Erasmus School of Economics

Bachelor Thesis Econometrics, Quantitative Finance

July 4, 2020

Evaluating Individual and Combined Density Forecasting Models

---

Author:
Vera Roersma
Student ID number: 471416

Supervisor:
Dr. M.D. Zaharieva
Second Assessor:
B. van Os MSc

Abstract

With simulation-based density forecast evaluation methods using auxiliary particle filters, this paper evaluates several density forecasting models for S&P 500 index returns, namely log stochastic volatility models and one- and two-factor affine jump diffusion models. From the individual models, combined models are constructed via the optimal pooling method. The empirical results show that jumps in volatility attain importance during the global financial crisis of 2007 until 2009. Furthermore, the combined models and two-factor models with jumps perform relatively good a propos of value-at-risk evaluations, but the one-factor affine diffusion model with jumps in returns and volatility has the most consistent relatively good performance in all other evaluation methods, consisting of an informal study of the probability integral transforms and Hong and Li, Berkowitz, Kolmogorov–Smirnov, and Diebold-Mariano test statistics.

Contents

1 Introduction

Forecasting the density of asset returns has gained much attention in the field of financial econometrics. The volatility of asset returns is considered a measure of risk, for which investors want to receive a premium. The distributions of returns are therefore important, for example, for optimal portfolio allocation and the pricing of options. Daily asset returns are characterized by fat tails and a higher peak at the mean (leptokurtosis), hence they are not easy to capture by, for example, a normal distribution. This paper focuses on stochastic volatility models for forecasting the density of S&P 500 index returns.

Taylor (1982) introduced the first stochastic volatility model, namely a discrete-time stochastic volatility model for the logarithm of the variance (LSV). It was less popular than the GARCH (Engle (1982), Bollerslev (1986)) models, because it was difficult to estimate due to the presence of unobservable stochastic volatilities in its model specification (Yun, 2020).

Later on, affine jump diffusion (AJD) models became popular, because they offer the possibility to price options via a closed form formula. Heston (1993) was the first to use the stochastic volatility model without jumps (SV) for this purpose, which is therefore also called the Heston model. Because the SV model seemed unable to explain large outliers in returns, such as during the October 1987 crash, Bates (1996) extends this model by allowing for jumps in returns (SVJ). Duffie et al. (2000) introduce a model with an additional jump in the volatility (SVCJ). Eraker et al. (2003) study this model extensively and find strong evidence for jumps in returns and volatility. They argue that jumps in volatility are important as they allow the volatility to rapidly increase, what happens in times of market distress. The jumps in returns remain important, because they can generate large crashlike movements. Yun (2020) finds that these models are capable of capturing the leverage effect (the asymmetric return-volatility relation), the conditional fat tail (through the jump components), and the asymmetric conditional distribution through nonzero mean jump size.

Bates (2000) proposes the the addition of a diffusive variance factor. Where the one-factor AJD models assume that the volatility factor reverts to a constant long-term mean, in the so-called two-factor affine jump models, this long-term mean is modelled as an additional diffusive latent variance factor. Hence, the long-term mean is allowed to vary over time. Bates (2000) argues that the additional factor can capture the volatility term structure in a more realistic way than one-factor models.

Besides density forecasts that are based on only one forecasting method, combinations of methods have also been investigated. Durham & Geweke (2013) compare different individual models

for density forecasting, and their combinations. They combine SV models, exponential generalized autoregressive conditional heteroscedastic (EGARCH) models, high-frequency models and option prices models, using the newly proposed "optimal pooling" method that combines the different density forecasts and evaluate the different (combined) models using likelihood values. They find that the optimal pools outperform all of the individual models in terms of log-likelihood. Apart from combining density forecasts with different information, Zhao (2013) also combines models within their own "class". He finds that the combination of the models within the class of SV models, consisting of four SV models with different distributions for the error term in the formula for the return, outperform three out of four individual models.

Over the past decades, in addition to the model specifications themselves, the evaluation of density forecasts has also been given much attention. Diebold et al. (1998) introduce the use of the probability integral transform (PIT), as introduced by Rosenblatt (1952), to evaluate GARCH density forecast. The idea of their evaluation is based on the fact that, given that the sequence of density forecasts $p_t(y_t)$ is equal to the data generating process $f(y_t)$, the density of the PIT z_t, $q(z_t)$, simply is the $U(0,1)$ density. The formulation of the PIT is as follows:

$$z_t = \int_{-\infty}^{y_t} p_t(u)du = P_t(y_t), \tag{1}$$

where $P_t(y_t)$ is the cumulative density of the estimated density. The PITs can be used to test if the density forecasts equate to the "true" density. To formally test the uniformity of the PITs, the Hong & Li (2005), Berkowitz (2001) and Kolmogorov–Smirnov can be drawn. Diebold et al. (1998) argue that formal tests should be accompanied by an informal study, to provide guidance into the question as to why the tests were rejected. For this purpose, they study the PITs visually by means of a histogram and correlograms of $(z_t - \bar{z})$ and its powers.

Since the introduction by Diebold et al. (1998), the PIT has has been widely used in the evaluation of density forecasts (Bao et al. (2007), Gneiting & Raftery (2007), Amisano & Giacomini (2007), among others). Yun (2020) uses the PIT to evaluate not only GARCH, but also stochastic volatility models (LSV, AJD and two-factor volatility models). As it is not possible to compute the PIT analytically for those models, he proposes a simulation based method to approximate it. This method builds on the auxiliary particle filter algorithm as proposed by Pitt & Shephard (1999), which was already applied to stochastic volatility models by Johannes et al. (2009). Moreover, this method makes it possible to approximate the likelihood values of the different models, which allows the models to be compared via the Diebold-Mariano test.

Using the simulation based density forecast evaluation from Yun (2020), this paper investigates

several stochastic volatility models. Following Yun (2020), I estimate LSV models, one-factor AJD models (SV, SVJ, and SVCJ) and the two-factor affine model without jumps (SVM). His simulation based density forecast evaluation is used to evaluate and compare the models. Furthermore, two-factor models with jumps in returns (SVMJ) and jumps in returns and volatility (SVMCJ) are estimated and evaluated in the same way, in order to examine whether these models can lead to an improvement compared to the models previously examined. The algorithm of Yun (2020) makes it possible to construct several model combinations using the "optimal pooling method" of Durham & Geweke (2013). This method provides weights, which can be used to compute the PITs and likelihoods of the combined models in order to evaluate all models with the same metrics. In addition, the weights provide better insight into the relative performance of the various models over time.

The first result of this paper is that the weight distribution of the combined models changes in favor of models with jumps in return ánd volatility during the global financial crisis of 2007 until 2009. Thus, it is useful to allow for abrupt changes in volatility during periods of market distress, which is in line with the findings in Eraker et al. (2003). Furthermore, during this period, the weight of SVMCJ rises at the expense of the weight of SVCJ. This suggests that the long-term average was not constant during this period. Next, the performance of the models differed across tests. SVCJ has the most consistent relatively good performance. However, for value-at-risk analysis, the two-factor models with jumps and the combined models outperformed the other models.

The structure of this paper is as follows. Section 2 introduces all the models that are analyzed. Subsequently, the methods to evaluate and compare the models are discussed in Section 3. The models are then estimated for S&P 500 index returns over the in-sample period from 1988 until 2000, and evaluated over two out-of-sample periods, from 2001 until 2007 and from 2001 until 2014 in Section 4. Finally, the main findings are highlighted in Section 5.

2 Models for Density Forecasting

In this section, several discrete- and continious-time stochastic volatility models are considered. To be able to compare the results with the results from Yun (2020), his model specifications are used. In what follows, y_t represents the log return in percentage at time t, i.e. $y_t = 100 * \ln\left(\frac{S_t}{S_{t-1}}\right)$, where S_t is the stock price. In order to estimate and evaluate the models, the continuous-time models have to be discretized. The discretized schemes and the estimation methods can be found in Appendix B.

2.1 Log Stochastic Volatility (LSV) Models

Yun (2020) formulates the LSV model as introduced by Taylor (1982) as follows:

$$\begin{aligned} y_t &= \mu + \sqrt{h_t}\varepsilon_t \\ \ln h_t &= \alpha + (1+\beta)\ln h_{t-1} + \sigma_h \rho \varepsilon_{t-1} + \sigma_h \sqrt{(1-\rho^2)}\eta_t. \end{aligned} \tag{2}$$

where h_t is the conditional variance, ρ the parameter that captures the leverage effect, and ε_{t-1} and η_t are both Gaussian and uncorrelated, i.e. $(\varepsilon_{t-1}, \eta_t) \sim i.i.d.N(0, I_2)$. The volatility of the volatility, σ_h, is restricted to be positive, and $-1 < (1+\beta) < 1 \iff -2 < \beta < 0$ to the ensure stationarity of the log of the variance. The model that does not take into account the leverage effect, which means that ρ is set to zero, will be referred to as LSV0.

2.2 Affine Jump Diffusion (AJD) Models

Second, continuous-time Affine Jump Diffusion (AJD) models are considered. The general framework is the SVCJ model, as formulated by Yun (2020):

$$\begin{aligned} d\ln S_t &= \mu dt + \sqrt{V_t} dW_t^s + Z_t^S dN_t \\ dV_t &= \kappa(\theta - V_t)dt + \sigma_v \sqrt{V_t} dW_t^V + Z_t^v dN_t, \end{aligned} \tag{3}$$

where W_t^i, for $i = S, V$, are standard Brownian Motions with $\mathrm{Cov}(\mathrm{W}_t^S, \mathrm{W}_t^V) = \rho \mathrm{d}t$. $Z_t^S \sim i.i.d.N(\mu_S, \sigma_S^2)$ denotes the size of the jump in the price, and $Z_t^V \sim i.i.d.\exp(\mu_V)$ the size of the jump in the variance. $N_t \sim Poi(\lambda)$ denotes the jump timing. From the general framework, the SV model is obtained by setting $Z_t^S dN_t = Z_t^v dN_t = 0$, and the SVJ model by setting $Z_t^v dN_t = 0$. The parameters can be interpreted as follows. κ is the speed of mean reversion, θ the mean reversion level and σ_v the volatility of volatility (Drimus, 2012). V_t denotes the spot variance. Because volatility cannot be negative, σ_v and V_t are restricted to be non-negative.

2.3 Two-Factor Models (TFM)

The aforementioned AJD models consider only one volatility factor. In two-factor affine jump diffusion models (TFM), this value is allowed to vary over time, and is modelled as an additional diffusive latent variance factor. Following Yun (2020), who analyzed the SVM model, the models are implemented using formulations of Kaeck & Alexander (2012). The general framework is the SVMCJ model, which is formulated as follows:

$$\begin{aligned} d\ln S_t &= \mu dt + \sqrt{V_t} dW_t^s + Z_t^S dN_t \\ dV_t &= \kappa(M_t - V_t)dt + \sigma_v \sqrt{V_t} dW_t^V + Z_t^v dN_t \\ dM_t &= \kappa_M(\theta_M - M_t)dt + \sigma_M \sqrt{M_t} dW_t^M, \end{aligned} \tag{4}$$

where W_t^i, for $i = S, V, M$, are standard Brownian Motions with $\text{Cov}(\dot{W}_t^S, \dot{W}_t^V) = \rho dt$, and W_t^M is independent of both W_t^S and W_t^V. $Z_t^S \sim i.i.d.N(\mu_S, \sigma_S^2)$ denotes the size of the jump in the price, $Z_t^V \sim i.i.d.\exp(\mu_V)$ the size of the jump in the variance. $N_t \sim Poi(\lambda)$ denotes the jump timing. From the general framework, the SVM model is obtained by setting $Z_t^S dN_t = Z_t^v dN_t = 0$, and the SVMJ model by setting $Z_t^v dN_t = 0$. M_t represents the long-term mean of the spot variance. This factor in turn exhibits mean reversion: κ_M is the speed of mean reversion, θ_M the mean reversion level, and σ_M is the volatility of the long-term mean of the spot variance M_t. All volatilities (σ_v, σ_m, V_t, and M_t) are restricted to be non-negative.

2.4 Combined Density Forecasts

The eight models as introduced above provide a sequence of conditional probability densities for the log return in percentage:

$$p_t(y_t|y^{t-1}, A_m), \tag{5}$$

where A_m denotes the specific method for density forecasting, $A_m \in \{LSV0, LSV, SV, SVJ, SVCJ, SVM, SVMJ, SVMCJ\}$, and $y^{t-1} = \{y_1 ... y_{t-2}, y_{t-1}\}$. For the forecast combination, I use the "optimal pooling method", which was introduced by Durham & Geweke (2013). The optimality of this combination method lies in the fact that it does not assume that one model is the "true" model, because all density forecasts are, in essence, false (Durham & Geweke, 2013). The different forecasts $p_t(y_t|y^{t-1}, A_m)$ are combined into a single forecast: $p_t(y_t|y^{t-1}, \boldsymbol{w_{t-1}})$. For day t, the weights are computed at the close of trading day $t-1$, and are chosen in such a way that its log-likelihood f_{t-1} is maximized:

$$f_{t-1}(\mathbf{w}_{t-1}) = \sum_{s=q+1}^{t-1} \log[\sum_{m=1}^{M} w_{t-1,m} p(y_s|y^{s-1}, A_m)], \tag{6}$$

where M is the number of models that is being combined. The optimal weight vector $\mathbf{w^*}_{t-1}$ satisfies

$$\sum_{m=1}^{M} w_{t-1,m}^* = 1 \qquad\qquad w_{t-1,m}^* \geq 0, \text{ for } m = 1, ...M. \tag{7}$$

Maximization of (6) is a regular convex programming problem (Durham & Geweke, 2013). The number q represents the number of observations in the in-sample period. For each t in the out-of-sample period, the weight for that period is computed by (6). Apart from maximizing the log-likelihood, this weight vector furthermore minimizes the Kullback–Leibler information criterion, i.e. the distance between the combined forecast density and the true density (Hall & Mitchell, 2007).

The weights are used to compute the predictive densities over two out-of-sample periods:

$$p(y_t|y^{t-1}, \boldsymbol{w_{t-1}}) = \sum_{1}^{M} w_{t-1,m} p(y_t|y^{t-1}, A_m). \tag{8}$$

Using $\boldsymbol{w *_t -1}$, the log-likelihood value at time t for the combined model can be calculated. Furthermore, the PIT for the combined models is computed as the weighted average of the PIT for the models that are combined, where $z_{t,m}$ is the PIT for model m:

$$z_t = \sum_{1}^{M} w_{t-1,m} z_{t,m}. \tag{9}$$

This paper considers six model combinations, that can be found in Table 1. In addition to the possible improvement that combining models can lead to, the weights provide better insight into the relative performance of the models over time. By studying plots of the weights, it becomes clear in which period a specific model performs better in terms of the log-likelihood, because it gets assigned more weight.

Table 1: Model combinations

Combination model name	Models
CLSV	LSV0, LSV
CAJD	SV, SVJ, SVCJ
CTFM	SVM, SVMJ, SMVCJ
CBEST3	SVJ, SVCJ, SVMCJ
CAJDTFM	SV, SVJ, SVCJ, SVM, SVMJ, SMVCJ
CALL	LSV0, LSV, SV, SVJ, SVCJ, SVM, SVMJ, SMVCJ

3 Methods for Model Evaluation and Comparison

In the first place, the models in this paper are evaluated using the fact that the PITs should be standard uniformly distributed in the case that a model is correctly specified (Rosenblatt, 1952). Yun (2020) shows that analytic methods are almost inapplicable to compute the PITs for models discussed in Section 2, because of multiple integrations. The models are also evaluated by means of their likelihood, for which the same problems apply when you try to solve them analytically. Therefore, he proposes a method that approximates the PITs and likelihood values via Monte-Carlo integration using the auxiliary particle filter algorithm.

The PITs of each model are evaluated using the informal study of Diebold et al. (1998). In addition, the formal Hong and Li (HL) test, Berkowitz test, and Kolmogorov–Smirnov (KS) test are

conducted[1]. Yun (2020) investigates the size-performance of these tests when his auxiliary particle filter algorithm is used to approximate the PITs. His findings are summarized in Appendix E. Next to evaluations based on the PIT, likelihood-based evaluations are also often used for evaluating predictive densities. Using the approximated log-likelihood values, the Diebold-Mariano test is used to compare the out-of-sample performance of the models.

This section first gives a brief introduction to auxiliary particle filter algorithms, and briefly discusses the steps of the specific algorithm of Yun (2020). Then all tests are discussed.

3.1 Simulation-Based Dynamic Probability Integral Transform

The auxiliary particle filter algorithm was introduced by Pitt & Shephard (1999). They define particle filters as the class of simulation filters that recursively approximate the latent random variable $L_t|y^t = (y_1,...y_t)'$ by "particles" $L_t^1,...L_t^M$, with *discrete* probability mass of $\pi_t^1,...\pi_t^M$. Thus, a continuous variable is approximated by a discrete one with random support, where the discrete points L_t^i are viewed as samples from $f(L_t|Y_t)$. As $M \to \infty$, the particles can be used to increasingly well approximate the density $f(L_t|y^t)$. Johannes et al. (2009) first used the auxiliary particle filter in the setting of stochastic volatility models. They discuss the various possibilities offered by the algorithm. Among other things, it is possible to assess the fit of a density forecast model. In order to evaluate the density forecasts, I use the auxiliary particle filter algorithm as described in Yun (2020), who extended the algorithm of Johannes et al. (2009) and made it possible approximate the PIT and likelihood.

In this section, I briefly go over each step of the auxiliary particle filter algorithm for the LSV models and the one- and two-factor AJD models, as described in detail by Yun (2020). This section briefly mentions the steps the algorithm consists of, and tries to explain the intuition behind the algorithm. For the exact details I refer to Yun (2020), who describes the steps in detail.

3.1.1 Log Stochastic Volatility Models

The LSV0 and LSV model are defined in (2), where $\rho = 0$ for LSV0. For these models, the latent variable L_t for which I aim to estimate the density of $L_t|y^t$ consists of the volatility h_t. The algorithm starts with an initial set of N particles $\{h_0^{(i)}\}_{i=1}^N$, drawn from a model-implied marginal distribution for h_t, where N is set to $25,000$. The following steps will be iterated.

In the first step, the auxiliary variable is introduced as the conditional expectation of h_t given $\hat{h}_{t-1}^{(i)}$ and y_t, which follows from (2). For the mathematical expression of the auxiliary variable, I refer

[1] To compute test statistics for the HL, Berkowitz and KS test, functions are readily available in R.

to Yun (2020). The variable is "auxiliary", because it is present simply to aid the task of simulation (Pitt & Shephard, 1999). In the second step, the particles are resampled with weights w_t. After resampling, the volatility particles are tilted toward those that are likely to have generated y_t. For example, if $|y_t|$ is large, resampling results in higher particles h_t (Johannes et al., 2009). This reduces the cost of sampling many times from particles with very low likelihoods of being resampled in the next step, which improves statistical efficiency. In the third step, the particles are resampled for the second time. Second stage resampling is necessary, because the first-stage resampling is not exact (Johannes et al., 2009). In the fourth step, the PIT z_{t+1}, the log-likelihood l_{t+1} and likelihood $p(y_{t+1}|y^t)$ are computed. For the model combination, it is required to store all likelihoods.

3.1.2 One- and two-factor Affine Jump Diffusion Models

The particle filter algorithm requires the functional forms for both the likelihood function and the latent state process to be known. Therefore, the discretized processes in (16) and (20) in Appendix B are used in the algorithm for the AJD and the TFM models. I cover the most extensive models (SVCJ and SVMCJ). The less elaborate models follow from this: the steps with the jump in volatility, or both jumps can be taken out. For the one-factor AJD models, the latent variable L_t for which I aim to estimate the conditional density $L_t|y^t$ consists of the volatilities in the previous period V_{t-1}, the jump times J_t, the size of the shock in returns Z_t^S and the size of the shock in volatilities in the previous period Z_{t-1}^V. For the two-factor models, there is one additional latent variable: the long-term mean of the spot variance M_t. The algorithm starts with an initial set of N particles at time $t-1$ $\{L_{t-1}^{(i)}\}_{i=1}^N$, drawn from a model-implied marginal distribution of L_{t-1} given all information until $t-1$, $y^{t-1} = (y_1, ...y_{t-1})$, where N is set to 25,000. The following steps will be iterated.

In the first step, the auxiliary variable is introduced as the expectation of the spot variance V_t conditional on the particles from the previous iteration. For the two-factor models, an additional auxiliary variable is introduced as the expectation of the long-term mean of the spot variance M_t conditional on the particles from the previous iteration. For the mathematical expression of the auxiliary variables, I refer to Appendix C. Using the auxiliary variables, the particles are resampled in the second step. In the third step, a jump occurrence is generated for each resampled particle, using a probability that follows from the model and the auxiliary variables. In the fourth and fifth step, jump-in-price and jump-in-variance sizes are simulated for each resampled particle. In the sixth step, the particles are resampled with second-stage weights to overcome sample impoverishment. In the seventh step, the PIT z_t, log-likelihood l_t and likelihood value $p(y_t|y^{t-1})$ are computed. After this, the particles are updated.

3.2 Hong and Li (HL) test

The idea behind the test of Hong & Li (2005) is to measure the distance between a model-implied transition density and the true transition density. This is done by comparing a kernel estimator for the joint density of the pair of PITs $\{z_t, z_{t-j}\}$ with the product of two $U(0,1)$ densities, where j is the lag order. This follows from the idea that the PITs should be uniformly distributed.

The HL test requires specification of the kernel $k(\cdot)$, and the bandwith h. Following Hong & Li (2005) and Yun (2020), the quartic kernel is chosen, and the bandwith is chosen such that it attains the optimal rate for the bivariate density estimation: $h = \widehat{S_Z} T^{-\frac{1}{6}}$, where $\widehat{S_Z}$ is the sample standard deviation of $\{z_t\}_{t=1}^{T}$ (Scott, 2015). The test statistics $\widehat{Q_j}$ provide insight in the exact lag orders where there is a departure from $U(0,1)$. However, as the goal of this paper is to evaluate each model as a whole, it is more convenient to use a single test statistic for each model. For this purpose, Hong et al. (2007) propose a portmenteau test statistic that combines p lag truncation orders: $\widehat{W}(p)$. Both test statistics $\widehat{Q_j}$ and $\widehat{W}(p)$ converge to $N(0,1)$. The test statistics for the HL test can be found in Appendix 3.2. Even when the models that are being compared are all rejected according to the $\widehat{W}(p)$ test statistics, it can be concluded that the model with the smallest test statistic is a better density forecast model (Hong et al., 2007). Due to non-trivial correlations between $\widehat{Q_j}$ in finite samples, this test statistic tends to over-reject. Therefore, the empirical critical values from Yun (2020) are used.

3.3 Berkowitz Test

If the PITs z_t are $i.i.d.U(0,1)$, it should hold that the transformed variable $x_t = \Phi^{-1}(z_t)$ is $i.i.d.N(0,1)$, where $\Phi(\cdot)$ denotes the CDF of the standard normal distribution. This corresponds to the null hypothesis of the Berkowitz test, in which $x_t = \eta_t$, with $\eta_t \sim i.i.d.N(0,1)$. The alternative hypothesis is that x_t follows the AR(1) process $x_t - \mu = \rho(x_{t-1} - mu) + \varepsilon_t$, where $\varepsilon_t \sim i.i.d.N(0, \sigma^2)$. Under the null hypothesis, $\mu = \rho = 0$ and $\sigma^2 = 1$. In order to test this, the likelihood ratio

$$LR_{BW} = -2(L(0,1,0) - L(\mu, \sigma^2, \rho)) \tag{10}$$

is computed. Under the null-hypothesis, $LR_{BW} \sim \chi^2(3)$, because of the three restrictions.

3.4 Kolmogorov–Smirnov (KS) Test

The KS test is the most straight-forward, and tests if $z_t \sim U(0,1)$, while assuming that $\{z_t\}$ is $i.i.d$. The test statistic KS is computed as follows

$$KS = \max_{0<z<1} |\widehat{CDF}(z) - z|. \tag{11}$$

3.5 Diebold-Mariano test for likelihood comparison

The log-likelihood for each model m over a period running from 1 to T can be defined as

$$L_T^{(m)} = \sum_{t=1}^{T} l^{(m)}(y_t|y^{t-1}), \tag{12}$$

where $l^{(m)}(y_t|y^{t-1})$ is the likelihood of model m at time t. Because the model with the highest likelihood is the best prediction model in terms of the Kullback-Leibler information criterion, conclusions can be drawn about the relative performance of a density forecast model even if all models are misspecified. The log-likelihoods are compared using the Diebold-Mariano test for likelihood comparison of model m and n:

$$DM = \frac{\bar{d_T}}{\widehat{\sigma_d}/\sqrt{T}}, \tag{13}$$

where $\bar{d_T}$ is the sample average of $\{d_t\}_1^T$, where $d_t = l^{(m)}(y_t|y^{t-1}) - l^{(n)}(y_t|y^{t-1})$, and $\widehat{\sigma_d}$ is the standard deviation of d_t. Under the null hypothesis, DM is standard normally distributed.

3.6 Value-at-Risk evaluations

Because large negative returns in the left tail of the distribution are related to portfolio loss, predicting the VaR can be important for decision takers. The VaR at $1-q\%$ is the return that is expected to be exceeded with probability $1-q$. For this purpose, this paper evaluates the VaR of all models using the likelihood ratio test of Christoffersen (1998) and the dynamic quantile (DQ) test of Engle & Manganelli (2004). Both tests use an indicator variable $I_t = (y_t < VaR_t(1-q))$, where $VaR_t(1-q)$ represents the q-th quantile of the distribution of returns. Hence, the approximated PIT can be used to directly evaluate the VaR predictive performance without having to calculate actual VaR measures, because $I(y_t < VaR_t(1-q)) = I(z_t < (1-q))$. The VaR evaluation of Christoffersen (1998) consists of three test-statistics. LR_{uc} tests for correct unconditional coverage, i.e. the average correctness to account for higher order dynamics. LR_{ind} tests for the independence of VaR violations over time, and LR_{cc} is a joint test statistic for the sum of LR_{uc} and LR_{ind}, and tests for correct conditional coverage. For the test statistics, I refer to Christoffersen (1998). The DQ test is a joint test for both unconditional coverage and independence of VaR-violations. The DQ test-statistic that is used can be found in Appendix F. This paper tests for $q = 95\%$ and $q = 90\%$.

Table 2: Parameter estimates for the density forecast models

Log Stochastic Volatility Models

Parameter	LSV0	LSV
μ	0.0681 (0.0126)	0.0469 (0.0129)
α	-0.0083 (0.0037)	-0.0121 (0.0044)
β	-0.0180 (0.0054)	-0.0261 (0.0065)
σ	0.1505 (0.0193)	0.1858 (0.0219)
ρ	-	-0.4977 (0.0601)
Log-likelihood	-4089.14	-4083.43

Affine Jump Diffusion Models

	One-factor models			Two-factor models		
Parameter	SV	SVJ	SVCJ	SVM	SVMJ	SVMCJ
μ	0.0386 (0.0129)	0.0435 (0.0128)	0.0441 (0.0127)	0.0316 (0.0129)	0.0378 (0.0129)	0.0309 (0.0125)
κ	0.0368 (0.0084)	0.0214 (0.0056)	0.0258 (0.0058)	0.0254 (0.0054)	0.0087 (0.0060)	0.0131 (0.0055)
θ	0.8998 (0.0958)	0.9115 (0.1292)	0.7309 (0.0989)	-	-	-
σ_V	0.1950 (0.0234)	0.1420 (0.0170)	0.1365 (0.0156)	0.1695 (0.0183)	0.1247 (0.0182)	0.1306 (0.0154)
ρ	-0.5827 (0.0538)	-0.5827 (0.0538)	-0.6094 (0.0555)	-0.4916 (0.0397)	-0.6122 (0.0562)	-0.6323 (0.0536)
λ	-	0.0099 (0.0047)	0.0078 (0.0033)	-	0.0115 (0.0055)	0.0054 (0.0019)
μ_S	-	-2.0201 (0.9539)	-2.8405 (1.0178)	-	-1.9333 (0.9753)	-3.6419 (0.8359)
σ_S	-	2.2106 (0.3952)	2.1904 (0.3922)	-	2.1117 (0.3578)	2.0691 (0.3781)
μ_V	-	-	1.9277 (0.4883)	-	-	2.0885 (0.4061)
κ_M				0.6024 (0.1858)	0.5897 (0.2097)	0.5923 (0.1981)
θ_M				1.0310 (0.0508)	1.8320 (0.6185)	1.2616 (0.3379)
σ_M				0.3118 (0.01286)	0.3883 (0.0575)	0.3402 (0.0401)
Log-likelihood	-4074.92	4041.64	-4039.43	-4084.76	-4048.48	-4047.39

Notes: All models are estimated for daily S&P index data from 1988 until 2000 (3586 observations). The mean and standard deviation of the posterior distribution of each parameter for each model are reported. The models have been estimated using 110,000 iterations, with a burn-in period of 10,000 iterations. An overview of the abbreviations for the models and their meanings can be found in Appendix 6.

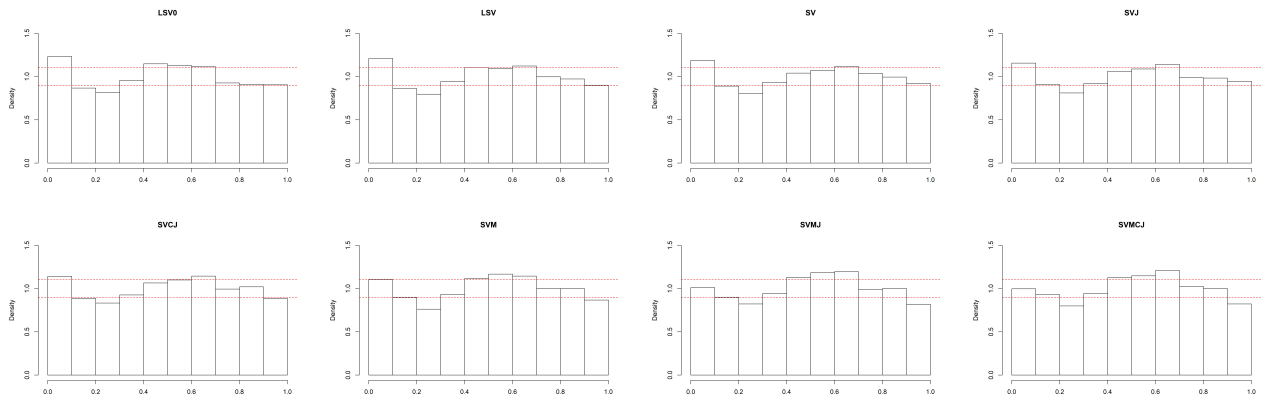
4 The auxiliary particle filter algorithm applied to the index returns of the S&P 500

For comparison reasons, this paper examines the same in-sample and out-of-sample periods as Yun (2020). To estimate the models, daily S&P 500 price indices from January 1988 until December 2000 (retrieved from Datastream) are used to calculate the log returns in percentage. To obtain only the trading days from this sample, I used the package "bizdays" in R. This resulted in 3286 observations, which is one observation more than in Yun (2020), possibly because I used the last price index from 1987 to calculate the first return in 1988. The first out-of-sample period spans from January 2001 until December 2007 (1764 observations), and the second from January 2001 until December 2014 (3529 observations). The amount of observations differs slightly from Yun

(2020), which might be caused by using a different method for determining the trading days.

Table 2 shows the parameter estimation results for each model. Each model is estimated using Bayesian MCMC, using 110,000 iterations and discarding the first 10,000 iterations. Appendix B deals with the model estimation in detail, and also mentions which prior distributions have been used. The reported log-likelihood values are approximated using the auxiliary particle filter algorithm as discussed in Section 3.1. The results for the LSV models and the AJD models are similar to those of Yun (2020). For the SVM model, however, the estimated values for κ, κ_M and σ_M are significantly different. The cause of this big difference could be a misspecification of the model that is unknown to me, but it does yield different results with respect to the in-sample performance.

For the LSV models, the inclusion of leverage effects results in a higher in-sample log-likelihood. For the AJD models, the log-likelihood values of the SVJ and SVCJ models are higher than for SV. In contrast to Yun (2020), SVM has a lower in-sample likelihood than SV. Adding jumps in the both one-factor and two-factor models leads to an increase in the in-sample log-likelihood. The log-likelihood values for SVMJ and SVMCJ are higher than those of SV, and are not far from SVJ and SVCJ, respectively. It is therefore possible that, when the fit of the two-factor models is improved, these models perform better in sample than the one-factor AJD models with jumps.

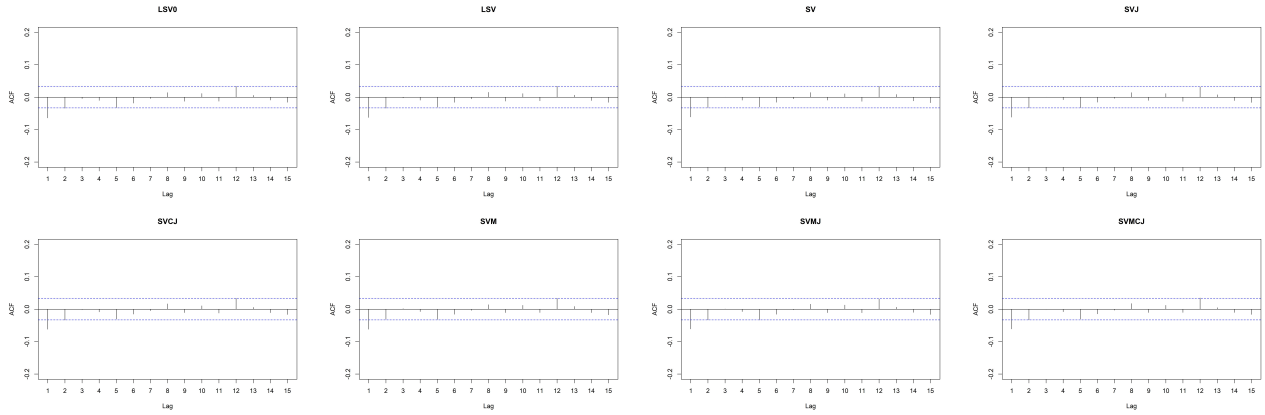


*Notes: Out-of-sample 2 is used to calculate the PITs for each model. The dotted lines are the 95% interval under the null hypothesis of a uniform distribution (computed as 1 +/- 1.96 * 1/$\sqrt{3529/10}$). An overview of the abbreviations for the models and their meanings can be found in Appendix 6.*

Figure 1: Histograms of the PITs for the different density forecast models.

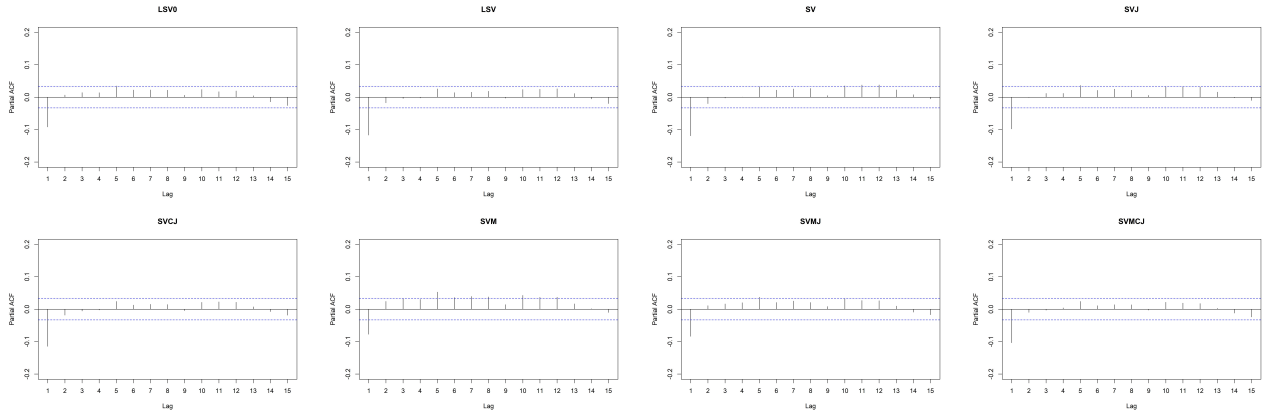
4.1 Informal Investigation

With the parameters from Table 2, the series of probability integral transforms $\{z_t\}$ and of log-likelihoods $\{l_t\}$ can be obtained via the auxiliary particle filter algorithm as explained in Section 3.1.



Notes: Out-of-sample 2 is used to calculate the series of PITs $\{z_t\}$ for each model. An overview of the abbreviations for the models and their meanings can be found in Appendix 6.

Figure 2: Autocorrelation function of $(z - \bar{z})$



Notes: Out-of-sample 2 is used to calculate the series of PITs $\{z_t\}$ for each model. An overview of the abbreviations for the models and their meanings can be found in Appendix 6.

Figure 3: Autocorrelation function of $(z - \bar{z})^2$

This research uses codes provided by Yun (2020) (with a few minor adjustments) to implement the auxiliary particle filter algorithm and approximate the PITs and likelihoods for the LSV models, AJD model and SVM. To evaluate the two-factor models, jumps were integrated in the existing code for SVM. Following Diebold et al. (1998), the histograms of z are examined to investigate if the PITs are standard uniformly distributed. To evelauate wheter z is *i.i.d.*, its correlograms are examined. The correlogram of $(z - \bar{z})$ provides information about dependence patterns through the conditional mean, and the correlogram of $(z - \bar{z})^2$ through the conditional variance (Diebold et al., 1998).

Figure 1 shows similar-looking histograms for all models. Except for SVMJ and SCMCJ, all models exhibit a significant peak at the left end, which is highest for LSV0 and LSV. Hence, too many realizations fall in the left tail of the forecast densities relatively to what is expected for

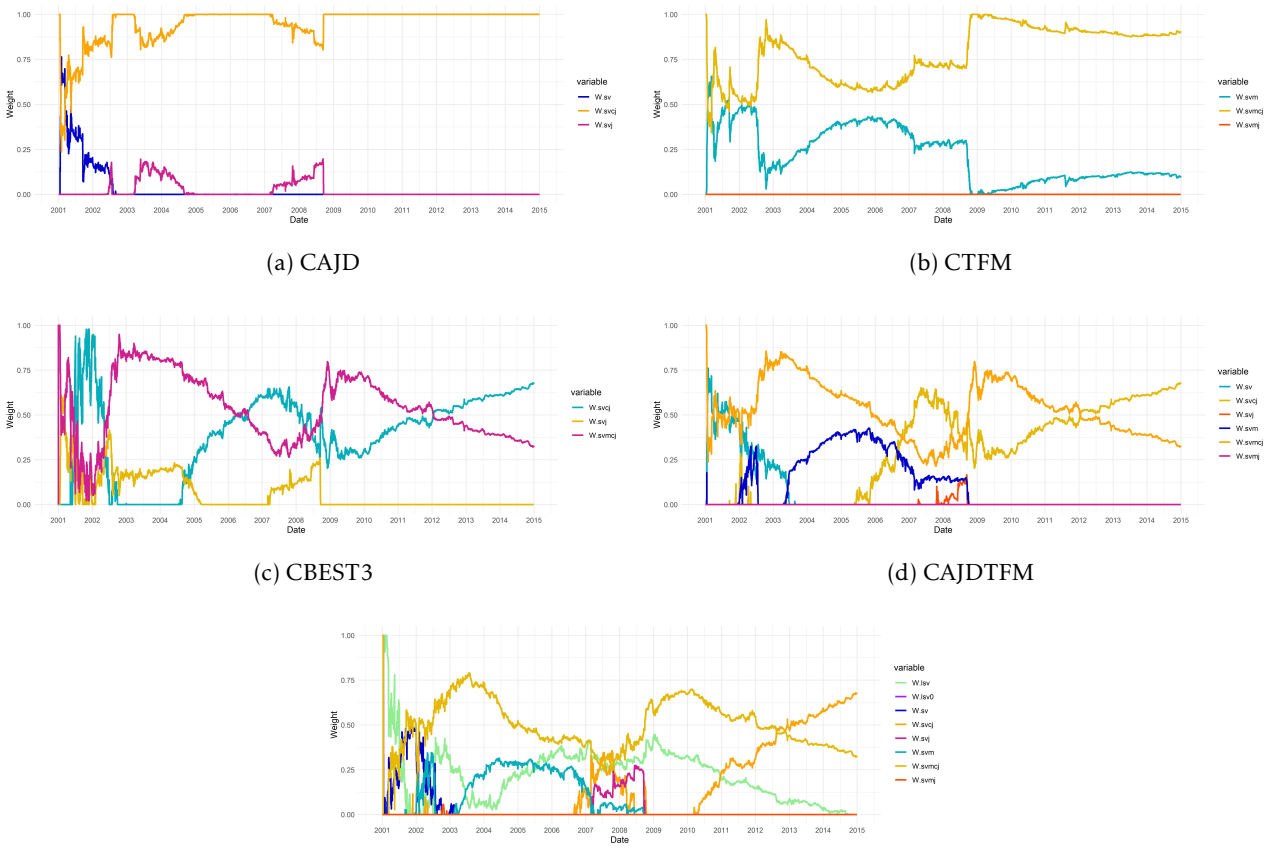
the models. Furthermore, all models have a hump in the middle, around 0.6, indicating that too many realizations fall in the middle of the forecast densities relatively to what is expected for the models. This hump is most significant for SVMJ. The histograms thus show us that the models do not entirely succeed in capturing the leptokurtosis of the daily returns, but many of the bins are within the 95% confidence interval or just at the boundary of significance.

Figure 2 shows the autocorrelation function of $(z - \bar{z})$. The same pattern can be observed for all models. The autocorrelation for the first lag is significant for all models, and for the twelfth lag it is on the boundary of significance. Thus, the models perform similar in regard of conditional mean dynamics: the correlation in the series indicates the density forecasting models do not fully succeed in capturing the mean dynamics, because of the significant first lag. Figure 3 shows the autocorrelation function of $(z - \bar{z})^2$. Here, it can be seen that there are differences between the models. All models have a significant negative autocorrelation for the first lag, and thus do not fully succeed in capturing the conditional volatility dynamics. Unlike for the other models, none of the autocorrelations for higher lags are significant or on the boundary of significant for SVCJ and SVMCJ, indicating that those models most adequately capture the conditional volatility dynamics.

4.2 Combination results

In this section, the results of the model combination as described in Section 2.4 are discussed. Using the approximated likelihoods $p_t(y_t|y^{t-1})$, the weights for the different model combinations are computed using (6). For each combined model, for the first out-of-sample observation a weight of one is given to the model with the highest in-sample-likelihood, as the weight for this observation cannot be computed. One might expect that the best model would simply be assigned a weight of one. However, this was only the case for CLSV, which is therefore excluded from the combination scheme. The weight distributions for the other models are displayed in Figure 4.

CAJD first mixes SV and SVCJ, but the weight of SV decreases and is zero from 2002. The weight of the most elaborate model, SVCJ, increases to one in this period and is mixed with SVJ from 2003 until 2005 and from 2007 until mid 2008. Starting from September 11, 2008, the little weight that SVJ received declines steeply to zero. At the same time, the weight of SVCJ increases to one, which remains the same throughout the remaining out-of-sample period. The change in weights takes place during the worst period global financial crisis. The financial crisis started in 2007, and culminated with the Bankruptcy of Lehman Brothers on September 15, 2008. In line with Eraker et al. (2003), the distribution of the weights shows that adding jumps in volatility is useful for forecasting the density of the S&P 500 returns during this period of market distress.



(a) CAJD

(b) CTFM

(c) CBEST3

(d) CAJDTFM

(e) CALL

Notes: An overview of the abbreviations for the models and their meanings can be found in Appendix 6.

Figure 4: Weights for the different model combinations

CTFM only mixes SVM and SVMCJ. This is interesting, because the in-sample log-likelihood value for SVMJ is higher than for SVM. Similar as for CAJD, the weight distribution changes abrupt from September 2008. Thus, in line with previous findings, the addition of jumps in returns and volatility is useful for modelling the distribution of the S&P 500 returns.

CBEST3, the model that combines the three models with the highest in-sample-likelihoods, namely SVJ, SVCJ and SVMCJ, mixes all three models, giving more weight to the models with jumps in return and volatility. The weight distribution again changes around September 2008, where the weight of the model without a jump in volatility, SVJ, drops to zero. Starting at October 17, 2007, the weight for SVCJ decreases and the weight for SVMCJ increases until December 2008. The beginning of this period roughly corresponds to the beginning of the financial crisis in the United States. The change in the weight distribution could be due to the fact that the long-term mean during this period was not constant, and certainly not equal to the long term mean during

the in-sample period. The increase in the weight of SVMCJ could then be explained by the fact that this model includes the long-term mean as a latent volatility factor. CAJDTFM shows almost the same pattern as CBEST3.

For CALL, the previously observed pattern is also visible: the increase and decrease of SVMCJ and SVCJ around October 17, 2007, respectively. What is also interesting is that one of the more simple models, LSV, receives a positive weight over the whole out-of-sample period. During most of the out-of-sample period, SVMCJ gets assigned the highest weight.

4.3 Results for tests based on the probability integral transform

Test statistcis for the HL, Berkowitz, and KS test and the log-likelihood values for out-of-sample 1 can be found in Table 3. Because the results are similar, the results for out-of-sample 2 can be found in Table 7 in Appendix G. In terms of the HL test, all models are rejected[2]. $W(p)$ statistics for $p = 5, 10$ and 20 are reported. In the discussion of the HL test results, I focus on $W(5)$ because the other test statistics show similar results.

Table 3: Hong and Li (*HL*), Berkowitz (LR_{BW}), and Kolmogorov–Smirnov (*KS*) test statistics and log-likelihood

Out-of-sample 1 (2001-2007)							
Model	**W(5)**	**W(10)**	**W(20)**	LR_{BW}	**KS**	**Likelihood**	
LSV0	14.18	18.18	24.42	13.24 (0.004)	0.038 (0.013)	-2348.4	0.0
LSV	10.27	13.41	18.11	9.35 (0.025)	0.031(0.066)	-2332.3	16.0
SV	10.38	12.96	16.51	8.06 (0.045)	0.025 (0.208)	-2337.0	11.4
SVJ	8.89	11.76	15.41	8.24 (0.041)	0.027 (0.166)	-2326.9	21.5
SVCJ	9.63	12.52	16.36	10.27 (0.016)	0.031 (0.066)	-2323.2	25.1
SVM	12.94	17.10	23.09	8.42 (0.038)	0.032 (0.051)	-2335.1	13.3
SVMJ	13.51	18.54	25.39	25.73 (0.000)	0.037 (0.017)	-2330.1	18.3
SVMCJ	12.70	16.99	22.92	21.23 (0.000)	0.032 (0.057)	-2324.8	23.5
CAJD	9.42	12.21	15.83	9.51 (0.023)	0.030 (0.076)	-2324.0	24.4
CTFM	12.53	16.88	22.73	17.28 (0.001)	0.032 (0.052)	-2324.9	23.5
CBEST3	10.81	14.34	19.01	15.07 (0.002)	0.031 (0.073)	-2324.9	23.5
CAJDTFM	11.18	14.93	19.72	15.31 (0.002)	0.031 (0.069)	-2324.8	23.5
CALL	10.61	14.15	18.66	13.08 (0.004)	0.031 (0.071)	-2325.6	22.7

Notes: The Likelihood column reports likelihood-values for each density forecast model and log-likelihood values in excess of the benchmark model LSV0. For the HL test, W(p) statistics with lag truncuation order p=5, 10 and 20 are reported. The numbers in parentheses are p-values for the corresponding test statistics. An overview of the abbreviations for the models and their meanings can be found in Appendix 6.

[2]I used the empirical critical values from Yun (2020). For out-of-sample 1, the critical values for 10%, 5% and 1% are 2.55, 3.40 and 5.45, respectively. For out-of-sample 2, the critical values for 10%, 5% and 1% are 2.57, 3.58 and 5.65, respectively. As the performance for SVM is underestimated for te HL test (see Appendix E), this might also be the case for the two-factor models with jumps.

For out-of-sample 1, the $W(5)$ values are the lowest for SVJ, SVCJ, CAJD, LSV and SV, and those models are thus closer to the true density forecast model (Hong & Li, 2005). It can be seen that the addition of jumps in the returns in the AJD model SV leads to a decrease in the HL statistics. The addition jumps in volatility does not decrease the statistic further, and $W(5)$ is even slightly higher for SVCJ than for the SVJ. For the two-factor models, SVMJ performs worse than SVM. The $W(5)$ value of SVMCJ is lower than for SV. CAJD outperforms SV and SVCJ with respect to the HL $W(5)$ statistic, but does not yield to a performance improvement with respect to SVJ. For the two-factor models, the combined model CTFM slightly outperforms all models it combines. Unfortunately, CBEST3, CAJDTFM and CALL fail to outperform the best model in the category they combine. For out-of-sample 2, all HL test statistics increase substantially with respect to out-of-sample 1, indicating that the density forecast performance deteriorates for all models. This could be due to a structural change in parameters, or because the global financial crisis is included in out-of-sample 2. $W(5)$ values are lowest for SVJ [17.02], CAJD [17.29], SVCJ [17.34], LSV [18.30] and CALL [18.63], a slightly different composition when you compare it to out-of-sample 1. None of the combined models are able to outperform the models they combine in terms of the HL test.

For out-of-sample 1, all models are rejected by the Berkowitz test at 5% significance. The values of LR_{BW} are lowest for SV, SVJ, SVM, CAJD, SVCJ and LSV. Note that the values of LR_{BW} for SVMJ and SVMCJ are substantially higher than for the other models. All models except LSV0 pass the KS test at 5% significance. In out-of-sample 2, all models are rejected by the Berkowitz test at 5% significance. However, it is notable that the Berkowitz test statistics are now lowest for CAJD [12.91], SVCJ [12.99], CALL [15.37], CAJDTFM [17.04], and CBEST3 [17.07], so many of the combined models perform better in terms of the Berkowitz test than the models they combine. LSV and SV are the only models not rejected by the KS test.

Table 8 in Appendix G shows the results of the 1% and 5% VaR evaluations on the density forecast models for out-of-sample 1, which are obtained by using the approximated PITs $\{z_t\}$. None of the models is rejected in out-of-sample 1. The VaR results for out-of-sample 2 can be found in Table 4. All combined models, the two-factor models with jumps and SVCJ show satisfactory 1% VaR prediction results in terms of the Christofferson test, but only SVMCJ is not rejected in terms of the DQ test. For the 5% VaR, SVMJ is the only model that is not rejected by both the Christofferson and the DQ test. It can thus be concluded that models with jumps in volatility, and combined models outperform models without jumps in volatility with respect to their VaR test statistics.

Table 4: VaR-evaluations

Out of sample 2 (2001-2014)								
	1% VaR				**5% VaR**			
Model	**LRuc**	**Lrind**	**LRcc**	**DQ**	**LRuc**	**Lrind**	**LRcc**	**DQ**
LSV0	0.000	0.699*	0.000	0.000	0.000	0.516*	0.000	0.000
LSV	0.000	0.748*	0.000	0.000	0.000	0.085*	0.000	0.000
SV	0.000	0.837*	0.000	0.000	0.000	0.371*	0.000	0.000
SVJ	0.013	0.226*	0.022	0.000	0.000	0.150*	0.000	0.000
SVCJ	0.346*	0.326*	0.396*	0.002	0.001	0.080*	0.001	0.002
SVM	0.000	0.618*	0.000	0.000	0.000	0.464*	0.001	0.000
SVMJ	0.774*	0.376*	0.648*	0.000	0.377*	0.497*	0.538*	0.303*
SVMCJ	0.572*	0.444*	0.636*	0.743*	0.512*	0.032	0.080	0.057*
CAJD	0.207*	0.309*	0.269*	0.003	0.001	0.080*	0.001	0.002
CTFM	0.774*	0.376*	0.648*	0.000	0.208*	0.048	0.064	0.012
CBEST3	0.961*	0.402*	0.703*	0.008	0.102*	0.032	0.027	0.030
CAJDTFM	0.774*	0.376*	0.648*	0.000	0.075*	0.028	0.018	0.024
CALL	0.435*	0.466*	0.565*	0.016	0.014	0.082	0.011	0.024

*Notes: p-values for the tests are reported. * indicates that it the test is **not** significantly rejected at the 5% level. An overview of the abbreviations for the models and their meanings can be found in Appendix 6.*

Table 5: Diebold Mariano test statistics

Out of sample 2 (2001-2014)												
	LSV0	LSV	SV	SVJ	SVCJ	SVM	SVMJ	SVMCJ	CAJD	CTFM	CBEST3	CAJDTFM
LSV	-5.62***											
SV	-3.50***	0.14										
SVJ	-3.24****	-0.17	-0.48									
SVCJ	-6.40***	-3.42***	-3.73***	-3.68***								
SVM	-1.08	2.06**	3.47***	3.19**	4.55***							
SVMJ	-2.97**	0.10	-0.01	0.54	3.91***	-2.34**						
SVMCJ	-5.62***	-2.60**	-2.50**	-2.48**	0.81	-3.84***	-3.99***					
CAJD	-6.39***	-3.36***	-3.70***	-3.60***	1.06	-4.51***	-3.77***	-0.62				
CTFM	-5.94***	-2.63**	-2.56**	-2.47**	1.07	-4.01***	-3.91***	0.66	0.90			
CBEST3	-6.16***	-3.08**	-3.08**	-3.11**	0.36	-4.23***	-4.31***	-1.26	0.10	-1.73*		
CAJDTFM	-6.31***	-3.14**	-3.13**	-3.08**	0.40	-4.32***	-4.24***	-1.01	0.16	-1.88**	0.17	
CALL	-6.70***	-3.46***	-3.02**	-2.76**	0.78	-4.19***	-3.63***	-0.30	0.59	-0.75	0.71	0.78

*Notes: T-statistics for Diebold-Mariano tests are reported. A negative value is reported when a "row" model is superior to a "column" model. *, **, *** indicate statistical significance at the 10%, 5% and 1% level, respectively. An overview of the abbreviations for the models and their meanings can be found in Appendix 6.*

4.4 Test results for tests based on the Likelihood

This section discusses likelihood-based density forecast evaluations for out-of-sample 2. Because the results for out-of-sample 1 are similar (yet less significant), the results for out-of-sample

1 can be found in Table 9 in Appendix 3.5. Strictly speaking, for all models but the discrete-time LSV0 and LSV, density forecast evaluations are conducted for the Euler-discretized version of each continuous-time AJD or TFM model and its combinations. For the statistical comparison of log-likelihood values, the Diebold-Mariano test is conducted. Table 5 shows the test results for each pair of models. SVCJ shows the best performance, significantly outperforming LSV0, LSV, SV, SVJ, SVM and SVMJ. SVMCJ, CAJD, CBEST3, CAJDTFM and CALL are not significantly outperformed by any other model. The addition of jumps in two-factor model improves density forecast performance apropos of log-likelihood value, as SVMJ and SVMCJ both statistically outperform SVM. Furthermore, the combined models fail to (statistically) outperform the models they are combining.

5 Conclusion

In this paper, I have investigated several stochastic volatility models using simulation-based dynamic probability integral transform and likelihood evaluation procedures from Yun (2020). Furthermore, the approximated likelihood values were used to combine the different models by maximizing the weighted log-likelihood. The weight-distributions of the combined models have provided better insight into the relative performance of each model during the out-of-sample period.

Following Yun (2020), the simulation-based method is combined with existing forecast evaluation methods. Using the informal study of Diebold et al. (1998), histograms and autocorrelation functions of the PITs are used to evaluate the models. The HL, Berkowitz and KS test were used to formally evaluate the models. VaR forecast performance was evaluated using the PIT, without directly having to compute VaR measures. Finally, the models were compared by means of log-likelihood using the DM-test.

For the LSV models and the AJD models, the model estimation and evaluation results in this paper are similar to the results in Yun (2020). The two-factor models with jumps have promising in-sample log-likelihoods, which are substantially higher than the log-likelihood value for SVM, hence improving the fit of the two-factor models could possibly lead to an even better performance of these models.

Unfortunately, the combined density forecasts do not improve performance with respect to the tests conducted in this paper. Future research could focus on a different criterion for the weight computation. In this paper, the log-likelihood of the combined model is maximized. Another method proposed by Hall & Mitchell (2007) uses the PITs to minimize Berkowitz' likelihood ra-

tio test statistics. Although they argue that the combination that minimizes the log-likelihoods is better, it might be different in the research field of this paper. Although, as far as I know, it does not yet appear in the literature, minimizing other test statistics discussed in this paper could also be used to calculate the weights. Combining elaborate affine jump diffusion methods with methods based on different information sources, such as realized volatility and option prices, as Zhao (2013) and Durham & Geweke (2013) have done for more simplistic stochastic volatility models, is also an option. These combination methods could be addressed in future studies.

The weight distributions do give better insight into the relative performance of the models: it can be observed that, during the global financial crisis of 2007 until 2009, the weight distributions change in favor of models with jumps in returns ánd volatility. This is line with Eraker et al. (2003), who argue that the jumps in volatility are important when the volatility rapidly increases, what happens in times of market distress.

Furthermore, in CBEST3 and CAJDTFM, SVMCJ gained a higher weight than SVCJ during the global financial crisis, which may indicate that the long-term average changed during this period. Because the PITs and likelihoods of the combined models can be computed, they can be compared to the individual models on a fair basis. In terms of the HL test, the one-factor AJD models show good performance for both out-of-sample periods, as well as some of the combined models. Berkowitz test statistics tend to differ a bit between the two periods. The relative performance of the combined models improves when looking at out-of-sample 2. For out-of-sample 1, only LSV0 and SVMJ are rejected by the KS statistic, and for out-of-sample 2, only LSV and SV are not rejected. The VaR evaluations show that none of the models are rejected in out-of-sample 1. In out-of-sample 2, the one-factor model with jumps in returns and variance, SVCJ, the two-factor models with jumps and the combined models all show relatively good VaR performance.

In terms of the log-likelihood, SVCJ performs best. Both SVMCJ and the combined models show relatively good performance, most of them not being statistically outperformed.

So, in conclusion, different models are preferred depending on the purpose. SVCJ showed consistently relatively good performance in all tests. But also the much simpler LSV showed relatively good performance in the tests based on the PIT. For VaR analysis over a long out-of-sample period, the two-factor models with jumps and the combined models outperformed the one-factor AJD and LSV models, and would be therefore be the preferred choice for this purpose.

References

Amisano, G., & Giacomini, R. (2007). Comparing density forecasts via weighted likelihood ratio tests. *Journal of Business & Economic Statistics*, *25*(2), 177–190.

Bao, Y., Lee, T.-H., & Saltoğlu, B. (2007). Comparing density forecast models. *Journal of Forecasting*, *26*(3), 203–225.

Bates, D. S. (1996). Jumps and stochastic volatility: Exchange rate processes implicit in deutsche mark options. *The Review of Financial Studies*, *9*(1), 69–107.

Bates, D. S. (2000). Post-'87 crash fears in the s&p 500 futures option market. *Journal of econometrics*, *94*(1-2), 181–238.

Berkowitz, J. (2001). Testing density forecasts, with applications to risk management. *Journal of Business & Economic Statistics*, *19*(4), 465–474.

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

Christoffersen, P. F. (1998). Evaluating interval forecasts. *International economic review*, 841–862.

Diebold, F. X., Gunther, T. A., & Tay, A. S. (1998). Evaluating density forecasts with applications to financial risk management. *International Economic Review*, *39*(4), 863–883.

Drimus, G. G. (2012). Options on realized variance by transform methods: a non-affine stochastic volatility model. *Quantitative Finance*, *12*(11), 1679–1694.

Duffie, D., Pan, J., & Singleton, K. (2000). Transform analysis and asset pricing for affine jump-diffusions. *Econometrica*, *68*(6), 1343–1376.

Durham, G., & Geweke, J. (2013). Improving asset price prediction when all models are false. *Journal of Financial Econometrics*, *12*(2), 278–306.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the Econometric Society*, 987–1007.

Engle, R. F., & Manganelli, S. (2004). Caviar: Conditional autoregressive value at risk by regression quantiles. *Journal of Business & Economic Statistics*, *22*(4), 367–381.

Eraker, B., Johannes, M., & Polson, N. (2003). The impact of jumps in volatility and returns. *The Journal of Finance*, *58*(3), 1269–1300.

Gneiting, T., & Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, *102*(477), 359–378.

Hall, S. G., & Mitchell, J. (2007). Combining density forecasts. *International Journal of Forecasting*, *23*(1), 1–13.

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies*, *6*(2), 327–343.

Hong, Y., & Li, H. (2005). Nonparametric specification testing for continuous-time models with applications to term structure of interest rates. *The Review of Financial Studies*, *18*(1), 37–84.

Hong, Y., Li, H., & Zhao, F. (2007). Can the random walk model be beaten in out-of-sample density forecasts? evidence from intraday foreign exchange rates. *Journal of Econometrics*, *141*(2), 736–776.

Johannes, M. S., Polson, N. G., & Stroud, J. R. (2009). Optimal filtering of jump diffusions: Extracting latent states from asset prices. *The Review of Financial Studies*, *22*(7), 2759–2799.

Kaeck, A., & Alexander, C. (2012). Volatility dynamics for the s&p 500: Further evidence from non-affine, multi-factor jump diffusions. *Journal of Banking & Finance*, *36*(11), 3110–3121.

Meyer, R., & Yu, J. (2000). Bugs for a bayesian analysis of stochastic volatility models. *The Econometrics Journal*, *3*(2), 198–215.

Pitt, M. K., & Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, *94*(446), 590–599.

Rosenblatt, M. (1952). Remarks on a multivariate transformation. *The annals of mathematical statistics*, *23*(3), 470–472.

Scott, D. W. (2015). *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons.

Taylor, S. J. (1982). Financial returns modelled by the product of two stochastic processes-a study of the daily sugar prices 1961-75. *Time series analysis: theory and practice*, *1*, 203–226.

Yu, J. (2005). On leverage in a stochastic volatility model. *Journal of Econometrics*, *127*(2), 165–178.

Yun, J. (2020). Density forecast evaluations via a simulation-based dynamic probability integral transformation. *Journal of Financial Econometrics*, *18*(1), 24–58.

Zhao, Y. (2013). Forecasting the stock return distribution using macro-finance variables. *Job Market Paper*.

Appendices

A Abbreviations

Table 6: Abbreviations

Abbrevation	Meaning
LSV	Log Stochastic Volatility Model
LSV0	Log Stochastic Volatility Model without leverage effect
SV	Stochastic Volatility Model
SVJ	Stochastic Volatility Model with jumps in returns
SVCJ	Stochastic Volatility Model with contemporaneous jumps in returns and variance
SVM	Two-Factor Stochastic Volatility Model
SVMJ	Two-Factor Stochastic Volatility Model with jumps in returns
SVMCJ	Two-Factor Stochastic Volatility Model with contemporaneous jumps in returns and variance
AJD	Affine Jump Diffusion (SV, SVJ, SVCJ)
TFM	Two-Factor affine Model (SVM, SVMJ, SVMCJ)
CAJD	Model that combines Affine Jump Diffusion models
CTFM	Model that combines Two-Factor affine Models
CBEST3	Model that combines 3 best in-sample models
CAJDTFM	Model that combines AJD and TFM models
CALL	Model that combines all models

B Model estimation

Because the models contain many unobserved latent variables, such as the volatility and jumps, it is difficult to estimate them analytically by using, for example, Maximum Likelihood Estimation. Therefore, Bayesian Markov Chain Monte Carlo (MCMC) methods are used to estimate the parameters. MCMC generates many random draws from the posterior distribution, rather than deriving the distribution analytically.

The Bayesian MCMC for the LSV models is implemented using Stan. For the AJD and TFM models, the Bayesian MCMC is implemented in JAGS. For the Stan and JAGS programs, the python package "PyStan" and the R package "rjags" are used, respectively. Both Stan and JAGS require prior distributions for all parameters, and likelihood functions for the returns and the volatility factors. This section explains exactly which distributions have been used for this purpose. In what follows, y_t represents the log return of the S&P 500 in percentage.

B.1 Log Stochastic Volatility (LSV) Models

For the estimation of the LSV models, the following priors from Yun (2020) are used, which were also used in other research (e.g. Yu (2005)): $\beta^* \sim Beta(20, 1.5)$, where $\beta^* = \frac{\beta+1}{2}$, $\alpha^* \sim N(0, 25)$, where $\alpha^* = \frac{\alpha}{1+\beta}$, $rho \sim U(-1, 1)$ and $\sigma_h^2 \sim IG(2.5, 0.025)$, where U, $Beta$, N and IG represent the Uniform, Beta, Normal and Inverse Gamma distribution, respectively. As no prior is given for μ by Yun, I use the non-informative prior $\mu \sim N(0, 25)$. The initial value for V0 is set to one for all AJD models. For the two-factor models, the initial value for M0 is also set to one. Denoting $V_t = \ln h_t$, (2) becomes:

$$\begin{aligned} y_t &= \mu + \exp\left(\frac{V_t}{2}\right)\varepsilon_t \\ V_t &= \alpha + (1+\beta)V_{t-1} + \sigma_h \rho \varepsilon_{t-1} + \sigma_h \sqrt{1-\rho^2}\eta_t, \end{aligned} \tag{14}$$

where ρ is set to zero in the LSV0 model. The LSV models are estimated using the following distributions for y_t, and V_t, obtained from Meyer & Yu (2000), which follow from the fact that y_t and V_t are bivariate normally distributed:

$$\begin{aligned} V_{t+1}|V_t, \alpha, \beta, \sigma_v^2 &\sim N(\alpha + (1+\beta)V_t, \sigma_v^2) \\ y_t|V_{t+1}, V_t, \alpha, \beta, \sigma_v^2 &\sim N(\mu + \frac{\rho}{\sigma_v}\exp\left(\frac{V_t}{2}\right)(V_{t+1} - \alpha - (1+\beta)V_t), \exp(V_t)(1-\rho^2)). \end{aligned} \tag{15}$$

I used the following initialization for V_0: $V_0 \sim N(\alpha, \frac{\sigma_h^2}{1-(1+\beta)^2})$, as log-volatilties are stationary.

B.2 Affine Jump Diffusion (AJD) models

For the model estimation and the particle filter algorithm (Section 5.1), (3) must be discretized and the Brownian Motions estimated, for which I consider the following Euler-discretized and Bernoulli-approximated SVCJ model:

$$\begin{aligned} y_t &= \mu + \sqrt{V_{t-1}}\varepsilon_{1,t} + J_t Z_t^S \\ V_t &= V_{t-1} + \kappa(\theta - V_{t-1}) + \rho\sigma_V\sqrt{V_{t-1}}\varepsilon_{1,t} + \sigma_v\sqrt{(1-\rho^2)}\sqrt{V_{t-1}}\varepsilon_{2,t} + J_t Z_t^V, \end{aligned} \tag{16}$$

where $(\varepsilon_{1,t}, \varepsilon_{2,t})' \sim i.i.d.N(0, 1)$, $J_t \sim i.i.d.Ber(\lambda)$, $Z_t^S \sim i.i.d.N(\mu_S, \sigma_S)$ and $Z_t^V \sim exp(\mu_V)$. This is the same approximation scheme as Yun (2020), who obtained it from Eraker et al. (2003).

Priors are obtained from Yun (2020), who obtained them from Eraker et al. (2003). The following priors are used for the estimation of the AJD models $\mu \sim N(1, 25)$, $k\theta \sim N(0, 1)$, $k \sim N(0, 1)$, $\sigma_v^2 \sim IG(2.5, 0.1)$, $\rho \sim U(-1, 1)$, $\lambda \sim Beta(2, 40)$, $\mu_s \sim N(0, 100)$, $\sigma_s^2 \sim IG(5.0, 20)$ and $\mu_v \sim G(20, 10)$, where G represents the Gamma distribution.

B.2.1 SV model

For the SV model, $J_t Z_t^S$ and $J_t Z_t^V$ in (16) are set to 0. The model is implemented in JAGS using the following distributions:

$$\begin{aligned} y_t|V_{t-1},\mu,\rho,\kappa,\theta &\sim N(\mu, V_{t-1}) \\ V_t|y_t.V_{t-1},\mu,\rho,\kappa,\theta &\sim N(V_{t-1}+\kappa(\theta-V_{t-1})+\rho\sigma_V(y_t-\mu),\sigma_v^2(1-\rho^2)V_{t-1}) \end{aligned} \tag{17}$$

B.2.2 SVJ model

For the SV model, $J_t Z_t^V$ in (16) is set to 0. The model is implemented in JAGS using the following distributions:

$$\begin{aligned} y_t|V_{t-1}\mu,J_t,Z_t^S &\sim N(\mu+J_t Z_t^S, V_{t-1}) \\ V_t|y_t,V_{t-1},\mu,\rho,\kappa,\theta,J_t,Z_t^S,Z_t^V &\sim N(V_{t-1}+\kappa(\theta-V_{t-1})+\rho\sigma_v(y_t-\mu-J_t Z_t^S),\sigma_v^2(1-\rho^2)V_{t-1}) \end{aligned} \tag{18}$$

B.2.3 SVCJ model

The SVCJ model is represented by (16). The model is implemented in JAGS using the following distributions:

$$\begin{aligned} y_t|V_{t-1},\mu,\rho,\kappa,\theta,J_t,Z_t^S &\sim N(\mu+J_t Z_t^S, V_{t-1}) \\ V_t|y_t,V_{t-1},\mu,\rho,\kappa,\theta,J_t,Z_t^S,Z_t^V &\sim N(V_{t-1}+\kappa(\theta-V_{t-1})+\rho\sigma_v(y_t-\mu-J_t Z_t^S)+J_t Z_t^V,\sigma_v^2(1-\rho^2)V_{t-1}) \end{aligned} \tag{19}$$

B.3 Two-Factor Models (TFM)

For the model estimation and the simulation based PIT and likelihood, (4) must be discretized and the Brownian Motions estimated, for which I consider the following Euler-discretized and Bernoulli-approximated two-factor model:

$$\begin{aligned} y_t &= \mu+\sqrt{V_{t-1}}\varepsilon_{1,t} \\ M_t &= M_{t-1}+\kappa_M(\theta_M-M_{t-1})+\sigma_M\sqrt{M_{t-1}}\eta_t \\ V_t &= V_{t-1}+\kappa(M_t-V_{t-1})+\rho\sigma_v\sqrt{V_{t-1}}\varepsilon_{1,t}+\sigma_v\sqrt{1-\rho^2}\sqrt{V_{t-1}}\varepsilon_{2,t}, \end{aligned} \tag{20}$$

where $(\varepsilon_{1,t},\varepsilon_{2,t},\eta_t) \sim i.i.d.N(0,I)$. This is the same approximation scheme as in Yun (2020).

The two-factor models are estimated using the discretized model as specified in (20). The priors for $\mu,\kappa,\sigma_v^2,\rho,\lambda,\mu_s,\sigma_s^2 and \mu_v$ are the same as for the AJD models. For the additional parameters, I use the uninformative priors of Yun (2020): $\kappa_M \sim N(0,1), \theta_M \sim N(1,1)$, and $\sigma_M^2 \sim \frac{1}{16} * IG(2.5, 0.1)$.

$$M_t|M_{t-1},\kappa_m,\theta_M,\sigma_M \sim N(M_{t-1}+\kappa_M(\theta_M-M_{t-1}),\sigma_M^2 M_{t-1}) \tag{21}$$

I consider three types of two factor models. The TFM without jumps (SVM) as in Yun (2020), the TFM with jumps in the return (SVMJ) and the TFM with contemporaneous jumps in return and variance (SVMCJ). The distributions for y_t and V_t are the same as specified in (17), (18) and (19), respectively, where θ is substituted by M_t.

C Auxiliary Particles TFM

For the formulas of the auxiliary particles for the LSV models and the one-factor AJD models, I refer to Yun (2020). The auxiliary variables for the most elaborete two-factor AJD model, SVMCJ, are given by:

$$\begin{aligned}\hat{M}_{t-1}^{(i)} &= \hat{M}_{t-2}^{(i)} + \kappa_M\left(\theta_M - \hat{M}_{t-2}^{(i)}\right)\\ \hat{V}_{t-1}^{(i)} &= \hat{V}_{t-2}^{(i)} + \kappa\left(\hat{M}_{t-1}^{(i)} - \hat{M}_{t-2}^{(i)}\right) + \rho\sigma_V\left(y_{t-1} - \mu - J_{t-1}^{(i)}Z_{t-1}^{(i)}\right) + J_{t-1}^{(i)}\mu_V\end{aligned} \tag{22}$$

D Hong and Li (HL) test

This section introduces the different HL test statistics. The kernel estimator of the joint density for any positive integer j is given by

$$\hat{g}_j(u_1,u_2) = (T-j)^{-1}\sum_{t=j+1}^{T} K_h(u_1,z_t)K_h(u_2,z_{t-j}), \tag{23}$$

where z_t is evaluated at any $\sqrt{T}$-consistent estimator for the true model parameter, and the boundary-modified kernel $K_h(u,z_t)$ is defined as follows:

$$K_h(x,y) = \begin{cases} h^{-1}k\left(\frac{x-y}{h}\right)/\int_{(x/h)}^{1} k(u)du & \text{if } x\in[0,h)\\ h^{-1}k\left(\frac{x-y}{h}\right) & \text{if } x\in[h,1-h)\\ h^{-1}k\left(\frac{x-y}{h}\right)/\int_{-1}^{(1-x)/h} k(u)du & \text{if } x\in[1-h,1]),\end{cases} \tag{24}$$

where $k(\cdot)$ is a pre-specified symmetric probability density and $h = h(T)$ is a bandwith such that, as $T\to\infty$, h goes to zero and Th goes to infinity. The HL test statistic $\hat{Q}(j)$ is given by

$$\hat{Q}(j) \equiv \left[(T-j)h\int_0^1\int_0^1 [\hat{g}_j(u_1,u_2)-1]^2 du_1 du_2 - hA_h^0\right]/V_0^{\frac{1}{2}}, \tag{25}$$

where

$$A_h^0 = \left[(h^{-1}-2)\int_{-1}^{1} k^2(u)du + 2\int_0^1\int_{-1}^{1} k_b^2(u)dudb\right]^2 - 1 \tag{26}$$

$$V_0 \equiv 2\left[\int_{-1}^{1}\left[\int_{-1}^{1}, k(u+v)k(v)dv\right]^2\right]^2, \tag{27}$$

and $k_b(\cdot) \equiv k(\cdot)/\int_{-1}^{b} k(v)dv$. Under the null hypothesis of correct model specification, $\hat{Q}_j$ converges to $N(0,1)$ for any $j > 0$.

The portmenteau evaluation test statistic from Hong et al. (2007) that is used in this paper is related to $\hat{Q}_j$ as follows:

$$\hat{W}(p) = \frac{1}{\sqrt{p}} \sum_{j=1}^{p} \hat{Q}_j, \tag{28}$$

where p is the lag truncation order. This test statistic also converges to $N(0,1)$.

E Size performance of the HL, Berkowitz and KS test statistics using particle filters

Yun (2020) analyzes the size performance of the HL, Berkowitz and KS test using his particle filter via a simulation study. For the discrete-time LSV models, 1000 sample paths are generated. For the continuous-time AJD models and the two-factor SVM model, each sample path is generated using five sub-intervals per sample interval. One out of five observations is kept, leaving observations at daily frequency. Test statistics for the aforementioned tests are computed for three sample sizes: $T = 250, T = 500$ and $T = 1000$. This simulation method can give rise to measurement errors from different sources: the particle filter algorithm, parameter estimation and discretization in the case of the continuous-time stochastic volatility models. Yun (2020) analyzes errors from the particle filter algorithm and discretization.

He finds that, except for SVM, it appears that the approach shows reasonable size performance. For the AJD models, a little overrejection is found for the W(5) and the KS test statistics when $T = 1000$. For SVM, the HL tests overrejects significantly when $T = 1000$, and so does the KS test (to a lesser extent). The Berkowitz test shows reasonably good size performance for the SVM model. By also conducting a simulation with the discretized version of the SVM as data generating process, Yun (2020) finds that the overrejection of the HL test is due to accumulation of discretization bias. Because of this overrejection, care should be taken for the evaluation of the SVM model, as it is possible that the performance of the model is underestimated. To account for this, Yun (2020) computes size-adjusted critical values.

For computational reasons, this paper does not replicate this size-performance test. In the evaluation, we take into account that the size performance of the test was reasonably good for the LSV and AJD models. Because there was overrejection for the SVM model, it is assumed that this is also the case for the two-factor models with jumps, thus I take into account that the HL and KS test for

these models can underestimate the performance.

F Dynamic Quantile (DQ) test

The DQ test of Engle & Manganelli (2004) tests for unconditional coverage and independence of VaR violations. For their test, the "adjusted hit" is defined as $I_t \equiv I_t - p$. The following ordinary least squares regression is considered:

$$I_t = X\beta + \varepsilon_t, \tag{29}$$

where the first column of X consists of ones, and the other columns of additional explanatory variables. Following Engle & Manganelli (2004), I include four lags ($l = 4$) of I_t. The test statistic is then defined as follows:

$$DQ = \frac{\beta'_{OLS}(X'X)\beta_{OLS}}{p(1-p)}. \tag{30}$$

Under de null hypothesis, $DQ \sim \chi^2(l+1)$.

G PIT test results

Table 7: Hong and Li (HL), Berkowitz (LR_{BW}), and Kolmogorov–Smirnov (KS) test statistics and log-likelihood

Out-of-sample 2 (2001-2014)							
Model	**W(5)**	**W(10)**	**W(20)**	LR_{BW}	**KS**	**Likelihood**	
LSV0	23.04	29.21	37.56	27.58 (0.000)	0.027 (0.011)	-4949.3	0.0
LSV	18.30	22.62	30.18	18.32 (0.000)	0.023 (0.053)	-4906.1	43.2
SV	19.30	23.05	30.21	22.14 (0.000)	0.023 (0.053)	-4907.6	41.7
SVJ	17.02	20.82	27.29	18.11 (0.000)	0.023 (0.048)	-4903.7	45.5
SVCJ	17.34	20.36	26.39	12.99 (0.005)	0.025 (0.023)	-4873.0	76.3
SVM	23.18	29.66	38.85	18.80 (0.000)	0.032 (0.001)	-4935.1	14.2
SVMJ	21.28	26.23	34.27	23.50 (0.000)	0.035 (0.000)	-4907.4	41.8
SVMCJ	22.43	27.05	35.24	24.95 (0.000)	0.036 (0.000)	-4877.1	72.1
CAJD	17.29	20.30	26.26	12.91 (0.005)	0.025 (0.025)	-4873.8	75.5
CTFM	22.44	27.47	35.75	21.26 (0.000)	0.035 (0.000)	-4878.3	70.9
CBEST3	19.11	22.79	29.48	17.07 (0.001)	0.031 (0.002)	-4874.2	75.1
CAJDTFM	19.34	23.28	30.06	17.04 (0.001)	0.032 (0.002)	-4874.4	74.9
CALL	18.63	22.56	29.34	15.37 (0.002)	0.029 (0.005)	-4876.0	73.3

Notes: The Likelihood column reports likelihood-values for each density forecast model and log-likelihood values in excess of the benchmark model LSV0, across the two out-of-sample periods. For the HL test, W(p) statistics with lag truncuation order p=5, 10 and 20 are reported. The numbers in parentheses are p-values for the corresponding test statistics. An overview of the abbreviations for the models can be found in Appendix 6.

Table 8: VaR-evaluations

Out-of-sample 1 (2001-2007)								
	1% VaR				5% VaR			
Model	**LRuc**	**Lrind**	**LRcc**	**DQ**	**LRuc**	**Lrind**	**LRcc**	**DQ**
LSV0	0.2205*	0.3051*	0.2790*	0.1265*	0.0744*	0.1314*	0.0652*	0.2324*
LSV	0.5803*	0.2235*	0.4091*	0.2125*	0.4005*	0.9356*	0.7000*	0.2063*
SV	0.6901*	0.1245*	0.2838*	0.0562*	0.0591*	0.4908*	0.1329*	0.1268*
SVJ	0.5167*	0.6241*	0.7187*	0.2824*	0.4005*	0.5716*	0.5985*	0.7902*
SVCJ	0.2441*	0.6603*	0.4608*	0.1675*	0.4629*	0.6049*	0.6681*	0.4213*
SVM	0.9316*	0.5538*	0.8361*	0.5019*	0.2925*	0.7850*	0.5536*	0.4119*
SVMJ	0.0877*	0.7101*	0.2173*	0.0594*	0.1726*	0.8910*	0.3909*	0.4538*
SVMCJ	0.0877*	0.7101*	0.2173*	0.7495*	0.2561*	0.3905*	0.3629*	0.3125*
CAJD	0.5167*	0.6241*	0.7187*	0.2824*	0.4629*	0.6049*	0.6681*	0.4213*
CTFM	0.2441*	0.6603*	0.4608*	0.1675*	0.6438*	0.6000*	0.7832*	0.4396*
CBEST3	0.1520*	0.6851*	0.3302*	0.1054*	0.6438*	0.6000*	0.7832*	0.4396*
CAJDTFM	0.2441*	0.6603*	0.4608*	0.1675*	0.7251*	0.5673*	0.7981*	0.4339*
CALL	0.5167*	0.1066*	0.2204*	0.0322*	0.8955*	0.9000*	0.9836*	0.4568*

*Notes: p-values for the tests are reported. * indicates that it the test is **not** significantly rejected at the 5% level. An overview of the abbreviations for the models and their meanings can be found in Appendix 6.*

H Likelihood test results

Table 9: Diebold Mariano test statistics for out-of-sample 1

Out of sample 1 (2001-2007)												
	LSV0	LSV	SV	SVJ	SVCJ	SVM	SVMJ	SVMCJ	CAJD	CTFM	CBEST3	CAJDTFM
LSV	-3.44***											
SV	-1.60	0.86										
SVJ	-2.19**	-0.63	-1.68*									
SVCJ	-2.71**	-1.16	-2.36**	-1.17								
SVM	-2.02**	0.45	-0.45	1.16	1.60							
SVMJ	-1.76**	-0.24	-0.87	0.76	1.57	-0.63						
SVMCJ	-2.43**	-0.91	-1.74	-0.45	0.55	-1.35	-1.93					
CAJD	-2.67**	-1.08	-2.31**	-0.95	0.95	-1.53	-1.33	-0.27				
CTFM	-2.66**	-0.97	-1.94**	-0.46	0.57	-1.59	-1.69*	0.04	0.31			
CBEST3	-2.50**	-0.93	-1.90*	-0.55	0.89	-1.39	-1.62	0.04	0.45	0.00		
CAJDTFM	-2.61**	-0.97	-2.03**	-0.55	0.71	-1.52	-1.54	0.00	0.39	-0.04	-0.05	
CALL	-2.78**	-1.00	-2.04**	-0.30	0.89	-1.48	-1.06	0.27	0.67	0.35	0.34	0.54

*Notes: T-statistics for Diebold-Mariano tests are reported. A negative value is reported when a "row" model is superior to a "column" model. *, **, *** mean statistical significance at the 10%, 5% and 1%, respectively. An overview of the abbreviations for the models can be found in Appendix 6.*

I Information about the code

This section contains information about the code, that has been submitted via a separate zip-file. The information can also be found there, in the README file.

```
-------------------------------------------------------------------------------
This section contains information about the different code files I used for my research.
For each folder, a description of each file is discussed. The abbreviations for the
model names can be found in Appendix A.


-------------------------------------------------------------------------------
Folder: Data
The file "SPdata.csv" contains the price indices of the S&P 500. With "compute_y.R",
frirst the businessdays are extracted. Subsequently, the log returns in percentage
are calculated.


-------------------------------------------------------------------------------
Folder: JAGS code for model estimation
For the estimation of the AJD and TFM models, I used the R package 'rjags', which is
```

used to estimate the parameters using MCMC. To estimate the models, rjags requires
a textfile where the parameters, priors and likelihoods of the models are stated.
This folder contains such a text file for each model.

Folder: STAN code for model estimation
For the estimation of the LSV models, I used the R package 'rstan', which is
used to estimate the parameters using MCMC. To estimate the models, rstan requires
a stan file where the parameters, priors and likelihoods of the models are stated.

Folder: Particle filter algorithm
This folder contains code for the particle filter of Yun (2020), for each model.
These codes are available on https://doi.org/10.1093/jjfinec/nby030.
SVMJ and SVMCJ were not covered in his model. However, the code for the particle
filter algorithm followed from the code for SVM and the one-factor models with jumps,
SVJ and SVCJ.

With "PIT and Likelihood - VERA - insample.R" the in-sample likelihoods for each model
can be calculated, with the estimated parameters in this research.
With "PIT and Likelihood - VERA - insample.R" the out-of sample PITs and likelihood,
that are used for the model evaluations, can be calculated.

Folder: Model combination
This folder contains the codes to compute the weights for the combined models, and to
construct the PITs and likelihoods for the combined models.

With "density_combination_weights_loop.m", the weights for the different combined models
are computed using a subset of the likelihoods (which can be found in "DEF_LHS_ALL_
outofsample.csv), of the models that are combined For example, for CAJD, the input matrix
likelihoods_outofsample consists of the columns "SV", "SVJ", and "SVCJ" from the file).

The resulting weights can be found in the "W_MODELNAME_DEFINITIEF.csv" files. These
are used as input for W in "compute_combined_llh". With this file, the combined likelihoods
and combined PITs can be computed. The likelihoods and pits of the individual models can be

found in "DEF_GRS_ALL_outofsample.csv" and "DEF_LHs_ALL_outofsample.csv".

Folder: Model evaluation

This folder contains the code for the formal tests that are conducted. The code for these tests has been published by Yun (2020), and were only slightly adjusted. Furthermore, some comments have been added to explain the code.

In "VaR functions.R", the functions for the Christoffersen and DQ test are given. The DQ test has been adjusted in comparison to Yun (2020).

In "VaR Tests.R", the test statistics are calculated.

In "Tests_with_GRs.R", the test statistics for the Hong and Li, Berkowitz and Kolmogorov{Smirnov are computed, using the package "rugarch", in which these test statistics are build in.

In "Likelihood Analysis using DM test.R", the DM test statistics for each pair of models is computed.

For the informal test, the code "histograms and acfs.R" plots the histograms and autocorrelation functions that are examined in the paper, using the package "forecast".

Furthermore, the PITs and likelihoods that have been used for the tests can be found in this folder. (DEF_GRS_ALL_outofsample.csv contains all the PIT's for out-of-sample 2, and DEF_GRS_ALL_outofsample_2007.csv for out-of-sample 1. DEF_LHS_ALL_outofsample.csv and DEF_LHS_ALL_outofsample_2007.csv contain the likelihoods for out-of-sample 2 and out-of-sample 1, respectively.