: non-adapted, jump-size adapted, jump-occurrence adapted, and fully jump adapted in order to assess whether the adapted filters provide better estimates of the latent states.

Figure 1 and Figure 2 show the simulated and the filtered latent state time series for one simulation run, for the un-adapted and the fully jump adapted particle filter.

Figure 1 – Simulated vs. filtered time series for the un-adapted particle filter

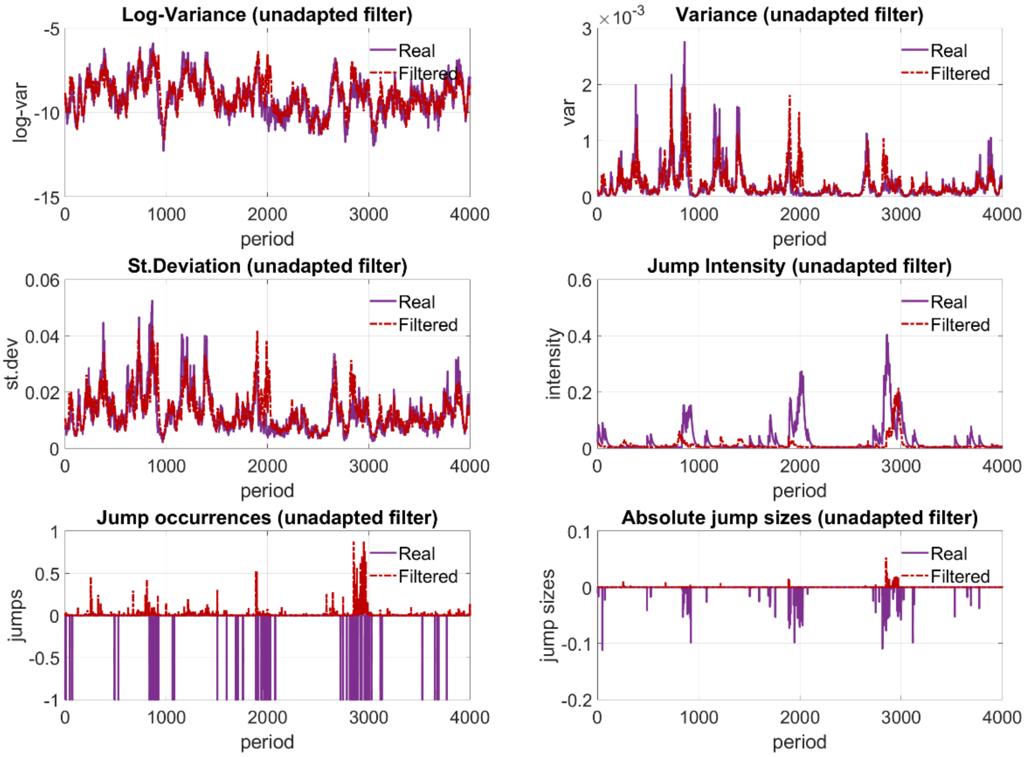
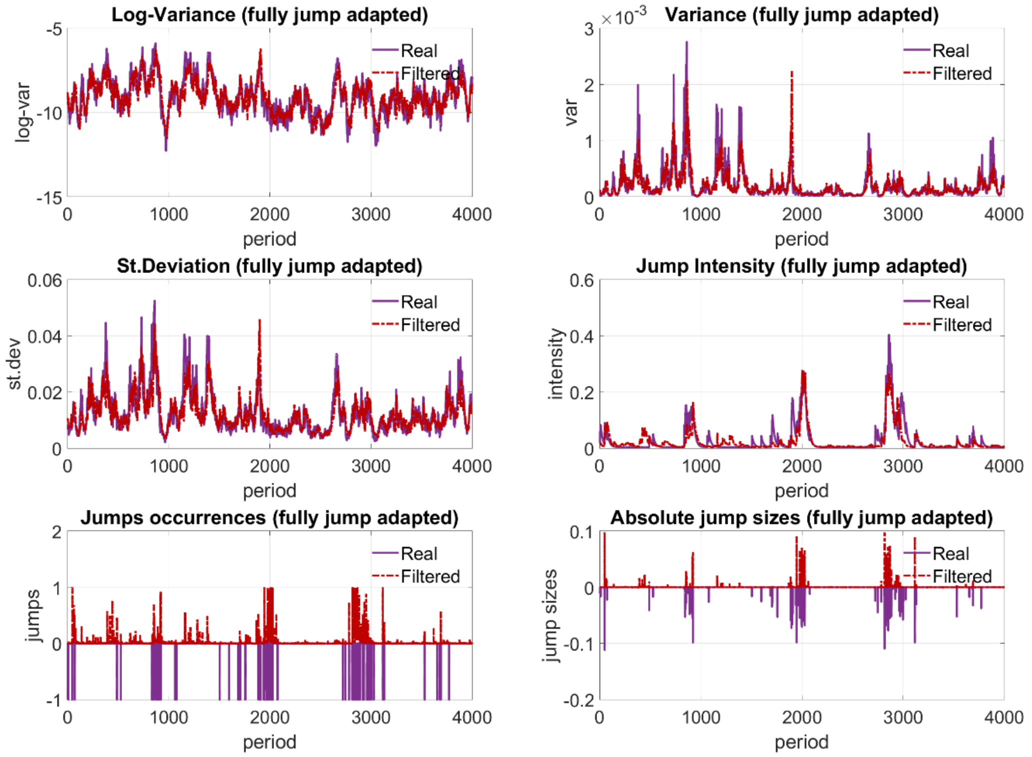


Figure 2 - Simulated vs. filtered time series for the jump-adapted particle filter

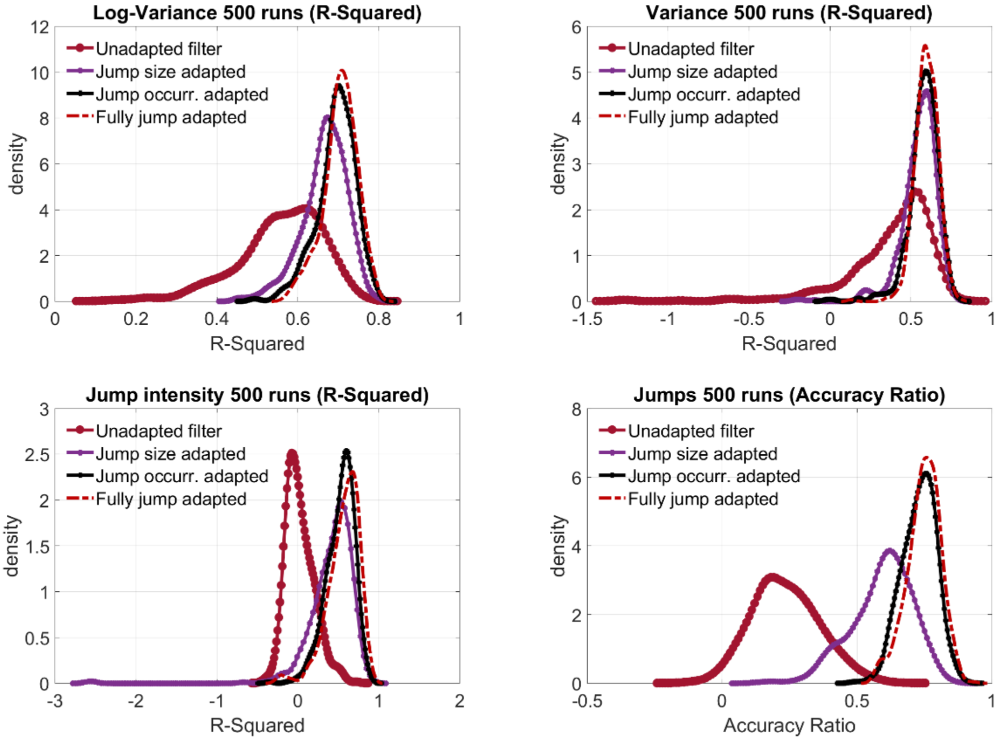


We can see that the un-adapted particle filter failed to accurately estimate most of the simulated jumps, resulting in a poor estimation of the underlying Hawkes process jump intensities, especially with regards to the jump clusters around the 2000th and the 3000th period. The adapted particle filter, on the other hand, managed to estimate the jumps much more efficiently (the Bayesian jump probabilities of occurrence are often close to one), and it also managed to capture the two main jump intensity clusters around the 2000th and the 3000th period.

In the next step, the simulation experiment was repeated 500 times, in order to assess the performance differences between the filters more quantitatively. For each simulation, 4 particle filters with different adaptation schemes were run, and the accuracy of their filtering estimates was assessed with the R-Squared (for the log-variance, variance and jump-intensity estimates) and the Accuracy Ratio (for jump occurrence estimates, where the target variable is only binary).

Figure 3 shows the distributions of the R-Squared and the Accuracy Ratios for all of the tested particle filters over the 500 simulation runs.

Figure 3 – R-Squared and Accuracy Ratio distributions of the filtered estimates of the adapted and un-adapted particle filters over 500 simulations (SVJD model with self-exciting jumps in returns)



It is apparent that the un-adapted particle filter achieved on average the worst performance, especially with regards to the jump occurrences and the jump intensities, on which its predictive power is close to zero. The best results were achieved by the fully jump adapted particle filter, followed by the filter adapted only to the jump occurrences, and then the filter adapted only to the jump sizes. The relatively small differences in performance between the fully-jump-adapted and the jump-occurrence adapted particle filter provide an important result, as for the more complex SVJD models, the exact adaptation of the jump sizes is often not possible, and we will thus have to use the jump-occurrence adapted filter in their case. The results of the simulation show that the drop of predictive power (compared to the theoretically ideal fully-jump adapted filter) should be relatively small.

Table 1 further shows the average values of the R-Squared and the Accuracy Ratios achieved by the adapted and un-adapted filters over the 500 simulation runs. The results clearly show the superiority of the fully jump-adapted filter and the jump-occurrence adapted filter against the other ones.

Table 1 – Average R-Squared and Accuracy Ratios achieved by different particle filters in 500 simulations of the SJVD process with self-exciting price jump

Particle Filter	Un-Adapted filter	Jump-size adapted	Jump-occurr. adapted	Fully jump-adapted
log-variance R2	0.5587	0.6656	0.6939	0.7046
variance R2	0.3847	0.5504	0.5801	0.5960
jump intensity R2	0.0262	0.4177	0.5246	0.5644
jump occurrence AR	0.2362	0.5928	0.7301	0.7471

4.1.2. Filtering in SVJD and SVJD-RV-Z models with volatility jumps

While the estimation of price jumps is itself a difficult task, the estimation of volatility jumps is even more problematic, as the volatility itself is unobservable. This turns out to be even more problematic in the case of Particle Filters, than what is the case for MCMC, as the particle filters work only with the information about the returns from $t = 1, \ldots, t$ when estimating the latent states at time t. The thus have only the information about the return at time t in order to tell, whether a jump (in prices or volatility) occurred, which makes it very difficult to distinguish price and volatility jumps from each other.

In the performed simulation test, we will work only with jump-occurrence adapted particle filters, and the goal will be to compare their filtering performance when applied to either the SVJD model with self-exciting jumps in returns and volatility, or a SVJD-RV-Z model with the same features. As the two models are set to be identical with respect to the assumed dynamics of the price process, they can be applied to the same simulated time series, with the only difference being that the SVJD-RV-Z model will use the realized variance and the Z-Estimator as additional sources of information, which should improve the ability of the particle filter to identify price and volatility jumps in the time series.

The models and particle filters used for the comparison are:

- SVJD model with latent states filtered with the jump-adapted particle filter
- SVJD-RV-Z model with latent states filtered with the jump-adapted particle filter (described in section ???)
- SVJD-RV-Z model with latent states filtered with the approximately fully adapted particle filter (described in the appendix ???)

The time series were simulated according to the process defined in ????, with parameters (expressed in daily values): $\mu = \frac{0.05}{252}$, $\mu_J = -0.01$, $\sigma_J = 0.04$, $V_{LT} = e^{\frac{\alpha}{1-\beta}} =$

0.01^2, $\beta = 0.98$, $\gamma = 0.2$, $\lambda_{LT} = \frac{\alpha_J}{1-\beta_J-\gamma_J} = 0.02$, $\beta_J = 0.95$, $\gamma_J = 0.04$, $\mu_{JV} = 0.3$, $\sigma_J = 0.5$, $\lambda_{LT,V} = \frac{\alpha_{JV}}{1-\beta_{JV}-\gamma_{JV}} = 0.02$, $\beta_{JV} = 0.95$, $\gamma_{JV} = 0.04$. The simulations were run on a 15-minute frequency (with appropriately adjusted parameters), so that we can ex-post calculate the time series of RV and Z to be used in the estimation. Values of σ_{RV}, μ_Z, ξ_Z and σ_Z were then set to the empirically observed values in the simulation.

Figure 4 and Figure 5 show the filtered latent states for one simulation run, estimated for the SVJD model and the SVJD-RV-Z model with the jump-adapted particle filter.

Figure 4 – Simulated vs. filtered time series for the SVJD model with jumps in prices and volatility (filtered with the jump-adapted particle filter)

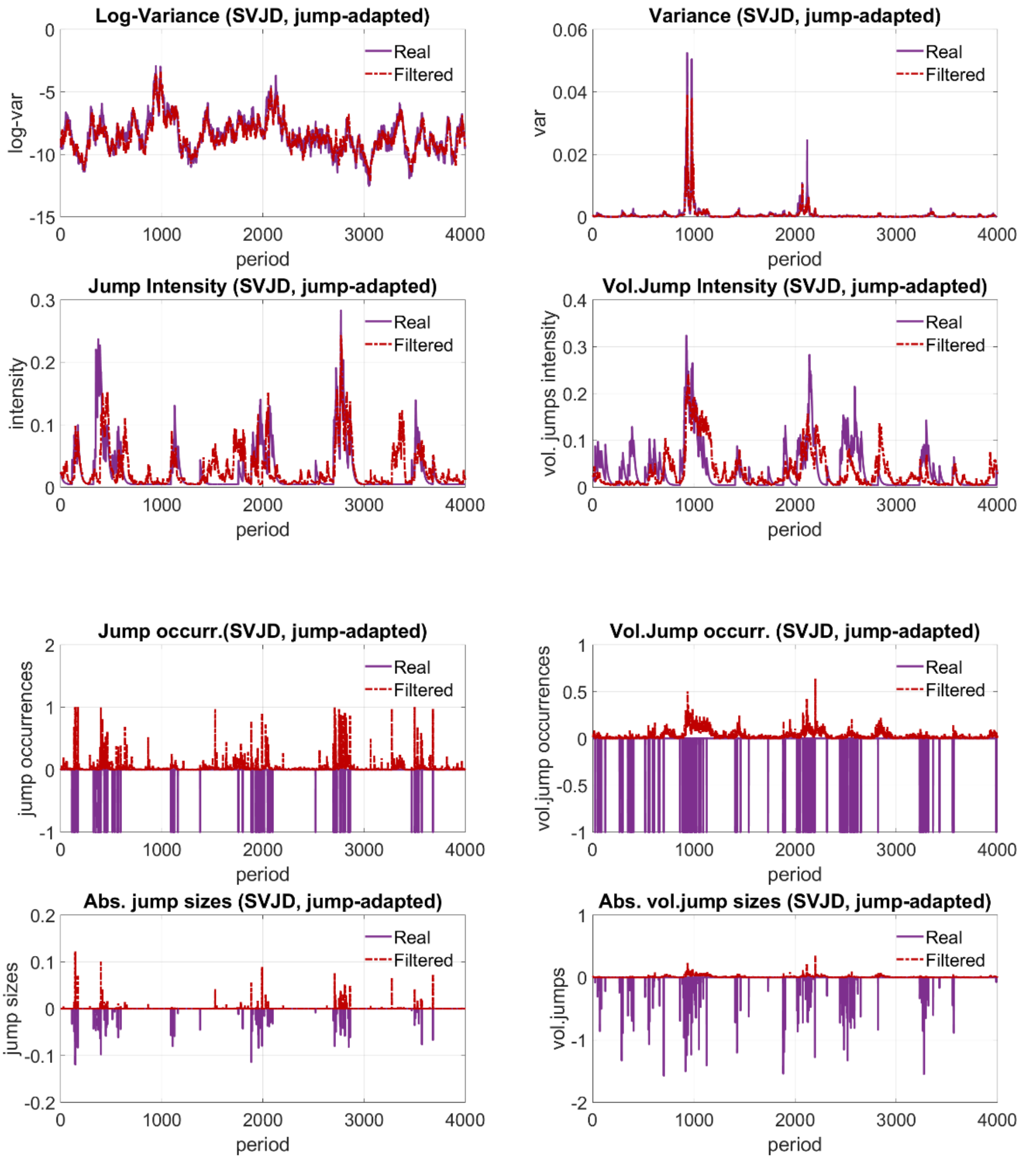
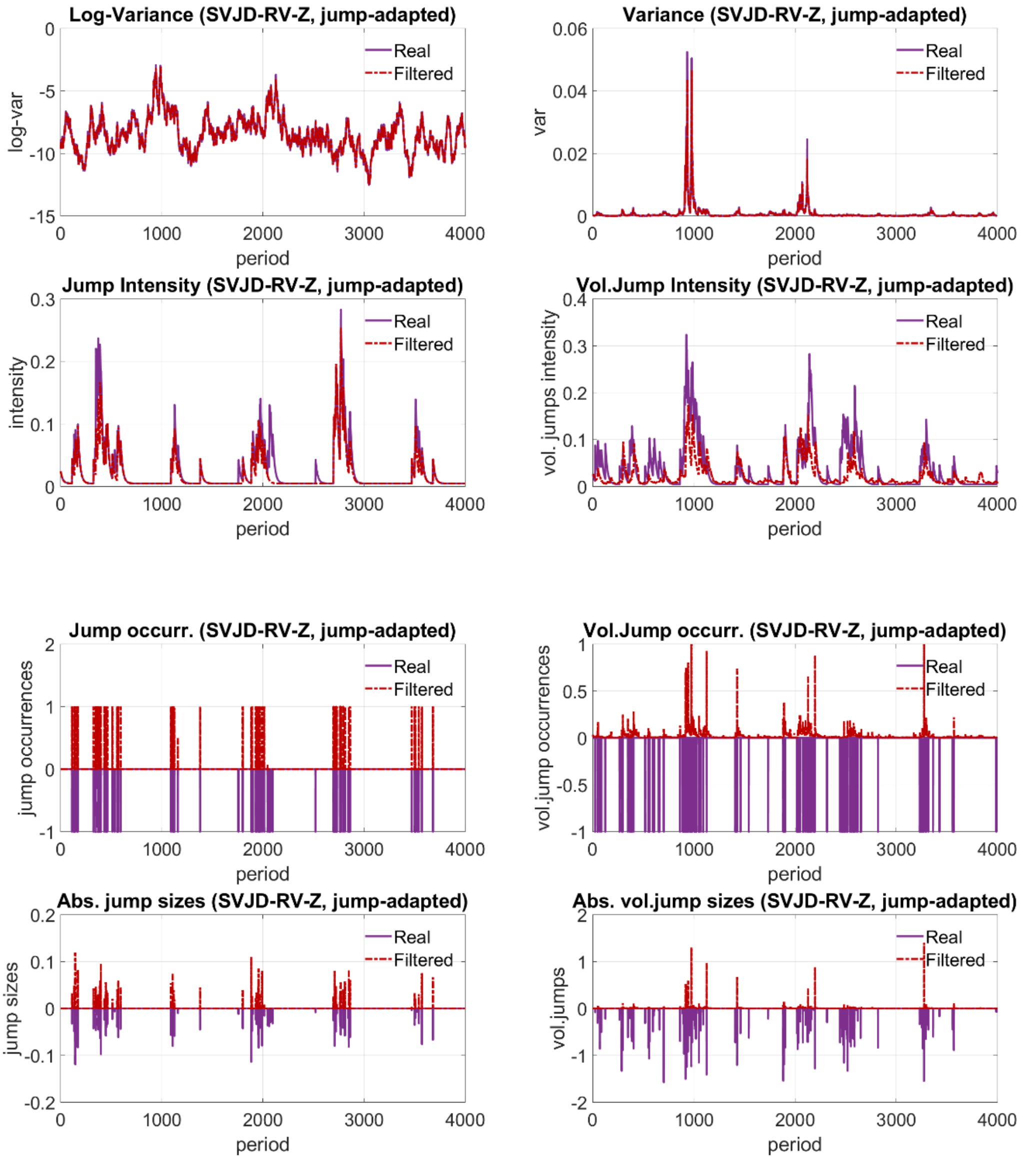


Figure 5 – Simulated vs. filtered time series for the SVJD-RV-Z model with jumps in prices and volatility (filtered with the jump-adapted particle filter)



We can see that while both of the models seem to provide decent filtering results in the case of the stochastic volatility, the jump intensities and the price jumps, the SVJD model was not able to capture most of the volatility jumps, with none of the Bayesian probabilities of occurrence being higher than 0.5-0.6. The SVJD-RV-Z model, on the other hand, was able estimate at least some of the volatility jumps, and its performance with regards to the price jumps is almost perfect. This is caused by the fact that the Z-Estimator and the realized variance provide powerful information to the particle filter about what actually occurred when a large daily return is observed, and whether it was caused by a price jump (manifesting in a large value of the Z-Estimator), or a volatility jump (corresponding to an increase of the realized variance with no significant increase of the Z-Estimator).

As in the previous chapter, 500 simulations of the SVJD process were generated, and the three models/filters were applied to each of them. The values of the R-Squared (for the log-variance, variance, jump intensity and volatility jump intensity) and the Accuracy Ratios (for the return jump and volatility jump occurrences) were then computed for each simulation and each filter. Figure 6 shows the distribution of these performance metrics over the 500 simulation runs, while Table 2 shows their averages.

Figure 6 – R-Squared and Accuracy Ratio distributions of the filtered estimates of the adapted and un-adapted particle filters over 500 simulations (SVJD and SVJD-RV-Z models with self-exciting jumps in volatility and returns)

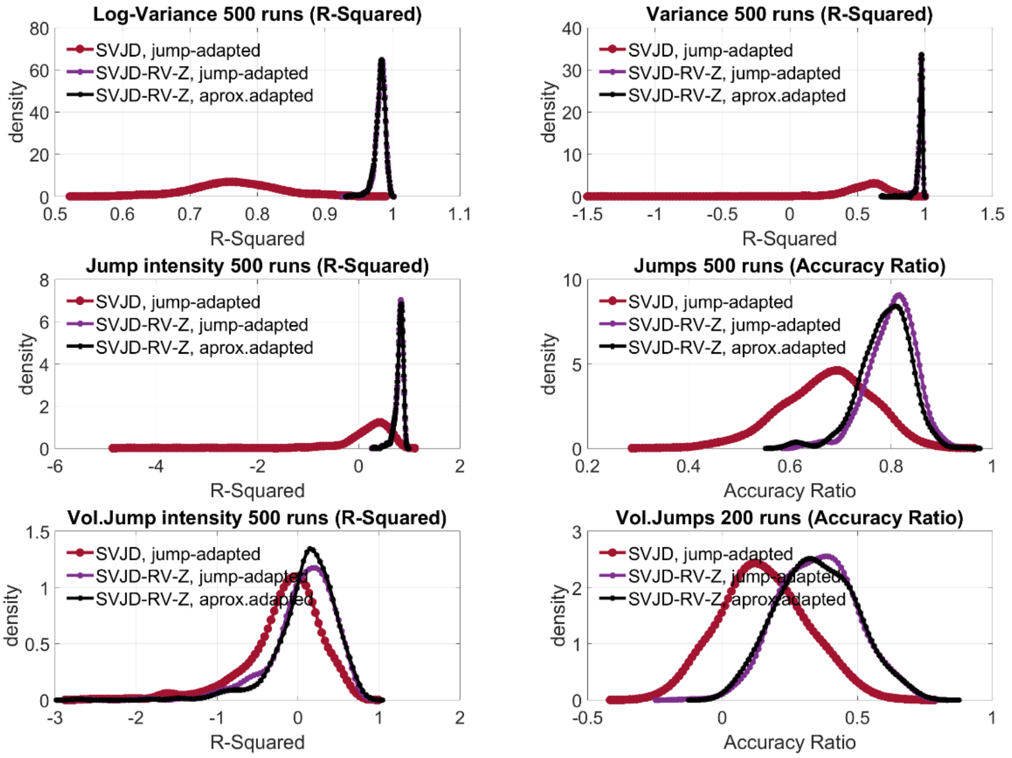


Table 2 – Average performance metrics achieved by different particle filters in 500 simulation runs of the SJVD model with self-exciting jumps in volatility and returns

	SVJD, jump-adapted	SVJD-RV-Z, jump-adapted	SVJD-RV-Z, approx.adapt.
log-variance R2	0.7672	0.9821	0.9817
variance R2	0.5490	0.9661	0.9660
jump intensity R2	0.1395	0.8048	0.8063
jump occurrence AR	0.6745	0.8014	0.7885
vol.jump intensity R2	-0.1648	0.0584	0.0996
vol.jump occurrence AR	0.1531	0.3529	0.3503

It is apparent from Figure 6 and Table 2 that the filtering estimates of the latent states in a SVJD-RV-Z model are significantly more accurate than in the SVJD model. This is to be expected as the high-frequency estimators provide powerful additional information about the underlying volatility and jumps that the standard SVJD model lacks. At the same time, it is necessary to mention that the performed simulation exercise is somewhat idealized, and in practical settings the power-variation estimators used in the SVJD-RV-Z model may be plagued by microstructure noise and other imperfections. The real differences in filtering accuracy therefore might be slightly lower.

Another interesting result of the study is that the approximately fully adapted particle filter was not able to significantly outperform the simple jump-adapted particle filter for the SVJD-RV-Z model. This indicates that the approximations that we used to derive the approximately fully adapted proposal densities (i.e. ignoring some of the information during the derivation of the proposal densities, in order to make them tractable) were probably too strong, and the derived proposal densities are thus no better than the ones derived exactly for the jump-adapted particle filter.

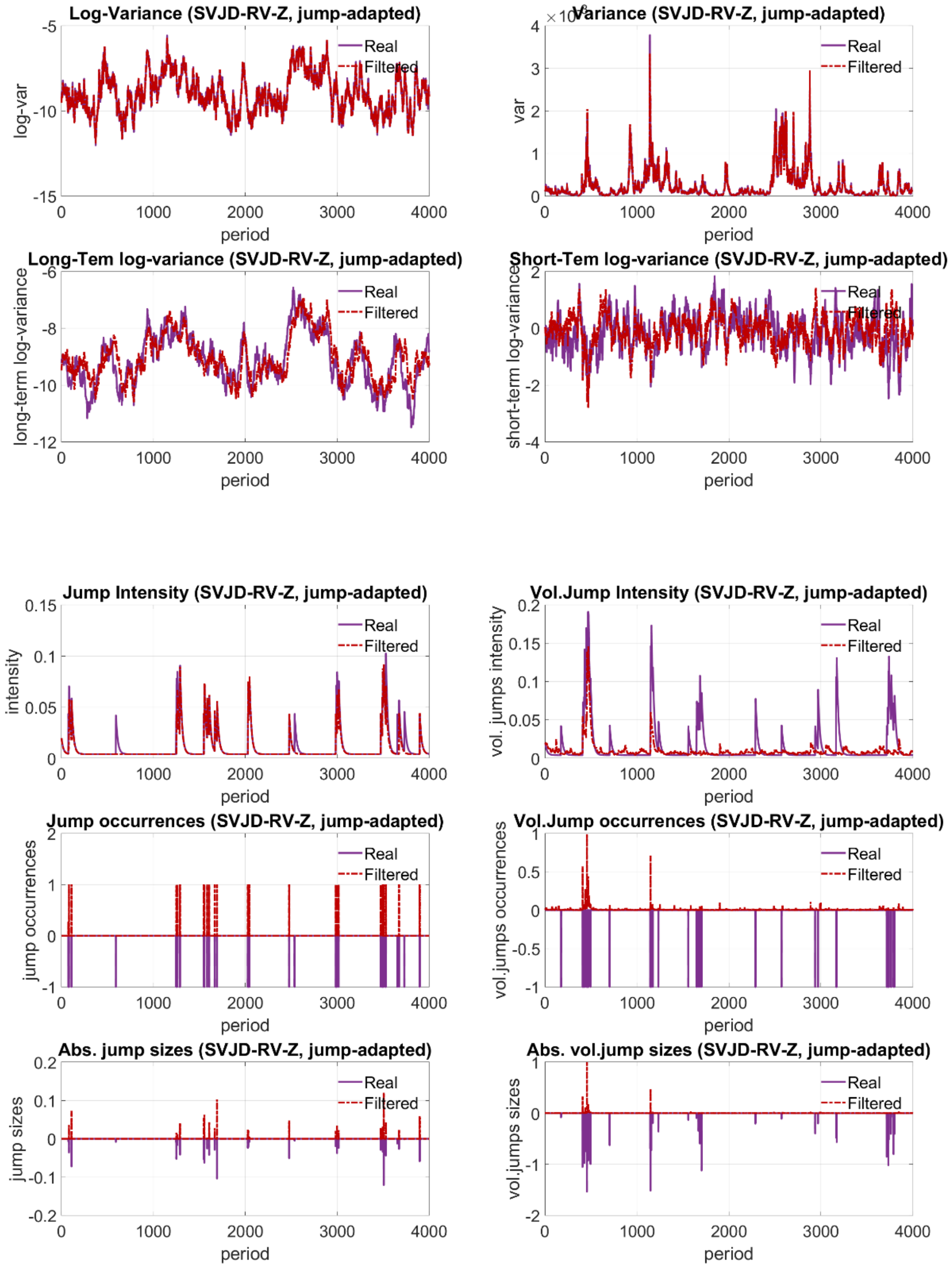
4.1.3. Filtering in a 2-Component SVJD-RV-Z model

The most complex filtering task pursued in this thesis is the filtering of the latent states in a 2-component SVJD model with self-exciting jumps in returns and volatility. Due to the difficulty of the task, we will perform the simulation study only on the SVJD-RV-Z model, and compare the performance of the jump-adapted particle filter with the approximately fully adapted particle filter.

Parameters used for the simulation are as follows: $\mu = \frac{0.05}{252}$, $\mu_J = -0.01$, $\sigma_J = 0.04$, $V_{LT} = e^{\frac{\phi_0}{1-\phi_1}} = 0.01^2$, $\phi_1 = 0.995$, $\phi_2 = 0.1$, $\beta = 0.95$, $\gamma = 0.2$, $\lambda_{LT} = \frac{\alpha_J}{1-\beta_J-\gamma_J} = 0.02$, $\beta_J = 0.95$, $\gamma_J = 0.04$, $\mu_{JV} = 0.3$, $\sigma_J = 0.5$, $\lambda_{LT,V} = \frac{\alpha_{JV}}{1-\beta_{JV}-\gamma_{JV}} = 0.02$, $\beta_{JV} = 0.95$, $\gamma_{JV} = 0.04$. As in the previous chapter, the time series were simulated on a 15-minute frequency (with adjusted parameters) and the values of σ_{RV}, μ_Z, ξ_Z and σ_Z were then calculated based on the values observed ex-post for each simulation.

Figure 7 shows the simulated and the filtered latent state time series for one simulation run for the case of the jump-adapted particle filter (we do not show the results for the approximately fully-adapted filter as they are visually indistinguishable).

Figure 7 – Simulated vs. filtered time series for the 2-Component SVJD-RV-Z model with jumps in prices and volatility (filtered with the jump-adapted particle filter)



We can see from Figure 7 that the filter managed to relatively well disentangle the short-term and the long-term volatility component, although the estimates of the short-term component tend to sometimes differ from the real values. The filter also missed most of the volatility jumps, especially in the second half of the time series.

500 simulations from the model were generated and the two tested particle filters (jump-adapted and approximately fully adapted) were applied to each of them. The distribution of the values of the R-Squared and the Accuracy Ratios (for the price and volatility jump occurrences) can be seen in Figure 8, while Table 3 shows their averages.

Figure 8 – R-Squared and Accuracy Ratio distributions of the filtered estimates of the adapted and un-adapted particle filters over 500 simulations

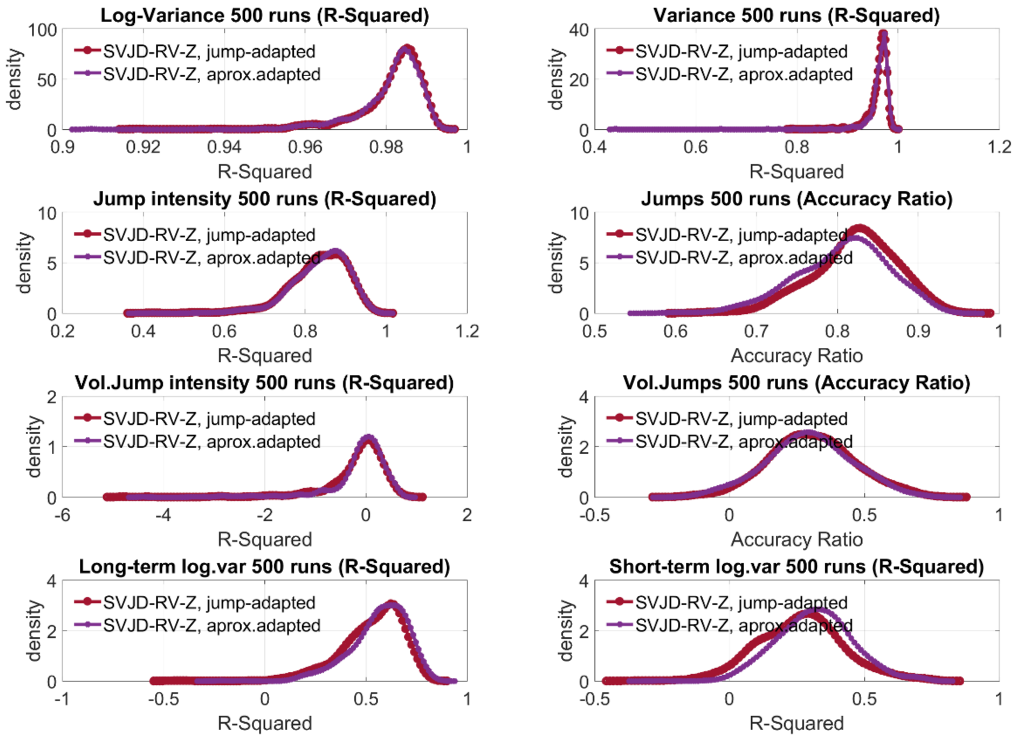


Table 3 – Performance metrics achieved by particle filters in 500 simulation runs of the 2-Component SJVD model with self-exciting price and volatility jumps

	SVJD-RV-Z, jump-adapted	SVJD-RV-Z, approx.adapted
log-variance R2	0.9816	0.9812
variance R2	0.9624	0.9603
jump intensity R2	0.8344	0.8361
jump occurrence AR	0.8227	0.8065
vol.jump intensity R2	-0.1511	-0.1300
vol.jump occurrence AR	0.3061	0.2952
long-term log-var R2	0.5310	0.5639
short-term log-var R2	0.2646	0.3149

We can see that both tested particle filters exhibit significantly positive R-Squared with respect to the short-term and the long-term volatility component, indicating that they were able to disentangle the two components of the volatility process. Unfortunately, the R-Squared of the volatility jump intensity estimates is in both cases negative, which sheds doubt on how accurately can the tested particle filters estimate volatility jumps in 2-component SVJD models.

We can further see that there are no significant differences between the accuracy of the jump-adapted and the approximately fully adapted particle filter. The only meaningful differences in R-Squared values seem to occur with respect to the two volatility components, which the approximately fully adapted filter estimates slightly better.

4.2. Simulation tests of the SGPF algorithm

The filtering task in the previous section was greatly simplified compared to the real-world problem, where the parameters of the underlying SVJD process are unknown and need to be estimated together with the latent states. In the empirical part of the study we will perform the estimation with the Sequential Gibbs Particle Filter (SGPF). In order to evaluate the performance of the SGPF algorithm with respect to parameter estimation and latent state filtering, we will apply to simulated time series.

Due to the large number of charts and limited scope of this thesis, we will discuss the results only for the most challenging case of a 2-Component SVJD-RV-Z model with self-exciting jumps in prices and volatility.

4.2.1. Parameter learning in a 2-Component SVJD-RV-Z model

Time series was simulated with the 2-Component SVJD-RV-Z model with self-exciting jumps in prices and volatility, using the following parameters: $\mu = \frac{0.05}{252}$, $\mu_J = -0.01$, $\sigma_J = 0.04$, $V_{LT} = e^{\frac{\phi_0}{1-\phi_1}} = 0.01^2$, $\phi_1 = 0.995$, $\phi_2 = 0.1$, $\beta = 0.95$, $\gamma = 0.2$, $\lambda_{LT} = \frac{\alpha_J}{1-\beta_J-\gamma_J} = 0.02$, $\beta_J = 0.95$, $\gamma_J = 0.04$, $\mu_{JV} = 0.3$, $\sigma_J = 0.5$, $\lambda_{LT,V} = \frac{\alpha_{JV}}{1-\beta_{JV}-\gamma_{JV}} = 0.02$, $\beta_{JV} = 0.95$, $\gamma_{JV} = 0.04$.

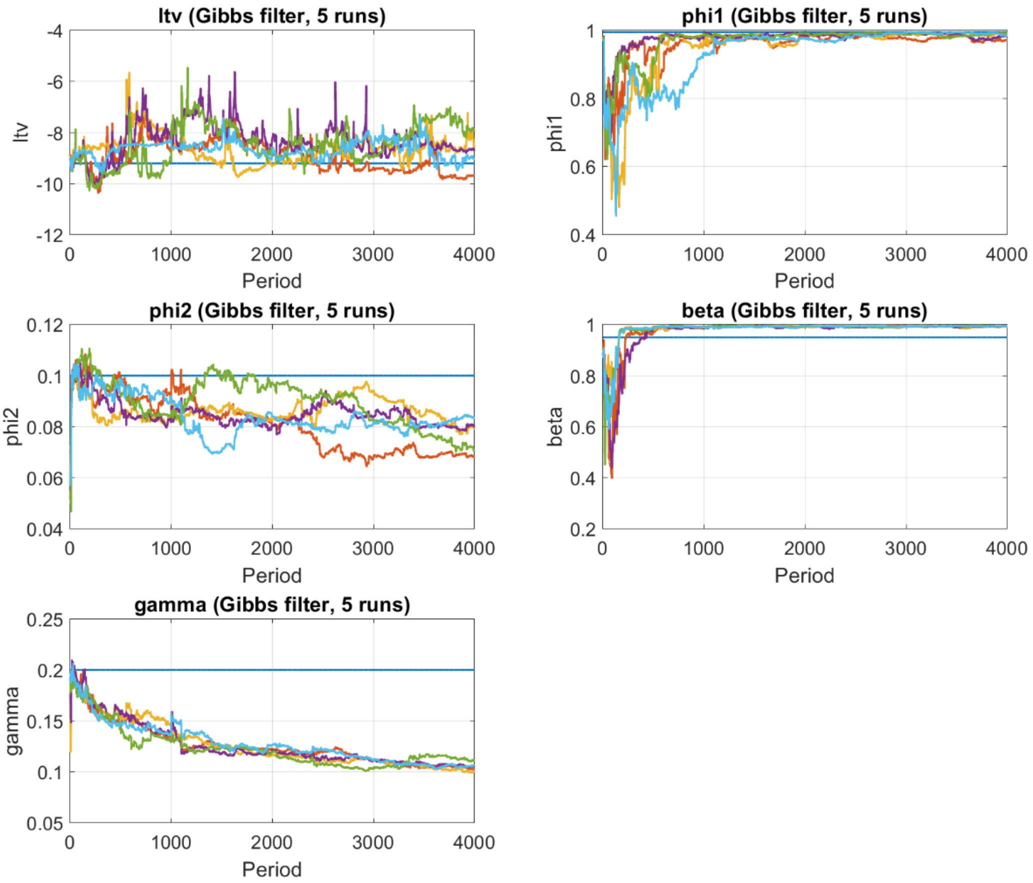
The SGPF algorithm was then run on the simulated time series 5 times in a row, each time with 100 parameter particles (50-particle threshold for resampling), and with 100 latent states particles (with daily re-sampling). The following (relatively wide) priors were used for some of the parameters to improve the convergence of the algorithm:

Table 4 – Priors used for the inverse gamma distributions of the parameters for the 2-Component model used in the simulation test

	a	b
sigmaJ	5	5*9*(0.01^2)
gamma	10	10*(0.2^2)
sigmaJV	5	5*9*(0.2^2)
phi3	10	10*(0.1^2)
sigmaRV	50	50*(0.6)^2

The convergence of the SGPF algorithm parameter estimates through the 4 000 periods of the time series is shown for the parameters related to the stochastic log-variance process on Figure 9. The blue line denotes the values used in the simulation.

Figure 9 – Convergence of the volatility parameters of the 2-Component SVJD-RV-Z model during 5 independent runs of the SGPF algorithm (simulated series)

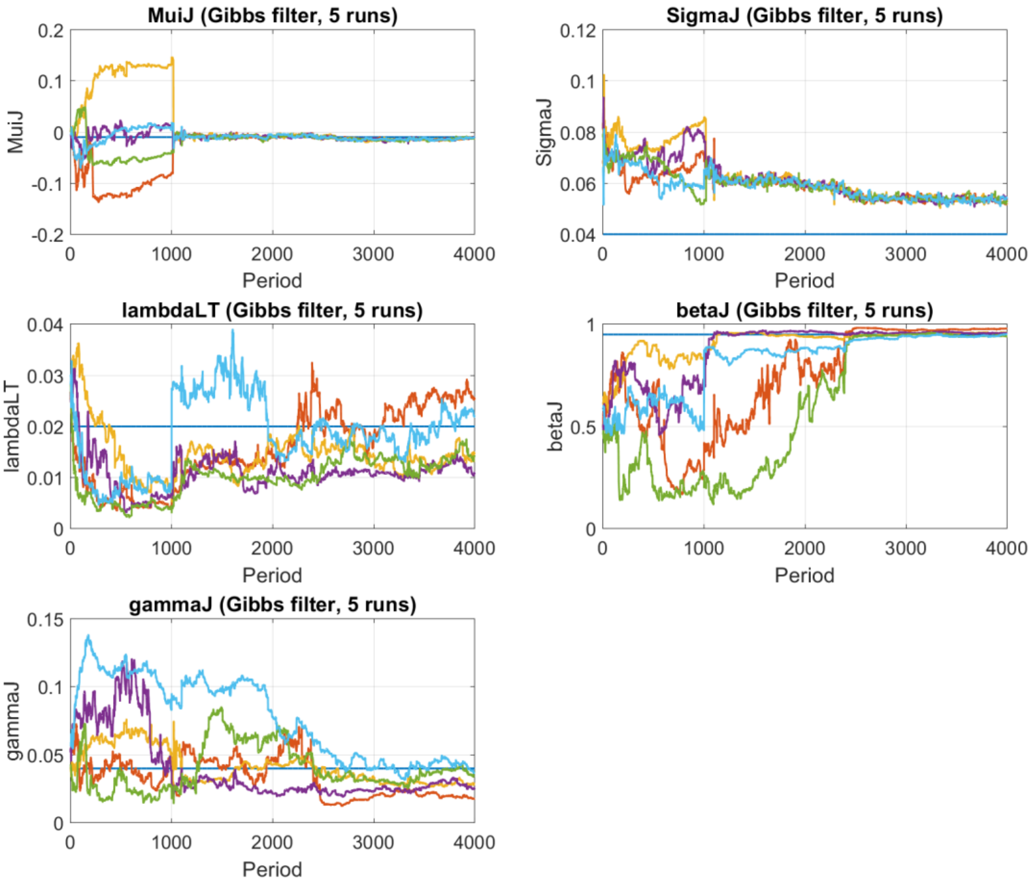


We can see from Figure 9 that the parameters of the long-term log-variance component ($V_{LT} = e^{\frac{\phi_0}{1-\phi_1}}$, ϕ_1 and ϕ_2), converged approximately to the values used in the simulation, although the final values of ϕ_2 are clearly slightly below simulation ones.

The results are worse for the short-term component, as while the β was converged very quickly, it converged to values that are above the ones used to simulate the time series, and also above ϕ_1, thus de-facto making the h_{ST} into the long-term component. At the same time, γ converged unanimously towards lower values that the ones used in the simulation.

It is worth noting that the convergence to the values used in the simulation cannot be expected to be perfect. While certain parameters were used to simulate the time series, the simulation is a realization of a random process and even the best possible estimate of the parameters will thus always be different from the real values. This is especially true for highly complex models in which different components can substitute for each other.

Figure 10 – Convergence of the jump parameters of the 2-Component SVJD-RV-Z model during 5 independent runs of the SGPF algorithm (simulated series)



From Figure 10 we can see that the convergence of the price-jump related parameters was relatively good, and all of the parameters converged either to the values used in the simulation or to values that are close to them (such as in the case of σ_J, which is only slightly above the value of 0.04, which can be caused by the fact that the simulated jumps were just higher than expected). A positive result is that convergence occurred even in the case of the decay parameter β_J, which is notoriously difficult to estimate, due to its high negative correlation with γ_J. This is apparent from some of the runs in which the values of γ_J initially moved towards overly high values, which subsequently caused the β_J to converge upwards slower, than in the runs in which γ_J was initially estimated as low.

Figure 11 – Convergence of the volatility jump parameters of the 2-Component SVJD-RV-Z model during 5 independent runs of the SGPF algorithm (simulated series)

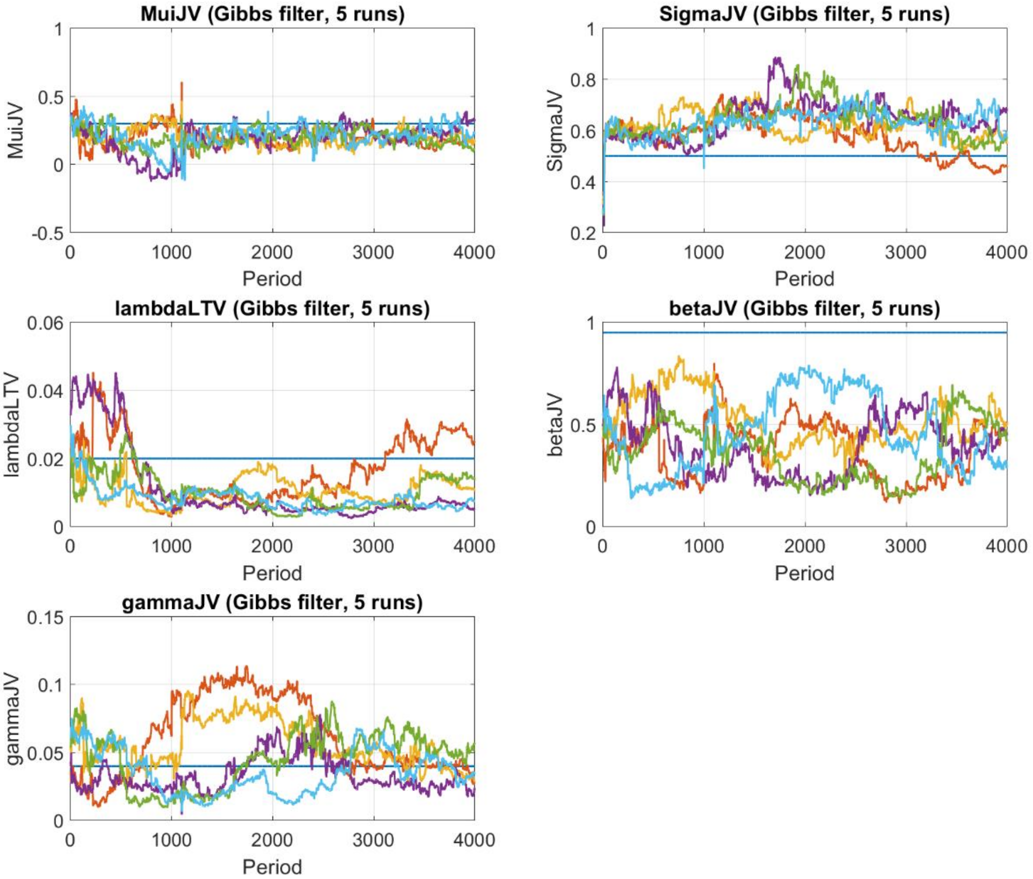


Figure 11 shows the convergence of the parameters related to the volatility jumps. While the estimation of volatility jump dynamics is especially challenging, we can see that most of the parameters still managed to converge towards the correct values, with the notable exception of β_{JV}. The lack of convergence in the case of β_{JV} can probably be explained by the fact that the simulated time series is too short and not enough volatility jump managed to occur in order for the self-exciting behaviour to be estimated properly. This is a common problem when estimating Hawkes process parameters, together with the fact that high correlation between β_{JV} and γ_J often leads the algorithm to converge towards local optima (as discussed in Fičura and Witzany, 2015).

Figure 12 – Convergence of the remaining parameters of the 2-Component SVJD-RV-Z model during 5 independent runs of the SGPF algorithm (simulated series)

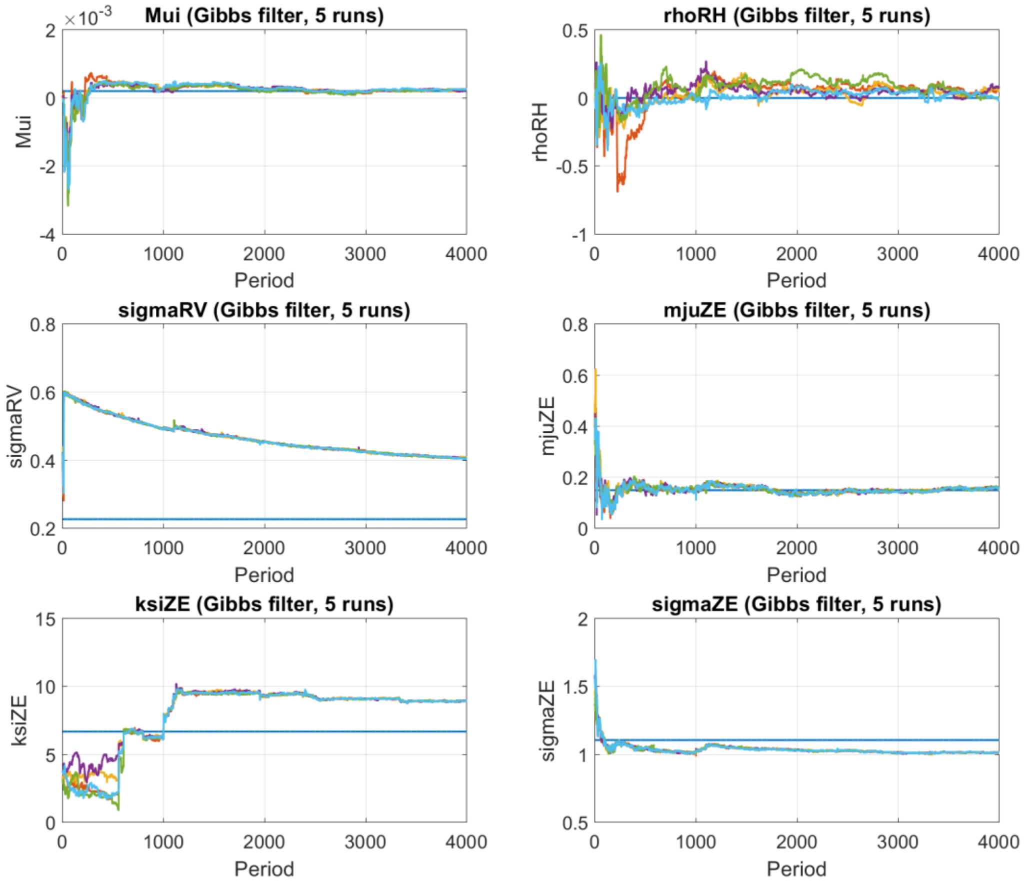


Figure 12 shows the remaining parameters of the 2-Component SVJD-RV-Z model. We can see that most of these parameters converged quickly to the correct values, with the exception of σ_{RV}, ξ_Z and σ_Z, which converged to different values than the ones used in the simulation. In the case of σ_{RV} this was probably caused by the high value of the prior which was put onto this parameter, in order to improve the convergence of some of the other parameters. The overestimation of ξ_Z could be caused by the inability of the algorithm to estimate some of the smaller jumps, due to the high diffusive volatility in the time series.

Table 5 – Final periods SGPF parameter estimates and Bayesian standard errors, compared with values used in the simulation of the 2-Component SVJD-RV-Z model

Model	SimValue	Run1	Run2	Run3	Run4	Run5
mju	**0.0002**	0.0002	0.0002	0.0002	0.0002	0.0003
		4.30E-05	4.64E-05	5.95E-05	4.58E-05	4.50E-05
mjuJ	**-0.01**	-0.0106	-0.0113	-0.0121	-0.0125	-0.0107
		0.0082	0.0063	0.0083	0.0072	0.0097
sigmaJ	**0.04**	0.0517	0.0532	0.0538	0.0531	0.0544
		0.0038	0.0056	0.0047	0.0055	0.0048
ltv	**-9.2103**	-9.6795	-8.4667	-8.6489	-7.8941	-9.0291
		0.1485	0.6250	0.3400	0.4903	0.3554
beta	**0.9500**	0.9946	0.9930	0.9930	0.9923	0.9946
		0.0017	0.0030	0.0034	0.0035	0.0042
gamma	**0.2000**	0.1024	0.1002	0.1052	0.1111	0.1064
		0.0026	0.0038	0.0034	0.0057	0.0051
lambdaLT	**0.0200**	0.0252	0.0146	0.0110	0.0141	0.0219
		0.0054	0.0062	0.0030	0.0046	0.0039
betaJ	**0.9500**	0.9788	0.9589	0.9571	0.9400	0.9515
		0.0025	0.0054	0.0070	0.0098	0.0066
gammaJ	**0.0400**	0.0177	0.0286	0.0258	0.0370	0.0369
		0.0026	0.0056	0.0047	0.0078	0.0065
mjuJV	**0.3000**	0.1649	0.1685	0.3291	0.1109	0.2317
		0.0993	0.1392	0.2087	0.1069	0.1344
sigmaJV	**0.5000**	0.4619	0.5496	0.6811	0.5712	0.5925
		0.0514	0.0805	0.1038	0.0854	0.0972
lambdaLTV	**0.0200**	0.0250	0.0108	0.0049	0.0141	0.0075
		0.0065	0.0023	0.0018	0.0056	0.0024
betaJV	**0.9500**	0.5075	0.5053	0.4514	0.4245	0.2989
		0.2259	0.1152	0.1741	0.1894	0.1584
gammaJV	**0.0400**	0.0272	0.0293	0.0234	0.0567	0.0344
		0.0137	0.0145	0.0154	0.0246	0.0137
phi1	**0.9950**	0.9714	0.9919	0.9837	0.9918	0.9849
		0.0140	0.0058	0.0069	0.0040	0.0036
phi2	**0.1000**	0.0682	0.0794	0.0800	0.0707	0.0833
		0.0028	0.0047	0.0039	0.0050	0.0039
sigmaRV	**0.2266**	0.4069	0.4071	0.4068	0.4066	0.4041
		0.0053	0.0059	0.0039	0.0045	0.0052
mjuZE	**0.1492**	0.1586	0.1536	0.1575	0.1575	0.1590
		0.0130	0.0106	0.0143	0.0184	0.0120
ksiZE	**6.6790**	8.9452	8.9298	8.9296	8.9458	8.9425
		0.1625	0.1289	0.1387	0.1336	0.1302
sigmaZE	**1.1055**	1.0144	1.0096	1.0130	1.0141	1.0148
		0.0122	0.0105	0.0116	0.0147	0.0119
CorrRH	**0.0000**	0.0513	0.0591	0.0727	0.0009	-0.0181
		0.0440	0.0735	0.0642	0.0500	0.0395

4.3. Application of the SGPF algorithm on EUR/USD

In the next step, we will apply the SGPF algorithm for parameter learning on the EUR/USD foreign exchange rate time series. Two tests will be performed. In the first test, the convergence of the SGPF algorithm is compared with the parameter estimates constructed with an MCMC algorithm, on a single-component SVJD-RV-Z model with self-exciting jumps in returns. In the second test, the SGPF algorithm is applied for the estimation of a SVJD-RV-Z model with self-exciting jumps in returns and volatility, in order to see, whether it tends to converge towards similar values when it is run multiple times in a row.

4.3.1. MCMC vs. SGPF estimation of SVJD-RV-Z model on EUR/USD

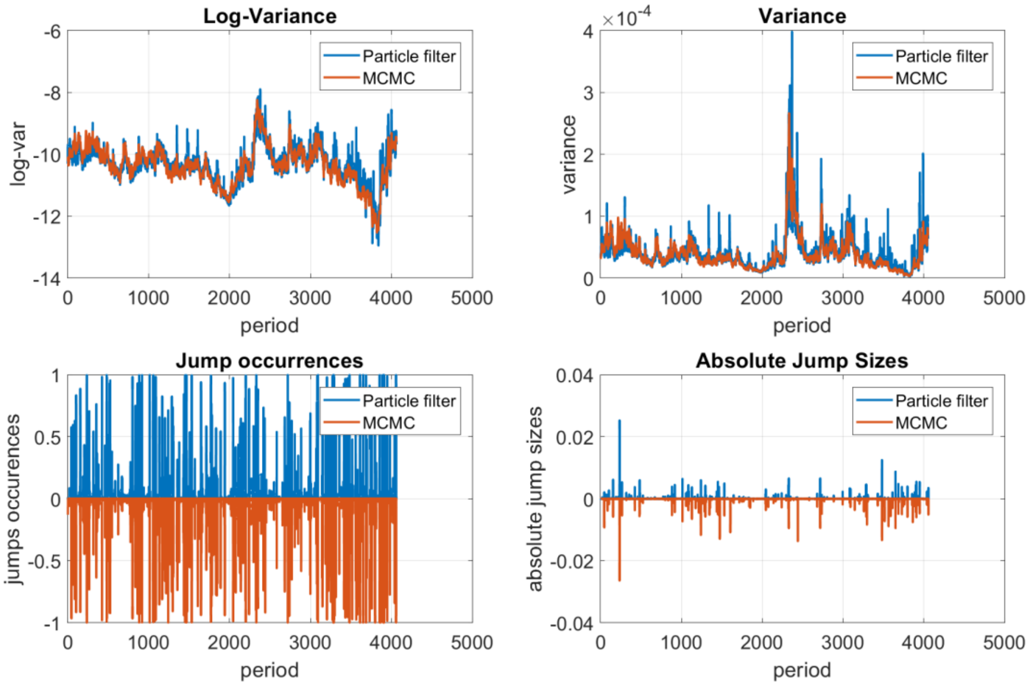
The parameter estimates from the SGPF algorithm will be compared with the MCMC estimates on the SVJD-RV-Z model with self-exciting jumps in prices that was analysed in Fičura and Witzany (2017). The application is performed on the EUR/USD time series in the period from 1.10.1999 to 15.06.2015. The realized variance and the Z-Estimator were computed from 15-minute returns, based on a dataset from ForexHistoryDatabase.com.

The MCMC method was run for 20 000 iterations, with the first 5 000 discarded and the remaining 15 000 ones used for parameter and latent states estimation based on posterior means. The Gibbs Particle Filter was run with 100x100 particles (100 parameter particles, each with 100 latent state particles), daily re-sampling of the latent states, and a 50 particle re-sampling threshold for the parameters.

Contrary to the specification of the other SVJD-RV-Z models in this thesis, we will use μ_{RV} as an additional parameter in this test, quantifying the potential bias of the logarithmic realized variance with respect to the logarithmic stochastic variance. It will be seen that the parameter is estimated as insignificantly different from zero, which is the reason why it is not used in the other SVJD-RV-Z models in this thesis.

Figure 13 compares the filtered log-variances, variances, jump occurrences and absolute jump sizes, estimated with the SGPF particle filter and the MCMC algorithm for the SVJD-RV-Z model with self-exciting jumps in prices on EUR/USD.

Figure 13 – SJVD-RV-Z model filtered latent states from the Gibbs Particle Filter compared with the MCMC latent state estimates



We can from Figure 13 that both methods estimated the jumps in the EUR/USD time series very similarly (jump probabilities of occurrence as well as the jump sizes). Regarding the log-variances and variances, it is apparent that the particle filter estimates are more noisy than the MCMC estimates, which can be attributed to the fact that the MCMC performs smoothing and uses thus all of the information from the time series, while the particle filter utilizes only the information set available at time t. As the MCMC algorithm sees also what happened after time t (i.e. in the periods $t+1$, $t+2$, up to T), it does not tend to produce so many false spikes in volatility in the cases when the future volatility did not actually increase.

Figure 14, Figure 15 and Figure 16 compare the parameter estimates of the MCMC method with the sequential estimates of the SGPF particle filter. Each plot shows the evolution of the Particle Filter estimates through the time (over the 4 063 days of the sample), together with the full sample MCMC parameter estimates, constructed based on the posterior means, together with their 95% Bayesian confidence intervals. In the ideal case, the SGPF parameter estimates at time $T = 4063$, should have converged towards similar values as the MCMC full sample estimates.

Figure 14 – Convergence of SGPF parameter estimates for the SVJD-RV-Z model, compared with the MCMC estimates and their 95% confidence intervals (1)

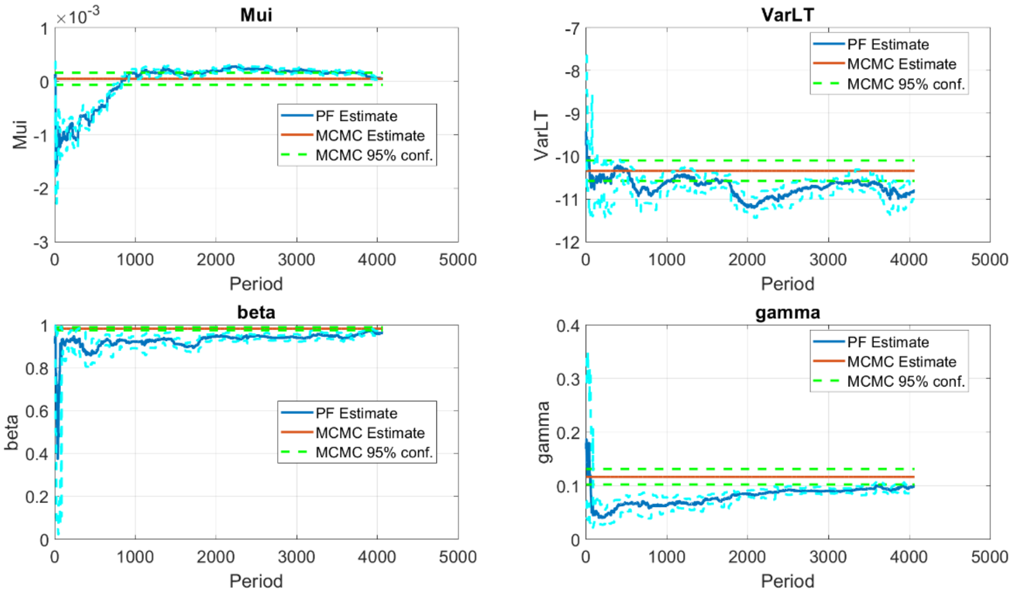


Figure 15 – Convergence of SGPF parameter estimates for the SVJD-RV-Z model, compared with the MCMC estimates and their 95% confidence intervals (2)

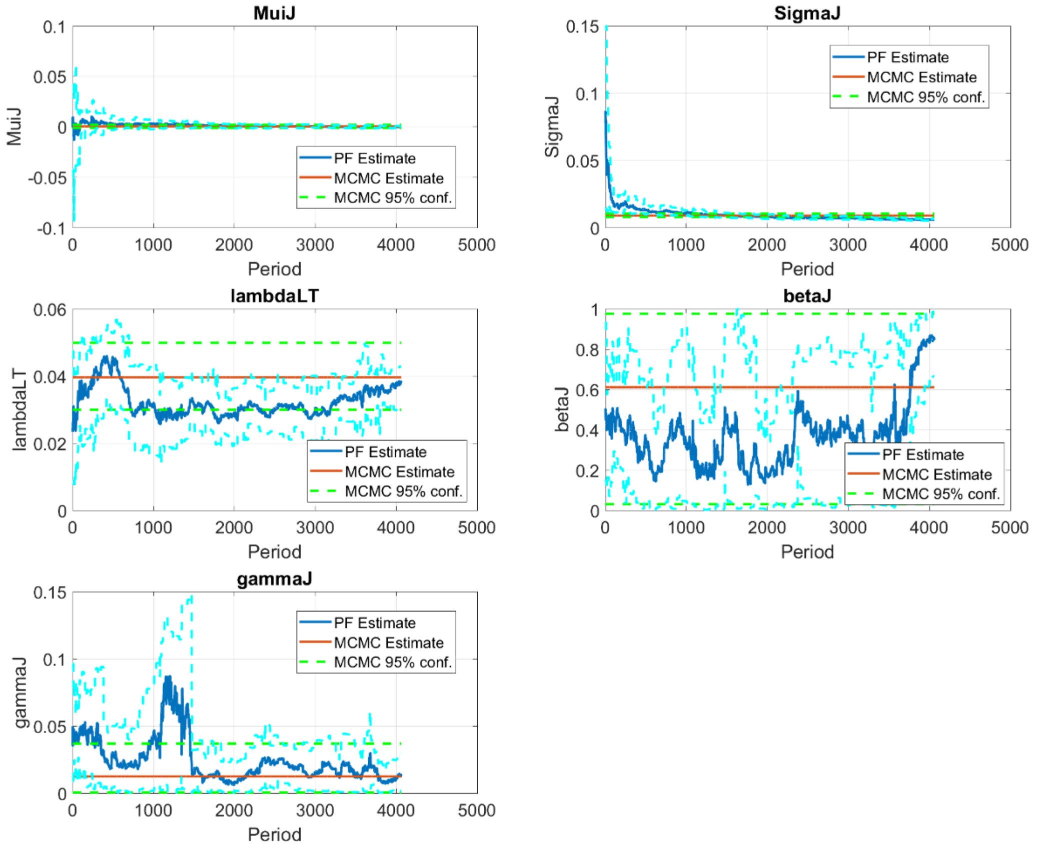
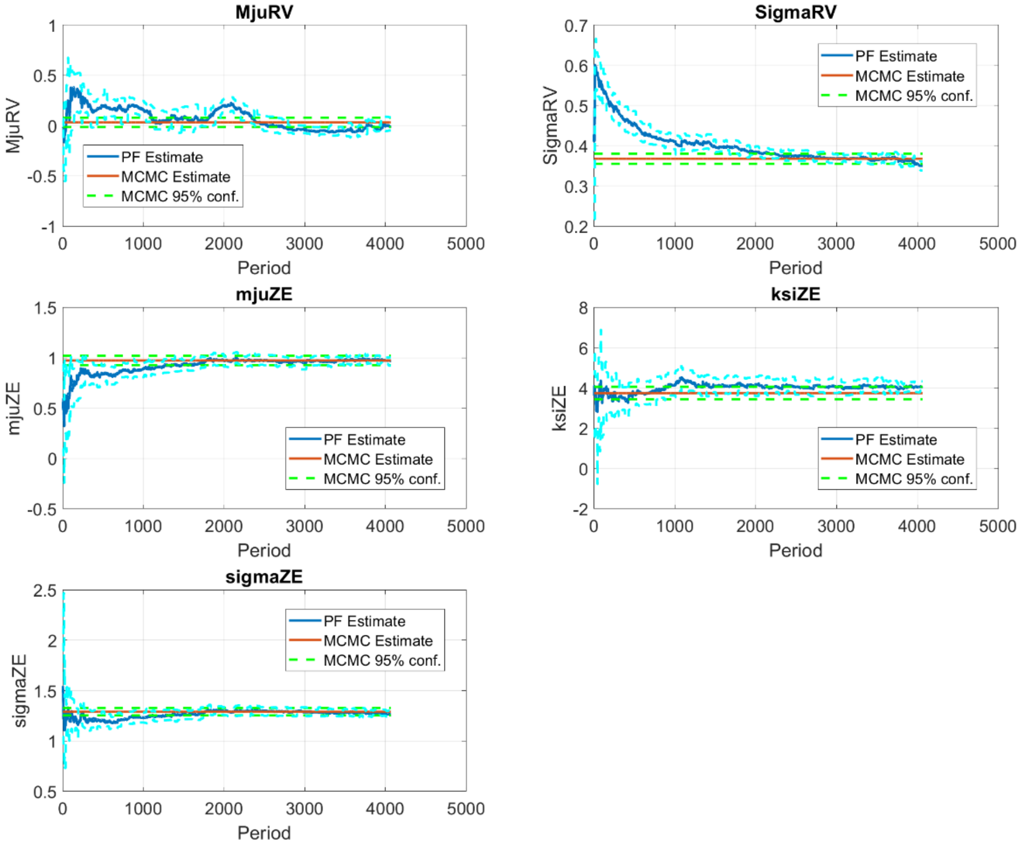


Figure 16 – Convergence of SGPF parameter estimates for the SVJD-RV-Z model, compared with the MCMC estimates and their 95% confidence intervals (3)



We can see that the SGPF sequential parameter estimates converged accurately to the MCMC parameter estimates for most of the parameters of the SVJD-RV-Z model with self-exciting jumps in returns.

Summary of the parameter estimates can be seen in Table 6. We can see that for almost all of the parameters do the MCMC 95% confidence intervals and the SGPF 95% confidence intervals (at $T = 4063$) intersect.

Table 6 – MCMC and SGPF final period parameter estimates for the SVJD-RV-Z model with self-exciting price jumps on EUR/USD

Model	SGPF(5%)	SGPF	SGPF(95%)	MCMC(5%)	MCMC	MCMC(95%)	Intersect
mju	0.0000	0.0001	0.0001	-0.0001	0.0000	0.0002	TRUE
mjuJ	-0.0012	-0.0004	0.0005	-0.0011	0.0004	0.0020	TRUE
sigmaJ	0.0052	0.0057	0.0065	0.0078	0.0089	0.0103	FALSE
ltv	-10.9986	-10.8058	-10.6700	-10.5769	-10.3417	-10.0976	FALSE
beta	0.9511	0.9629	0.9706	0.9764	0.9835	0.9901	FALSE
gamma	0.0853	0.0987	0.1052	0.1024	0.1166	0.1312	TRUE
lambdaLT	0.0298	0.0380	0.0428	0.0300	0.0396	0.0499	TRUE
betaJ	0.6642	0.8542	0.9843	0.0304	0.6108	0.9751	TRUE
gammaJ	0.0016	0.0132	0.0299	0.0005	0.0124	0.0368	TRUE
mjuRV	-0.0488	-0.0073	0.0825	-0.0151	0.0316	0.0790	TRUE
sigmaRV	0.3387	0.3501	0.3608	0.3550	0.3677	0.3800	TRUE
mjuZE	0.9218	0.9711	1.0029	0.9265	0.9726	1.0181	TRUE
ksiZE	3.8835	4.0600	4.3277	3.4411	3.7405	4.0580	TRUE
sigmaZE	1.2311	1.2616	1.3029	1.2528	1.2896	1.3268	TRUE

4.3.2. 2-Component SVJD-RV-Z model parameter learning on EUR/USD

In order to assess the convergence properties of the SGPF algorithm in the case of more complex SVJD models, we will run it 5 times for the 2-component SVJD model with self-exciting jumps in prices and volatility on the EUR/USD time series and observe whether different runs of the algorithm converge to the same values. The algorithm is again run with 100 parameter particles, 50-particle threshold for parameter resampling, and 100 latent state particles (re-sampled daily). A jump-adapted filter is used for the filtering of the latent states.

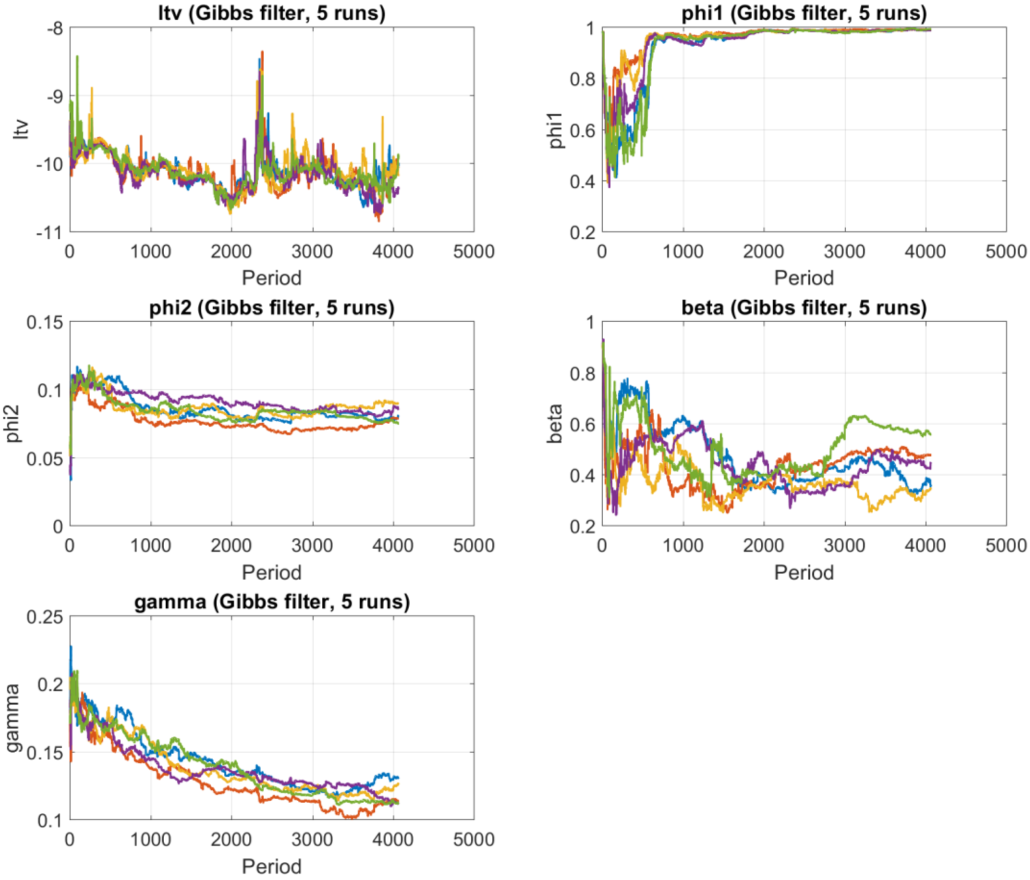
In order to improve the convergence of the algorithm, we have put priors on the values of some of the parameters. Table 7 shows the utilized priors, which were all put on the inverse-gamma distributions of the variability related parameters.

Table 7 - Priors used for the inverse gamma distributions of the parameters for the 2-Component model used in convergence test on EUR/USD

	a	b
sigmaJ	5	5*9*(0.01^2)
gamma	10	10*(0.2^2)
sigmaJV	5	5*9*(0.2^2)
phi3	10	10*(0.1^2)
sigmaRV	50	50*(0.6)^2

Figure 17 shows the convergence of the algorithm for the parameters related to the two components of the log-variance process.

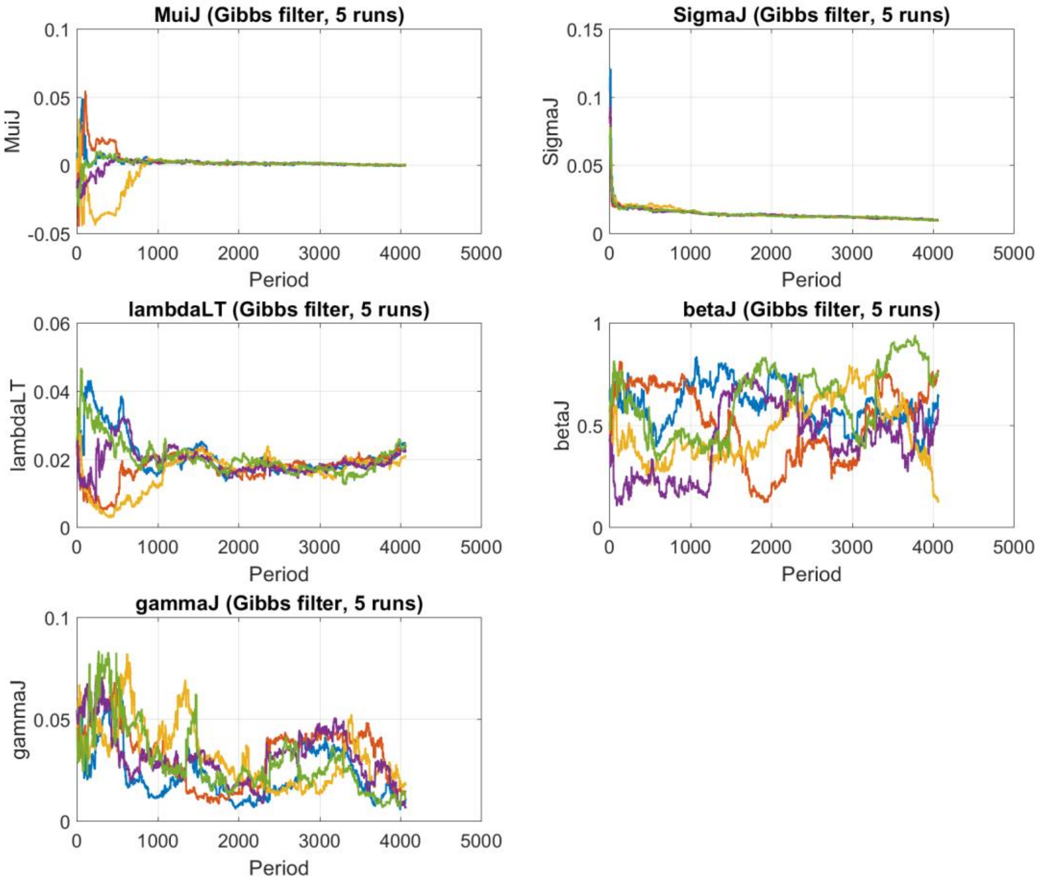
Figure 17 – Convergence of the volatility parameters of the 2-Component SVJD-RV-Z model during 5 independent runs of the SGPF algorithm (EUR/USD)



We can see from Figure 17 that the algorithm converged towards similar parameter values in all 5 runs of the algorithm. The value of ϕ_1 corresponds to a very high persistence of the long-term log-variance component, and it is also much higher than β, so the persistence relationship between the two components are in accordance with our expectations.

Figure 18 shows the convergence results for the parameters related to the jumps in returns.

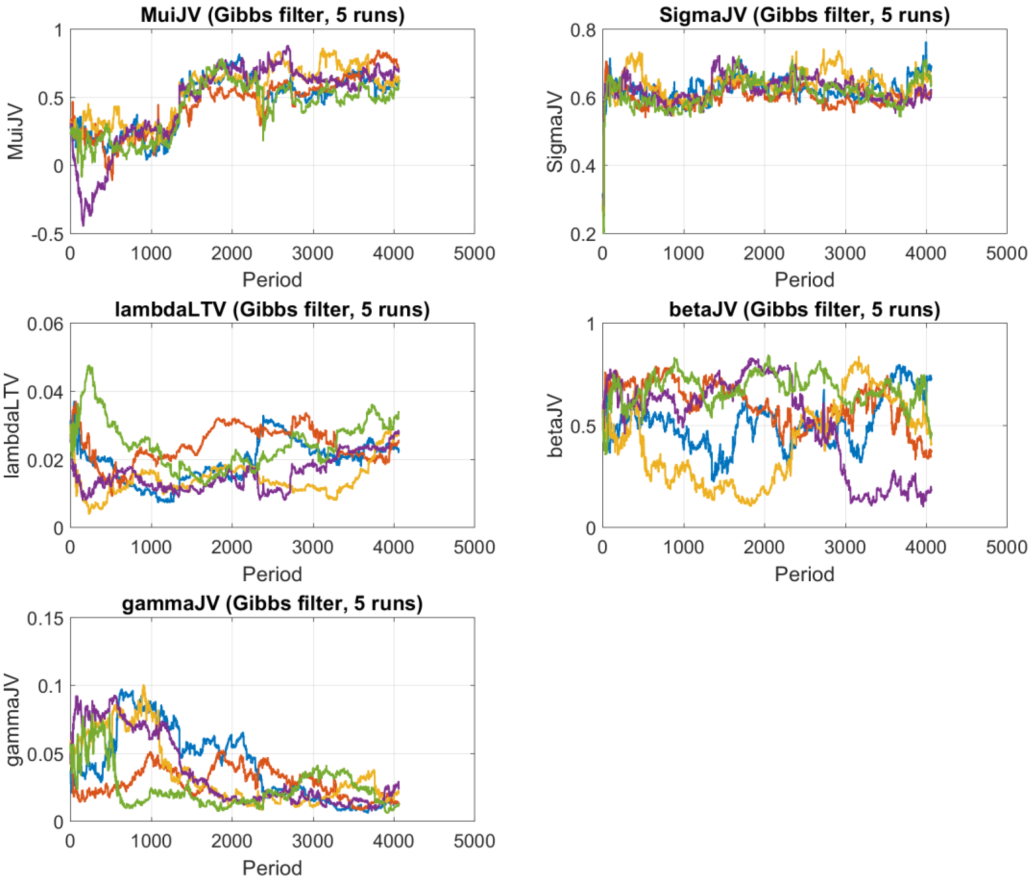
Figure 18 – Convergence of the jump parameters of the 2-Component SVJD-RV-Z model during 5 independent runs of the SGPF algorithm (EUR/USD)



We can see from Figure 18 that while for μ_J, σ_J, λ_{LT} and γ_J, the SGPF algorithm converged quickly towards similar values in all of the runs, for the Hawkes process intensity decay parameter β_J it did not manage to converge towards similar values. This can be either due to low levels of jump intensity persistence in the analysed time series, or due the problems in convergence caused by a high correlation between γ_J and β_J. As will be shown later in Table 8, the values of γ_J are for most of the runs only 1-1.5 standard errors away from zero, indicating that the self-exciting effects for the EUR/USD may not be statistically significant.

Figure 19 shows the convergence results for the volatility-jump related parameters.

Figure 19 – Convergence of the volatility jump parameters of the 2-Component SVJD-RV-Z model during 5 independent runs of the SGPF algorithm (EUR/USD)



We can see from Figure 19, that while the μ_{JV}, σ_{JV}, $\lambda_{LT,V}$ and γ_{JV} managed to again converge towards similar values in all of the runs, the intensity decay parameter β_{JV} was again more problematic, with one of the runs ending at a far lower value than the others. This can again be caused by a relatively low statistical significance of the self-exciting behaviour of volatility jumps in EUR/USD.

Figure 20 shows the convergence of the remaining parameters of the 2-Component SVJD-RV-Z model. We can see that for all of these parameters the SGPF algorithm quickly converged towards similar values in all of the runs. The spike observed in one of the runs on σ_{JV} corresponds to the crisis period and was probably caused by a drop in efficient number of particles due to extreme events on the market, which was, however, quickly reversed in subsequent re-sampling of the particles.

Figure 20 – Convergence of the remaining parameters of the 2-Component SVJD-RV-Z model during 5 independent runs of the SGPF algorithm (EUR/USD)

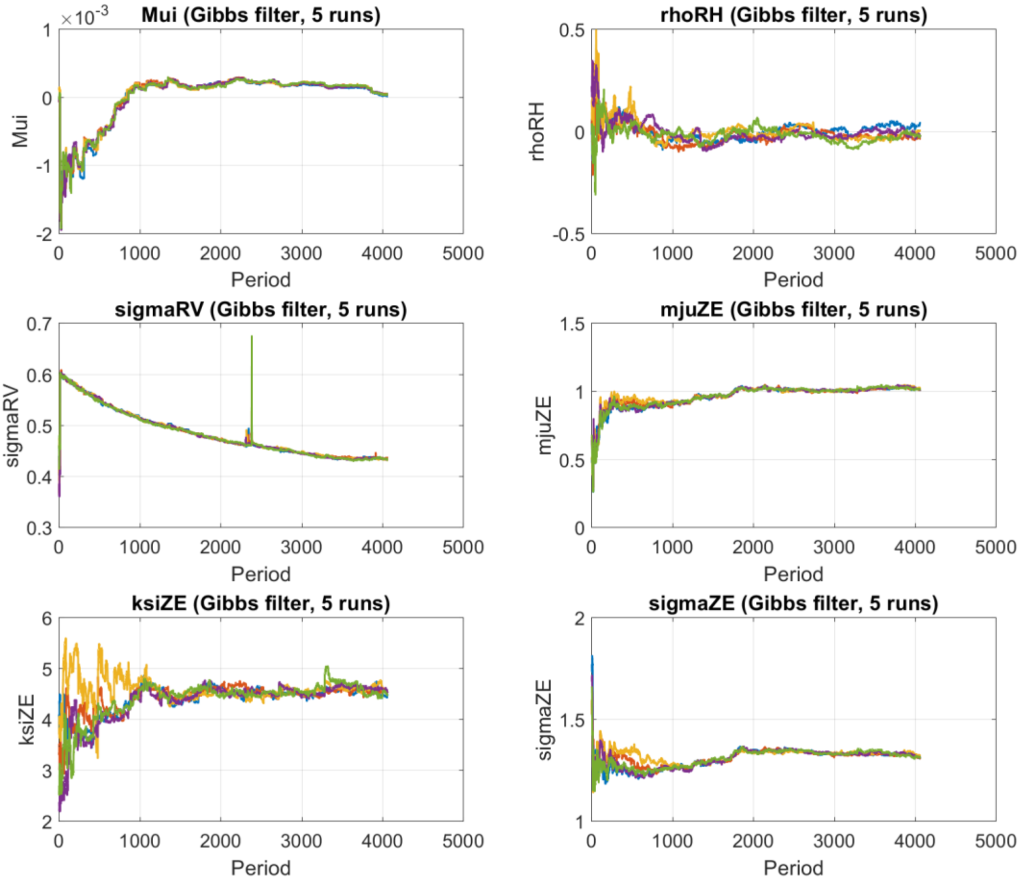


Table 8 shows the parameter estimates from the SGPF particle in the final period ($T = 4063$), together with the Bayesian standard errors, constructed by re-sampling the parameter particles in the final period and computing the standard deviation of their values.

Most parameters are in accordance with out expectation, with the exception of γ_J and γ_{JV}, which are for most of the runs less than 2 standard deviations away from zero, indicating that the self-exciting behaviour of price and volatility jumps on the EUR/USD time series is only weak.

Table 8 – Final periods SGPF parameter estimates and Bayesian standard errors, compared with values used in the simulation of the 2-Component SVJD-RV-Z model

Model	Run1	Run2	Run3	Run4	Run5
mju	0.0000	0.0001	0.0000	0.0000	0.0001
	2.15E-05	1.68E-05	2.32E-05	2.71E-05	2.16E-05
mjuJ	-0.0003	0.0004	-0.0001	0.0001	-0.0001
	0.0010	0.0011	0.0011	0.0010	0.0015
sigmaJ	0.0097	0.0098	0.0100	0.0098	0.0097
	0.0008	0.0010	0.0007	0.0007	0.0009
ltv	-10.0398	-10.0438	-10.1843	-10.3766	-9.9420
	0.5654	0.5605	0.4817	0.3764	0.7709
beta	0.3583	0.4776	0.3439	0.4491	0.5593
	0.0594	0.0186	0.0461	0.0503	0.0344
gamma	0.1312	0.1143	0.1261	0.1117	0.1121
	0.0074	0.0039	0.0088	0.0049	0.0045
lambdaLT	0.0238	0.0241	0.0208	0.0221	0.0235
	0.0033	0.0039	0.0030	0.0034	0.0033
betaJ	0.6360	0.7611	0.1356	0.5818	0.7439
	0.2353	0.1597	0.1164	0.2909	0.1292
gammaJ	0.0108	0.0144	0.0187	0.0067	0.0111
	0.0105	0.0086	0.0122	0.0062	0.0062
mjuJV	0.6309	0.7124	0.6538	0.6844	0.6062
	0.1681	0.1207	0.1948	0.1298	0.1201
sigmaJV	0.6848	0.6114	0.6431	0.6240	0.6351
	0.0673	0.0730	0.1207	0.0748	0.0560
lambdaLTV	0.0227	0.0254	0.0280	0.0277	0.0338
	0.0048	0.0035	0.0047	0.0044	0.0047
betaJV	0.7256	0.3726	0.4056	0.1829	0.4681
	0.1798	0.2489	0.2124	0.1371	0.3231
gammaJV	0.0133	0.0133	0.0209	0.0245	0.0128
	0.0138	0.0125	0.0140	0.0247	0.0083
phi1	0.9909	0.9909	0.9895	0.9878	0.9934
	0.0037	0.0046	0.0043	0.0049	0.0044
phi2	0.0806	0.0791	0.0896	0.0851	0.0753
	0.0062	0.0032	0.0024	0.0065	0.0030
sigmaRV	0.4345	0.4360	0.4318	0.4336	0.4325
	0.0039	0.0048	0.0040	0.0057	0.0036
mjuZE	1.0156	1.0116	1.0252	1.0117	1.0156
	0.0283	0.0214	0.0201	0.0214	0.0215
ksiZE	4.4919	4.5438	4.5501	4.5452	4.4862
	0.2367	0.2201	0.2158	0.1402	0.1643
sigmaZE	1.3120	1.3107	1.3201	1.3164	1.3169
	0.0212	0.0200	0.0207	0.0196	0.0149
CorrRH	0.0401	-0.0327	0.0015	-0.0195	-0.0174
	0.0379	0.0323	0.0343	0.0527	0.0494

5. Empirical test of the predictive power of SVJD models

In this final section we will apply the proposed SVJD and SVJD-RV-Z models, estimated with the SGPF particle filter, for the task of realized volatility forecasting, on the time series of 7 currency exchange rates and 10 ETF/ETN securities. The performance of the models will be compared with benchmark models from the GARCH and HAR model classes, as well as with Echo State Neural Network models.

5.1. Applied benchmark models

The models used as benchmark include three most commonly used ARCH/GARCH models, specifically the GARCH(1,1), EGARCH(1,1) and GJR-GARCH(1,1) model, and a wide variety of most commonly used HAR model specifications.

The HAR model variants that were tested include the standard HAR model, HARQ model (using realized quarticity as an additional predictor), HARJ model (using the bi-power variation based estimator of jumps), AHAR model (using daily returns) and SHAR model (using the realized semi-variance). In preliminary tests, the HAR models were estimated in their simple form, with only the last 1-day values of the additional predictors, as well as in the full HAR specification, using the aggregated values over the last 1-Day, 1-Week (5 days) and 1-Month (22 days) for all predictors. As the full HAR specifications achieved on average better results, they were chosen as the ones to be applied in the thesis.

All of the HAR models were additionally estimated in their standard, logarithmic and square root form, as these do often perform better than standard HAR models (especially the logarithmic form). The logarithmic and square root forms of the HAR models will be denoted as logHAR and sqrtHAR in the tables (and analogically for the extended versions of the HAR model, i.e. for example logHARQ and sqrtHARQ for the HARQ model).

In order to assure that the results of the benchmark models are comparable with the sequential estimates and predictions of the SVJD models, a walk-forward estimation with a 22-day moving window is used. All models were initially estimated on the first 1 000 days of the time series (in-sample), and then re-estimated after each 22 days (on the full dataset available at that time-point). The performed estimates are thus always up to date and should be comparable with the way of how the SGPF algorithm works when estimating the SVJD models.

5.2. Applied Echo State Neural Network models

Echo State Neural Network (ESN) model was chosen as an additional approach to be tested with the goal of validating its predictive performance reported in Fičura (2017). Unlike the version of the model applied in Fičura (2017), the networks were applied to differentiated time-series of the realized variance (and the other predictors). The reason for this modification is that when applied to the levels of the realized variance, the models tended to underestimate the volatility during the crisis, due to the poor ability of the neural networks to approximate beyond the value range that they have observed during training.

In addition to the standard ESN model, using only the past realized variance from day $t-1$ as a predictor, versions of the model enhanced with additional high-frequency estimators were tested. The additional predictors are analogical to the predictors used in the different versions of the HAR model (i.e. HARQ, HARJ and SHAR), with the difference that in the ESN models, only the predictors from $t-1$ are used, as the ESN network should be able to learn the long-range dependencies by itself, due to its recurrent nature. The second difference with respect to the HAR models is that all predictors of the ESN models are differentiated (as well as the target variable).

Specifically, we will test the ESNQ model, using in addition to the realized variance also the realized quarticity, the ESNJ model, using the bi-power variation based estimates of the jump variance, and the SESN model, using the realized semi-variance as an additional predictor. Further versions, with multiple additional predictors were tested as well, but they generally did not perform better than the versions mentioned above. An exception was the SESNQ model, utilizing the realized semi-variance and realized quarticity together. The model will thus be included in the final testing as well.

Similarly as in the case of the HAR models, all ESN models were formulated in the standard, logarithmic and the square root form, in which all predictors, as well as the target variable, are subject to the given transformation. The resulting models are denoted as logESN and sqrtESN (and analogically for the extended versions).

As in the case of the other models, the Echo State Neural Networks were estimated with a walk-forward estimation approach, with a moving window of 22 days, in order to assure that the results are comparable with the SGPF sequential estimates of the SVJD models where the parameter estimates get constantly updated.

Regarding the meta-parameters, the expertly chosen values, based on the recommendation in ESN literature were used (Lukoševičius, 2012). All ESN models were trained with 100 neurons in the recurrent layer (reservoir), 100% connectivity, logistic activation function, spectral radius equal to 1, input scaling equal to 1 (regarding to the normalized values of the predictors), and penalization of the ridge regression (used for training of the output layer) equal to 1. Additionally, to improve the long-memory properties of the neural networks, smoothing of the recurrent layer was performed with a vector α, containing 100 equally spaced values ranging from 0.05 to 1, each corresponding to one of the neurons (i.e. each neuron will use different level of smoothing). This is done in order to enhance the ability of the neural network to learn dependencies that occur at different frequencies, and to learn the ones that are most informative with regards to the future evolution of the realized variance.

The training was performed in a standard way. In the first step, the explanatory variables were standardized by subtracting the mean and dividing them with the standard deviation. The weights of the input and recurrent weight matrix of the ESN network were then randomly generated from an uniform distribution scaled to the interval of [-1,1]. The recurrent matrix was then re-scaled to assure a spectral radius equal to 1, the values of the reservoir (recurrent layer) were computed for all historical periods, and the output weight vector was estimated with a penalized Ridge regression with penalization parameter equal to 1.

To assure that the results are not depending on the specific initialization of the neural network, an envelope of 100 ESN networks was trained, and their predictions averaged. Preliminary tests showed that at this envelope size, the random component of the neural network training gets mostly averaged out and the variability of the final R-Squared values of the model predictions is smaller than 1%. This assures that the results are replicable (up to a 1% margin of error).

5.3. Applied SVJD and SVJD-RV-Z models

The four SVJD and SVJD-RV-Z models that were used in the simulation studies will also be used in the empirical test. They will be denoted as follows:

- **SVJD** – SVJD model with self-exciting jumps in returns
- **SVJD-RV-Z** – SVJD-RV-Z model with self-exciting jumps in returns
- **SVJJD-RV-Z** – SVJD-RV-Z model with self-exciting jumps in returns and volatility
- **2SVJJD-RV-Z** – 2-Component SVJD-RV-Z model with self-exciting jumps in returns and volatility

The estimation of model parameters and latent states was performed with the Sequential Gibbs Particle Filter (SGPD) with 100 parameter particles, 50-particle threshold for re-sampling, 100 latent state particles, and daily re-sampling of the latent states. In order to prevent convergence towards local optima, the following priors were used for the variance-related parameters in all of the models.

Table 9 – Inverse-gamma density priors used for the variability-related parameters of the SVJD models used in the empirical test

	a	b
sigmaJ	5	5*9*(0.01^2)
gamma	10	10*(0.2^2)
sigmaJV	5	5*9*(0.2^2)
phi3	10	10*(0.1^2)
sigmaRV	50	50*(0.6)^2

The priors are especially needed in the case of the 2-Component SVJD model as due to its complexity it sometimes tended to converge to inferior local optimums. Overall, the setting of the priors corresponds to a situation as if we previously observed 10 jumps in prices with standard deviation 3%, 10 volatility jumps with standard deviation 60%, 20 day evolution of the stochastic log-variance in which the short-term component exhibited standard deviation of 20% and the long-term component 10%, and 100 days in which the logaritmized realized variance (adjusted for price jumps) exhibited 60% variability around the stochastic log-variance. As the lengths of the analysed time series are around 4 000 days, the role of the priors should have a negligible role on the final parameter estimates.

5.4. Forecasting currency exchange rate volatility

In this section the predictive power of selected SVJD models is compared with a series of benchmark models (GARCH, HAR, HARQ, etc.) and with models based on Echo State Neural Networks, on a dataset of 7 major currency exchange rates.

5.4.1. Currency dataset description

The models are applied to foreign exchange rate time series of 7 major currency pairs (EUR/USD, GBP/USD, USD/CHF, USD/JPY, USD/CAD, AUD/USD and NZD/USD), over a period from 1.10.1999 to 15.06.2015 (4 063 observations). The first 1 000 observations are used as an in-sample period, while the remaining observations are used as out-sample period for model comparison. The high-frequency non-parametric estimators (realized variance, Z-Estimator and realized quarticity) were computed from 15-minute data provided by ForexHistoryDatabase.com.

Table 10 shows summary statistic of the daily currency exchange rate time series that were used in the empirical study. In addition to the moment statistics it also includes the estimated number of jumps with the Z-Estimator (Barndorff-Nielsen and Shephard 2004), on a 99.99% confidence level, constructed from 15-minute returns.

Table 10 – Summary statistics about the foreign exchange rates dataset

	Start date	End date	Days	Mean	Std	Skewness	Kurtosis	Jumps
EURUSD	1-Nov-99	15-Jun-15	4063	0.0000	0.0065	0.0444	4.4318	222
GBPUSD	1-Nov-99	15-Jun-15	4063	0.0000	0.0057	-0.3406	5.6476	163
USDCHF	1-Nov-99	15-Jun-15	4063	-0.0001	0.0076	-3.8224	115.5912	204
USDJPY	1-Nov-99	15-Jun-15	4063	0.0000	0.0064	-0.0617	6.7908	219
USDCAD	1-Nov-99	15-Jun-15	4063	0.0000	0.0058	0.2100	6.3418	126
AUDUSD	1-Nov-99	15-Jun-15	4063	0.0000	0.0085	-0.5148	15.2365	167
NZDUSD	1-Oct-99	15-Jun-15	4071	0.0001	0.0084	-0.3561	6.5063	161

5.4.2. SVJD model parameter estimates

The four SVJD models were estimated on each of the time series with the use of Sequential Gibbs Particle Filter (Fičura and Witzany, 2018) with 100x100 particles (100 parameter particles, each with 100 latent state particles), with daily re-sampling of the latent state particles and a 50-particle threshold for re-sampling of the parameter particles.

The final parameter estimates from the SGPF algorithm in the last period of the sample are shown in the tables below. Take note that some of the parameters were transformed, specifically $\lambda_{LT} = \alpha_J/(1-\beta_J-\gamma_J)$ and $\lambda_{LT,V} = \alpha_{JV}/(1-\beta_{JV}-\gamma_{JV})$, and further $ltv = \phi_{LT} = \alpha/(1-\beta)$ for one-component models, and $ltv = \phi_{LT} = \phi_0/(1-\phi_1)$ for the two-component model.

Table 11 shows the final period parameter estimates for the SVJD model.

Table 11 – SVJD model parameter estimates and Bayesian standard errors from the SGPF algorithm on the exchange rate time series

Model	EURUSD	GBPUSD	USDCHF	USDJPY	USDCAD	AUDUSD	NZDUSD
mju	0.0001	0.0001	0.0001	0.0001	-0.0001	0.0003	0.0005
	4.26E-05	4.77E-05	3.29E-05	4.68E-05	5.43E-05	7.15E-05	9.73E-05
mjuJ	0.0002	-0.0483	-0.0034	-0.0016	0.0089	-0.0244	-0.0104
	0.0060	0.0019	0.0075	0.0017	0.0037	0.0179	0.0100
sigmaJ	0.0166	0.0243	0.0379	0.0149	0.0338	0.0252	0.0229
	0.0027	0.0029	0.0045	0.0011	0.0040	0.0049	0.0033
ltv	-10.2592	-10.1054	-10.0914	-10.5440	-10.5018	-9.8861	-9.8211
	0.7482	2.4003	1.1843	0.3155	2.3668	1.6847	0.2108
beta	0.9854	0.9887	0.9901	0.9903	0.9880	0.9830	0.9533
	0.0072	0.0062	0.0044	0.0053	0.0065	0.0086	0.0152
gamma	0.1197	0.1032	0.0807	0.0622	0.0882	0.1454	0.2249
	0.0056	0.0103	0.0032	0.0023	0.0043	0.0061	0.0116
lambdaLT	0.0074	0.0538	0.0080	0.0433	0.0014	0.0032	0.0047
	0.0045	0.0076	0.0028	0.0055	0.0005	0.0024	0.0026
betaJ	0.5150	0.9267	0.9073	0.2527	0.1966	0.3182	0.1885
	0.2344	0.0000	0.0546	0.1292	0.1597	0.2072	0.1285
gammaJ	0.0171	0.0833	0.0295	0.1097	0.0472	0.0557	0.1093
	0.0199	0.0000	0.0098	0.0183	0.0390	0.0291	0.0239
CorrRH	0.0116	-0.2481	-0.3134	0.2402	0.1417	-0.1832	-0.2104
	0.0614	0.0956	0.0352	0.1067	0.1439	0.0847	0.0610

We can see from Table 11 that the parameters are at the expected value in most of the cases. The log-variance process exhibits very strong persistence with the β typically being around 0.98-0.99. The jump sizes, measured by σ_j are usually around 2%, while the long-term jump intensity is at levels below 1% for most of the time series with the exception of GBP/USD and USD/JPY, where it is substantially higher. The jumps were estimated as persistently self-exciting on GBP/USD and USD/CHF (high β_j and significant γ_j), while on the USD/JPY and NZD/USD the co-jumps effects can be observed (low β_j with high and significant γ_j). Correlation between volatility and returns ρ was for most of the series close to zero, although on some of them still statistically significant.

Table 12 shows the final period parameter estimates for the SVJD-RV-Z model.

Table 12 – SVJD-RV-Z model parameter estimates and Bayesian standard errors from the SGPF algorithm on the exchange rate time series

Model	EURUSD	GBPUSD	USDCHF	USDJPY	USDCAD	AUDUSD	NZDUSD
mju	0.0001	0.0000	0.0000	0.0001	-0.0001	0.0002	0.0002
	2.65E-05	1.83E-05	2.28E-05	2.13E-05	1.26E-05	2.12E-05	2.12E-05
mjuJ	0.0003	-0.0002	0.0024	0.0000	0.0005	0.0020	-0.0004
	0.0011	0.0012	0.0019	0.0012	0.0013	0.0018	0.0016
sigmaJ	0.0101	0.0092	0.0151	0.0113	0.0097	0.0142	0.0133
	0.0008	0.0009	0.0014	0.0009	0.0009	0.0012	0.0012
ltv	-10.2496	-10.3219	-10.1108	-9.9740	-10.0851	-9.2515	-9.4362
	0.1937	0.4400	0.2231	0.4031	0.6150	1.1490	1.5014
beta	0.9789	0.9894	0.9800	0.9821	0.9912	0.9928	0.9946
	0.0072	0.0053	0.0057	0.0076	0.0038	0.0050	0.0033
gamma	0.1224	0.0810	0.1093	0.1171	0.0738	0.0846	0.0635
	0.0091	0.0046	0.0022	0.0108	0.0037	0.0036	0.0056
lambdaLT	0.0223	0.0190	0.0174	0.0220	0.0166	0.0162	0.0211
	0.0032	0.0035	0.0021	0.0029	0.0027	0.0040	0.0032
betaJ	0.2529	0.7750	0.7203	0.2276	0.2830	0.9844	0.5983
	0.2009	0.2155	0.1029	0.1640	0.1613	0.0573	0.1830
gammaJ	0.0095	0.0210	0.0047	0.0185	0.0250	0.0105	0.0074
	0.0085	0.0129	0.0053	0.0163	0.0143	0.0046	0.0065
sigmaRV	0.4458	0.4264	0.4642	0.4818	0.4223	0.4461	0.4687
	0.0056	0.0043	0.0050	0.0066	0.0052	0.0059	0.0062
mjuZE	1.0208	0.8496	0.9638	1.0175	0.5898	0.8705	0.9197
	0.0232	0.0214	0.0212	0.0217	0.0228	0.0226	0.0211
ksiZE	4.4789	4.2166	4.8647	4.3043	4.7713	4.9951	4.5499
	0.2101	0.2027	0.2246	0.1974	0.2300	0.1986	0.1907
sigmaZE	1.3218	1.2808	1.3463	1.3442	1.3236	1.3309	1.2376
	0.0185	0.0171	0.0177	0.0180	0.0167	0.0152	0.0161
CorrRH	-0.0464	-0.0690	-0.1681	-0.0982	0.0266	-0.2055	-0.1480
	0.0505	0.0548	0.0299	0.0301	0.0275	0.0365	0.0388

We can see from Table 12 that by introducing intraday data, the absolute jump sizes (σ_j) decreased to about 1%, which corresponds to the fact that the Z-Statistics enables us to estimate smaller jumps than in the case when only daily returns are used. The persistence of the log-variance remained at similar levels as in the SVJD model, while the long-term jump intensity (λ_{LT}) mostly increased, corresponding to the fact that the model estimates larger number of small jumps. The jumps were estimated as persistently self-exciting (high β_j and significant γ_j) only for the AUD/USD, while for the other series is the γ_j insignificant.

Table 13 shows the parameter estimates for the SVJJD-RV-Z model.

Table 13 – SVJJD-RV-Z model parameter estimates and Bayesian standard errors from the SGPF algorithm on the exchange rate time series

Model	EURUSD	GBPUSD	USDCHF	USDJPY	USDCAD	AUDUSD	NZDUSD
mju	0.0000	0.0000	0.0000	0.0001	-0.0001	0.0002	0.0003
	1.94E-05	1.63E-05	2.17E-05	1.87E-05	1.53E-05	2.81E-05	2.26E-05
mjuJ	-0.0001	0.0000	-0.0002	-0.0002	0.0001	0.0018	-0.0005
	0.0011	0.0011	0.0018	0.0013	0.0011	0.0018	0.0014
sigmaJ	0.0095	0.0092	0.0142	0.0105	0.0088	0.0138	0.0128
	0.0007	0.0009	0.0027	0.0011	0.0007	0.0012	0.0009
ltv	-10.7570	-10.8689	-10.3811	-10.7193	-10.8599	-10.0404	-9.9632
	0.2407	0.2067	0.0832	0.0890	0.3171	0.1272	0.1118
beta	0.9787	0.9856	0.9450	0.9378	0.9892	0.9717	0.9750
	0.0066	0.0030	0.0082	0.0090	0.0041	0.0061	0.0051
gamma	0.0797	0.0309	0.1109	0.0775	0.0417	0.0884	0.0667
	0.0043	0.0020	0.0030	0.0048	0.0028	0.0055	0.0060
lambdaLT	0.0253	0.0187	0.0206	0.0186	0.0233	0.0167	0.0227
	0.0032	0.0027	0.0020	0.0030	0.0061	0.0020	0.0026
betaJ	0.3803	0.2493	0.1448	0.2492	0.9744	0.3517	0.2579
	0.1606	0.1905	0.0885	0.1697	0.0221	0.1828	0.2346
gammaJ	0.0104	0.0480	0.0771	0.0139	0.0119	0.0164	0.0097
	0.0095	0.0281	0.0111	0.0121	0.0048	0.0127	0.0093
mjuJV	0.4955	0.3324	0.6635	0.5525	0.5765	0.5527	0.2224
	0.1409	0.1668	0.1283	0.0966	0.1548	0.1564	0.1001
sigmaJV	0.7558	0.5998	0.8104	0.7711	0.7172	0.6939	0.5696
	0.0840	0.0837	0.1055	0.0672	0.1250	0.0852	0.0633
lambdaLTV	0.0273	0.0235	0.0235	0.0636	0.0124	0.0164	0.0441
	0.0048	0.0083	0.0026	0.0101	0.0036	0.0060	0.0069
betaJV	0.3271	0.4669	0.0929	0.2790	0.6084	0.3491	0.1972
	0.2170	0.2795	0.0926	0.1683	0.2246	0.2065	0.1282
gammaJV	0.0204	0.0220	0.0224	0.0283	0.0286	0.0213	0.0186
	0.0140	0.0187	0.0104	0.0184	0.0230	0.0123	0.0119
sigmaRV	0.4384	0.4235	0.4450	0.4561	0.4198	0.4268	0.4364
	0.0053	0.0059	0.0059	0.0051	0.0056	0.0058	0.0060
mjuZE	1.0075	0.8487	0.9625	1.0207	0.5730	0.8696	0.9174
	0.0233	0.0209	0.0227	0.0224	0.0228	0.0193	0.0179
ksiZE	4.4594	4.2969	4.9003	4.7043	4.5981	5.0914	4.4947
	0.1804	0.1833	0.1676	0.2057	0.1934	0.1999	0.1570
sigmaZE	1.3079	1.2842	1.3348	1.3383	1.3009	1.3321	1.2282
	0.0186	0.0179	0.0168	0.0174	0.0187	0.0177	0.0148
CorrRH	0.0538	-0.0111	-0.1582	-0.0765	0.0065	-0.1143	-0.0765
	0.0514	0.0551	0.0296	0.0385	0.0372	0.0408	0.0351

Table 14 – 2SVJJD-RV-Z model parameter estimates and Bayesian standard errors from the SGPF algorithm on the exchange rate time series

Model	EURUSD	GBPUSD	USDCHF	USDJPY	USDCAD	AUDUSD	NZDUSD
mju	0.0000	0.0001	0.0000	0.0002	-0.0001	0.0003	0.0003
	2.82E-05	2.03E-05	2.45E-05	1.96E-05	1.68E-05	2.73E-05	2.88E-05
mjuJ	0.0000	-0.0001	-0.0008	0.0004	0.0005	0.0014	-0.0003
	0.0011	0.0011	0.0022	0.0017	0.0014	0.0018	0.0015
sigmaJ	0.0099	0.0085	0.0175	0.0109	0.0093	0.0133	0.0129
	0.0008	0.0007	0.0026	0.0010	0.0008	0.0014	0.0011
ltv	-10.1827	-10.4108	-10.4515	-10.4178	-11.1461	-10.1302	-9.6145
	0.7084	0.6026	0.3686	0.5099	0.0140	0.0051	0.0139
beta	0.6367	0.6550	0.6588	0.5370	0.9877	0.9706	0.9674
	0.0352	0.0460	0.0566	0.0181	0.0054	0.0069	0.0077
gamma	0.1150	0.1076	0.1249	0.1180	0.1054	0.1157	0.1213
	0.0095	0.0044	0.0072	0.0056	0.0040	0.0051	0.0067
lambdaLT	0.0229	0.0312	0.0187	0.0182	0.0177	0.0217	0.0219
	0.0034	0.0070	0.0029	0.0031	0.0030	0.0082	0.0031
betaJ	0.5459	0.9569	0.6319	0.4146	0.3324	0.9914	0.5500
	0.1703	0.1218	0.2364	0.1652	0.2541	0.0013	0.1940
gammaJ	0.0228	0.0107	0.0149	0.0233	0.0429	0.0065	0.0104
	0.0154	0.0041	0.0092	0.0154	0.0211	0.0014	0.0095
mjuJV	0.6972	0.4666	0.9210	0.7786	0.4982	0.6554	0.1838
	0.1391	0.2038	0.1476	0.1266	0.1532	0.1716	0.1732
sigmaJV	0.6459	0.6182	0.5995	0.6374	0.6542	0.6625	0.6421
	0.1184	0.0866	0.0640	0.0718	0.0947	0.1044	0.0895
lambdaLTV	0.0266	0.0145	0.0272	0.0640	0.0149	0.0146	0.0146
	0.0059	0.0053	0.0077	0.0089	0.0050	0.0040	0.0075
betaJV	0.2778	0.2746	0.5191	0.6051	0.3548	0.5548	0.3713
	0.2618	0.1986	0.3018	0.1583	0.1931	0.2079	0.1847
gammaJV	0.0241	0.0212	0.0127	0.0260	0.0210	0.0100	0.0442
	0.0149	0.0141	0.0092	0.0176	0.0141	0.0084	0.0146
phi1	0.9924	0.9898	0.9854	0.9850	0.1272	0.2717	0.4936
	0.0042	0.0042	0.0049	0.0063	0.0538	0.0946	0.0968
phi2	0.0738	0.0757	0.0788	0.0892	0.0809	0.0679	0.0721
	0.0038	0.0045	0.0043	0.0045	0.0064	0.0030	0.0033
sigmaRV	0.4301	0.4117	0.4361	0.4411	0.4093	0.4211	0.4311
	0.0050	0.0043	0.0057	0.0047	0.0050	0.0054	0.0048
mjuZE	1.0175	0.8397	0.9635	1.0260	0.5854	0.8633	0.9184
	0.0206	0.0208	0.0225	0.0227	0.0216	0.0208	0.0217
ksiZE	4.5410	4.0568	4.9947	4.7551	4.8102	5.0234	4.5826
	0.1945	0.1585	0.2212	0.2390	0.1886	0.1834	0.2003
sigmaZE	1.3178	1.2735	1.3364	1.3469	1.3184	1.3190	1.2347
	0.0179	0.0154	0.0175	0.0195	0.0172	0.0168	0.0155
CorrRH	0.0191	-0.0654	-0.1090	-0.0719	0.0421	-0.1572	-0.1212
	0.0653	0.0487	0.0342	0.0331	0.0275	0.0369	0.0243

We can see from Table 13 that with the introduction of volatility jumps, the persistence of the log-variance slightly decreased when compared with the SVJD-RV-Z model. Persistent clustering of the price jumps (high β_j and significant γ_j) was in the SVJJD-RV-Z model estimated only for the USD/CAD, while the USD/CHF exhibits co-jumps (low β_j and significant γ_j). The volatility jumps were estimated to occur in 1-6% of the days (depending on the currency) and to be significantly upwards biased with μ_{JV} ranging from 0.33 to 0.66. Significant volatility jump clustering is observed only on the USD/CHF, in the form of volatility co-jumps (low β_{jV} and significant γ_{jV})

Table 14 shows the results for the 2SVJJD-RV-Z model. It is apparent that while the persistence of the two log-variance components was estimated in accordance with our expectations ($\phi_1 > \beta$) on EUR/USD, GBP/USD, USD/CHF and USD/JPY, on the reminding time series (USD/CAD, AUD/USD and NZD/USD) it was estimated inversely. Persistent returns-jump clustering was estimated for GBP/USD and AUD/USD, and returns co-jumps for USD/CAD. Significant co-jumps in volatility were further estimated for the NZD/USD.

5.4.3. Exchange rate volatility forecasting results

Realized variance for each of the 7 currency time series was forecasted in the 1-day, 5-day and the 22-day horizon and the historical accuracy of the forecasts was assessed with the R-Squared criterion. The multi-period forecasts were performed with direct projection for the HAR and ESN models (as in Bollerslev et al, 2015), in which the model is re-estimated with the multi-period realized variance as the target. For the logarithmic and square root transformations of the models, forecasts were appropriately transformed in order to get forecasts of the realized variance. For the SVJD model, forecasts were constructed by randomly sampling the parameter and latent state particles from the Particle Filter, constructing 10 000 simulations of the latent states into the future (for the respective horizon), computing the quadratic variation for each simulation and then their average.

Table 15 shows the R-Squared values for all of the models and currency time series for the 1-day forecast horizon. The last two columns show the average R-Squared and the average rank of the model over all of the forecasted series. The colouring indicates the size of the R-Squared (green is better, red is worse) which is applied separately for each column and corresponds to the relative performance of the model when compared to other models.

Table 15 – R-Squared of the 1-Day realized variance forecasts (currencies)

Type	Model	EURUSD	GBPUSD	USDCHF	USDJPY	USDCAD	AUDUSD	NZDUSD	AvgR2	AvgRank
GARCH	GARCH	0.0574	0.4140	-0.0296	0.2221	0.5020	0.4515	0.3280	0.2779	31.79
	EGARCH	0.3635	0.2569	-0.0024	0.0096	0.4200	0.2910	0.1408	0.2114	34.14
	GJRGARCH	0.0574	0.4140	-0.0296	0.2221	0.4862	0.5480	0.3280	0.2894	31.36
HAR	HAR	0.5135	0.6704	-0.0242	0.3397	0.5811	0.6057	0.6285	0.4735	14.57
	LogHAR	0.4838	0.5356	0.0162	0.2650	0.4504	0.4923	0.5429	0.3980	26.43
	SqrtHAR	0.5146	0.6344	0.0071	0.3224	0.5571	0.5822	0.6086	0.4609	19.29
	HARQ	0.5305	0.4730	-0.0007	0.2620	0.5297	0.3898	0.4360	0.3743	26.29
	LogHARQ	0.5091	0.5550	0.0166	0.2891	0.4705	0.5305	0.5838	0.4221	23.29
	SqrtHARQ	0.5204	0.4800	-0.0064	0.3203	0.5346	0.3593	0.4372	0.3779	25.14
	HARJ	0.5406	0.6692	-0.0039	0.3409	0.5805	0.6306	0.6457	0.4862	8.29
	logHARJ	0.4881	0.4825	0.0092	0.2643	0.4533	0.4895	0.5168	0.3862	27.29
	sqrtHARJ	0.5243	0.6338	0.0099	0.3256	0.5534	0.5934	0.6248	0.4665	16.14
	AHAR	0.5128	0.6713	-0.0243	0.3482	0.5815	0.6081	0.6310	0.4755	13.71
	logAHAR	0.4807	0.5430	0.0154	0.2770	0.4469	0.5231	0.5495	0.4051	25.71
	sqrtAHAR	0.5125	0.6369	0.0074	0.3318	0.5563	0.5915	0.6122	0.4641	18.14
	SHAR	0.5164	0.6719	-0.0164	0.3501	0.5811	0.6181	0.6284	0.4785	12.57
	logSHAR	0.4844	0.5372	0.0158	0.2728	0.4508	0.4924	0.5431	0.3995	25.71
	sqrtSHAR	0.5158	0.6361	0.0068	0.3362	0.5570	0.5908	0.6091	0.4645	17.86
ESN	ESN	0.5397	0.6602	-0.7465	0.2049	0.5033	0.5907	0.6328	0.3407	21.14
	LogESN	0.5318	0.6811	0.0077	0.3267	0.5815	0.6104	0.6218	0.4802	11.57
	SqrtESN	0.5437	0.6857	-0.5316	0.2962	0.5480	0.6394	0.6469	0.4040	10.71
	ESNQ	0.5242	0.6563	-0.7588	0.2055	0.4974	0.5778	0.6450	0.3353	23.14
	LogESNQ	0.5424	0.6841	0.0074	0.3283	0.5813	0.6305	0.6403	0.4877	7.71
	SqrtESNQ	0.5484	0.6828	-0.6616	0.2781	0.5409	0.6239	0.6652	0.3825	12.86
	ESNJ	0.5167	0.6575	-0.7607	0.2140	0.4980	0.5778	0.6266	0.3328	25.14
	LogESNJ	0.5298	0.6618	-0.3723	0.2422	0.5557	0.6158	0.6210	0.4077	19.14
	SqrtESNJ	0.5412	0.6880	-0.5611	0.3001	0.5531	0.6179	0.6548	0.3991	12.29
	SESN	0.5172	0.6593	-0.7510	0.1994	0.4963	0.5750	0.6229	0.3313	25.71
	LogSESN	0.5343	0.6822	0.0074	0.3388	0.5739	0.6234	0.6235	0.4833	10.29
	SqrtSESN	0.5384	0.6866	-0.5521	0.3093	0.5388	0.6251	0.6446	0.3987	12.71
	SESNQ	0.5339	0.6536	-0.7585	0.1790	0.4901	0.5796	0.6438	0.3316	23.29
	LogSESNQ	0.5428	0.6856	0.0057	0.3379	0.5748	0.6360	0.6426	0.4893	7.29
	SqrtSNESNQ	0.5490	0.6842	-0.6553	0.2776	0.5327	0.6241	0.6652	0.3825	12.57
SVJD	SVJD	0.4632	0.6015	0.0073	0.1965	0.5245	0.4532	0.4063	0.3789	28.71
	SVJD-RV-Z	0.5181	0.6019	0.0105	0.2753	0.5319	0.4603	0.4497	0.4068	23.00
	SVJJD-RV-Z	0.5255	0.6885	0.0177	0.3344	0.5723	0.6303	0.6417	0.4872	7.29
	2SVJJD-RV-Z	0.5322	0.6562	0.0115	0.3339	0.5790	0.6209	0.6405	0.4820	10.71

We can see from Table 15 that the best average R-Squared was achieved by the logarithmic form of the SESNQ model (Echo State Neural network, using the realized variance, realized quarticity and realized semi-variance as the predictors). Only slightly lower is the average R-Squared of the SVJJD-RV-Z model. These two models do also have jointly the best average rank (7.29).

The SVJJD-RV-Z model significantly outperformed the version of the model without volatility jumps (SVJD-RV-Z), and it slightly outperformed even the more complex 2SVJJD-RV-Z model.

Among the ESN models, the LogSESNQ model was followed by the LogSESN, LogESNQ and the standard LogESN. Generally, the logarithmic specifications performed the best, followed by the square root specifications which were almost as good on all of the time series except for the USDCHF on which they performed poorly.

The best HAR model was HARJ, which extends the basic HAR model by including the bi-power variation based jump variance estimates as an additional predictor. HARJ performed the best among the HAR models based on both, the average R-Squared as well as the average rank criterions, and its performance is comparable to the best ESN and SVJD models. Other well-performing HAR models were SHAR, AHAR and the standard HAR. Surprisingly the untransformed and the square root HAR specifications outperformed the logarithmic ones, which is unexpected as usually the logarithmic forms of the HAR models perform best.

The GARCH models performed worst, underperforming even the standard SVJD model with no intraday data.

All of the models achieved very low R-Squared on the USDCHF, which can be attributed to the enormous jumps that occurred at the start and the end of the Swiss central bank monetary interventions.

The performance of the models in the 5-Day forecast horizon is shown in Table 16.

We can see that the 2SVJJD-RV-Z model was in this case the best one among the SVJD models, but it still got slightly outperformed by some of the HAR models, with the HARJ model performing best. Among other well-performing HAR models is again the AHAR, SHAR, and the standard HAR.

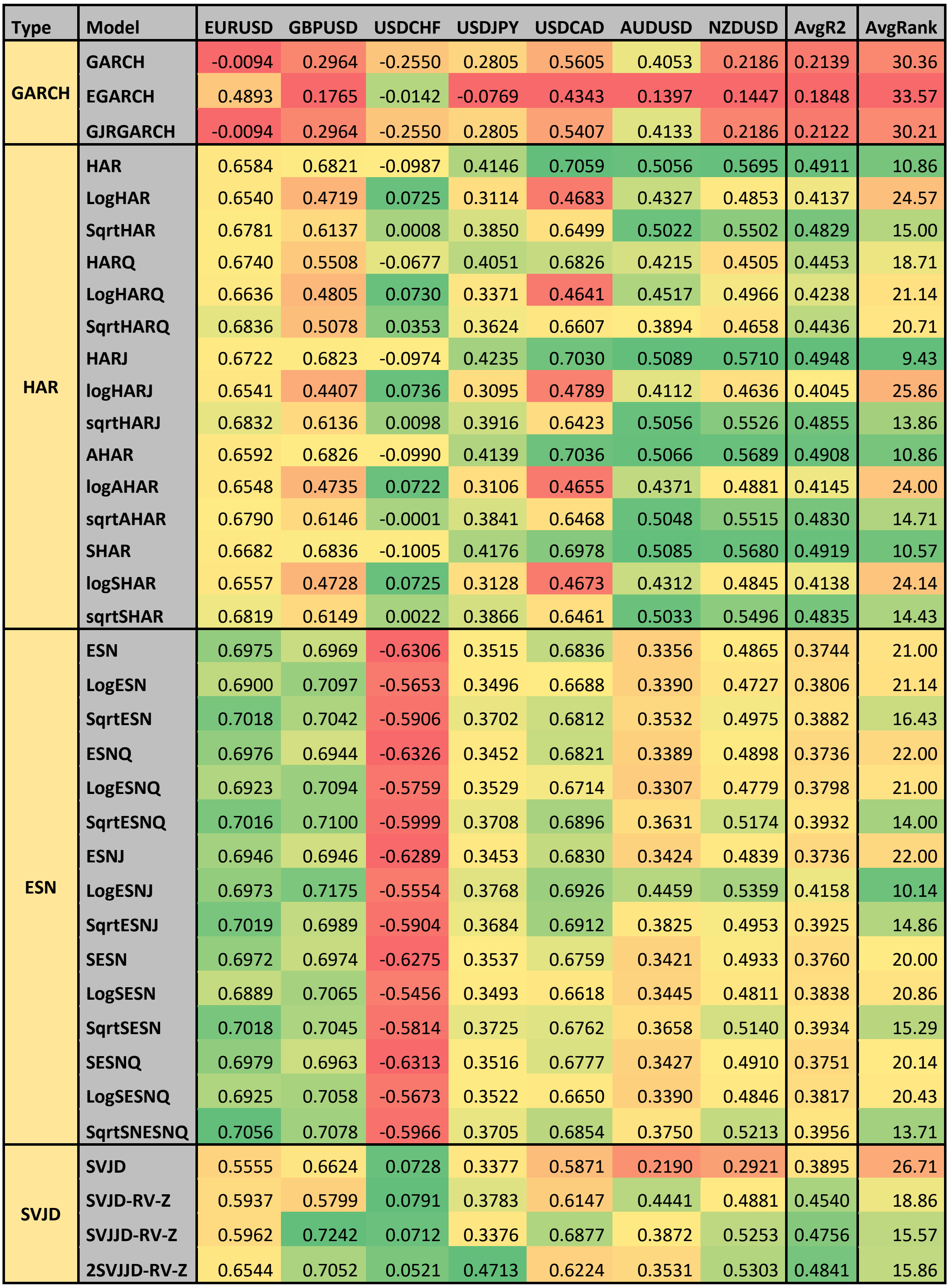
The ESN models achieved average performance on most of the time series, with the exception of USDCHF on which their performance was very poor, which has had a significant impact on the average R-Squared and average rank statistics. The best performing ESN model was again the logarithmic form of SESNQ.

Table 16 – R-Squared of the 5-Day realized variance forecasts (currencies)

Type	Model	EURUSD	GBPUSD	USDCHF	USDJPY	USDCAD	AUDUSD	NZDUSD	AvgR2	AvgRank
GARCH	GARCH	0.0089	0.4413	-0.1503	0.3001	0.6303	0.4944	0.3395	0.2949	31.07
	EGARCH	0.4887	0.2722	-0.0056	-0.0332	0.5196	0.2722	0.1618	0.2394	33.71
	GJRGARCH	0.0089	0.4413	-0.1503	0.3001	0.6089	0.5706	0.3395	0.3027	29.36
HAR	HAR	0.6539	0.7391	-0.0835	0.4001	0.7167	0.5998	0.6321	0.5226	9.57
	LogHAR	0.6430	0.5785	0.0343	0.3270	0.5431	0.4882	0.5430	0.4510	25.43
	SqrtHAR	0.6671	0.6959	-0.0036	0.3851	0.6839	0.5806	0.6132	0.5174	14.00
	HARQ	0.6730	0.6076	-0.0035	0.3844	0.6940	0.4369	0.4843	0.4681	18.43
	LogHARQ	0.6613	0.5876	0.0358	0.3557	0.5550	0.5151	0.5618	0.4675	20.43
	SqrtHARQ	0.6753	0.6084	0.0239	0.3564	0.6750	0.4013	0.4884	0.4612	19.71
	HARJ	0.6727	0.7390	-0.0268	0.4106	0.7116	0.6108	0.6397	0.5368	6.00
	logHARJ	0.6471	0.5558	0.0336	0.3201	0.5386	0.4623	0.5156	0.4390	26.57
	sqrtHARJ	0.6741	0.6958	0.0050	0.3905	0.6782	0.5875	0.6186	0.5214	12.29
	AHAR	0.6546	0.7403	-0.0835	0.4053	0.7159	0.6016	0.6349	0.5242	8.43
	logAHAR	0.6435	0.5843	0.0343	0.3339	0.5431	0.5010	0.5517	0.4560	24.43
	sqrtAHAR	0.6676	0.6984	-0.0029	0.3910	0.6834	0.5864	0.6182	0.5203	11.86
	SHAR	0.6619	0.7410	-0.0638	0.4132	0.7133	0.6086	0.6326	0.5295	6.86
	logSHAR	0.6440	0.5800	0.0339	0.3344	0.5432	0.4876	0.5430	0.4523	24.71
	sqrtSHAR	0.6698	0.6975	-0.0020	0.3960	0.6829	0.5858	0.6139	0.5206	12.29
ESN	ESN	0.6543	0.7287	-0.7471	0.2479	0.6610	0.5307	0.5970	0.3818	25.29
	LogESN	0.6443	0.7411	-0.4460	0.3136	0.6832	0.5790	0.6155	0.4472	17.14
	SqrtESN	0.6661	0.7373	-0.6371	0.3042	0.6766	0.5747	0.6225	0.4206	18.14
	ESNQ	0.6565	0.7274	-0.7457	0.2472	0.6631	0.5268	0.5952	0.3815	25.00
	LogESNQ	0.6473	0.7437	-0.4971	0.3086	0.6812	0.5961	0.6264	0.4437	15.29
	SqrtESNQ	0.6667	0.7379	-0.6355	0.3155	0.6731	0.5686	0.6200	0.4209	18.29
	ESNJ	0.6461	0.7282	-0.7496	0.2275	0.6601	0.5260	0.5923	0.3758	28.00
	LogESNJ	0.6553	0.7138	-0.3100	0.3264	0.6798	0.5363	0.5740	0.4537	20.00
	SqrtESNJ	0.6678	0.7375	-0.6306	0.2945	0.6789	0.5684	0.6198	0.4195	17.86
	SESN	0.6474	0.7282	-0.7455	0.2546	0.6617	0.5310	0.5949	0.3817	25.57
	LogSESN	0.6503	0.7391	-0.4681	0.3150	0.6802	0.5978	0.6229	0.4482	15.71
	SqrtSESN	0.6680	0.7372	-0.6177	0.3174	0.6733	0.5738	0.6261	0.4254	16.71
	SESNQ	0.6493	0.7277	-0.7463	0.2531	0.6620	0.5264	0.5914	0.3805	26.57
	LogSESNQ	0.6524	0.7406	-0.5050	0.3129	0.6805	0.6025	0.6277	0.4445	14.57
	SqrtSNESNQ	0.6675	0.7372	-0.6307	0.3242	0.6739	0.5681	0.6181	0.4226	18.43
SVJD	SVJD	0.5786	0.6976	0.0258	0.2910	0.6622	0.4592	0.4051	0.4456	27.14
	SVJD-RV-Z	0.6569	0.6757	0.0333	0.3760	0.6742	0.5102	0.5187	0.4921	19.71
	SVJJD-RV-Z	0.6335	0.7702	0.0330	0.3294	0.6755	0.6093	0.6443	0.5279	11.14
	2SVJJD-RV-Z	0.6610	0.7402	0.0282	0.4194	0.6854	0.5840	0.6442	0.5375	7.29

The results for the 22-Day forecast horizon are in Table 17.

Table 17 – R-Squared of the 22-Day realized variance forecasts (currencies)



Type	Model	EURUSD	GBPUSD	USDCHF	USDJPY	USDCAD	AUDUSD	NZDUSD	AvgR2	AvgRank
GARCH	GARCH	-0.0094	0.2964	-0.2550	0.2805	0.5605	0.4053	0.2186	0.2139	30.36
	EGARCH	0.4893	0.1765	-0.0142	-0.0769	0.4343	0.1397	0.1447	0.1848	33.57
	GJRGARCH	-0.0094	0.2964	-0.2550	0.2805	0.5407	0.4133	0.2186	0.2122	30.21
HAR	HAR	0.6584	0.6821	-0.0987	0.4146	0.7059	0.5056	0.5695	0.4911	10.86
	LogHAR	0.6540	0.4719	0.0725	0.3114	0.4683	0.4327	0.4853	0.4137	24.57
	SqrtHAR	0.6781	0.6137	0.0008	0.3850	0.6499	0.5022	0.5502	0.4829	15.00
	HARQ	0.6740	0.5508	-0.0677	0.4051	0.6826	0.4215	0.4505	0.4453	18.71
	LogHARQ	0.6636	0.4805	0.0730	0.3371	0.4641	0.4517	0.4966	0.4238	21.14
	SqrtHARQ	0.6836	0.5078	0.0353	0.3624	0.6607	0.3894	0.4658	0.4436	20.71
	HARJ	0.6722	0.6823	-0.0974	0.4235	0.7030	0.5089	0.5710	0.4948	9.43
	logHARJ	0.6541	0.4407	0.0736	0.3095	0.4789	0.4112	0.4636	0.4045	25.86
	sqrtHARJ	0.6832	0.6136	0.0098	0.3916	0.6423	0.5056	0.5526	0.4855	13.86
	AHAR	0.6592	0.6826	-0.0990	0.4139	0.7036	0.5066	0.5689	0.4908	10.86
	logAHAR	0.6548	0.4735	0.0722	0.3106	0.4655	0.4371	0.4881	0.4145	24.00
	sqrtAHAR	0.6790	0.6146	-0.0001	0.3841	0.6468	0.5048	0.5515	0.4830	14.71
	SHAR	0.6682	0.6836	-0.1005	0.4176	0.6978	0.5085	0.5680	0.4919	10.57
	logSHAR	0.6557	0.4728	0.0725	0.3128	0.4673	0.4312	0.4845	0.4138	24.14
	sqrtSHAR	0.6819	0.6149	0.0022	0.3866	0.6461	0.5033	0.5496	0.4835	14.43
ESN	ESN	0.6975	0.6969	-0.6306	0.3515	0.6836	0.3356	0.4865	0.3744	21.00
	LogESN	0.6900	0.7097	-0.5653	0.3496	0.6688	0.3390	0.4727	0.3806	21.14
	SqrtESN	0.7018	0.7042	-0.5906	0.3702	0.6812	0.3532	0.4975	0.3882	16.43
	ESNQ	0.6976	0.6944	-0.6326	0.3452	0.6821	0.3389	0.4898	0.3736	22.00
	LogESNQ	0.6923	0.7094	-0.5759	0.3529	0.6714	0.3307	0.4779	0.3798	21.00
	SqrtESNQ	0.7016	0.7100	-0.5999	0.3708	0.6896	0.3631	0.5174	0.3932	14.00
	ESNJ	0.6946	0.6946	-0.6289	0.3453	0.6830	0.3424	0.4839	0.3736	22.00
	LogESNJ	0.6973	0.7175	-0.5554	0.3768	0.6926	0.4459	0.5359	0.4158	10.14
	SqrtESNJ	0.7019	0.6989	-0.5904	0.3684	0.6912	0.3825	0.4953	0.3925	14.86
	SESN	0.6972	0.6974	-0.6275	0.3537	0.6759	0.3421	0.4933	0.3760	20.00
	LogSESN	0.6889	0.7065	-0.5456	0.3493	0.6618	0.3445	0.4811	0.3838	20.86
	SqrtSESN	0.7018	0.7045	-0.5814	0.3725	0.6762	0.3658	0.5140	0.3934	15.29
	SESNQ	0.6979	0.6963	-0.6313	0.3516	0.6777	0.3427	0.4910	0.3751	20.14
	LogSESNQ	0.6925	0.7058	-0.5673	0.3522	0.6650	0.3390	0.4846	0.3817	20.43
	SqrtSNESNQ	0.7056	0.7078	-0.5966	0.3705	0.6854	0.3750	0.5213	0.3956	13.71
SVJD	SVJD	0.5555	0.6624	0.0728	0.3377	0.5871	0.2190	0.2921	0.3895	26.71
	SVJD-RV-Z	0.5937	0.5799	0.0791	0.3783	0.6147	0.4441	0.4881	0.4540	18.86
	SVJJD-RV-Z	0.5962	0.7242	0.0712	0.3376	0.6877	0.3872	0.5253	0.4756	15.57
	2SVJJD-RV-Z	0.6544	0.7052	0.0521	0.4713	0.6224	0.3531	0.5303	0.4841	15.86

We can see from Table 17 that the best model on the 22-Day forecast horizon was again the HARJ model, followed by SHAR, AHAR and standard HAR.

The ESN models achieved bad average R-Squared statistics due to their low performance on USDCHF. Nevertheless, the models performed quite well on the other series, with the LogESNJ being the second best model (after HARJ) according to the average rank criterion. The ESN models also clearly outperformed the HAR models on the EUR/USD and the GBP/USD pair.

The SVJD models achieved slightly above-average performance based on the average rank, while their average R-Squared is relatively close to the best of the HAR models. The best SVJD models were again the SVJJD-RV-Z and the 2SVJJD-RV-Z, which shows that the including of volatility jumps plays an important role in volatility forecasting.

5.5. Forecasting ETF/ETN volatility

In the second empirical study, the proposed models (ARCH/GARCH, HAR, ESN and SVJD based) were applied to 10 most commonly traded Exchange Traded Funds (ETF) and Exchange Traded Notes (ETN).

5.5.1. ETF/ETN dataset description

The ETF/ETN for the study were selected based on average volume on 1th June 2018, as reported by etfdb.com. The time series vary in their length. In spite of this, the first 1 000 days will be used for the in-sample and the rest of the dataset as out-sample. The high-frequency data used for power-variation estimator calculation were provided by Kibot.com. The 15-minute frequency was used for the calculation of all of the estimators.

Table 18 shows information about the selected ETF/ETN.

Table 18 – ETF and ETN used in the empirical study

	Name	Issuer	Type	Class	Assets
EEM	iShares MSCI Emerging Markets ETF	iShares	ETF	Large-Cap Equity	Emerging Market Equities
EFA	iShares MSCI EAFE ETF	iShares	ETF	Large-Cap Equity	Foreign Developed Market Equities
FXI	iShares China Large-Cap ETF	iShares	ETF	Large-Cap Equity	China Equities
GDX	VanEck Vectors Gold Miners ETF	VanEck	ETF	Multi-Cap Equity	Gold Miner Equities
IWM	iShares Russell 2000 ETF	iShares	ETF	Small-Cap Equity	US Equities
QQQ	Invesco QQQ	Invesco	UIT	Large-Cap Equity	US Technology Equities
SPY	SPDR S&P 500 ETF	State Street SPDR	UIT	Large-Cap Equity	US Equities
USO	United States Oil Fund	USCF	Pool	Commodity Futures	Light Sweet Crude Oil
VXX	iPath S&P 500 VIX Short-Term Futures ETN	Barclays Capital	ETN	Volatility	US Volatility (VIX)
XLF	Financial Select Sector SPDR Fund	State Street SPDR	ETF	Large-Cap Equity	US Financial Equities

Source: Constructed based on information from etfdb.com

We can see that the dataset includes large-cap and small-cap equities, from domestic (US) as well as foreign markets, and from different sectors (technology, gold mining, finance). Additionally, commodity based (USO) and volatility based (VXX) ETFs are also included. The results will thus enable us to evaluate the model performance with respect to different asset classes.

Table 19 shows the summary statistics of the ETFs and ETNs included in the dataset, as well the numbers of jumps, estimated with the Z-Estimator on the 99.99% level.

Table 19 – Summary statistics about the analyzed ETF and ETN

	Start date	End date	Days	Mean	Std	Skewness	Kurtosis	Jumps
EEM	23-Apr-03	1-Jun-18	3795	0.0004	0.0191	0.3360	24.5088	673
EFA	27-Aug-01	1-Jun-18	4215	0.0002	0.0143	-0.1706	16.7685	732
FXI	8-Oct-04	1-Jun-18	3436	0.0004	0.0220	0.4389	16.3244	747
GDX	22-May-06	1-Jun-18	3029	-0.0001	0.0270	-0.0342	7.4723	160
IWM	2-Jun-00	1-Jun-18	4526	0.0003	0.0155	-0.0584	10.1336	253
QQQ	10-Mar-99	1-Jun-18	4839	0.0003	0.0188	0.0638	11.5924	356
SPY	2-Jan-98	1-Jun-18	5137	0.0003	0.0130	0.1469	16.4962	525
USO	10-Apr-06	1-Jun-18	3058	-0.0005	0.0216	-0.2311	5.5322	234
VXX	30-Jan-09	1-Jun-18	2351	-0.0034	0.0413	1.6027	19.4639	184
XLF	22-Dec-98	1-Jun-18	4891	0.0002	0.0202	0.3923	23.9160	289

Source: Computed based on data from Kibot.com

5.5.2. SVJD model parameter estimates

The four tested SVJD models were estimated on the ETF/ETN time series with the Sequential Gibbs Particle Filter with 100x100 particles, daily re-sampling of the latent state particles and a 50 particle threshold for re-sampling of the parameter particles.

The final period parameter estimates from the particle filter can be seen in the tables below. Take note that some of the parameters were transformed, specifically $\lambda_{LT} = \alpha_J/(1-\beta_J-\gamma_J)$ and $\lambda_{LT,V} = \alpha_{JV}/(1-\beta_{JV}-\gamma_{JV})$, and further $ltv = \phi_{LT} = \alpha/(1-\beta)$ for one-component models, and $ltv = \phi_{LT} = \phi_0/(1-\phi_1)$ for the two-component model.

Table 20 shows the parameter estimates and the Bayesian standard errors from the SGPF algorithm for the SVJD model on the ETF/ETN time series.

Table 20 – SVJD model parameter estimates and Bayesian standard errors from the SGPF algorithm on the ETF/ETN time series

Model	EEM	EFA	FXI	GDX	IWM	QQQ	SPY	USO	VXX	XLF
mju	0.0012	0.0008	0.0007	-0.0002	0.0010	0.0010	0.0008	0.0006	-0.0075	0.0009
	1.74E-04	8.85E-05	1.39E-04	2.26E-04	1.66E-04	8.85E-05	6.78E-05	2.53E-04	4.37E-04	1.13E-04
mjuJ	-0.0279	-0.0170	-0.0137	-0.0295	-0.0188	-0.0086	-0.0076	-0.0166	0.0898	0.0035
	0.0251	0.0115	0.0272	0.0403	0.0158	0.0141	0.0083	0.0230	0.0371	0.0280
sigmaJ	0.0592	0.0503	0.0642	0.0806	0.0479	0.0533	0.0361	0.0600	0.1192	0.0848
	0.0106	0.0070	0.0115	0.0116	0.0071	0.0085	0.0044	0.0086	0.0166	0.0106
ltv	-8.1649	-8.9520	-8.3518	-7.1184	-8.4320	-7.3679	-8.8122	-7.8036	-6.7582	-8.5909
	0.7431	0.5316	1.0517	2.5315	1.2665	1.3975	1.4810	1.0941	0.2007	0.4040
beta	0.9838	0.9807	0.9839	0.9944	0.9751	0.9928	0.9843	0.9823	0.9235	0.9760
	0.0084	0.0072	0.0073	0.0043	0.0121	0.0035	0.0068	0.0086	0.0169	0.0071
gamma	0.1518	0.1878	0.1685	0.1129	0.1599	0.1509	0.1933	0.1246	0.3601	0.2002
	0.0151	0.0163	0.0123	0.0092	0.0125	0.0142	0.0097	0.0113	0.0275	0.0106
lambdaLT	0.0036	0.0070	0.0052	0.0035	0.0037	0.0066	0.0093	0.0068	0.0081	0.0030
	0.0022	0.0028	0.0037	0.0026	0.0015	0.0023	0.0028	0.0037	0.0028	0.0013
betaJ	0.2950	0.3888	0.3107	0.4004	0.5985	0.6681	0.2225	0.5747	0.5802	0.8487
	0.1688	0.1965	0.1915	0.1900	0.2027	0.1510	0.1461	0.1800	0.1007	0.1141
gammaJ	0.0698	0.0902	0.0606	0.0460	0.0534	0.0480	0.1776	0.0791	0.0959	0.0189
	0.0320	0.0365	0.0207	0.0366	0.0218	0.0184	0.0317	0.0201	0.0300	0.0125
CorrRH	-0.3808	-0.2905	-0.0067	-0.0770	-0.3643	-0.2852	-0.2640	-0.4231	0.3193	-0.3578
	0.1171	0.0524	0.0688	0.0871	0.0969	0.0504	0.0535	0.1380	0.0447	0.0604

We can see from Table 21 that similarly to the exchange rate time series, the log-variance was estimated as very persistent, with β typically around 0.98. The jump sizes exhibit far greater volatility than for the currencies (σ_J around 5%-8%), and have for most of the time series negative mean value (μ_J), with the exception of VXX, which is a VIX ETN for which significantly positive jump sizes are observed. The long-term jump intensity for most of the time series is very low ($\lambda_{LT} < 1\%$) and the jumps do not exhibit long-term clustering (β_J is much smaller than one), although on most of the time series we can observe statistically significant co-jump effects (γ_J is significant), which seem to be especially pronounced on SPY. In accordance with the expectation for stock-based ETFs/ETNs, the correlation between volatility and returns (ρ) is negative for all of the time series with the exception of VXX, on which a positive correlation is to be expected.

Table 21 shows the parameter estimates and standard errors for the SVJD-RV-Z model on the ETF/ETN time series.

Table 21 – SVJD-RV-Z model parameter estimates and Bayesian standard errors from the SGPF algorithm on the ETF/ETN time series

Model	EEM	EFA	FXI	GDX	IWM	QQQ	SPY	USO	VXX	XLF
mju	0.0013	0.0010	0.0012	0.0001	0.0009	0.0011	0.0009	0.0004	-0.0072	0.0006
	8.57E-05	5.42E-05	1.15E-04	2.41E-04	7.42E-05	5.57E-05	3.47E-05	1.61E-04	3.22E-04	5.16E-05
mjuJ	-0.0004	-0.0013	0.0032	0.0022	0.0026	0.0062	0.0030	-0.0004	-0.0022	0.0049
	0.0047	0.0101	0.0061	0.0058	0.0040	0.0049	0.0028	0.0134	0.0134	0.0081
sigmaJ	0.0368	0.0449	0.0417	0.0472	0.0306	0.0362	0.0219	0.0513	0.0925	0.0510
	0.0040	0.0062	0.0045	0.0046	0.0029	0.0030	0.0017	0.0064	0.0105	0.0053
ltv	-7.3439	-9.1184	-7.8466	-6.5331	-8.4526	-7.5363	-9.0017	-7.5400	-6.7218	-8.5499
	0.7440	0.1942	0.3896	1.8190	0.2777	0.6768	0.1900	0.7413	0.1228	0.4601
beta	0.9894	0.9572	0.9777	0.9896	0.9798	0.9907	0.9746	0.9833	0.9144	0.9877
	0.0047	0.0105	0.0059	0.0074	0.0052	0.0038	0.0056	0.0091	0.0145	0.0041
gamma	0.0993	0.2767	0.1560	0.0976	0.1358	0.1398	0.1831	0.1221	0.2346	0.1797
	0.0052	0.0261	0.0221	0.0064	0.0066	0.0049	0.0081	0.0268	0.0091	0.0084
lambdaLT	0.0215	0.0058	0.0193	0.0217	0.0157	0.0170	0.0209	0.0082	0.0319	0.0110
	0.0052	0.0018	0.0039	0.0048	0.0032	0.0025	0.0022	0.0028	0.0043	0.0035
betaJ	0.6180	0.2531	0.2973	0.5446	0.2564	0.4794	0.3604	0.6663	0.8658	0.2455
	0.1384	0.1797	0.1379	0.1818	0.1322	0.1358	0.1626	0.2012	0.0425	0.1435
gammaJ	0.1046	0.1006	0.0635	0.0426	0.0800	0.0871	0.0553	0.0399	0.0526	0.1269
	0.0272	0.0263	0.0229	0.0290	0.0165	0.0151	0.0202	0.0207	0.0097	0.0241
sigmaRV	0.7800	0.7099	0.8055	0.6253	0.6133	0.5920	0.5796	0.5971	0.6423	0.6161
	0.0164	0.0114	0.0251	0.0087	0.0074	0.0068	0.0073	0.0254	0.0096	0.0069
mjuZE	1.9081	1.9166	2.0775	1.0965	1.2015	1.2304	1.4450	1.4264	1.2163	1.2194
	0.0330	0.0278	0.0368	0.0268	0.0202	0.0228	0.0216	0.0274	0.0337	0.0205
ksiZE	3.4643	3.7684	3.9397	4.0953	4.1896	4.5866	4.4357	4.7535	4.6205	3.9162
	0.3447	0.4390	0.2862	0.1960	0.2253	0.2231	0.1773	0.3528	0.2879	0.2664
sigmaZE	1.7118	1.7167	1.8277	1.2950	1.3340	1.4225	1.4850	1.4303	1.4376	1.3529
	0.0239	0.0175	0.0263	0.0216	0.0156	0.0164	0.0181	0.0212	0.0275	0.0165
CorrRH	-0.6990	-0.3534	-0.4755	-0.3747	-0.4842	-0.5525	-0.5232	-0.4132	0.6285	-0.2334
	0.0698	0.0508	0.1086	0.1285	0.0497	0.0405	0.0366	0.0949	0.0609	0.0378

We can see from Table 21 that by introducing intraday estimators into the model, the size of the jumps (σ_J) decreased to about 2%, while the long-term jump intensity (λ_{LT}) increased to 1-3%. Strong co-jumps effects (significant γ_J) are observed on most of the time series, while the VXX exhibits even some moderately persistent jump-clusters ($\beta_J = 0.87$). The absolute correlation between volatility and returns (ρ) did increase for of the time series (when compared to the SVJD model), towards levels of about -0.5 for the stock ETF/ETN, and 0.6 for the VXX.

Table 22 shows the parameter estimates for the SVJJD-RV-Z model.

Table 22 – SVJJD-RV-Z model parameter estimates and Bayesian standard errors from the SGPF algorithm on the ETF/ETN time series

Model	EEM	EFA	FXI	GDX	IWM	QQQ	SPY	USO	VXX	XLF
mju	0.0013	0.0009	0.0010	0.0004	0.0009	0.0009	0.0008	0.0004	-0.0070	0.0004
	8.80E-05	5.13E-05	1.07E-04	1.36E-04	1.09E-04	4.01E-05	3.58E-05	1.18E-04	3.45E-04	4.62E-05
mjuJ	-0.0015	-0.0064	0.0073	0.0030	0.0024	0.0021	0.0025	0.0016	-0.0037	0.0020
	0.0038	0.0153	0.0163	0.0065	0.0039	0.0046	0.0030	0.0117	0.0146	0.0050
sigmaJ	0.0326	0.0488	0.0519	0.0468	0.0296	0.0332	0.0224	0.0539	0.0914	0.0347
	0.0034	0.0073	0.0076	0.0044	0.0033	0.0027	0.0023	0.0069	0.0102	0.0039
ltv	-4.9905	-10.1616	-9.0898	-7.6460	-9.0928	-7.6393	-9.2693	-7.4244	-6.8827	-10.2529
	1.0643	0.2994	0.0617	0.2589	0.1022	0.7862	0.1486	0.6880	0.0925	0.4943
beta	0.9678	0.9417	0.8237	0.9822	0.9412	0.9926	0.9572	0.9840	0.9030	0.9798
	0.0076	0.0091	0.0121	0.0074	0.0123	0.0027	0.0051	0.0064	0.0189	0.0044
gamma	0.0946	0.2265	0.1426	0.0918	0.1317	0.0633	0.1279	0.1110	0.2045	0.0734
	0.0078	0.0139	0.0068	0.0041	0.0195	0.0047	0.0071	0.0062	0.0138	0.0025
lambdaLT	0.0221	0.0042	0.0058	0.0230	0.0134	0.0149	0.0153	0.0070	0.0160	0.0124
	0.0050	0.0016	0.0026	0.0034	0.0030	0.0028	0.0026	0.0021	0.0035	0.0030
betaJ	0.8880	0.5557	0.2776	0.5059	0.3141	0.3720	0.2526	0.7510	0.6561	0.3039
	0.0718	0.2059	0.1480	0.1320	0.1611	0.2164	0.1461	0.1855	0.1327	0.1712
gammaJ	0.0483	0.0681	0.0775	0.0629	0.0580	0.0461	0.0718	0.0149	0.0750	0.0610
	0.0182	0.0280	0.0263	0.0201	0.0324	0.0222	0.0404	0.0216	0.0242	0.0192
mjuJV	-0.8485	1.0730	1.0693	0.4996	1.5968	0.1302	0.1792	-0.0254	0.9770	0.7795
	0.0518	0.0551	0.0548	0.2499	0.1178	0.0842	0.1283	0.1322	0.1667	0.2103
sigmaJV	0.2647	0.3284	0.3465	0.3304	0.3496	0.8027	0.8070	0.2902	0.3252	0.8581
	0.0230	0.0534	0.0370	0.0531	0.0639	0.0646	0.0756	0.0368	0.0543	0.1763
lambdaLTV	0.1188	0.0593	0.0799	0.0119	0.0185	0.0614	0.0570	0.0609	0.0146	0.0509
	0.0096	0.0110	0.0153	0.0057	0.0028	0.0076	0.0102	0.0087	0.0050	0.0065
betaJV	0.6637	0.3375	0.9746	0.5569	0.5570	0.3370	0.6683	0.3591	0.3050	0.2683
	0.2187	0.2180	0.0015	0.1466	0.2932	0.1954	0.2400	0.1738	0.1473	0.1727
gammaJV	0.0043	0.0102	0.0220	0.0125	0.0123	0.0171	0.0158	0.0505	0.0202	0.0187
	0.0052	0.0082	0.0016	0.0105	0.0091	0.0110	0.0138	0.0194	0.0143	0.0102
sigmaRV	0.6881	0.6913	0.7832	0.6210	0.5882	0.5740	0.5674	0.5902	0.6414	0.6079
	0.0098	0.0108	0.0129	0.0103	0.0092	0.0078	0.0076	0.0068	0.0104	0.0139
mjuZE	1.8862	1.9222	2.1324	1.0973	1.2072	1.2349	1.4577	1.4375	1.2469	1.2096
	0.0323	0.0262	0.0336	0.0231	0.0199	0.0249	0.0211	0.0301	0.0311	0.0227
ksiZE	3.8359	4.2136	4.4856	4.0484	4.5980	4.7994	4.8688	4.9249	5.3080	4.1538
	0.2513	0.5894	0.5046	0.1896	0.2615	0.2396	0.2100	0.4451	0.3095	0.2308
sigmaZE	1.6869	1.7269	1.8786	1.2920	1.3366	1.4275	1.4995	1.4319	1.4574	1.3371
	0.0238	0.0207	0.0254	0.0195	0.0160	0.0174	0.0154	0.0197	0.0222	0.0187
CorrRH	-0.7733	-0.3166	-0.3030	-0.5707	-0.4476	-0.3815	-0.5046	-0.4417	0.5992	-0.1131
	0.0726	0.0364	0.0949	0.0622	0.0490	0.0349	0.0331	0.0657	0.0378	0.0451

Table 23 – 2SVJJD-RV-Z model parameter estimates and Bayesian standard errors from the SGPF algorithm on the ETF/ETN time series

Model	EEM	EFA	FXI	GDX	IWM	QQQ	SPY	USO	VXX	XLF
mju	0.0014	0.0009	0.0008	0.0001	0.0009	0.0011	0.0008	0.0007	-0.0064	0.0006
	6.83E-05	4.49E-05	9.35E-05	1.30E-04	6.50E-05	5.25E-05	3.89E-05	1.24E-04	2.10E-04	4.78E-05
mjuJ	-0.0062	-0.0015	0.0016	0.0006	0.0016	0.0019	0.0028	0.0024	-0.0047	0.0041
	0.0081	0.0102	0.0144	0.0059	0.0058	0.0042	0.0028	0.0123	0.0176	0.0076
sigmaJ	0.0403	0.0448	0.0522	0.0455	0.0311	0.0326	0.0233	0.0515	0.0894	0.0473
	0.0059	0.0067	0.0105	0.0043	0.0036	0.0028	0.0020	0.0062	0.0100	0.0094
ltv	-10.3855	-8.4020	-9.3009	-7.4888	-7.6328	-8.6302	-8.5384	-8.5792	-6.9574	-9.5875
	0.0710	0.0125	0.7401	0.1006	0.0153	3.6194	0.0061	0.7436	5.8169	0.0198
beta	0.9946	0.9820	0.8944	0.9706	0.9939	0.9978	0.9674	0.9732	0.9986	0.9212
	0.0018	0.0070	0.0179	0.0101	0.0039	0.0022	0.0060	0.0118	0.0010	0.0115
gamma	0.0973	0.2238	0.2613	0.1606	0.1547	0.0911	0.1497	0.1264	0.1199	0.1634
	0.0040	0.0077	0.0121	0.0086	0.0071	0.0066	0.0047	0.0075	0.0052	0.0106
lambdaLT	0.0101	0.0239	0.0078	0.0242	0.0103	0.0159	0.0146	0.0080	0.0158	0.0109
	0.0039	0.0071	0.0045	0.0058	0.0026	0.0024	0.0024	0.0022	0.0036	0.0044
betaJ	0.6372	0.9783	0.7005	0.8635	0.3274	0.4696	0.4141	0.5665	0.5885	0.1538
	0.1461	0.0004	0.1880	0.1966	0.1780	0.1987	0.2169	0.3682	0.1695	0.1113
gammaJ	0.0863	0.0209	0.0407	0.0137	0.0905	0.0607	0.0351	0.0201	0.0319	0.1456
	0.0268	0.0008	0.0223	0.0100	0.0331	0.0265	0.0135	0.0156	0.0159	0.0238
mjuJV	1.3315	0.2011	1.1801	0.2147	0.6784	0.3884	-0.1401	0.1386	0.6714	0.4768
	0.1121	0.1291	0.1680	0.2156	0.2167	0.1626	0.0435	0.1012	0.1730	0.0588
sigmaJV	0.3060	0.8158	0.3791	0.3281	0.7807	0.6625	0.3567	0.3056	0.4399	0.2931
	0.0445	0.1097	0.0427	0.0617	0.2047	0.1077	0.0366	0.0491	0.1344	0.0508
lambdaLTV	0.0239	0.0505	0.0765	0.0314	0.0275	0.0463	0.0912	0.0572	0.0496	0.0624
	0.0038	0.0057	0.0062	0.0095	0.0057	0.0111	0.0076	0.0085	0.0088	0.0156
betaJV	0.6489	0.1983	0.2126	0.4265	0.5295	0.2472	0.2161	0.3839	0.2610	0.9260
	0.1803	0.1311	0.1551	0.1557	0.2463	0.1464	0.1526	0.2639	0.2157	0.0054
gammaJV	0.0121	0.0104	0.0091	0.0354	0.0131	0.0189	0.0302	0.0207	0.0352	0.0583
	0.0092	0.0082	0.0108	0.0170	0.0102	0.0167	0.0151	0.0156	0.0254	0.0067
phi1	0.6107	0.5581	0.9863	0.8028	0.2054	0.9985	0.5776	0.9862	0.9991	0.4405
	0.0857	0.0423	0.0087	0.0745	0.1074	0.0022	0.0270	0.0116	0.0021	0.0668
phi2	0.1088	0.1065	0.0831	0.0999	0.0892	0.0886	0.0980	0.0969	0.1128	0.0763
	0.0110	0.0035	0.0073	0.0041	0.0073	0.0099	0.0033	0.0076	0.0085	0.0051
sigmaRV	0.7367	0.6956	0.7039	0.5983	0.5808	0.5700	0.5768	0.5799	0.6458	0.6058
	0.0116	0.0097	0.0119	0.0087	0.0080	0.0071	0.0072	0.0077	0.0121	0.0069
mjuZE	1.9375	1.9178	2.1273	1.0874	1.2194	1.2276	1.4611	1.4338	1.2512	1.2204
	0.0329	0.0272	0.0339	0.0306	0.0194	0.0243	0.0251	0.0236	0.0346	0.0222
ksiZE	4.5078	3.9850	4.5869	4.1097	4.8427	4.7545	4.7284	4.8436	5.2725	4.1471
	0.3767	0.4125	0.5183	0.2013	0.2998	0.2006	0.2298	0.3809	0.2959	0.3154
sigmaZE	1.7344	1.7119	1.8763	1.2861	1.3490	1.4188	1.5041	1.4261	1.4603	1.3515
	0.0248	0.0179	0.0293	0.0253	0.0158	0.0164	0.0162	0.0203	0.0249	0.0194
CorrRH	-0.7810	-0.2604	-0.1028	-0.1953	-0.3319	-0.6168	-0.5456	-0.5559	0.7738	-0.2349
	0.0421	0.0337	0.0491	0.0516	0.0410	0.0726	0.0414	0.0887	0.0428	0.0437

We can see from Table 22 that the price jumps in the SVJJD-RV-Z model are on average larger than in the SVJD-RV-Z model, with the σ_J being around 5% for most of the stock ETF/ETN and 9% for VXX. Jumps in returns also seem to occur quite often, with λ_{LT} being around 1-2% for most of the time series. Persistent clustering of the jumps in returns is observed on EEM ($\beta_J = 0.89$), while most of the other ETF/ETN exhibit co-jumps (significant γ_J). Volatility jumps occur mostly upwards (positive μ_{JV}) and the occur in about 1-8% of the days ($\lambda_{LT,V}$), depending on the respective time series. The volatility jumps exhibit highly persistent and significant clustering on FXI ($\beta_{JV} = 0.97$ with $\gamma_J = 0.022$), and statistically significant co-jump effects (significant γ_J with low β_{JV}) on USO.

Table 23 shows the parameter estimates for the 2SVJJD-RV-Z model. We can see that similarly to the currency time series, the value of ϕ_1 is not always higher than β, indicating that the SGPF algorithm estimated the persistence of the two components oppositely than what we expected. This is not particularly problematic per-se, and it instead indicates that the volatility jumps and correlation between volatility and jumps that we assumed for the short-term log-variance component do rather occur in the long-term component. The results further show that EFA exhibits highly persistent price-jump clustering ($\beta_J = 0.98$ with $\gamma_J = 0.02$), while the XLF exhibits persistent volatility-jump clustering ($\beta_{JV} = 0.93$ with $\gamma_{JV} = 0.06$). Several other series were estimated to contain statistically significant co-jumps in volatility, including the SPY.

5.5.3. ETF/ETN volatility forecasting results

Realized variance for each of the ETF/ETN time series was forecasted in the 1-day, 5-day and the 22-day horizon and the historical accuracy of the forecasts was assessed with the R-Squared criterion. The multi-period forecasts were performed with direct projection for the HAR and ESN models (as in Bollerslev et al, 2015), in which the model is re-estimated with the multi-period realized variance as the target. For the logarithmic and square root transformations of the model the forecasts were then transformed back into realized variance forecasts for the computation of the R-Squared values. For the SVJD model the forecasts were constructed by randomly sampling parameter and latent state particles from the Particle Filter, constructing 10 000 simulations of the evolution of the latent states into the future (for the respective horizons), computing the quadratic variation for each simulation and then the average of all simulations.

Table 24 shows the R-Squared values for all of the models and ETF/ETN time series for the 1-day forecast horizon. In the last two columns is the average R-Squared and the average rank of the model over all of the forecasted series. The colouring indicates the size of the R-Squared (green is higher and red lower) and is applied separately for each ETF/ETN.

Table 24 – R-Squared of the 1-Day realized variance forecasts (ETF/ETN)

Type	Model	EEM	EFA	FXI	GDX	IWM	QQQ	SPY	USO	VXX	XLF	AvgR2	AvgRank
GARCH	GARCH	0.3770	0.2031	0.3790	0.2318	0.2010	0.2711	0.2163	0.3407	0.1982	0.3534	0.2771	22.40
	EGARCH	0.2011	0.1581	0.3982	0.1978	0.1612	0.4282	0.2440	0.3557	0.2534	0.2343	0.2632	23.30
	GJRGARCH	0.4027	0.2176	0.4028	0.2176	0.2448	0.2676	0.2767	0.3656	0.1865	0.3548	0.2937	17.70
HAR	HAR	0.4008	0.1837	0.3896	0.2090	0.2965	0.1691	0.2359	0.3268	0.2465	0.3194	0.2777	21.80
	LogHAR	0.1647	0.1439	0.3496	0.2390	0.2264	0.4392	0.1984	0.3730	0.1440	0.2964	0.2575	26.40
	SqrtHAR	0.3648	0.1916	0.4003	0.2514	0.2855	0.4198	0.2429	0.3604	0.2192	0.3443	0.3080	18.20
	HARQ	0.1000	0.1767	0.3695	0.1005	0.1149	0.4603	0.2410	0.3501	0.1635	0.3432	0.2420	25.20
	LogHARQ	0.2132	0.1614	0.3469	0.2550	0.2393	0.4620	0.2432	0.3824	0.1460	0.3406	0.2790	19.90
	SqrtHARQ	0.0744	0.1051	0.3221	0.2440	0.1081	0.4862	0.1043	0.3733	0.1921	0.3628	0.2372	25.00
	HARJ	0.3988	0.1467	0.3854	0.0711	0.3320	0.1873	0.2832	0.3181	0.2971	0.4353	0.2855	18.70
	logHARJ	0.1557	0.1357	0.3641	0.2398	0.1569	0.4537	0.1729	0.3595	0.1601	0.3080	0.2507	26.80
	sqrtHARJ	0.3879	0.1572	0.4021	0.2457	0.3187	0.4430	0.2758	0.3550	0.2786	0.4504	0.3314	13.90
	AHAR	0.4062	0.1963	0.4128	0.0821	0.2960	0.2591	0.2694	0.3474	0.2732	0.2709	0.2813	19.30
	logAHAR	0.4416	0.2051	0.3980	0.2449	0.3436	0.5174	0.4468	0.3885	0.3340	0.4509	0.3771	5.90
	sqrtAHAR	0.4046	0.2086	0.4362	0.2551	0.3064	0.4912	0.3153	0.3806	0.2719	0.3975	0.3467	7.40
	SHAR	0.4452	0.1596	0.2837	0.1229	0.2971	0.3191	0.2339	0.3417	0.2732	0.3482	0.2825	21.40
	logSHAR	0.1760	0.1376	0.3501	0.2459	0.2356	0.4552	0.2175	0.3732	0.1770	0.3154	0.2683	23.80
	sqrtSHAR	0.4080	0.1722	0.4063	0.2526	0.2983	0.4566	0.2546	0.3699	0.2549	0.3320	0.3205	14.60
ESN	ESN	0.1989	-0.0185	0.3102	0.2062	0.2132	0.3911	-0.0085	0.3392	0.0878	0.3732	0.2093	30.00
	LogESN	0.4353	0.2011	0.3616	0.2472	0.2920	0.4710	0.2892	0.3788	0.1400	0.4565	0.3273	14.00
	SqrtESN	0.3631	0.1262	0.3483	0.2619	0.2949	0.4897	0.2225	0.3799	0.1482	0.4184	0.3053	17.50
	ESNQ	0.2269	-0.2447	0.3150	0.2023	0.1953	0.3856	0.0031	0.3363	0.0815	0.3294	0.1831	31.80
	LogESNQ	0.4334	0.2039	0.3769	0.2527	0.3043	0.4808	0.3183	0.3790	0.1511	0.4715	0.3372	10.20
	SqrtESNQ	0.3364	0.0932	0.3625	0.2643	0.3014	0.4890	0.1977	0.3788	0.0989	0.3573	0.2880	19.50
	ESNJ	0.1694	-0.0950	0.2418	0.1625	0.1107	0.3573	0.0331	0.3011	0.1280	0.3327	0.1742	33.60
	LogESNJ	0.4763	0.1924	0.3077	0.2506	0.2471	0.4702	0.2470	0.3760	0.1775	0.4231	0.3168	15.90
	SqrtESNJ	0.3322	0.0483	0.3444	0.2535	0.2677	0.4910	0.2877	0.3739	0.2298	0.4127	0.3041	17.30
	SESN	0.1805	-0.0463	0.1788	0.1989	0.1887	0.4263	-0.0327	0.3308	0.1301	0.3556	0.1911	31.30
	LogSESN	0.4549	0.2010	0.3882	0.2468	0.2994	0.4915	0.3090	0.3777	0.1492	0.4718	0.3389	10.50
	SqrtSESN	0.3623	0.1214	0.3531	0.2605	0.3025	0.5255	0.2228	0.3743	0.2125	0.4362	0.3171	15.10
	SESNQ	0.2188	-0.0349	0.2700	0.1974	0.2069	0.4196	0.0025	0.3309	0.0905	0.3826	0.2084	30.40
	LogSESNQ	0.4443	0.2027	0.3958	0.2535	0.3076	0.4958	0.3240	0.3767	0.1609	0.4837	0.3445	8.10
	SqrtSNESNQ	0.3485	0.1046	0.3316	0.2668	0.3097	0.5173	0.2049	0.3679	0.1457	0.4417	0.3039	17.80
SVJD	SVJD	0.4041	0.2316	0.3743	0.2305	0.2019	0.4255	0.2211	0.3704	0.1295	0.4020	0.2991	20.20
	SVJD-RV-Z	0.4430	0.2015	0.4322	0.2650	0.1954	0.4719	0.2316	0.3952	0.1029	0.4639	0.3203	12.90
	SVJJD-RV-Z	0.4155	0.2110	0.3632	0.2751	0.2690	0.4920	0.2535	0.4130	0.2318	0.4734	0.3398	9.00
	2SVJJD-RV-Z	0.4257	0.2244	0.4055	0.2841	0.2947	0.5165	0.2555	0.4217	0.2477	0.4720	0.3548	6.20

We can see from Table 24 that the best average R-Squared and the best average rank were achieved by the logarithmic version of the AHAR model (logAHAR), which managed to slightly outperform even the SVJD models.

SVJD models were the second best (after the logAHAR). The best performing SVJD model was 2SVJJD-RV-Z, followed by the SVJJD-RV-Z and then the SVJD-RV-Z. The performance of the standard SVJD model was slightly better than of the GARCH model.

Among the HAR models, the sqrtAHAR models was the second best after logAHAR. The HARJ model, which was the best on the currency time series, achieved only mediocre results on the ETF/ETN time series. Overall, the good performance of AHAR model specification is to be expected on the ETF/ETN time series, due to the strong correlation between volatility and returns.

The best out of the ESN models was the LogSESNQ model, which achieved comparable results to the best of the HAR models and SVJD models. The LogSESNQ model was also the best ESN model on the currency time series, which indicates its robustness. Similarly as in the case of currencies, the ESN models using the logarithmic transformation performed best, followed by the square root transformation. In addition to the LogSESNQ model, the LogESNQ and LogSESN models performed also very well.

Among the GARCH models, the GJR-GARCH was the best, managing to outperform even some of the HAR models.

The results for the 1-Week (5-Day) horizon are shown in Table 25. We can see that the SVJD models outperformed all of the other models on 5-day horizon, achieving the highest R-Squared (SVJD-RV-Z) as well as the lowest average rank (2SVJJD-RV-Z).

The best HAR models were the logarithmic and square root versions of the AHAR model (the same as on the 1-Day horizon), followed by the sqrtHARJ model and the sqrtSHAR model.

ESN models, such as the LogSESNQ achieved only mediocre performance on the 5-day forecast horizon, mainly due to their bad results on VXX.

Table 25 – R-Squared of the 5-Day realized variance forecasts (ETF/ETN)

Type	Model	EEM	EFA	FXI	GDX	IWM	QQQ	SPY	USO	VXX	XLF	AvgR2	AvgRank
GARCH	GARCH	0.5599	0.4098	0.5826	0.3751	0.3247	0.2596	0.3942	0.4580	-0.0087	0.3708	0.3726	19.60
	EGARCH	0.2126	0.2929	0.6181	0.2967	0.2342	0.5118	0.3503	0.5060	0.1019	0.1939	0.3318	21.10
	GJRGARCH	0.5466	0.4271	0.6128	0.3166	0.3734	0.2245	0.4694	0.4991	-0.2165	0.3734	0.3627	18.50
HAR	HAR	0.5650	0.3059	0.6005	0.3393	0.4221	0.0448	0.3575	0.4493	0.0705	0.2595	0.3414	20.60
	LogHAR	0.2491	0.2811	0.5336	0.4142	0.3495	0.5378	0.3147	0.5317	0.0871	0.4134	0.3712	19.80
	SqrtHAR	0.5205	0.3141	0.6034	0.3968	0.4163	0.4714	0.3714	0.5047	0.0796	0.3922	0.4070	17.00
	HARQ	0.0840	0.3544	0.4567	0.1422	0.0546	0.4675	0.3453	0.4718	0.0661	0.5652	0.3008	25.60
	LogHARQ	0.2883	0.3112	0.5152	0.4312	0.3591	0.5807	0.3445	0.5430	0.0840	0.4699	0.3927	17.30
	SqrtHARQ	0.1632	0.2285	0.3604	0.3502	0.1371	0.5793	0.2186	0.5127	0.0708	0.4900	0.3111	23.30
	HARJ	0.5763	0.2968	0.6326	0.0149	0.4730	0.0361	0.3864	0.4461	0.0678	0.6203	0.3550	16.90
	logHARJ	0.3107	0.3102	0.5643	0.4233	0.3104	0.5663	0.2959	0.5169	0.0788	0.4466	0.3823	18.40
	sqrtHARJ	0.5460	0.2517	0.6151	0.3469	0.4657	0.4833	0.3964	0.5019	0.0861	0.6257	0.4319	12.50
	AHAR	0.5541	0.3480	0.6249	0.2311	0.4135	0.0826	0.3949	0.5033	0.1093	0.4294	0.3691	16.10
	logAHAR	0.4350	0.3359	0.5544	0.4183	0.4193	0.5831	0.4705	0.5645	0.1135	0.5579	0.4452	9.70
	sqrtAHAR	0.5390	0.3666	0.6310	0.3952	0.4236	0.5032	0.4370	0.5480	0.1021	0.5235	0.4469	9.50
	SHAR	0.5820	0.3225	0.5738	0.2498	0.4262	0.1820	0.3602	0.4904	0.0563	0.3548	0.3598	19.20
	logSHAR	0.2534	0.2787	0.5379	0.4189	0.3567	0.5509	0.3342	0.5214	0.0926	0.4350	0.3780	18.60
	sqrtSHAR	0.5490	0.3199	0.6080	0.3997	0.4294	0.5021	0.3851	0.5225	0.0772	0.4227	0.4216	13.10
ESN	ESN	0.5317	0.0680	0.5449	0.1838	0.3084	0.4014	0.2001	0.4796	-0.2709	0.4703	0.2917	29.10
	LogESN	0.5795	0.2616	0.5196	0.2996	0.4169	0.4961	0.3547	0.4653	-0.1718	0.6002	0.3822	21.10
	SqrtESN	0.5711	0.2108	0.5509	0.3100	0.3820	0.5011	0.3020	0.4906	-0.1903	0.5024	0.3631	22.00
	ESNQ	0.5301	0.0888	0.5533	0.1831	0.3084	0.3926	0.2040	0.4767	-0.3085	0.4490	0.2877	29.70
	LogESNQ	0.5901	0.2543	0.5465	0.3166	0.4249	0.5046	0.3746	0.4673	-0.1731	0.6104	0.3916	17.60
	SqrtESNQ	0.5484	0.2125	0.5546	0.3112	0.3690	0.5140	0.2872	0.4923	-0.1327	0.4487	0.3605	22.20
	ESNJ	0.5297	0.0652	0.5395	0.1193	0.3115	0.3374	0.2072	0.4751	-0.2920	0.4583	0.2751	30.60
	LogESNJ	0.5073	0.2343	0.5252	0.2994	0.4070	0.5373	0.3454	0.4842	0.0581	0.5775	0.3976	21.00
	SqrtESNJ	0.5700	0.1927	0.5483	0.2952	0.3859	0.5016	0.3108	0.4838	-0.1333	0.5213	0.3676	22.30
	SESN	0.5336	0.0474	0.5494	0.1379	0.3154	0.3456	0.2063	0.4784	-0.2836	0.4760	0.2806	29.00
	LogSESN	0.5882	0.2674	0.5177	0.3028	0.4200	0.5153	0.3594	0.4577	-0.1563	0.6188	0.3891	19.40
	SqrtSESN	0.5732	0.1640	0.5789	0.3084	0.3926	0.5258	0.3171	0.4902	-0.1523	0.5374	0.3735	19.30
	SESNQ	0.5327	0.0640	0.5534	0.1401	0.3134	0.3524	0.2085	0.4781	-0.2940	0.4531	0.2802	29.10
	LogSESNQ	0.6004	0.2678	0.5346	0.3205	0.4281	0.5163	0.3731	0.4589	-0.1548	0.6223	0.3967	16.20
	SqrtSNESNQ	0.5606	0.1896	0.5654	0.3125	0.3738	0.5350	0.3002	0.4934	-0.1316	0.5002	0.3699	19.90
SVJD	SVJD	0.5989	0.4645	0.5334	0.3671	0.3240	0.5493	0.3737	0.5122	0.1013	0.5444	0.4369	12.60
	SVJD-RV-Z	0.6194	0.3536	0.6353	0.4467	0.2816	0.5919	0.3431	0.5592	0.0904	0.6409	0.4562	8.30
	SVJJD-RV-Z	0.5899	0.3622	0.4588	0.4459	0.3819	0.5888	0.3754	0.5681	0.1205	0.5974	0.4489	9.30
	2SVJJD-RV-Z	0.5957	0.4151	0.5583	0.4481	0.4455	0.6005	0.3807	0.5552	-0.1616	0.6153	0.4453	7.50

The results for the 1-Month (22-Days) horizon are shown in Table 26.

Table 26 – R-Squared of the 22-Day realized variance forecasts (ETF/ETN)

Type	Model	EEM	EFA	FXI	GDX	IWM	QQQ	SPY	USO	VXX	XLF	AvgR2	AvgRank
GARCH	GARCH	0.4174	0.4825	0.4569	0.3534	0.2784	-0.5112	0.3466	0.4069	-0.1898	0.1671	0.2208	19.90
	EGARCH	0.0703	0.2711	0.6411	0.2996	0.1658	0.1658	0.2311	0.5090	0.0048	0.0641	0.2423	22.80
	GJRGARCH	0.3321	0.4932	0.5093	0.2595	0.2870	-0.5650	0.3972	0.4816	-0.5154	0.1687	0.1848	21.30
HAR	HAR	0.4349	0.3857	0.6044	0.1842	0.3737	-1.1510	0.2754	0.4201	-0.0485	0.1726	0.1652	18.40
	LogHAR	0.0955	0.2906	0.5849	0.4063	0.2991	0.3830	0.2509	0.4807	-0.0163	0.3413	0.3116	19.00
	SqrtHAR	0.3316	0.3914	0.6155	0.3512	0.3619	0.1068	0.2992	0.4457	-0.0354	0.3155	0.3183	17.00
	HARQ	0.0207	0.1555	0.2910	0.0221	0.0495	0.0412	0.2552	0.3119	-0.0133	0.4385	0.1572	29.30
	LogHARQ	0.1122	0.3158	0.5463	0.4604	0.2966	0.4281	0.2591	0.4925	-0.0197	0.3890	0.3280	16.60
	SqrtHARQ	0.0198	0.1631	0.3077	0.2496	0.0728	0.4445	0.1698	0.4235	-0.0092	0.4140	0.2255	24.90
	HARJ	0.5101	0.4273	0.6086	-0.6422	0.4327	-1.2086	0.3039	0.4189	-0.0713	0.5934	0.1373	14.80
	logHARJ	0.1148	0.3475	0.5430	0.4219	0.2820	0.4595	0.2335	0.4737	-0.0263	0.3813	0.3231	18.00
	sqrtHARJ	0.3886	0.4153	0.6115	0.1634	0.4294	0.0490	0.3203	0.4576	-0.0504	0.5349	0.3320	15.20
	AHAR	0.4178	0.3850	0.5896	-1.0197	0.3447	-1.2039	0.2709	0.5175	-0.0129	0.3967	0.0686	16.60
	logAHAR	0.1424	0.3712	0.4849	0.4029	0.3107	0.3723	0.3212	0.5399	-0.0025	0.4797	0.3423	14.80
	sqrtAHAR	0.3328	0.4248	0.5827	0.2557	0.3410	0.0541	0.3147	0.5285	-0.0193	0.4848	0.3300	15.20
	SHAR	0.4609	0.3917	0.5067	-0.2844	0.3761	-0.8151	0.2687	0.5421	-0.0515	0.2537	0.1649	18.00
	logSHAR	0.1023	0.2857	0.5875	0.4044	0.3033	0.3832	0.2571	0.4356	-0.0152	0.3629	0.3107	18.70
	sqrtSHAR	0.3487	0.3931	0.6055	0.2722	0.3690	0.1532	0.2952	0.5017	-0.0452	0.3607	0.3254	16.20
ESN	ESN	0.4102	0.3177	0.5266	0.3573	0.2365	0.3502	0.1562	0.3715	-0.4690	0.6193	0.2877	23.00
	LogESN	0.3935	0.3423	0.3944	0.3939	0.2706	0.3807	0.1406	0.3494	-0.6399	0.6198	0.2645	24.30
	SqrtESN	0.4560	0.3730	0.5607	0.4181	0.2987	0.3973	0.2262	0.3761	-0.4312	0.6000	0.3275	16.00
	ESNQ	0.4094	0.3207	0.5170	0.3553	0.2357	0.3511	0.1483	0.3587	-0.4858	0.6155	0.2826	24.40
	LogESNQ	0.3943	0.3506	0.3845	0.4034	0.2821	0.3879	0.1668	0.3390	-0.6266	0.6295	0.2712	21.70
	SqrtESNQ	0.4516	0.3747	0.5795	0.4239	0.3080	0.4150	0.2389	0.3780	-0.3940	0.5959	0.3372	14.20
	ESNJ	0.4116	0.3210	0.5293	0.3111	0.2367	0.2818	0.1550	0.3587	-0.4636	0.6131	0.2755	24.00
	LogESNJ	0.4856	0.3790	0.6295	0.4000	0.3812	0.4078	0.2805	0.3572	-0.4282	0.6122	0.3505	12.10
	SqrtESNJ	0.4601	0.3643	0.5968	0.4243	0.2985	0.3862	0.2403	0.3650	-0.4282	0.6038	0.3311	15.20
	SESN	0.4010	0.3211	0.5265	0.3383	0.2359	0.2968	0.1568	0.3768	-0.4655	0.6059	0.2794	23.90
	LogSESN	0.4007	0.3459	0.3631	0.3950	0.2742	0.3857	0.1408	0.3452	-0.6548	0.6328	0.2629	23.80
	SqrtSESN	0.4420	0.3610	0.5742	0.4159	0.2918	0.3902	0.2409	0.3799	-0.4396	0.5983	0.3255	16.60
	SESNQ	0.4117	0.3250	0.5173	0.3354	0.2375	0.3025	0.1561	0.3663	-0.4613	0.6117	0.2802	23.30
	LogSESNQ	0.4058	0.3475	0.3582	0.4028	0.2839	0.3897	0.1648	0.3367	-0.6412	0.6373	0.2685	22.10
	SqrtSNESNQ	0.4508	0.3771	0.5763	0.4173	0.2996	0.4041	0.2533	0.3772	-0.3882	0.5973	0.3365	14.50
SVJD	SVJD	0.3463	0.4841	0.2052	0.3241	0.2920	0.3790	0.2849	0.4695	0.0084	0.4269	0.3220	17.40
	SVJD-RV-Z	0.0000	0.3862	0.3468	0.4862	0.4776	0.2172	0.4791	-0.0270	0.0000	0.0180	0.2384	18.80
	SVJJD-RV-Z	0.3816	0.4083	0.3668	0.4603	0.2690	0.3940	0.3005	0.4983	0.0181	0.3681	0.3465	14.40
	2SVJJD-RV-Z	0.1296	0.5182	-0.3609	0.4504	0.4010	0.3722	0.3119	0.4742	-3.2983	0.5789	-0.0423	16.60

We can see that the SVJJD-RV-Z model was the second best, while the other SVJD models performed rather poorly on the 22-day forecast horizon, especially the 2SVJJD-RV-Z model, due to its poor performance on the VXX and the FXI time series.

The best results on the 22-Day horizon were achieved by the LogESNJ model, in spite of its highly negative R-Squared on VXX. The performance of all models on VXX was rather poor on the 22-Day horizon, which can be attributed to the fact that VXX is a volatility ETN that simulates a position in the futures on the volatility index VIX. The volatility of VXX is thus de-facto the volatility of the volatility of S&P500, which is very challenging to predict, especially in the longer forecast horizons.

Among other notable models is the SqrtESNQ model and the log and sqrt transformations of the AHAR model, which was also performing well on the shorter horizons, making it probably the best HAR model for the ETF/ETN time series. Other successful HAR model specifications were the square root forms of the HARJ and the SHAR model.

6. Conclusion

The thesis reviews the most commonly used volatility forecasting frameworks, with the main focus placed on Stochastic-Volatility Jump-Diffusion (SVJD) models, on the ways of how information from intraday data can be integrated into these models, and on Bayesian methods used for SVJD model estimation.

Among the main contributions of the thesis is the development of SVJD-RV-Z class of models that use the realized variance, computed from intraday returns, as an additional source of information in the estimation of the stochastic variance, and the Z-Estimator of jumps as an additional source of information for the estimation of price jumps. A further contribution is the development of adapted particle filters for more efficient filtering of the latent states (i.e. volatility and jumps) in SVJD models, and the Sequential-Gibbs Particle Filter (SGPF) algorithm, which can be used for the sequential estimation of SVJD model parameters. An additional contribution, not related to SVJD models, is the application of Echo State Neural networks (ESN) for realized volatility forecasting.

In the empirical part of the thesis four complex SVJD and SVJD-RV-Z models are applied to the time series of 7 currency exchange rates and 10 ETF/ETN securities with the goal of forecasting the future realized variance in the 1-Day, 1-Week and 1-Month horizon. The models are sequentially estimated with the SGPF algorithm and include many complex features such as self-exciting jumps in prices and volatility as well as multiple volatility components. The predictive power of the SVJD models is compared with three ARCH/GARCH models (GARCH, EGARCH and GJR-GARCH), 15 HAR models (HAR, AHAR, HARJ, SHAR and HARQ, and their square-root and logarithmic modifications), and 15 ESN models (with predictors analogical to the HAR models).

The results of the empirical study show that SVJD-RV-Z models with self-exciting jumps in prices and volatility provide volatility forecasts that are for most of the analysed time series comparable or better than the best benchmark models. Among the ESN models, the best performance was achieved on the shorter forecast horizons by the LogSESNQ model (ESN with realized variance, realized quarticity and realized semi-variance as predictors), while on the longer horizons the LogESNJ model proved superior (ESN with realized variance and bi-power variation based jump variance estimates as predictors). Among the HAR models, the HARJ model performed best on the currency exchange rate time series, while the logAHAR was the best on ETF/ETN time series.

In his future research, the author plans to focus more on the optimal SVJD model architecture. Specifically, the long-range dependencies in the volatility process may instead of the two-component approach be also modelled with a genuine LMSV model, with the Markov-Switching based approaches or with the Markov-Switching-Multifractal approach.

An additional extension would be to include more elaborate modelling of the interdependencies between the volatility components, returns and jumps. Both components of the two-component SVJD model can be assumed to be correlated with the returns. Similarly, the price and volatility jump occurrences and sizes can be made to be correlated, and the self-exciting jump intensities can be made to be cross-exciting. Copulas may further be used to capture the complex interdependencies between different components of the model.

As a third possible extension, the noise distribution in the return, volatility and jump processes can be assumed to follow a different distribution than the normal one.

Regarding the use of power-variation estimators in SVJD models, a natural next step would be to view the noise of the realized variance as heteroskedastic and utilize the realized quarticity for its estimation. Similarly, the realized semi-variance may be used to better capture the asymmetries of the volatility process, and the incorporation of the Z-Statistics into SVJD models may also be improved by more accurately modelling the dependency between the sizes of the Z-Statistics and the jump sizes. Finally, as an additional source of information, option-based volatility forecasts can be integrated into the SVJD model setting, with the possibility of modelling the volatility-risk premium as a stochastic latent state process.

Regarding the particle filter based estimation, an improvement of convergence may be achieved by approximating the intractable densities in a way similar to what was used in Jacquier et al. (1994) or Kim, Shephard and Chib (1998) in the MCMC estimation framework, while the volatility jump sizes may be approximately adapted to the returns in a way outlined in Fičura and Witzany (2018). Alternatively, approaches using the Auxiliary Particle Filter (Pitt and Shephard, 199), or the Approximate Rao-Blackwelization (Johansen, 2012) may be used to improve the sampling in models with intractable proposal densities.

7. Appendix – Approximate adaptations

In the following sections, we will derive approximate proposal densities for some of the latent states of the SVJD-RV-Z model, where the fully-adapted densities are intractable, but we can still at least partially utilize the information about the observed $r(t)$, $RV(t)$, $Z(t)$ and also $BV(t)$. Take note that even if the sampling proceeds from an approximate density that differs from the ideal fully adapted density, the particle filtering is still asymptotically unbiased, due to the re-weighting and the re-sampling step, which guarantee that the final sample will be from the true posterior density.

7.1. SVJD-RV-Z model with price jumps – Sampling of h_t^i

The sampling of h_t^i involves two major hurdles. The first one is that the distribution $p(h_t^i|r_t)$ is intractable due to the non-linearity of the relationship between h_t^i and r_t. The second hurdle is related to the distribution $p(h_t^i|RV_t)$. As the assumed relationship between h_t^i and RV_t in the analysed models is given by $\log[RV(t) - J^2(t)Q(t)] = h(t) + \sigma_{RV}\varepsilon_{\mathrm{RV}}(t)$, it is apparent that as long as we know the values of J_t^i and Q_t^i, we can easily express the density $p(h_t^i|RV_t, J_t^i, Q_t^i)$. Unfortunately, as long as we want to sample the values of J_t^i and Q_t^i in an adapted way, we need to know the value of h_t^i before. The values of J_t^i and Q_t^i are thus unavailable during the sampling of h_t^i, and the marginalized density $p(h_t^i|RV_t)$ is unfortunately intractable.

In order to sample the values of h_t^i from a density that is at least approximately adapted to the observed values of RV_t, we will utilize an adaptation scheme in which h_t^i will be sampled from a mix of two Gaussian densities. The proposal distribution can be derived as follows:

1. Calculate provisional probabilities of jump occurrence $p(Q_t^i = 1|Z_t^i)$. These can be calculated from the likelihood densities with respect to the Z_t^i, given as:

$$p(Z_t^i|Q_t^i = 0) \sim N(Z_t; \mu_Z, \sigma_Z)(1 - \lambda_t^i)$$

$$p(Z_t^i|Q_t^i = 1) \sim N(Z_t; \mu_Z + \xi_Z, \sigma_Z)\lambda_t^i$$

The provisional jump probability $p(Q_t^i = 1|Z_t^i)$ can thus be computed as:

$$p(Q_t^i = 1|Z_t^i) = \frac{p(Z_t^i|Q_t^i = 1)}{p(Z_t^i|Q_t^i = 0) + p(Z_t^i|Q_t^i = 1)}$$

The probability of no jump is then equal to $p\left(Q_t^i = 0|Z_t^i\right) = 1 - p\left(Q_t^i = 1|Z_t^i\right)$.

We will further denote $p_0^* = p\left(Q_t^i = 0|Z_t^i\right)$ and $p_1^* = p\left(Q_t^i = 1|Z_t^i\right)$

2. In the no-jump case of $Q_t^i = 0$, we could derive the proposal density for h_t^i, adapted to RV_t, based on:

$$p\left(h_t^i|RV_t, h_{t-1}^i, Q_t^i = 0\right) \propto N\left(h_t^i; \alpha + \beta h_{t-1}^i, \gamma\right) N\left(\log(RV_t); h_t^i, \sigma_{RV}\right)$$

The two Gaussian densities can then be combined to get:

$$p\left(h_t^i|RV_t, h_{t-1}^i, Q_t^i = 0\right) = N\left(h_t^i; \frac{\log(RV_t)\gamma^2 + (\alpha + \beta h_{t-1}^i)\sigma_{RV}^2}{\sigma_{RV}^2 + \gamma^2}, \frac{\sigma_{RV}\gamma}{\sqrt{\sigma_{RV}^2 + \gamma^2}}\right)$$

In the case of $Q_t^i = 1$, we will have to use the approximation that:

$$\log[RV(t) - J^2(t)Q(t)] \cong \log[BV(t)]$$

and thus

$$\log[BV(t)] \cong h(t) + \sigma_{RV}\varepsilon_{\mathrm{RV}}(t)$$

Analogically to the previous case, it will approximately hold that:

$$p\left(h_t^i|BV_t, h_{t-1}^i, Q_t^i = 1\right) \propto N\left(h_t^i; \alpha + \beta h_{t-1}^i, \gamma\right) N\left(\log(BV_t); h_t^i, \sigma_{RV}\right)$$

Which can be transformed to get:

$$p\left(h_t^i|BV_t, h_{t-1}^i, Q_t^i = 1\right) = N\left(h_t^i; \frac{\log(BV_t)\gamma^2 + (\alpha + \beta h_{t-1}^i)\sigma_{RV}^2}{\sigma_{RV}^2 + \gamma^2}, \frac{\sigma_{RV}\gamma}{\sqrt{\sigma_{RV}^2 + \gamma^2}}\right)$$

3. The value of h_t^i will then be sampled from the following Gassian mixture density:

$$g\left(h_t^i|RV_t, BV_t, h_{t-1}^i\right) = p\left(h_t^i|RV_t, h_{t-1}^i, Q_t^i = 0\right)p_0^* + p\left(h_t^i|BV_t, h_{t-1}^i, Q_t^i = 1\right)p_1^*$$

4. The weights during the weight update will then have to be multiplied (in addition to the other terms) with the ratio:

$$\frac{p\left(h_t^i|h_{t-1}^i\right)}{g\left(h_t^i|RV_t, BV_t, h_{t-1}^i\right)}$$

In order to account for the differences between the transition distribution and the proposal distribution.

7.2. SVJD-RV-Z with price and volatility jumps – Sampling JV_t^i and h_t^i

The approximate adaptation will be performed as follows:

1. Calculate provisional probabilities of price jump occurrence $p\left(Q_t^i = 1|Z_t^i\right)$. These can be calculated from the likelihood densities with respect to the Z_t^i, given as:

$$p\left(Z_t^i|Q_t^i = 0\right) \sim N(Z_t; \mu_Z, \sigma_Z)\left(1 - \lambda_t^i\right)$$

$$p\left(Z_t^i|Q_t^i = 1\right) \sim N(Z_t; \mu_Z + \xi_Z, \sigma_Z)\lambda_t^i$$

The provisional jump probability $p\left(Q_t^i = 1 | Z_t^i\right)$ can thus be computed as:

$$p\left(Q_t^i = 1 | Z_t^i\right) = \frac{p\left(Z_t^i | Q_t^i = 1\right)}{p\left(Z_t^i | Q_t^i = 0\right) + p\left(Z_t^i | Q_t^i = 1\right)}$$

The probability of no jump is then equal to $p\left(Q_t^i = 0 | Z_t^i\right) = 1 - p\left(Q_t^i = 1 | Z_t^i\right)$.

We will further denote $p_0^* = p\left(Q_t^i = 0 | Z_t^i\right)$ and $p_1^* = p\left(Q_t^i = 1 | Z_t^i\right)$

2. Calculate provisional probabilities of volatility jump occurrence $p\left(QV_t^i = 1 | RV_t^i, Q_t^i\right)$. For the $p\left(QV_t^i = 1 | RV_t^i, Q_t^i = 0\right)$ case we can compute the probability from the likelihoods:

$$p\left(RV_t^i | QV_t^i = 0, Q_t^i = 0\right) = N\left(\log(RV_t); \alpha + \beta h_{t-1}^i; \sqrt{\sigma_{RV}^2 + \gamma^2}\right)\left(1 - \lambda_{V,t}^i\right)$$

$$p\left(RV_t^i | QV_t^i = 1, Q_t^i = 0\right) = N\left(\log(RV_t); \alpha + \beta h_{t-1}^i + \mu_{JV}; \sqrt{\sigma_{RV}^2 + \gamma^2 + \sigma_{JV}^2}\right)\lambda_{V,t}^i$$

Probability of a volatility jump is then given by:

$$p\left(QV_t^i = 1 | RV_t^i, Q_t^i = 0\right) = \frac{p\left(RV_t^i | QV_t^i = 1, Q_t^i = 0\right)}{p\left(RV_t^i | QV_t^i = 1, Q_t^i = 0\right) + p\left(RV_t^i | QV_t^i = 1, Q_t^i = 0\right)}$$

We will further denote $p_{0,V,1}^* = p\left(QV_t^i = 1 | RV_t^i, Q_t^i = 0\right)$ and $p_{0,V,0}^* = 1 - p_{V,1}^*$

For the $p\left(QV_t^i = 1 | RV_t^i, Q_t^i = 1\right)$ case we will utilize the approximation that $\log[RV(t) - J^2(t)Q(t)] \cong \log[BV(t)]$ and can compute the probability from the likelihoods:

$$p\left(BV_t^i | QV_t^i = 0, Q_t^i = 1\right) = N\left(\log(BV_t); \alpha + \beta h_{t-1}^i; \sqrt{\sigma_{RV}^2 + \gamma^2}\right)\left(1 - \lambda_{V,t}^i\right)$$

$$p\left(BV_t^i | QV_t^i = 1, Q_t^i = 1\right) = N\left(\log(BV_t); \alpha + \beta h_{t-1}^i + \mu_{JV}; \sqrt{\sigma_{RV}^2 + \gamma^2 + \sigma_{JV}^2}\right)\lambda_{V,t}^i$$

Probability of a volatility jump is then given by:

$$p\left(QV_t^i = 1 | RV_t^i, Q_t^i = 1\right) = \frac{p\left(BV_t^i | QV_t^i = 1, Q_t^i = 1\right)}{p\left(BV_t^i | QV_t^i = 1, Q_t^i = 1\right) + p\left(BV_t^i | QV_t^i = 1, Q_t^i = 1\right)}$$

We will further denote $p_{1,V,1}^* = p\left(QV_t^i = 1 | BV_t^i, Q_t^i = 0\right)$ and $p_{1,V,0}^* = 1 - p_{1,V,1}^*$.

3. In order to sample the volatility jump sizes JV_t^i conditional on $QV_t^i = 1, Q_t^i = 0$, we need to use the following relationship:

$$p\left(JV_t^i | RV_t, h_{t-1}^i, QV_t^i = 1, Q_t^i = 0\right) \\ \propto N\left(JV_t^i; \mu_{JV}, \sigma_{JV}\right) N\left(h_t^i; \alpha + \beta h_{t-1}^i + JV_t^i, \gamma\right) N\left(\log(BV_t); h_t^i, \sigma_{RV}\right)$$

As all of the densities are Gaussian, they can be combined to get:

$$p\left(JV_t^i|RV_t,h_{t-1}^i,QV_t^i=1,Q_t^i=0\right)$$
$$=N\left(JV_t^i;\frac{(\log(RV_t)-\alpha-\beta h_{t-1}^i)\sigma_{JV}^2+\mu_{JV}(\sigma_{RV}^2+\gamma^2)}{\sigma_{RV}^2+\gamma^2+\sigma_{JV}^2},\frac{\sigma_{JV}\sqrt{\sigma_{RV}^2+\gamma^2}}{\sqrt{\sigma_{RV}^2+\gamma^2+\sigma_{JV}^2}}\right)$$

Similarly, we can get the density for the $QV_t^i=1,Q_t^i=1$ case as:

$$p\left(JV_t^i|BV_t,h_{t-1}^i,QV_t^i=1,Q_t^i=1\right)$$
$$=N\left(JV_t^i;\frac{(\log(BV_t)-\alpha-\beta h_{t-1}^i)\sigma_{JV}^2+\mu_{JV}(\sigma_{RV}^2+\gamma^2)}{\sigma_{RV}^2+\gamma^2+\sigma_{JV}^2},\frac{\sigma_{JV}\sqrt{\sigma_{RV}^2+\gamma^2}}{\sqrt{\sigma_{RV}^2+\gamma^2+\sigma_{JV}^2}}\right)$$

4. Volatility jump sizes JV_t^i will then be sampled from the following Gaussian mixture:

$$g\left(JV_t^i|RV_t,BV_t,h_{t-1}^i\right)=$$
$$\left(p_0^*p_{0,V,0}^*+p_1^*p_{1,V,0}^*\right)N\left(JV_t^i;\mu_{JV},\sigma_{JV}\right)+$$
$$p_0^*p_{0,V,1}^*p\left(JV_t^i|RV_t,h_{t-1}^i,QV_t^i=1,Q_t^i=0\right)+$$
$$p_1^*p_{1,V,1}^*p\left(JV_t^i|BV_t,h_{t-1}^i,QV_t^i=1,Q_t^i=1\right)$$

5. Before we proceed to sample h_t^i, it is necessary to re-calculate the provisional volatility jump occurrence probabilities $p\left(QV_t^i=1|RV_t^i,JV_t^i,Q_t^i\right)$, so that they correspond to the sampled values of JV_t^i.

For the $p\left(QV_t^i=1|RV_t^i,JV_t^i,Q_t^i=0\right)$ case we can compute the probability from the likelihoods:

$$p\left(RV_t^i|JV_t^i,QV_t^i=0,Q_t^i=0\right)=N\left(\log(RV_t);\alpha+\beta h_{t-1}^i;\sqrt{\sigma_{RV}^2+\gamma^2}\right)\left(1-\lambda_{V,t}^i\right)$$

$$p\left(RV_t^i|JV_t^i,QV_t^i=1,Q_t^i=0\right)=N\left(\log(RV_t);\alpha+\beta h_{t-1}^i+JV_t^i;\sqrt{\sigma_{RV}^2+\gamma^2}\right)\lambda_{V,t}^i$$

Probability of a volatility jump is then given by:

$$p\left(QV_t^i=1|RV_t^i,JV_t^i,Q_t^i=0\right)$$
$$=\frac{p\left(RV_t^i|JV_t^i,QV_t^i=1,Q_t^i=0\right)}{p\left(RV_t^i|JV_t^i,QV_t^i=1,Q_t^i=0\right)+p\left(RV_t^i|JV_t^iQV_t^i=1,Q_t^i=0\right)}$$

We will further denote $p_{0,V,1}^{**}=p\left(QV_t^i=1|RV_t^i,JV_t^i,Q_t^i=0\right)$ and $p_{0,V,0}^{**}=1-p_{V,1}^{**}$

Analogously for $p\left(QV_t^i=1|RV_t^i,JV_t^i,Q_t^i=1\right)$, we use the likelihoods:

$$p\left(BV_t^i|JV_t^i,QV_t^i=0,Q_t^i=1\right)=N\left(\log(BV_t);\alpha+\beta h_{t-1}^i;\sqrt{\sigma_{RV}^2+\gamma^2}\right)\left(1-\lambda_{V,t}^i\right)$$

$$p\left(BV_t^i|JV_t^i, QV_t^i = 1, Q_t^i = 1\right) = N\left(\log(BV_t); \alpha + \beta h_{t-1}^i + JV_t^i; \sqrt{\sigma_{RV}^2 + \gamma^2}\right)\lambda_{V,t}^i$$

Probability of a volatility jump is then given by:

$$p\left(QV_t^i = 1|JV_t^i, RV_t^i, Q_t^i = 0\right)$$
$$= \frac{p\left(BV_t^i|JV_t^i, QV_t^i = 1, Q_t^i = 1\right)}{p\left(BV_t^i|JV_t^i, QV_t^i = 1, Q_t^i = 1\right) + p\left(BV_t^i|JV_t^i, QV_t^i = 1, Q_t^i = 1\right)}$$

We will further denote $p_{1,V,1}^{**} = p\left(QV_t^i = 1|JV_t^i, BV_t^i, Q_t^i = 0\right)$ and $p_{1,V,0}^{**} = 1 - p_{1,V,1}^{**}$.

6. In order to sample the diffusive part of the stochastic variances $h_{Diff,t}^i = h_t^i - JV_t^i QV_t^i$ conditional on RV_t and JV_t^i, we need to derive the conditional densities for all possible combinations of the values of QV_t^i and Q_t^i. These can be derived from the relationships:

$$p\left(h_t^i|RV_t, h_{t-1}^i, JV_t^i, QV_t^i, Q_t^i = 0\right)$$
$$\propto N\left(h_t^i; \alpha + \beta h_{t-1}^i + JV_t^i QV_t^i, \gamma\right) N\left(\log(RV_t); h_t^i, \sigma_{RV}\right)$$

And

$$p\left(h_t^i|RV_t, h_{t-1}^i, JV_t^i, QV_t^i, Q_t^i = 1\right)$$
$$\propto N\left(h_t^i; \alpha + \beta h_{t-1}^i + JV_t^i QV_t^i, \gamma\right) N\left(\log(BV_t); h_t^i, \sigma_{RV}\right)$$

The resulting densities are:

$$p\left(h_{Diff,t}^i|RV_t, h_{t-1}^i, JV_t^i, Q_t^i = 0, QV_t^i = 0\right) = N(m_{0,V,0}^*, s^*)$$
$$p\left(h_{Diff,t}^i|RV_t, h_{t-1}^i, JV_t^i, Q_t^i = 0, QV_t^i = 1\right) = N(m_{0,V,1}^*, s^*)$$
$$p\left(h_{Diff,t}^i|RV_t, h_{t-1}^i, JV_t^i, Q_t^i = 1, QV_t^i = 0\right) = N(m_{1,V,0}^*, s^*)$$
$$p\left(h_{Diff,t}^i|RV_t, h_{t-1}^i, JV_t^i, Q_t^i = 1, QV_t^i = 1\right) = N(m_{1,V,1}^*, s^*)$$

Where

$$m_{0,V,0}^* = \frac{\log(RV_t)\gamma^2 + (\alpha + \beta h_{t-1}^i)\sigma_{RV}^2}{\sigma_{RV}^2 + \gamma^2}$$

$$m_{0,V,1}^* = \frac{\left(\log(RV_t) - JV_t^i\right)\gamma^2 + (\alpha + \beta h_{t-1}^i)\sigma_{RV}^2}{\sigma_{RV}^2 + \gamma^2}$$

$$m_{1,V,0}^* = \frac{\log(BV_t)\gamma^2 + (\alpha + \beta h_{t-1}^i)\sigma_{RV}^2}{\sigma_{RV}^2 + \gamma^2}$$

$$m_{1,V,1}^* = \frac{\left(\log(BV_t) - JV_t^i\right)\gamma^2 + (\alpha + \beta h_{t-1}^i)\sigma_{RV}^2}{\sigma_{RV}^2 + \gamma^2}$$

And

$$s^* = \frac{\sigma_{RV}\gamma}{\sqrt{\sigma_{RV}^2 + \gamma^2}}$$

7. The value of $h_{Diff,t}^{i}$ will then be sampled from the following Gaussian mixture density:

$$g\left(h_{Diff,t}^{i}|RV_t, h_{t-1}^{i}, JV_t^{i}\right) =$$
$$p_0^* p_{0,V,0}^{**} N\left(m_{0,V,0}^*, s^*\right) + p_0^* p_{0,V,1}^{**} N\left(m_{0,V,1}^*, s^*\right) +$$
$$p_1^* p_{1,V,0}^{**} N\left(m_{1,V,0}^*, s^*\right) + p_1^* p_{1,V,1}^{**} N\left(m_{1,V,1}^*, s^*\right)$$

8. In order to account for the utilized proposal densities for the sampling of JV_t^i and h_t^i, the weight update ratio will have to be multiplied with the following ratio:

$$\frac{p\left(h_{Diff,t}^{i}|h_{t-1}^{i}\right)N\left(JV_t^i;\mu_{JV},\sigma_{JV}\right)}{g\left(h_{Diff,t}^{i}|RV_t,h_{t-1}^{i},JV_t^i\right)g\left(JV_t^i|RV_t,BV_t,h_{t-1}^{i}\right)}$$

7.3. 2-Component SVJD-RV-Z with price and volatility jumps – Sampling of JV_t^i, $h_{ST,t}^i$ and $h_{LT,t}$

The approximate adaptation in the case of a two-component SVJD model with price and volatility jumps will be performed as follows:

1. Calculate provisional probabilities of price jump occurrence $p\left(Q_t^i = 1|Z_t^i\right)$. These can be calculated from the likelihood densities with respect to the Z_t^i, given as:

$$p\left(Z_t^i|Q_t^i = 0\right) \sim N(Z_t;\mu_Z,\sigma_Z)\left(1-\lambda_t^i\right)$$
$$p\left(Z_t^i|Q_t^i = 1\right) \sim N(Z_t;\mu_Z+\xi_Z,\sigma_Z)\lambda_t^i$$

The provisional jump probability $p\left(Q_t^i = 1|Z_t^i\right)$ can thus be computed as:

$$p\left(Q_t^i = 1|Z_t^i\right) = \frac{p\left(Z_t^i|Q_t^i = 1\right)}{p\left(Z_t^i|Q_t^i = 0\right) + p\left(Z_t^i|Q_t^i = 1\right)}$$

The probability of no jump is then equal to $p\left(Q_t^i = 0|Z_t^i\right) = 1 - p\left(Q_t^i = 1|Z_t^i\right)$.

We will further denote $p_0^* = p\left(Q_t^i = 0|Z_t^i\right)$ and $p_1^* = p\left(Q_t^i = 1|Z_t^i\right)$

2. Calculate provisional probabilities of volatility jump occurrence $p\left(QV_t^i = 1|RV_t^i, Q_t^i\right)$. For the $p\left(QV_t^i = 1|RV_t^i, Q_t^i = 0\right)$ case we can compute the probability from the likelihoods:

$$p\left(RV_t^i|QV_t^i = 0, Q_t^i = 0\right) =$$
$$N\left(\log(RV_t);\phi_0+\phi_1 h_{LT,t-1}^{i}+\beta h_{ST,t-1}^{i};\sqrt{\sigma_{RV}^2+\phi_2^2+\gamma^2}\right)\left(1-\lambda_{V,t}^{i}\right)$$

$$p\left(RV_t^i|QV_t^i = 1, Q_t^i = 0\right) =$$
$$N\left(\log(RV_t);\phi_0+\phi_1 h_{LT,t-1}^{i}+\beta h_{ST,t-1}^{i}+\mu_{JV};\sqrt{\sigma_{RV}^2+\phi_2^2+\gamma^2+\sigma_{JV}^2}\right)\lambda_{V,t}^{i}$$

Probability of a volatility jump is then given by:

$$p\left(QV_t^i = 1|RV_t^i, Q_t^i = 0\right) = \frac{p\left(RV_t^i|QV_t^i = 1, Q_t^i = 0\right)}{p\left(RV_t^i|QV_t^i = 1, Q_t^i = 0\right) + p\left(RV_t^i|QV_t^i = 1, Q_t^i = 0\right)}$$

We will further denote $p_{0,V,1}^* = p\left(QV_t^i = 1|RV_t^i, Q_t^i = 0\right)$ and $p_{0,V,0}^* = 1 - p_{V,1}^*$

For the $p\left(QV_t^i = 1|RV_t^i, Q_t^i = 1\right)$ case we will utilize approximation that $\log[RV(t) - J^2(t)Q(t)] \cong \log[BV(t)]$ and can compute the probability from the likelihoods:

$$p\left(BV_t^i|QV_t^i = 0, Q_t^i = 1\right) =$$

$$N\left(\log(BV_t); \phi_0 + \phi_1 h_{LT,t-1}^i + \beta h_{ST,t-1}^i; \sqrt{\sigma_{RV}^2 + \phi_2^2 + \gamma^2}\right)\left(1 - \lambda_{V,t}^i\right)$$

$$p\left(BV_t^i|QV_t^i = 1, Q_t^i = 1\right) =$$

$$N\left(\log(BV_t); \phi_0 + \phi_1 h_{LT,t-1}^i + \beta h_{ST,t-1}^i + \mu_{JV}; \sqrt{\sigma_{RV}^2 + \phi_2^2 + \gamma^2 + \sigma_{JV}^2}\right)\lambda_{V,t}^i$$

Probability of a volatility jump is then given by:

$$p\left(QV_t^i = 1|RV_t^i, Q_t^i = 1\right) = \frac{p\left(BV_t^i|QV_t^i = 1, Q_t^i = 1\right)}{p\left(BV_t^i|QV_t^i = 1, Q_t^i = 1\right) + p\left(BV_t^i|QV_t^i = 1, Q_t^i = 1\right)}$$

We will further denote $p_{1,V,1}^* = p\left(QV_t^i = 1|BV_t^i, Q_t^i = 0\right)$ and $p_{1,V,0}^* = 1 - p_{1,V,1}^*$.

3. In order to sample the volatility jump sizes JV_t^i conditional on $QV_t^i = 1, Q_t^i = 0$, we need to use the following relationship:

$$\begin{aligned}&p\left(JV_t^i|RV_t, h_{LT,t-1}^i, h_{ST,t-1}^i, QV_t^i = 1, Q_t^i = 0\right)\\ &\qquad \propto N\left(JV_t^i; \mu_{JV}, \sigma_{JV}\right)N\left(h_t^i; h_{LT,t}^i + \beta h_{ST,t-1}^i + JV_t^i, \gamma\right)N\left(h_{LT,t}^i; \phi_0\right.\\ &\qquad \left.+ \phi_1 h_{LT,t-1}^i, \phi_2\right)N\left(\log(BV_t); h_t^i, \sigma_{RV}\right)\end{aligned}$$

As all of the densities are Gaussian, they can be combined to get:

$$p\left(JV_t^i|RV_t, h_{LT,t-1}^i, h_{ST,t-1}^i, QV_t^i = 1, Q_t^i = 0\right)$$

$$= N\left(JV_t^i; \frac{(\log(RV_t) - \phi_0 - \phi_1 h_{LT,t-1}^i - \beta h_{t-1}^i)\sigma_{JV}^2 + \mu_{JV}(\sigma_{RV}^2 + \phi_2^2 + \gamma^2)}{\sigma_{RV}^2 + \phi_2^2 + \gamma^2 + \sigma_{JV}^2}, \frac{\sigma_{JV}\sqrt{\sigma_{RV}^2 + \phi_2^2 + \gamma^2}}{\sqrt{\sigma_{RV}^2 + \phi_2^2 + \gamma^2 + \sigma_{JV}^2}}\right)$$

Similarly, we can get the density for the $QV_t^i = 1, Q_t^i = 1$ case as:

$$p\left(JV_t^i|BV_t, h_{LT,t-1}^i, h_{ST,t-1}^i, QV_t^i = 1, Q_t^i = 1\right)$$

$$= N\left(JV_t^i; \frac{(\log(BV_t) - \phi_0 - \phi_1 h_{LT,t-1}^i - \beta h_{t-1}^i)\sigma_{JV}^2 + \mu_{JV}(\sigma_{RV}^2 + \phi_2^2 + \gamma^2)}{\sigma_{RV}^2 + \phi_2^2 + \gamma^2 + \sigma_{JV}^2}, \frac{\sigma_{JV}\sqrt{\sigma_{RV}^2 + \phi_2^2 + \gamma^2}}{\sqrt{\sigma_{RV}^2 + \phi_2^2 + \gamma^2 + \sigma_{JV}^2}}\right)$$

4. Volatility jump sizes JV_t^i can then be sampled from the following Gaussian mixture:

$$g(JV_t^i|RV_t, BV_t, h_{LT,t-1}^i, h_{ST,t-1}^i) =$$
$$(p_0^* p_{0,V,0}^* + p_1^* p_{1,V,0}^*) N(JV_t^i; \mu_{JV}, \sigma_{JV}) +$$
$$p_0^* p_{0,V,1}^* p(JV_t^i|RV_t, h_{LT,t-1}^i, h_{ST,t-1}^i, QV_t^i = 1, Q_t^i = 0) +$$
$$p_1^* p_{1,V,1}^* p(JV_t^i|BV_t, h_{LT,t-1}^i, h_{ST,t-1}^i, QV_t^i = 1, Q_t^i = 1)$$

5. Before we proceed to sample $h_{LT,t}^i$, it is necessary to re-calculate the provisional volatility jump occurrence probabilities $p(QV_t^i = 1|RV_t^i, JV_t^i, Q_t^i)$, so that they correspond to the sampled values of JV_t^i.

For the $p(QV_t^i = 1|RV_t^i, JV_t^i, Q_t^i = 0)$ case we can compute the probability from the likelihoods:

$$p(RV_t^i|JV_t^i, QV_t^i = 0, Q_t^i = 0) = N\left(\log(RV_t); \phi_0 + \phi_1 h_{LT,t-1}^i + \beta h_{ST,t-1}^i; \sqrt{\sigma_{RV}^2 + \phi_2^2 + \gamma^2}\right)(1 - \lambda_{V,t}^i)$$

$$p(RV_t^i|JV_t^i, QV_t^i = 1, Q_t^i = 0) = N\left(\log(RV_t); \phi_0 + \phi_1 h_{LT,t-1}^i + \beta h_{ST,t-1}^i + JV_t^i; \sqrt{\sigma_{RV}^2 + \phi_2^2 + \gamma^2}\right)\lambda_{V,t}^i$$

Probability of a volatility jump is then given by:

$$p(QV_t^i = 1|RV_t^i, JV_t^i, Q_t^i = 0) = \frac{p(RV_t^i|JV_t^i, QV_t^i = 1, Q_t^i = 0)}{p(RV_t^i|JV_t^i, QV_t^i = 1, Q_t^i = 0) + p(RV_t^i|JV_t^i QV_t^i = 1, Q_t^i = 0)}$$

We will further denote $p_{0,V,1}^{**} = p(QV_t^i = 1|RV_t^i, JV_t^i, Q_t^i = 0)$ and $p_{0,V,0}^{**} = 1 - p_{V,1}^{**}$

Analogously for $p(QV_t^i = 1|RV_t^i, JV_t^i, Q_t^i = 1)$, we use the likelihoods:

$$p(BV_t^i|JV_t^i, QV_t^i = 0, Q_t^i = 1) = N\left(\log(BV_t); \phi_0 + \phi_1 h_{LT,t-1}^i + \beta h_{ST,t-1}^i; \sqrt{\sigma_{RV}^2 + \phi_2^2 + \gamma^2}\right)(1 - \lambda_{V,t}^i)$$

$$p\left(BV_t^i|JV_t^i, QV_t^i=1, Q_t^i=1\right) = N\left(\log(BV_t); \phi_0+\phi_1 h_{LT,t-1}^i+\beta h_{ST,t-1}^i + JV_t^i; \sqrt{\sigma_{RV}^2+\phi_2^2+\gamma^2}\right)\lambda_{V,t}^i$$

Probability of a volatility jump is then given by:

$$p\left(QV_t^i=1|JV_t^i, RV_t^i, Q_t^i=0\right) = \frac{p\left(BV_t^i|JV_t^i, QV_t^i=1, Q_t^i=1\right)}{p\left(BV_t^i|JV_t^i, QV_t^i=1, Q_t^i=1\right)+p\left(BV_t^i|JV_t^i, QV_t^i=1, Q_t^i=1\right)}$$

We will further denote $p_{1,V,1}^{**}=p\left(QV_t^i=1|JV_t^i, BV_t^i, Q_t^i=0\right)$ and $p_{1,V,0}^{**}=1-p_{1,V,1}^{**}$.

6. In order to sample the long-term log-variance component $h_{LT,t}^i$, conditional on RV_t and JV_t^i, we need to derive the conditional densities for all possible combinations of the values of QV_t^i and Q_t^i. These can be derived from the relationships:

$$p\left(h_{LT,t}^i|RV_t, h_{LT,t-1}^i, h_{ST,t-1}^i, JV_t^i, QV_t^i, Q_t^i=0\right) \propto N\left(h_t^i; h_{LT,t}^i+\beta h_{ST,t-1}^i+JV_t^i QV_t^i, \gamma\right) N\left(h_{LT,t}^i; \phi_0 + \phi_1 h_{LT,t-1}^i, \phi_2\right) N\left(\log(RV_t); h_t^i, \sigma_{RV}\right)$$

And

$$p\left(h_{LT,t}^i|RV_t, h_{LT,t-1}^i, h_{ST,t-1}^i, JV_t^i, QV_t^i, Q_t^i=0\right) \propto N\left(h_t^i; h_{LT,t}^i+\beta h_{ST,t-1}^i+JV_t^i QV_t^i, \gamma\right) N\left(h_{LT,t}^i; \phi_0 + \phi_1 h_{LT,t-1}^i, \phi_2\right) N\left(\log(BV_t); h_t^i, \sigma_{RV}\right)$$

The resulting densities are:

$$p\left(h_{LT,t}^i|RV_t, h_{LT,t-1}^i, h_{ST,t-1}^i, JV_t^i, Q_t^i=0, QV_t^i=0\right)=N(m_{0,V,0}^*, s^*)$$
$$p\left(h_{LT,t}^i|RV_t, h_{LT,t-1}^i, h_{ST,t-1}^i, JV_t^i, Q_t^i=0, QV_t^i=1\right)=N(m_{0,V,1}^*, s^*)$$
$$p\left(h_{LT,t}^i|RV_t, h_{LT,t-1}^i, h_{ST,t-1}^i, JV_t^i, Q_t^i=1, QV_t^i=0\right)=N(m_{1,V,0}^*, s^*)$$
$$p\left(h_{LT,t}^i|RV_t, h_{LT,t-1}^i, h_{ST,t-1}^i, JV_t^i, Q_t^i=1, QV_t^i=1\right)=N(m_{1,V,1}^*, s^*)$$

Where

$$m_{0,V,0}^* = \frac{\left(\log(RV_t)-\beta h_{ST,t-1}^i\right)\phi_2^2+(\phi_0+\phi_1 h_{LT,t-1}^i)(\sigma_{RV}^2+\gamma^2)}{\sigma_{RV}^2+\gamma^2+\phi_2^2}$$

$$m_{0,V,1}^* = \frac{\left(\log(RV_t)-\beta h_{ST,t-1}^i-JV_t^i\right)\phi_2^2+(\phi_0+\phi_1 h_{LT,t-1}^i)(\sigma_{RV}^2+\gamma^2)}{\sigma_{RV}^2+\gamma^2+\phi_2^2}$$

$$m_{1,V,0}^* = \frac{\left(\log(BV_t)-\beta h_{ST,t-1}^i\right)\phi_2^2+(\phi_0+\phi_1 h_{LT,t-1}^i)(\sigma_{RV}^2+\gamma^2)}{\sigma_{RV}^2+\gamma^2+\phi_2^2}$$

$$m^*_{1,V,1} = \frac{\left(\log(BV_t) - \beta h^i_{ST,t-1} - JV^i_t\right)\phi_2^2 + (\phi_0 + \phi_1 h^i_{LT,t-1})(\sigma^2_{RV} + \gamma^2)}{\sigma^2_{RV} + \gamma^2 + \phi_2^2}$$

And

$$s^* = \frac{\phi_2\sqrt{\sigma^2_{RV} + \gamma^2}}{\sqrt{\sigma^2_{RV} + \gamma^2 + \phi_2^2}}$$

7. The value of $h^i_{LT,t}$ can then be sampled from the following Gaussian mixture density:

$$g\left(h^i_{LT,t} | RV_t, h^i_{LT,t-1}, h^i_{ST,t-1}, JV^i_t\right) =$$
$$p^*_0 p^{**}_{0,V,0} N\left(m^*_{0,V,0}, s^*\right) + p^*_0 p^{**}_{0,V,1} N\left(m^*_{0,V,1}, s^*\right) +$$
$$p^*_1 p^{**}_{1,V,0} N\left(m^*_{1,V,0}, s^*\right) + p^*_1 p^{**}_{1,V,1} N\left(m^*_{1,V,1}, s^*\right)$$

8. Before we proceed to sample $h^i_{ST,t}$, it is necessary to re-calculate the provisional volatility jump occurrence probabilities $p\left(QV^i_t = 1 | RV_t, h^i_{LT,t}, JV^i_t, Q^i_t\right)$, so that they correspond to the sampled values of JV^i_t and $h^i_{LT,t}$.

For the $p\left(QV^i_t = 1 | RV_t, h^i_{LT,t}, JV^i_t, Q^i_t = 0\right)$ case we can compute the probability from the likelihoods:

$$p\left(RV_t | h^i_{LT,t}, JV^i_t, QV^i_t = 0, Q^i_t = 0\right)$$
$$= N\left(\log(RV_t); h^i_{LT,t} + \beta h^i_{t-1}; \sqrt{\sigma^2_{RV} + \gamma^2}\right)\left(1 - \lambda^i_{V,t}\right)$$
$$p\left(RV_t | h^i_{LT,t}, JV^i_t, QV^i_t = 1, Q^i_t = 0\right)$$
$$= N\left(\log(RV_t); h^i_{LT,t} + \beta h^i_{t-1} + JV^i_t; \sqrt{\sigma^2_{RV} + \gamma^2}\right)\lambda^i_{V,t}$$

Probability of a volatility jump is then given by:

$$p\left(QV^i_t = 1 | RV_t, h^i_{LT,t}, JV^i_t, Q^i_t = 0\right)$$
$$= \frac{p\left(RV_t | h^i_{LT,t}, JV^i_t, QV^i_t = 1, Q^i_t = 0\right)}{p\left(RV_t | h^i_{LT,t}, JV^i_t, QV^i_t = 1, Q^i_t = 0\right) + p\left(RV_t | h^i_{LT,t}, JV^i_t QV^i_t = 1, Q^i_t = 0\right)}$$

We will further denote $p^{***}_{0,V,1} = p\left(QV^i_t = 1 | RV_t, h^i_{LT,t}, JV^i_t, Q^i_t = 0\right)$ and $p^{***}_{0,V,0} = 1 - p^{***}_{V,1}$

Analogously for $p\left(QV^i_t = 1 | BV_t, h^i_{LT,t}, JV^i_t, Q^i_t = 1\right)$, we use the likelihoods:

$$p\left(BV_t | h^i_{LT,t}, JV^i_t, QV^i_t = 0, Q^i_t = 1\right)$$
$$= N\left(\log(BV_t); h^i_{LT,t} + \beta h^i_{t-1}; \sqrt{\sigma^2_{RV} + \gamma^2}\right)\left(1 - \lambda^i_{V,t}\right)$$
$$p\left(BV_t | h^i_{LT,t}, JV^i_t, QV^i_t = 1, Q^i_t = 1\right)$$
$$= N\left(\log(BV_t); h^i_{LT,t} + \beta h^i_{t-1} + JV^i_t; \sqrt{\sigma^2_{RV} + \gamma^2}\right)\lambda^i_{V,t}$$

Probability of a volatility jump is then given by:

$$p\left(QV_t^i = 1|BV_t, h_{LT,t}^i, JV_t^i, Q_t^i = 0\right)$$
$$= \frac{p\left(BV_t|h_{LT,t}^i, JV_t^i, QV_t^i = 1, Q_t^i = 1\right)}{p\left(BV_t|h_{LT,t}^i, JV_t^i, QV_t^i = 1, Q_t^i = 1\right) + p\left(BV_t|h_{LT,t}^i, JV_t^i, QV_t^i = 1, Q_t^i = 1\right)}$$

We will further denote $p_{1,V,1}^{***} = p\left(QV_t^i = 1|BV_t, h_{LT,t}^i, JV_t^i, Q_t^i = 0\right)$ and $p_{1,V,0}^{***} = 1 - p_{1,V,1}^{***}$.

9. In order to sample the diffusive part of the short-term stochastic variances $h_{ST,Diff,t}^i = h_{ST,t}^i - JV_t^i QV_t^i$ conditional on RV_t, JV_t^i and $h_{LT,t}^i$, we need to derive the conditional densities for all possible combinations of the values of QV_t^i and Q_t^i. These can be derived from the relationships:

$$p\left(h_t^i|RV_t, h_{ST,t-1}^i, h_{LT,t}^i, JV_t^i, QV_t^i, Q_t^i = 0\right)$$
$$\propto N\left(h_t^i; h_{LT,t}^i + \beta h_{ST,t-1}^i + JV_t^i QV_t^i, \gamma\right) N\left(\log(RV_t); h_t^i, \sigma_{RV}\right)$$

And

$$p\left(h_t^i|RV_t, h_{ST,t-1}^i, h_{LT,t}^i, JV_t^i, QV_t^i, Q_t^i = 1\right)$$
$$\propto N\left(h_t^i; h_{LT,t}^i + \beta h_{ST,t-1}^i + JV_t^i QV_t^i, \gamma\right) N\left(\log(BV_t); h_t^i, \sigma_{RV}\right)$$

The resulting densities are:

$$p\left(h_{ST,Diff,t}^i|RV_t, h_{ST,t-1}^i, h_{LT,t}^i, JV_t^i, Q_t^i = 0, QV_t^i = 0\right) = N(m_{0,V,0}^{**}, s^{**})$$
$$p\left(h_{ST,Diff,t}^i|RV_t, h_{ST,t-1}^i, h_{LT,t}^i, JV_t^i, Q_t^i = 0, QV_t^i = 1\right) = N(m_{0,V,1}^{**}, s^{**})$$
$$p\left(h_{ST,Diff,t}^i|RV_t, h_{ST,t-1}^i, h_{LT,t}^i, JV_t^i, Q_t^i = 1, QV_t^i = 0\right) = N(m_{1,V,0}^{**}, s^{**})$$
$$p\left(h_{ST,Diff,t}^i|RV_t, h_{ST,t-1}^i, h_{LT,t}^i, JV_t^i, Q_t^i = 1, QV_t^i = 1\right) = N(m_{1,V,1}^{**}, s^{**})$$

Where

$$m_{0,V,0}^{**} = \frac{\left(\log(RV_t) - h_{LT,t}^i\right)\gamma^2 + \beta h_{ST,t-1}^i \sigma_{RV}^2}{\sigma_{RV}^2 + \gamma^2}$$

$$m_{0,V,1}^{**} = \frac{\left(\log(RV_t) - h_{LT,t}^i - JV_t^i\right)\gamma^2 + \beta h_{ST,t-1}^i \sigma_{RV}^2}{\sigma_{RV}^2 + \gamma^2}$$

$$m_{1,V,0}^{**} = \frac{\left(\log(BV_t) - h_{LT,t}^i\right)\gamma^2 + \beta h_{ST,t-1}^i \sigma_{RV}^2}{\sigma_{RV}^2 + \gamma^2}$$

$$m_{1,V,1}^{**} = \frac{\left(\log(BV_t) - h_{LT,t}^i - JV_t^i\right)\gamma^2 + \beta h_{ST,t-1}^i \sigma_{RV}^2}{\sigma_{RV}^2 + \gamma^2}$$

And

$$s^{**} = \frac{\sigma_{RV}\gamma}{\sqrt{\sigma_{RV}^2 + \gamma^2}}$$

10. The value of $h_{ST,Diff,t}^i$ will then be sampled from the following Gaussian mixture density:

$$g\left(h^{i}_{ST,Diff,t}|RV_t,h^{i}_{ST,t-1},h^{i}_{LT,t},JV^{i}_{t}\right)=$$
$$p^{*}_{0}p^{***}_{0,V,0}N\left(m^{**}_{0,V,0},s^{**}\right)+p^{*}_{0}p^{***}_{0,V,1}N\left(m^{**}_{0,V,1},s^{**}\right)+$$
$$p^{*}_{1}p^{***}_{1,V,0}N\left(m^{**}_{1,V,0},s^{**}\right)+p^{*}_{1}p^{***}_{1,V,1}N\left(m^{**}_{1,V,1},s^{**}\right)$$

11. In order to account for the utilized proposal densities for the sampling of JV^{i}_{t}, $h^{i}_{LT,t}$ and $h^{i}_{ST,Diff,t}$, the weight update ratio will have to be multiplied with the following ratio:

$$\frac{p\left(h^{i}_{ST,Diff,t}|h^{i}_{ST,t-1}\right)p\left(h^{i}_{LT,t}|h^{i}_{LT,t-1}\right)N\left(JV^{i}_{t};\mu_{JV},\sigma_{JV}\right)}{g\left(h^{i}_{ST,Diff,t}|RV_t,h^{i}_{ST,t-1},h^{i}_{LT,t},JV^{i}_{t}\right)g\left(h^{i}_{LT,t}|RV_t,h^{i}_{LT,t-1},h^{i}_{ST,t-1},JV^{i}_{t}\right)g\left(JV^{i}_{t}|RV_t,BV_t,h^{i}_{LT,t-1},h^{i}_{ST,t-1}\right)}$$

7.4. SVJD-RV-Z with price and volatility jumps – Sampling of J^{i}_{t}

The sampling of J^{i}_{t} should be performed after we have already sampled the $h^{i}_{Diff,t}$ and JV^{i}_{t}. The non-linearity in the relationship between RV_t and J^{i}_{t} unfortunately makes the fully adapted distribution intractable. Nevertheless, we can at least use the available information about r_t and Z_t in order to derive an approximately adapted distribution of J^{i}_{t}. The adaptation proceeds as follows:

1. Let us define $V^{i}_{Diff,t}=\exp(h^{i}_{Diff,t})$ and $V^{i}_{Jump,t}=\exp(h^{i}_{Diff,t}+JV^{i}_{t})$. We can then calculate the provisional probabilities of price and volatility jumps, conditional on r_t and Z_t, by using the likelihoods $l^{*}_{x,y}=p\left(r_t,Z_t|V^{i}_{Diff,t},Q^{i}_{t}=x,QV^{i}_{t}=y\right)$, defined as follows:

$$l^{*}_{0,0}=N\left(r_t;\mu,\sqrt{V^{i}_{Diff,t}}\right)N(Z_t;\mu_Z,\sigma_Z)\left(1-\lambda^{i}_{t}\right)\left(1-\lambda^{i}_{V,t}\right)$$

$$l^{*}_{0,1}=N\left(r_t;\mu,\sqrt{V^{i}_{Jump,t}}\right)N(Z_t;\mu_Z,\sigma_Z)\left(1-\lambda^{i}_{t}\right)\lambda^{i}_{V,t}$$

$$l^{*}_{1,0}=N\left(r_t;\mu+\mu_J,\sqrt{V^{i}_{Diff,t}+\sigma_J^2}\right)N(Z_t;\mu_Z+\xi_Z,\sigma_Z)\lambda^{i}_{t}\left(1-\lambda^{i}_{V,t}\right)$$

$$l^{*}_{1,1}=N\left(r_t;\mu+\mu_J,\sqrt{V^{i}_{Jump,t}+\sigma_J^2}\right)N(Z_t;\mu_Z+\xi_Z,\sigma_Z)\lambda^{i}_{t}\lambda^{i}_{V,t}$$

The provisional probabilities $p^{*}_{x,y}=p(Q^{i}_{t}=x,QV^{i}_{t}=y|r_t,Z_t,h^{i}_{Diff,t},JV^{i}_{t})$ are then given as:

$$p^{*}_{0,0}=l^{*}_{0,0}/(l^{*}_{0,0}+l^{*}_{0,1}+l^{*}_{1,0}+l^{*}_{1,1})$$
$$p^{*}_{0,1}=l^{*}_{0,1}/(l^{*}_{0,0}+l^{*}_{0,1}+l^{*}_{1,0}+l^{*}_{1,1})$$
$$p^{*}_{1,0}=l^{*}_{1,0}/(l^{*}_{0,0}+l^{*}_{0,1}+l^{*}_{1,0}+l^{*}_{1,1})$$
$$p^{*}_{1,1}=l^{*}_{1,1}/(l^{*}_{0,0}+l^{*}_{0,1}+l^{*}_{1,0}+l^{*}_{1,1})$$

2. We can then compute conditional densities for J_t^i for the two cases when $Q_t^i = 1$. These will be:

$$p\left(J_t^i \middle| r_t, Z_t, h_{Diff,t}^i, JV_t^i, Q_t^i = 1, QV_t^i = 0\right)$$

$$= N\left(J_t^i; \frac{(r_t - \mu)\sigma_j^2 + \mu_J V_{Diff,t}^i}{V_{Diff,t}^i + \sigma_j^2}, \frac{\sigma_j \sqrt{V_{Diff,t}^i}}{\sqrt{\sigma_j^2 + V_{Diff,t}^i}}\right)$$

$$p\left(J_t^i \middle| r_t, Z_t, h_{Diff,t}^i, JV_t^i, Q_t^i = 1, QV_t^i = 1\right)$$

$$= N\left(J_t^i; \frac{(r_t - \mu)\sigma_j^2 + \mu_J V_{Jump,t}^i}{V_{Jump,t}^i + \sigma_j^2}, \frac{\sigma_j \sqrt{V_{Jump,t}^i}}{\sqrt{\sigma_j^2 + V_{Jump,t}^i}}\right)$$

3. The values of J_t^i can then be sampled from the following Gaussian mixture proposal density:

$$g\left(J_t^i \middle| r_t, Z_t, h_{Diff,t}^i, JV_t^i\right) = \left(p_{0,0}^* + p_{0,1}^*\right) N\left(J_t^i; \mu_J, \sigma_j\right) +$$

$$p_{1,0}^*\, p\left(J_t^i \middle| r_t, Z_t, h_{Diff,t}^i, JV_t^i, Q_t^i = 1, QV_t^i = 0\right) +$$

$$p_{1,1}^* p\left(J_t^i \middle| r_t, Z_t, h_{Diff,t}^i, JV_t^i, Q_t^i = 1, QV_t^i = 1\right)$$

4. During the weight update step, the weight update ratio will then have to be multiplied with:

$$\frac{N\left(J_t^i; \mu_J, \sigma_j\right)}{g\left(J_t^i \middle| r_t, Z_t, h_{Diff,t}^i, JV_t^i\right)}$$

8. References

1. **AIT-SAHALIA, Y., CACHO-DIAZ, J. and LEAVEN, R., (2013).** "Modeling Financial Contagion Using Mutually Exciting Jump Processes", August 2014, Working Paper, p.1-47.
2. **ANDERSEN, Torben, (1994).** "Stochastic Autoregressive Volatility: A Framework for Volatility Modeling", *Mathematical Finance*, Vol.4, No.2, pp.75-102.
3. **ANDERSEN, T.G., BENZONI, L., LUND, J., (2002).** "An Empirical Investigation of Continuous-Time Equity Return Models", *The Journal of Finance*, June 2002, Vol.57, No.3, pp.1239-1284
4. **ANDERSEN, T.G. and BOLLERSLEV, T., (1998).** "Answering the Skeptics: Yes, Standard Volatility Models Do Provide Accurate Forecasts," *International Economic Review*, November 1998, Vol. 39, No.4, p.885-905.
5. **ANDERSEN, T.G., BOLLERSLEV, T., DIEBOLD, F.X., LABYS, P., (2003).** "Modeling and Forecasting Realized Volatility", *Econometrica*, March 2003, Vol. 71, No. 2, p. 579-625
6. **ANDERSEN, T.G., BOLLERSLEV, T., CHRISTOFFERSEN, P.F. and DIEBOLD, F.X., (2005).** "Volatility Forecasting", *National Bureau of Economic Research*, March 2005, NBER Working Paper No. 11188.
7. **ANDERSEN, T.G., BOLLERSLEV, T. and DIEBOLD, F.X., (2007).** "Roughing it up: Including jump components in the measurement, modeling and forecasting of return volatility", *Review of Economics and Statistics*, November 2007, Vol. 89, No. 4, p.701-720.
8. **ANDERSEN, T.G., DOBREV, D. and SCHAUMBURG, E., (2009).** "Duration-Based Volatility Estimation", *Institute of Economic Research*, Hitotsubashi University, Global COE Hi-Stat Discussion Paper Series, No.034, pp.1-32
9. **ANDERSEN, T.G., DOBREV, D. and SCHAUMBURG, E., (2014).** "A Robust Neighborhood Truncation Approach to Estimation of Integrated Quarticity", *Econometric Theory*, February 2014, Vol. 30, No.1, pp.3-59
10. **ANDRIEU, C., DOUCET, A. and HOLENSTEIN, R., (2010).** "Particle Markov Chain Monte Carlo Methods", *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol.72, No.3, pp.269-342
11. **ANDRIEU, C., DOUCET, A., SINGH, S.S., TADIC, V.B., (2004).** "Particle methods for change detection, systém identification, and control", *Proceedings of the IEEE*, Vol.92, No.3, pp.423-438
12. **BACRY, E., MASTROMATTEO, I., MUZY, J.-F., (2015).** "Hawkes processes in finance", *Market Microstructure and Liquidity*, 2015, Vol. 01, No. 01
13. **BAILIE, R. T., BOLLERSLEV, T., MIKKELSEN, H. O., (1996).** "Fractionally Integrated Generalized Autoregressive Conditional Heteroskedasticity", *Journal of Econometrics*, September 1996, Vol.74, No.1, pp.3-30.
14. **BAKSHI, Gurdip, KAPADIA, Nikunj, (2003).** "Delta-Hedged Gains and the Negative Market Volatility Risk Premium", *Review of Financial Studies*, April 2003, Vol.16, No.2, pp.527-566
15. **BARNDORFF-NIELSEN, O. E., HANSEN, P.R., LUNDE, A., SHEPARD, N., (2008).** "Designing realised kernels to measure the ex-post variation of equity prices in the presence of noise", *Econometrica*, November 2008, Vol.76, p.1481–1536.
16. **BARNDORFF-NIELSEN, O. E., KINNEBROUK, S., SHEPHARD, N., (2010).** "Measuring Downside Risk: Realised Semivariance", *Volatility and Time Series Econometrics: Essays in Honor of Robert F. Engle*, Edited by T. Bollerslev, J. Russel and M. Watson, Oxford University Press, p.117-136.

17. **BARNDORFF-NIELSEN, Ole E. and SHEPHARD Neil, (2002).** "Econometric analysis of realized volatility and its use in estimating stochastic volatility models", *Journal of the Royal Statistical Society: Series B*, Vol.64, No.2, pp. 253-280.

18. **BARNDORFF-NIELSEN, Ole E. and SHEPHARD Neil, (2004).** "Power and Bipower Variation with Stochastic Volatility and Jumps", *Journal of Financial Econometrics*, 2004, Vol.2, No.1, p.1-48

19. **BATES, David S., (1996).** "Jumps and Stochastic Volatility: Exchange Rate Processes Implicit in Deutsche Mark Options", *Review of Financial Studies*, January 1996, Vol.9, No.1, pp.69-107

20. **BEKIERMAN, Jeremias, GRIBISCH, Bastian, (2016).** "A Mixed Frequency Stochastic Volatility Model for Intraday Stock Market Returns", *SSRN Working Paper*, August 2016, pp.1-37

21. **BEKIERMAN, Jeremias, MANNER, Hans, (2018).** "Forecasting realized variance measures using time-varying coefficient models", *International Journal of Forecasting*, Vol.34, No.2, April-June 2018, pp. 276-287

22. **BLACK, Fischer, (1986).** "Noise", *The Journal of Finance*, July 1986, Vol.16, No.3, pp.529-543

23. **BLACK, Fischer, SCHOLES, Myron, (1973).** "The Pricing of Options and Corporate Liabilities", *Journal of Political Economy*, May-June 1973, Vol.81, No.3, pp.637-654

24. **BOLLERSLEV, Tim, (1986).** "Generalized Autoregressive Conditional Heteroskedasticity", *Journal of Econometrics*, April 1986, Vol.31, No.3, pp.307-327

25. **BOLLERSLEV, Tim, (2008).** "Glossary to ARCH (GARCH)", *Duke University*, CREATES Research Paper, p.1-44

26. **BOLLERSLEV, T., PATTON, A. J., QUAEDVLIEG, R., (2016).** "Exploiting the Errors: A Simple Approach for Improved Volatility Forecasting", *Journal of Econometrics*, May 2016, Vol.192, No.1, pp.1-18.

27. **BREIDT, J.F., CRATO, N., de LIMA, P., (1998).** "The Detection and Estimation of Long Memory in Stochastic Volatility", March-April 1998, Vol.83, No.1-2, pp.325-348

28. **BRITTEN-JONES, Mark, NEUBERGER, Anthony, (2000).** "Option Prices, Implied Price Processes, and Stochastic Volatility", *The Journal of Finance*, April 2000, Vol.55, No.2, pp.839-866

29. **CALVET, Laurent and FISHER, Adlai, (2001).** "Forecasting Multifractal Volatility", *Journal of Econometrics*, November 2001, Vol.105, No.1, p.27-58

30. **CARR, Peter, WU, Liuren, (2009).** "Variance Risk Premiums", *The Review of Financial Studies*, March 2009, Vol.22, No.3, pp.1311-1341

31. **CARVALHO, C., JOHANNES, M., LOPES, H., and POLSON, N., (2010).** “Particle Learning and Smoothing“, *Statistical Science*, Vol.25, No.1, pp. 88-106.

32. **CHEN, Ke and POON, Ser-Huang, (2013).** "Variance Swap Premium under Stochastic Volatility and Self-Exciting Jumps", *Manchester Business School, University of Manchester,* January 2013, Working paper, p.1-50

33. **CHERNOV, M., GALLANT, R.A, GHYSELS, E., TAUCHEN, G., (2003).** "Alternative Models for Stock Price Dynamics", *Journal of Econometrics*, September-October 2003, Vol.116, No.1-2, pp.225-257

34. **CHOPIN, N., IACOBUCCI, A., MARIN, J.M., MENGERSEN, K.L., ROBERT, C.P., RYDER, R., SCHAFER, C., (2010).** “On Particle Learning“, Ninth Valencia Meeting discussion paper, Benidorm, Spain

35. **CHRONOPOULOU, Alexandra, SPILIOPOULOS, Konstantinos (2017).** “Sequential Monte Carlo for Fractional Stochastic Volatility Models“, arXiv Working Paper, https://arxiv.org/pdf/1508.02651.pdf

36. **CLARK, Peter K., (1973).** "A Subordinated Stochastic Process Model with Finite Variance for Speculative Prices", *Econometrica*, January 1973, Vol.41, No.1, pp.135-155

37. **CORSI, Fulvio, (2004).** "A Simple Long Memory Model of Realized Volatility", *Institute of Finance, University of Lugano, Switzerland*, August 2004, Working paper, p.1-31

38. **CORSI, F., MITTNIK, S., PIGORSCH C., PIGORSCH, U., (2008).** "The Volatility of Realized Volatility", *Econometric Reviews*, March 2008, Vol.27, No.1-3, pp.1-33

39. **CORSI, Fulvio, RENO, Roberto, (2009).** "HAR Volatility Modelling with Heterogenous Leverage and Jumps", *SSRN Working Paper*, August 2009, p.1-31

40. **CORSI, F., PIRINO, D. and RENO, R., (2010).** “Threshold Bipower Variation and the Impact of Jumps on Volatility Forecasting”, *Journal of Econometrics*, December 2010, Vol.159, pp.276-288

41. **CRAINE, R., LOCHSTOER, L.A. and SYRTVEIT, K., (2000).** “Estimation of a Stochastic-Volatility Jump-Diffusion Model”, *Economic Analysis Review*, 2000, Vol.15, No.1, p.61-87, ISSN 0716-5927

42. **DEL MORAL, P. (2004).** "Feynman-Kac formulae", *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*, Springer, New York

43. **DEO, R., HURVICH, C., LU, Y., (2006).** "Forecasting realized volatility using a long-memory stochastic volatility model: estimation, prediction and seasonal adjustment", *Journal of Econometrics*, Vol.131, No.1, pp.29-58

44. **DANIELSSON, Jon, (1994).** “Stochastic Volatility in Asset Prices Estimation with Simulated Maximum Likelihood“, *Journal of Econometrics*, September-October 1994, Vol.64, No.1-2, pp.375-400

45. **DOBREV, D., and SZERSZEN, P., (2010).** “The information content of high-frequency data for estimating equity return models and forecasting risk“, Working Paper, Federal Reserve Board, Washington D.C.

46. **DOUCET, A., and JOHANSEN, A.M., (2009).** "A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later", *Handbook of nonlinear filtering*, pp.656-704

47. **DOVONON, P., GONCALVES, S., HOUNYO, U., MEDDAHI, N., (2014).** “Bootstrapping High-Frequency Jump Tests”, Discussion paper, Toulouse School of Economics

48. **DROST, Feike C., WERKER Bas J. M., (1996).** “Closing the GARCH Gap: Continuous Time GARCH Modeling“, *Journal of Econometrics*, September 1996, Vol.74, No.1, pp.31-57

49. **DUMITRU, Ana-Maria, URGA, Giovanni, (2010).** "Identifying Jumps in Financial Assets: A Comparison Between Nonparametric Jump Tests", *Journal of Business & Economic Statistics*, February 2012, Vol.30, No.2, pp.242-255

50. **ELMAN, J. L., (1990).** "Finding Structure in Time", *Cognitive Science 2*, Vol.14, No.2, pp.179-211

51. **ENGLE, Robert F., (1982).** "Autoregressive Conditional Heteroscedasticity with Estimates of Variance of United Kingdom Inflation", *Econometrica,* July 1982, Vol.50, No.4, pp.987-1008.

52. **ENGLE, Robert F., (1990).** "Stock Volatility and the Crash of ‘87", *The Review of Financial Studies*, March 1990, Vol.3, No.1, pp.103-106.

53. **ENGLE, R. F., LEE, G. J., (1999).** "A Permanent and Transitory Component Model of Stock Return Volatility", in ed. R.F. Engle and H. White, *Cointegration, Causality, and Forecasting: A Festschrift in Honor of Clive W.J. Granger*, Oxford University Press, 1999, pp. 475-497.

54. **ERAKER, Bjorn, (2004).** “Do Stock Prices and Volatility Jump? Reconciling Evidence from Spot and Option Prices.” *The Journal of Finance*, June 2004, Vol. 59, No. 3, p. 1367-1403.

55. **ERAKER, Bjorn, (2009).** “The Volatility Premium“, University of Wisconsin, Working Paper, October 2009, Available at: https://scholar.google.cz/scholar?oi=bibs&cluster=7638525025451491206&btnI=1&hl=en

56. **ERAKER, B., JOHANNES, M. and POLSON, N. G., (2003).** “The Impact of Jumping equity index Volatility and Returns.” *The Journal of Finance*, June 2003, Vol.58, No.3, pp. 1269-1300.

57. **FEARNHEAD, P., (2002).** “Markov chain Monte Carlo, sufficient statistics, and particle filters“, Journal of Computational and Graphical Statistics, Vol.11, No.4, pp. 848-862.

58. **FIČURA, Milan, (2014)**. "Analysis of factors influencing the size of the volatility risk premium of the EUR/USD exchange rate", *Business & IT*, December 2014, Vol.4, No.2, ISSN 1805-3777

59. **FIČURA, Milan, (2017)**. "Forecasting Stock Market Realized Variance with Echo State Neural Networks", European Financial and Accounting Journal, 2017, Vol.12, No.3, pp.145-156, ISSN 1802-2197, Available at: https://www.vse.cz/efaj/193

60. **FIČURA, Milan and WITZANY, Jiří (2015).** “Using high-frequency power-variation estimators in the Bayesian estimation of Stochastic-Volatility Jump-Diffusion models”, *The 9th International Days of Statistics and Economics*, Prague, September, 2015, pp. 423-434, ISBN 978-80-87990-09-3, Available at: https://msed.vse.cz/msed_2015/article/170-Ficura-Milan-paper.pdf

61. **FIČURA, Milan and WITZANY, Jiří (2016).** “Estimating Stochastic Volatility and Jumps Using High-Frequency Data and Bayesian Methods“, *Czech Journal of Economics and Finance*, 2016, Vol.66, No.4, pp.279-301.

62. **FIČURA, Milan and WITZANY, Jiří (2017).** “Estimation of SVJD Models with Bayesian Methods“, *Advanced methods of computational finance*, Prague: Oeconomica, 2017, pp.161-204, ISBN 978-80-245-2207-4.

63. **FIČURA, Milan and WITZANY, Jiří (2018).** “Sequential Gibbs Particle Filter Algorithm with an Application to Stochastic Volatility and Jumps Estimation“, 2018, Available at SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3194544

64. **FULOP, Andras, LI, Junye, (2013).** “Efficient Learning via Simulation: A Marginalized Resample-Move Approach“, *Journal of Econometrics*, October 2013, Vol.176, No.2, pp.146-161

65. **FULOP, Andras, LI, Junye and YU, Jun, (2015).** “Self-Exciting Jumps, Learning, and Asset Pricing Implications”, *The Review of Financial Studies*, March 2015, Vol.28, No.3, pp.876-912.

66. **FRANSES, Philip Hans and VAN DIJK, Dick, (2000).** *“Non-Linear Time Series Models in Empirical Finance”*, Cambridge University Press, Australia, 2000, p.18, ISBN 0-511-01100-8

67. **GALLANT, R. A., HSIEH, D., TAUCHEN, G., (1997).** *“Estimation of Stochastic Volatility Models with Diagnostics“*, Journal of Econometrics, November 1997, Vol.81, No.1, pp.159-162

68. **GILKS, W. R., and BERZUINI, C., (2001).** "Following a moving target – Monte Carlo inference for dynamic Bayesian models.", *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol.63, No.1, pp.127-146

69. **GOLIGHTLY, Andrew, (2009).** “Bayesian Filtering for Jump-Diffusions with Application to Stochastic Volatility”, *Journal of Computational and Graphical Statistics*, June 2009, Vol. 18, No. 2, p. 384-400

70. **GORDON, N. J., SALMOND, D. J. and SMITH, A. F., (1993).** "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", *IEE Proceedings F (Radar and Signal Processing)*, Vol.140, No.2, pp. 107-113

71. **GLOSTEN, L. R., JAGANNATHAN, R. and RUNKLE, D. E., (1993).** “On the Relation Between the Expected Value and the Volatility of the Nominal Excess Return on Stocks“, *Journal of Finance*, December 1993, Vol.48, No.5, pp.1779-1801

72. **HANSEN, P. R. and LUNDE, A., (2004).** "An Unbiased Measure of Realized Variance", London School of Economic: http://www.lse.ac.uk/fmg/documents/events/seminars/capitalMarket/2004/P_Hansen.pdf

73. **HANSEN, P. R., HUANG, Z., SHEK, H. H., (2012).** "Realized GARCH: A Joint Model for Returns and Realized Measures of Volatility", *Journal of Applied Econometrics*, September 2012, Vol.27, No.6, pp.877-906

74. **HARVEY, Andrew C., (1998).** "Long Memory in Stochastic Volatility", in eds. J. Knight and S. Satchell, *Forecasting Volatility in Financial Markets*, 307-320, Oxford: Butterworth-Heineman.

75. **HESTON, Steven L., (1993).** "A Closed Form Solution for Options with Stochastic Volatility, with Applications to Bond and Currency Options", *Review of Financial Studies*, April 1993, Vol.6, No.2, pp.327-343

76. **HULL, John and WHITE, Alan, (1987).** "The Pricing of Options on Assets with Stochastic Volatilities", *Journal of Finance*, June 1987, Vol.42, No.2, pp.281-300

77. **ISHIDA, Isao, WATANABE, Toshiaki, (2009).** "Modeling and Forecasting the Volatility of the Nikkei 225 Realized Volatility Using the ARFIMA-GARCH Model", *Institute of Economic Research*, Hitotsubashi University, Global COE Hi-Stat Discussion Paper Series, No.032, pp.1-32

78. **J.P. MORGAN, (1997).** *RiskMetrics, Technical Documents*, 4th Edition, New York

79. **JAEGER, H., HAAS, H., (2004).** "Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication", Science 2, Vol.304, No.5667, pp.78-80.

80. **JACKWERTH, Jens C., RUBINSTEIN, Mark, (1996).** "Recovering Probability Distributions from Option Prices", *The Journal of Finance*, December 1996, Vol.51, No.5, pp.1611-1631

81. **JACQUIER, E., JOHANNES, M. and POLSON, N., (2007).** "MCMC Maximum Likelihood for Latent State Models.", *Journal of Econometrics*, 2007, Vol.137, p. 615–640.

82. **JACQUIER, E., POLSON, N. and ROSSI, P., (1994).** "Bayesian Analysis of Stochastic Volatility Models", *Journal of Business & Economic Statistics*, October 1994, Vol.12, No.4, p.69-87.

83. **JIANG, George J., TIAN, Yisong S., (2005).** "The Model-Free Implied Volatility and Its Information Content", *The Review of Financial Studies*, December 2005, Vol.18, No.4, pp.1305-1342.

84. **JOHANNES, M. and POLSON, N., (2009).** "MCMC Methods for Financial Econometrics.", In Ait-Sahalia, Hansen, L. P., eds., *Handbook of Financial Econometrics*, p.1-72.

85. **JOHANNES, M. and POLSON, N., and STROUD, J., (2002).** "Nonlinear Filtering of Stochastic Diferential Equations with Jumps, Working Paper, Graduate School of Business, University of Chicago

86. **JOHANNES, M., WHITELEY, N., DOUCET, A., (2012).** "Exact Approximation of Rao-Blackwellised Particle Filters", *16th IFAC Symposium on System Identification*, The International Federation of Automatic Control, Brussels, Belgium, July 2012

87. **KIM, S., SHEPHARD, N., CHIB, S., (1998).** "Stochastic Volatility: Likelihood Inference and Comparison with ARCH Models", *The Review of Economic Studies*, July 1998, Vol.65, No.3, pp.361-393

88. **KOOPMAN, S. J., JUNGBACKER, B., HOI, E., (2005).** "Forecasting Daily Variability of the S&P 100 Stock Index Using Historical, Realised and Implied Volatility Measurements", June 2005, Vol.12, No.3, p.445-475

89. **KOOPMAN, S.J., and SCHARTH, M., (2013).** "The analysis of stochastic volatility in the presence of daily realized measures", *Journal of Financial Econometrics*, Vol.11, No.1, pp. 76-115.

90. **LANNE, Markku, (2006).** "Forecasting Realized Volatility by Decomposition", *European University Institute*, 2006, Economics Working Papers, ECO2006/20, p.1-26.

91. **LEE, S.S. and MYKLAND, P.A., (2008):** "Jumps in Financial Markets: A New Nonparametric Test and Jump Dynamics", Review of Financial Studies, 2008, vol. 21, no. 6, pp. 2535-2563

92. **LIU, J., and WEST, M., (2001).** "Combined parameters and state estimation in simulation-based filtering", *Sequential Monte Carlo Methods in Practice* (A.Doucet, N. de Freitas and N. Gordon, eds.), Springer, New York, pp. 197-223, ISBN-13: 978-1441928870

93. **LIU, Xiao-Bin and LI, Yong, (2014).** "Bayesian Testing Volatility Persistence in Stochastic Volatility Models with Jumps", *Quantitative Finance*, July 2014, Vol. 14, No. 8, p. 1415-1426

94. **LOPES, H., CARVALHO, C., JOHANNES, M., POLSON, N., (2010).** "Particle learning for sequential Bayesian computation (with discussion and rejoinder) ", *Bayesian Statistics 9*, Oxford University Press

95. **LUKOŠEVIČIUS, M., JAEGER, H., (2009).** "Reservoir Computing Approaches to Recurrent Neural Network Training", *Computer Science Review*, Vol.3, No.3, pp. 127-149.

96. **LUKOŠEVIČIUS, M., (2012).** "A Practical Guide to Applying Echo State Networks", In: Montavon G. Orr G. B., Muller K. R. (eds), *Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science*, Vol.7700, Springer, Berlin, Heidelberg, ISBN: 978-3-642-35288-1

97. **LUX, T., (2008).** "The Markov-Switching Multifractal Model of Asset Returns: GMM estimation and linear forecasting of volatility", *Journal of Business and Economic Statistics*, Vol.26, pp. 194-210.

98. **LYNCH, Scott M., (2007)**, *"Introduction to Applied Bayesian Statistics and Estimation for Social Scientists"*, Springer, August 2007, pp. 359, ISBN-13: 978-0387712642.

99. **MANEESOONTHORN, W., FORBES, C. S., MARTIN, G. M., (2017).** "Inference on Self-Exciting Jumps in Prices and Volatility Using High-Frequency Measures", *Journal of Applied Econometrics*, April/May 2017, Vol.32, No.3, pp.504-532.

100. **McALEER, Michael and MEDEIROS, Marcelo C., (2011).** "Forecasting Realized Volatility with Linear and Nonlinear Univariate Models", *Journal of Economic Surveys*, February 2011, Vol.25, No.1, pp.6-18.

101. **MINCER, Jacob, ZARNOWITZ, Victor, (1969).** "The Evaluation of Economic Forecasts", *Economic Forecasts and Expectations*, J. Mincer (Ed.), National Bureau of Economic Research (NBER), New York, 1969, ISBN:0-870-14202-X, pp.1-47.

102. **MOLNÁR, Peter, (2016).** "High-Low Range in GARCH Models of Stock Return Volatility", *Applied Economics*, April 2016, Vol.48, No.51, pp.4977-4991.

103. **MÜLLER, U. A., DACOROGNA, M. M., DAVÉ, R. D., OLSEN, R. B., PICTET, O. V., WEIZSÄCKER von, J. E., (1997).** "Volatilities of Different Time Resolutions – Analyzing the Dynamics of Market Components", *Journal of Empirical Finance*, June 1997, Vol.4, No.2-3, pp.213-239

104. **MUZZIOLI, Silvia, (2010).** "Option Based Forecasts of Volatility: An Empirical Study in the DAX-Index Options Market", *The European Journal of Finance*, April 2010, Vol.16, No.6, pp.561-586.

105. **NAKAJIMA, Jouchi, (2012).** "Bayesian Analysis of GARCH and Stochastic Volatility: Modeling Leverage, Jumps and Heavy-Tails for Financial Time Series", *Japanese Economic Review*, March 2012, Vol. 63, No. 1, p. 81-103.

106. **NELSON, D. B., (1991).** "Conditional Heteroskedasticity in Asset Returns: A New Approach", *Econometrica*, March 1991, Vol.59, No.2, pp.347-370.

107. **NEMETH, C., FEARNHEAD, P., MIHAYLOVA, L., (2013).** "Sequential Monte Carlo Methods for State and Parameter Estimation in Abruptly Changing Environments", *IEEE Transactions on Signal Processing*, Vol.62, No.5, pp. 1245-1255.

108.**PATTON, A. J., SHEPPARD, K., (2015).** "Good volatility, bad volatility: Signed jumps and the persistence of volatility", *Review of Economics and Statistics*, July 2015, Vol.97, No.3, pp. 683-697

109.**PONG, S., SHACKLETON, M. B., TAYLOR, S. J., XINZHONG, X., (2004).** "Forecasting Currency Volatility: A Comparison of Implied Volatilities and AR(F)IMA Models", *Journal of Banking & Finance*, October 2004, Vol.28, No.10, pp.2541-2563.

110.**PITT, Michael K., SHEPHARD, Neil, (1999).** "Filtering via Simulation: Auxiliary Particle Filters", *Journal of the American Statistical Association*, June 1999, Vol.94, No.446, pp.590-599

111.**RAGGI, D, and BORDIGNON, S., (2008).** "Sequential Monte Carlo methods for stochastic volatility models with jumps.", Preprint. URL: http://homes. stat. unipd. it/raggi

112.**SANDMANN, Gleb, KOOPMAN, Siem Jan, (1998).** "Estimation of Stochastic Volatility Models via Monte Carlo Maximum Likelihood", *Journal of Econometrics*, December 1998, Vol.87, No.2, pp.271-301

113.**SENTANA, Enrique, (1995).** "Quadratic ARCH Models", *Review of Economic Studies*, Vol.62, No.4, pp.639-661

114.**SHEPHARD, Neil**, **(2005).** "*Stochastic Volatility: Selected Readings (Advanced Texts in Econometrics)*", Oxford: Oxford University Press, May 2006, Edited by Neil Shephard

115.**SHI, Peng, (2009).** "Correcting Finite Sample Biases in Conventional Estimates of Power Variation and Jumps", *Duke University, Durham NC, 27708*, April 2009, Final Paper for Econ201FS, p.1-22

116.**SO, M.K.P., LAM, K., LI, W.K., (1998).** "A Stochastic Volatility Model with Markov Switching", *Journal of Business & Economics Statistics*, Vol.16, No.2, pp.244-253

117.**SPEEKENBRINK, Maarten, (2016).** "A Tutorial on Particle Filters", *Journal of Mathematical Psychology*, August 2016, Vol.73, pp.140-152

118.**STORVIK, G. (2002).** "Particle filters in state space models with the presence of unknown static parameters", *IEEE Transactions on Signal Processing*, Vol. 50, pp. 281-289

119.**STROUD, J. and JOHANNES, M. (2014).** "Bayesian Modeling and Forecasting of 24-Hour High-Frequency Volatility", *Journal of the American Statistical Association*, July 2014, Vol.109, No.508, pp.1368-1384

120.**SZERSZEN, Pawel J., (2009).** "Bayesian Analysis of Stochastic Volatility Models with Levy Jumps: Application to Risk Analysis", *Finance and Economics Discussion Series*, Division of Research & Statistics and Monetary Affairs, Federal Reserve Board, Washington, D.C., 2009.

121.**TAKAHASHI, M., OMORI, Y., WATANABE, T., (2009).** "Estimating Stochastic Volatility Models Using Daily Returns and Realized Volatility Simultaneously", *Computational Statistics & Data Analysis*, April 2009, Vol.53, No.6, pp.2404-2426.

122.**TAYLOR, Stephen J., (1986).** "*Modeling Financial Time Series*", Chichester, UK, John Wiley and Sons, ISBN: 978-981-277-084-4

123.**TODOROV, Viktor, (2009).** "Variance Risk Premium Dynamics: The Role of Jumps", *The Review of Financial Studies*, January 2010, Vol.23, No.1, p.345-383.

124.**VORTELINOS, Dimitrios I., (2017).** "Forecasting Realized Volatility: HAR Against Principal Components Combining, Neural Networks and GARCH", *Research in International Business and Finance*, January 2017, Vol.39, Part B, pp.824-839.

125.**WITZANY, Jiří, (2013).** "Estimating Correlated Jumps and Stochastic Volatilities", *Prague Economic Papers*, 2013, Vol.2013, No.2, p.251-283.

126.**YSUSI, Carla, (2006).** "Detecting Jumps in High-Frequency Financial Series Using Multipower Variation", *Banco de Mexico*, September 2006, Working paper, No.2006-10, p.1-35.

127. **ZAKOIAN, J. M., (1994).** “Threshold Heteroskedastic Models“, *Journal of Economic Dynamics and Control*, Vol.18, pp. 931-955

128. **ZHANG, L., MYKLAND, Per A. and AIT-SAHALIA, Y., (2005).** “A Tale of Two Time Scales: Determining Integrated Volatility With Noisy High-Frequency Data“, *Journal of the American Statistical Association*, December