

A COMPARISON OF BAYESIAN APPROACHES TO FORECASTING VALUE-AT-RISK

ELIS BERGLING

Bachelor's thesis
2026:K27



LUND INSTITUTE OF TECHNOLOGY
Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematical Statistics

Bachelor's Theses in Mathematical Sciences 2026:K27
ISSN 1654-6229
LUTFMS-4021-2026
Mathematical Statistics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden
<http://www.maths.lu.se/>

Abstract

Reliable Value-at-Risk forecasts are essential for quantifying the market risk a financial institution is exposed to. Utilizing a Bayesian approach treats the model's parameters as random variables, allowing for forecasts that consider parameter uncertainty as well as a measure of uncertainty through the spread of the forecasts, the epistemic uncertainty. The method of approximating the parameter uncertainty does, however, vary widely between low-dimensional parametric models and high-dimensional neural networks, and this thesis aims to compare them applied to a GARCH model using Markov chain Monte Carlo (MCMC) and an LSTM using Monte Carlo Dropout (MC Dropout). They are compared to their non-Bayesian counterparts using Christoffersen's conditional coverage framework and quantile loss over the 1% and 5% VaR levels, and evaluated on overall predictive performance using the discretized continuous ranked probability score (CRPS), on a one-day horizon across three assets. A characterization of the epistemic uncertainty and its relation to predictive performance is also performed.

The Bayesian model in the GARCH family exhibited better predictive accuracy in quantile loss and CRPS, whereas the Bayesian framework showed the opposite, but more weakly, when applied to the LSTM. As for tests within the conditional coverage framework, no results favoured either approach. The models performed worse on the loss-based scores when their epistemic uncertainty was high, even after accounting for the level of realized volatility, though the effect was weakened. The overall conclusion is that the propagation of uncertainty is only useful for the GARCH, and both signals of epistemic uncertainty are informative about the reliability of the models.

Keywords: Value-at-Risk, Bayesian inference, GARCH, LSTM, epistemic uncertainty

Acknowledgments

I want to thank my supervisor, Petar Jovanovski, for introducing me to the Bayesian paradigm and supplying great guidance. I also want to thank my family and girlfriend who have been very supportive, thank you.

Contents

Abbreviations	4
1 Introduction	5
2 Method	7
2.1 Data processing	7
2.2 Bayesian inference	9
2.2.1 Introduction to Bayesian inference	9
2.2.2 MCMC	9
2.2.3 Variational Inference and MC Dropout	12
2.3 Model Specifications	13
2.3.1 GARCH	13
2.3.2 LSTM	17
2.4 Evaluation methods	20
2.4.1 CRPS and quantile loss	20
2.4.2 Diebold-Mariano	22
2.4.3 VaR backtesting	23
2.4.4 Uncertainty analysis	25
3 Results	27
3.1 Predictive accuracy and Diebold-Mariano	27
3.2 VaR backtesting	28
3.3 Uncertainty analysis	30
4 Discussion	35
4.1 RQ1	35
4.2 RQ2	37
4.3 RQ3	38
5 Conclusion	40
Appendix	42

Abbreviations

Abbreviation	Meaning
ADF	Augmented Dickey–Fuller test
ARCH	Autoregressive Conditional Heteroskedasticity
B-GARCH	Bayesian GARCH
BPTT	Backpropagation Through Time
CAViaR	Conditional Autoregressive Value at Risk
CC	Conditional Coverage test
CRPS	Continuous Ranked Probability Score
DM	Diebold–Mariano test
EG	Engle–Granger cointegration test
ELBO	Evidence Lower Bound
ESS	Effective Sample Size
GARCH	Generalized Autoregressive Conditional Heteroskedasticity
HAC	Heteroskedasticity and Autocorrelation Consistent
HMC	Hamiltonian Monte Carlo
IND	Independence test
KL	Kullback–Leibler divergence
KPSS	Kwiatkowski–Phillips–Schmidt–Shin test
LR	Likelihood Ratio
LSTM	Long Short-Term Memory network
MC Dropout	Monte Carlo Dropout
MCD-LSTM	LSTM with Monte Carlo Dropout applied at inference
MCMC	Markov chain Monte Carlo
MH	Metropolis–Hastings
MLE	Maximum Likelihood Estimation
NN	Neural Network
NUTS	No-U-Turn Sampler
OHLCV	Open, High, Low, Close, Volume
OLS	Ordinary Least Squares
QL	Quantile Loss
RNN	Recurrent Neural Network
RV	Realized Volatility
UC	Unconditional Coverage test
VaR	Value-at-Risk
VI	Variational Inference

1 Introduction

In risk management, it is essential to know the market risk one is exposed to. Conventionally, this is quantified through Value-at-Risk (VaR), the loss level that a portfolio's return is expected to fall below with probability τ over a given horizon. The levels of interest are typically $\tau = 0.05$ and $\tau = 0.01$, capturing downside risk, and these are the primary subjects of this thesis, forecasted on a one-day horizon.

A standard approach for estimating VaR is the variance-covariance method. It relies on the assumption of normally distributed returns, allowing VaR to be obtained by scaling the estimated volatility (standard deviation) by the corresponding quantile of the standard normal distribution (Linsmeier & Pearson, 1999). In its simple form, the standard deviation is estimated by computing it over a rolling window, a method that adapts slowly to new volatility regimes, making it ill-suited for financial data characterized by volatility clustering. The GARCH model introduced by Bollerslev (1986) accounts for this by estimating the conditional volatility, effectively representing the current regime of an asset. In terms of modeling the conditional volatility, the GARCH(1,1) specification is hard to beat (Hansen & Lunde, 2005), and represents a simple parametric approach to VaR forecasting; hence its place in this thesis. It does, however, lack consistency over the tails due to the imposed distributional assumption of normality, where real data is generally more heavy-tailed (McNeil & Frey, 2000). There are several methods that forecast VaR without imposing any distributional assumptions. A standard such approach is historical simulation, which computes the empirical quantile over a rolling window and uses that as its forecast. It does resolve the issue of distributional assumptions, but shares the rolling window limitation of adapting slowly to changing volatility dynamics (Manganelli & Engle, 2001). Instead of reading the quantile off a window, it can be modeled as a function of the information set. The quantile regression framework of Koenker and Bassett (1978) allows this, commonly applied through the CAViaR model of Engle and Manganelli (2004), where a specific quantile is targeted directly via the quantile loss function as the output of the model. This model does, however, require the choice of a specification, the choice of which impacts the forecasts of the model, reintroducing assumptions into the result. Applying the quantile loss function directly to a neural network avoids this and maps the information set directly to the quantiles. This method has been established for the use case of financial returns since the paper of Taylor (2000). For this study, the LSTM (Hochreiter & Schmidhuber, 1997) will be used due to its sequential data processing and ability to capture long-term effects, and represents the deep learning approach to VaR forecasting of this paper.

As for the evaluation of the models, common VaR backtesting methods include the unconditional coverage test of Kupiec (1995), and the extension of the independence test, resulting in the conditional coverage framework of Christoffersen (1998). These are consistent with what the VaR measurement is meant to represent, but lack consideration of the sharpness of the probabilistic forecast, i.e., the precision of the quantile. A model can, for instance, trivially produce

statistically consistent forecasts of the 5% VaR by ignoring all information and produce an immensely high forecast on a random day in a twenty-day interval, and very low forecasts for the remaining days, which of course is worthless in reality. To account for this, the quantile loss function itself is used at the 1% and 5% VaR levels, providing a score whose expected value is uniquely minimized by consistently producing forecasts at the true quantile level. Both evaluations are, however, to a large extent dominated by a handful of observations near the tail. Evaluating the full predictive distribution instead lets every observation contribute, and is done here using a discretization of the continuous ranked probability score (Gneiting & Raftery, 2007), which is implemented using a total of nine forecasted quantiles each, with the 1% and 5% VaR levels being two of them.

This framework evaluates the reliability of VaR prediction meant to indicate the market risk an institution is exposed to. However, relying on that model itself constitutes an additional risk, a model risk. The above framework aims to reduce it, but does so by assessing reliability on realized outcomes over a test period. It does not allow for insight into the reliability of a forecasted value as the model is used in everyday operations. The Bayesian framework provides an answer by expressing a degree of belief in an event rather than a fixed value. This is accomplished by not treating the parameters or weights of the model as fixed, but rather as distributions themselves with their own uncertainty. This uncertainty is referred to as the epistemic uncertainty of the forecast, and relates to the uncertainty that can be reduced by additional data. The alternative form of uncertainty, aleatoric uncertainty, instead relates to the inherent noise of the data, which a model cannot reduce with sufficient data (Kendall & Gal, 2017). Through a measure of epistemic uncertainty, an actor can be informed of the underlying uncertainty of the parameters, but this is in itself not informative if it does not translate into an assessment of the prediction’s reliability. This raises the question of whether predictive performance changes under different levels of epistemic uncertainty, and whether it is independent of that caused by the level of volatility.

The additional application of the Bayesian paradigm is the propagation of parameter uncertainty into the forecast itself, which has been shown to be beneficial for the calibration of 1% VaR forecasts in GARCH-type models (Ardia et al., 2017). This thesis aims to answer how the propagation of uncertainty affects both the GARCH and the LSTM models, as they are dissimilar in architecture as well as in the procedure by which uncertainty itself is produced. This is accomplished through a comparison of four models: the GARCH and LSTM trained using standard procedures, as well as their Bayesian counterparts, sampled using Markov chain Monte Carlo (MCMC) and Monte Carlo Dropout (MC Dropout), respectively. These are evaluated against three assets of varying volatility dynamics to generalize the results and to avoid making them a direct function of the asset.

Purpose:

The purpose of this thesis is to evaluate the predictive accuracy and usefulness of Bayesian approaches to VaR forecasting, applied to the GARCH and LSTM architectures, across three assets with differing volatility characteristics, namely S&P 500, TSLA, and Brent crude oil.

Research Questions:

RQ1: How does the overall predictive performance compare between standard and Bayesian models, as well as between the two Bayesian models, evaluated through the discretized Continuous Ranked Probability Score and quantile loss functions at the 1% and 5% VaR levels, and are any such differences statistically significant?

RQ2: What models produce reliable downside risk forecasts for 1% and 5% VaR levels, evaluated through the unconditional coverage, independence, and conditional coverage backtests?

RQ3: How does the epistemic uncertainty produced by the two Bayesian models behave, and to what extent is it a distinct signal rather than a rescaling of the realized volatility level, and does this signal carry information about forecast reliability?

2 Method

Below follow the procedures used to build and evaluate the four models on their respective one-day ahead VaR forecasts over the complete set of quantiles $Q = \{0.01, 0.05, 0.10, 0.25, 0.50, 0.75, 0.90, 0.95, 0.99\}$. As well as specification of the derivation of epistemic uncertainty for the Bayesian models, and the characterisation of such.

2.1 Data processing

To retrieve the required asset-specific data, the Python Yahoo Finance package, `yfinance`, is used. The three assets used are S&P 500, Tesla Inc, and Brent Crude Oil Last Day Financial Futures, corresponding to the Yahoo Finance tickers `^GSPC`, `TSLA`, and `BZ=F`, respectively. The assets are downloaded for the earliest registered date, except for S&P 500, which is given the start date of the 3rd of January 2005. Downloading the asset data provides complete OHLCV (Open, High, Low, Close, Volume) data.

Asset	Ticker	Start date
S&P 500	^GSPC	2005-01-03
Tesla, Inc.	TSLA	2010-06-29
Brent Crude Oil Last Day Financial Futures	BZ=F	2007-07-30

Table 1: Asset data and start date

The data is inspected before further use to ensure it looks reasonable. Doing so reveals that five early days of the Brent series reported a lower high than low, which cannot occur by definition. For these days, the high-low range is set to zero. In addition, the Brent series has thirteen consecutive days missing, also early in the series. Both issues are part of the data used for training, and are therefore not treated further. No other issues with the data were found. As for the features, the high-low range is computed by $\ln(\frac{\text{High}_i}{\text{Low}_i})$, where High_i and Low_i are the high and low prices from the OHLCV data at time i and serves as volatility proxy, showing resemblance to the Parkinson volatility estimate (Parkinson, 1980). The complete feature set is as follows

Feature
Log return
Rolling volatility
Rolling skewness
Rolling kurtosis
High-low range
Log volume

Table 2: Feature set

All features are endogenous and computed from the OHLCV data. The log return is computed as $\ln\left(\frac{P_t}{P_{t-1}}\right)$, for the closing price P_t at time t , and serves as both target and feature. The rolling volatility is computed on a 20-day window, which roughly corresponds to a trading month. The rolling skewness and kurtosis are computed on a 60-day rolling window, resulting in the first 60 days of relevant feature columns being NaN, further resulting in the training period being set to 60 days after the start dates.

A training, validation, and test split is set in order to have well-defined fitting and evaluation periods. The training split consists of all dates prior to the 1st of January 2018. The validation period is set between the 1st of January 2018 and the 31st of December 2019, and the test split is between the 1st of January 2020 and the 24th of June 2026. The GARCH and Bayesian GARCH (B-GARCH) are fitted on both the training and validation periods, and the LSTM network is trained on the training split and uses the validation period for early stopping and hyperparameter selection. This allows for a fair comparison where each model has been informed on the data from the training and validation periods, and none has encountered the test period before evaluation. It should be explicitly

stated that neither GARCH model is refitted on a rolling window during testing, which is a common procedure, to preserve a fair evaluation symmetry.

2.2 Bayesian inference

As the comparison of the application of the Bayesian paradigm is at the forefront of this study, it is also of interest to present the underlying techniques used to approximate the parameter uncertainty.

2.2.1 Introduction to Bayesian inference

Bayesian inference is a method of statistical inference where a prior belief is updated through observed data. Bayes' theorem provides a formulation for the probability density of parameters θ conditioned on the data D , referred to as the posterior.

$$\underbrace{p(\theta | D)}_{\text{posterior}} = \frac{\overbrace{p(D | \theta)}^{\text{likelihood}} \overbrace{p(\theta)}^{\text{prior}}}{\underbrace{\int_{\Theta} p(D | \theta) p(\theta) d\theta}_{\text{marginal likelihood}}} \quad (1)$$

The prior belief of the distribution of a θ is incorporated in the prior $p(\theta)$. The prior is itself determined by a Bayesian practitioner and is, for convenience, almost exclusively an analytical expression. The likelihood provides a measure of how likely the parameter θ is at generating the data given the model. This is also an evaluation of analytical expressions. The marginal likelihood, however, requires evaluating the likelihood and prior for all possible sets of parameters, which is computationally intractable for most sets of likelihoods and priors (Martin et al., 2021). The marginal likelihood is however, the same for all θ and serves only as a normalization constant for the posterior to integrate to 1. This results in the posterior often being written as a proportionality.

$$p(\theta | D) \propto p(D | \theta) p(\theta) \quad (2)$$

The intractability of the normalization leads to the use of numerical methods to approximate the posterior. The two most common methods are Markov chain Monte Carlo (MCMC) and Variational Inference (VI), and will be presented. MCMC is the method of choice for approximating the posterior for Bayesian GARCH, and VI is fundamental to the approximate Bayesian inference method, MC Dropout, which the LSTM utilizes.

2.2.2 MCMC

Markov chain Monte Carlo (MCMC) denotes a family of algorithms for sampling from a probability distribution whose density can be evaluated pointwise up to a normalizing constant, precisely the situation of the posterior in equation (2). The idea is to construct a sequence of samples in which each depends only

on its predecessor, i.e a Markov chain, whose stationary distribution is the target distribution. Several such algorithms exist, the foundational one is the Metropolis algorithm (Metropolis et al., 1953), generalized by Hastings (1970) to the Metropolis-Hastings (MH) algorithm.

MH builds the chain from two components. First is a proposal distribution $q(\theta' | \theta)$, which suggests a candidate move from the current state. Second is an accept/reject step that corrects the proposals so that the chain approximates the correct distribution. Given current state θ , a candidate θ' is drawn from $q(\cdot | \theta)$ and accepted with probability

$$a(\theta, \theta') = \min \left\{ 1, \frac{\tilde{p}(\theta' | D) q(\theta | \theta')}{\tilde{p}(\theta | D) q(\theta' | \theta)} \right\}, \quad (3)$$

where $\tilde{p}$ denotes the unnormalized posterior. If rejected, the chain remains at θ . Importantly, the intractable normalizing constant appears in both the numerator and the denominator and cancels. The algorithm can be described using the following pseudo code.

Algorithm 1 Metropolis-Hastings

```

1: Input: unnormalized target  $\tilde{p}$ , proposal  $q$ , initial state  $\theta^{(0)}$ , number of
   iterations  $N$ 
2: for  $i = 1, \dots, N$  do
3:   Draw candidate  $\theta' \sim q(\cdot | \theta^{(i-1)})$ 
4:   Compute
       
$$a = \min \left\{ 1, \frac{\tilde{p}(\theta') q(\theta^{(i-1)} | \theta')}{\tilde{p}(\theta^{(i-1)}) q(\theta' | \theta^{(i-1)})} \right\}$$

5:   Draw  $u \sim \text{Uniform}(0, 1)$ 
6:   if  $u < a$  then
7:     Set  $\theta^{(i)} \leftarrow \theta'$ 
8:   else
9:     Set  $\theta^{(i)} \leftarrow \theta^{(i-1)}$ 
10:  end if
11: end for
12: return  $\{\theta^{(i)}\}_{i=1}^N$ 

```

Due to the inherent randomness of drawing from the proposal distribution, typically a Gaussian centered at the current state, standard random-walk proposals do not use information about the shape of the posterior. Therefore, if the proposal density is narrow, the proposed steps are generally small. This often leads to a high acceptance rate, but the chain explores the posterior slowly. Conversely, if the proposal density is too wide, the proposed steps are larger but frequently land in low-density regions and are rejected, causing the chain to remain in the same state. In both cases, consecutive samples tend to be highly autocorrelated. Adaptive MCMC methods account for this by tuning the proposal density between samples based on the cumulated information thus

far (Andrieu & Thoms, 2008; Haario et al., 2001). An alternative approach is to change the proposal mechanism and use first-order gradient information from the log-posterior to guide proposals toward high-probability regions. Hamiltonian Monte Carlo (HMC) methods use this approach: instead of making uninformed random-walk steps, they simulate a trajectory through the parameter space, allowing the chain to propose distant states that still have a high probability of acceptance. As a result, HMC can explore the posterior more efficiently and reduce autocorrelation between consecutive samples.

HMC introduces an auxiliary momentum variable ϕ of the same dimension as θ , drawn from a Gaussian, $\phi \sim \mathcal{N}(0, M)$, for each iteration. The mass matrix M denotes the covariance matrix of the multivariate normal distribution, and is often initialized as the identity matrix I . The pair (θ, ϕ) is then evolved under Hamiltonian dynamics, interpreting the negative log-posterior $U(\theta) = -\log \tilde{p}(\theta \mid D)$ as a potential-energy surface, the current state can be regarded as a particle with kinetic energy $K(\phi) = \frac{1}{2} \phi^\top M^{-1} \phi$ rolling along the surface. Simulating this motion for some trajectory length yields a proposal (θ', ϕ') that can lie far from the starting point while remaining in a region of high posterior density (Neal, 2011; Betancourt, 2017).

The dynamics conserve the total energy $H(\theta, \phi) = U(\theta) + K(\phi)$, and a proposal generated by exact dynamics would always be accepted. In practice, the trajectory is simulated numerically using the leapfrog integrator. The motion is discretized into L steps of size ϵ , each step updating the momentum by a half-step using the gradient $\nabla_\theta U$, moving the position a full step, and completing the momentum half-step. Discretization introduces a small error, which the Metropolis accept/reject step at the end of the trajectory corrects, so the chain still targets the posterior.

Algorithm 2 Hamiltonian Monte Carlo (one iteration)

- 1: **Input:** unnormalized target $\tilde{p}$, gradient $\nabla_\theta \log \tilde{p}$, step size ϵ , number of steps L , current state $\theta^{(i-1)}$
 - 2: Draw momentum $\phi \sim \mathcal{N}(0, M)$
 - 3: $(\theta', \phi') \leftarrow \text{LEAPFROG}(\theta^{(i-1)}, \phi; \epsilon, L, \nabla_\theta \log \tilde{p})$
 - 4: Compute

$$a = \min \left\{ 1, \exp \left(H(\theta^{(i-1)}, \phi) - H(\theta', \phi') \right) \right\}$$
 - 5: Draw $u \sim \text{Uniform}(0, 1)$
 - 6: **if** $u < a$ **then**
 - 7: Set $\theta^{(i)} \leftarrow \theta'$
 - 8: **else**
 - 9: Set $\theta^{(i)} \leftarrow \theta^{(i-1)}$
 - 10: **end if**
-

The drawback of HMC is the tuning of the step size ϵ and the trajectory length L . A too-short trajectory recovers random-walk behaviour, while a too-long trajectory wastes computation and can loop back toward its starting point.

The No-U-Turn Sampler (NUTS) addresses this (Hoffman & Gelman, 2014), and is used to sample the Bayesian GARCH posterior. It adaptively sets L by stopping when the trajectory starts to retrace previous states. To decide the step size ϵ , the `target accept` parameter is set. `Target accept` tells what the desired expected acceptance rate of the Metropolis step in the HMC algorithm is. Larger values yield smaller steps to minimize the error of the Leapfrog integrator, and vice versa. In practice, it is set high enough to avoid divergences, where the integration error increases.

2.2.3 Variational Inference and MC Dropout

Contrary to MCMC, which addresses the intractable posterior by sampling from it, variational inference (VI) addresses it by optimization (Jordan et al., 1999; Blei et al., 2017). For a neural network with weights $\mathbf{w}$, the posterior $p(\mathbf{w} \mid D)$ is intractable as before. In addition, it is of far too high a dimension for MCMC to be practical. VI instead regards a family of tractable distributions $q_\lambda(\mathbf{w})$, indexed by the variational parameter λ , and seeks the member closest to the posterior determined by the Kullback–Leibler (KL) divergence,

$$\lambda^* = \arg \min_{\lambda} \text{KL}(q_\lambda(\mathbf{w}) \parallel p(\mathbf{w} \mid D)), \quad (4)$$

where the KL divergence is defined as

$$\text{KL}(q(\mathbf{w}) \parallel p(\mathbf{w} \mid D)) = \int q(\mathbf{w}) \log \left(\frac{q(\mathbf{w})}{p(\mathbf{w} \mid D)} \right) d\mathbf{w}. \quad (5)$$

However, optimizing against the intractable posterior by itself does not progress approximations. The solution is a rewriting; the KL divergence decomposes as

$$\begin{aligned} \text{KL}(q_\lambda(\mathbf{w}) \parallel p(\mathbf{w} \mid D)) &= \\ \log p(D) - \underbrace{\left(\mathbb{E}_{q_\lambda} [\log p(D \mid \mathbf{w})] - \text{KL}(q_\lambda(\mathbf{w}) \parallel p(\mathbf{w})) \right)}_{\text{ELBO}(\lambda)}, \end{aligned} \quad (6) \quad (7)$$

where $\log p(D)$ does not depend on λ . Minimizing the KL divergence is therefore equivalent to maximizing the evidence lower bound (ELBO), which only contains computable quantities. Inference has thus been converted into an optimization problem.

Due to the use of an approximate distribution, the approximation can never be better than the family q_λ allows. Unlike MCMC, whose error vanishes asymptotically with more samples, the error of VI is a fixed property of the chosen family. This limitation is the theoretical root of the practical weaknesses of MC Dropout discussed below.

Monte Carlo Dropout (MC Dropout) (Gal & Ghahramani, 2016a) is a VI technique over a neural network’s (NN) weights that minimizes the loss function and negative ELBO simultaneously by regularizing with Dropout and L_2 . Let

the variational distribution be defined by multiplying each row of each weight matrix by an independent Bernoulli variable,

$$\mathbf{w}_i = z_i \cdot \mathbf{m}_i, \quad \mathbf{w}_i \sim q_\lambda(\mathbf{w}) \quad (8)$$

$$z_i \sim \text{Bernoulli}(1 - p), \quad (9)$$

where the weights without dropout mask, $\lambda = \mathbf{m}_i$, are the variational parameters and p is the dropout rate. A draw from q is thus a network in which each neuron has zero activation with probability p . During training, the objective of backpropagation is to minimize the cost $\mathcal{L}_{\text{dropout}}$, for the loss function $E(\cdot, \cdot)$, weight decay μ , biases $\mathbf{b}_i$, realized output $\mathbf{y}_i$, predicted output $\hat{\mathbf{y}}_i$ for N data points, and L NN layers.

$$\mathcal{L}_{\text{dropout}} := \underbrace{\frac{1}{N} \sum_{i=1}^N E(\mathbf{y}_i, \hat{\mathbf{y}}_i)}_{\text{Loss Function}} + \underbrace{\mu \sum_{i=1}^L (\|\mathbf{m}_i\|_2^2 + \|\mathbf{b}_i\|_2^2)}_{L_2 \text{ regularization}}. \quad (10)$$

The two terms of Equation (10) correspond to the two terms of the ELBO, resulting in a minimization of $\mathcal{L}_{\text{dropout}}$ being approximately equal to a maximization of the ELBO, which is the purpose of Variational Inference. The first term of the ELBO is the negative expectation of the log-likelihood over the weights under q_λ . It can not be evaluated analytically and is estimated using a single Monte Carlo sample, i.e., the network with a dropout mask applied, reducing the term to ordinary training using dropout. The second term of the ELBO measures the divergence from the prior, and for the zero-mean Gaussian prior implied by the weight decay, Gal and Ghahramani approximate it by a penalty proportional to $\sum_i \|\mathbf{m}_i\|_2^2$, which is the L_2 term.

For a Recurrent Neural Network (RNN), and therefore a Long Short-Term Memory (LSTM) model, the dropout needs to be applied following Gal and Ghahramani (2016b), for the above interpretation to hold. A single dropout mask is drawn per sequence and used for all time steps, instead of the naive approach of drawing a new one per time step. This is the scheme implemented for the LSTM (Section 2.3.2).

A note on the quality of the approximation is worth addressing. MC Dropout can underestimate uncertainty, particularly under distribution shift (Ovadia et al., 2019). This relates to additional research (Le Folgoc et al., 2021), which suggests that the uncertainty provided by MC Dropout is not necessarily a faithful interpretation of posterior uncertainty. Despite these limitations, it remains an inexpensive uncertainty approximation for NNs, which motivates its place in this thesis.

2.3 Model Specifications

2.3.1 GARCH

The Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model, introduced by Bollerslev (1986) as a generalization of the Autoregressive Con-

ditional Heteroskedasticity (ARCH) model (Engle, 1982), serves as this study’s parametric model. The general GARCH(p, q) model models the conditional volatility by considering the past p squared returns and q past conditional variances. Letting $p = q = 1$, results in the GARCH(1, 1) model, which is effective at capturing volatility dynamics and is therefore the model of choice in several academic papers, as well as this (Hansen & Lunde, 2005).

Returns are modelled as $r_t = \mu + \varepsilon_t$, where μ is the mean return and ε_t the innovation. When modelling the returns on a one-day horizon, the magnitude of the mean return is negligible compared to the magnitude of the conditional standard deviation and is therefore disregarded (Andersen et al., 2003). This assumption yields the following formulation of the GARCH(1, 1) model with normal innovations.

$$r_t = \varepsilon_t \quad \text{for } t = 1, \dots, T \quad (11)$$

$$\varepsilon_t | \mathcal{F}_{t-1} \sim \mathcal{N}(0, \sigma_t^2) \quad (12)$$

$$\sigma_t^2 = \omega + \alpha r_{t-1}^2 + \beta \sigma_{t-1}^2, \quad (13)$$

where $\omega > 0, \alpha \geq 0$, and $\beta \geq 0$ to ensure positive variance, and $\mathcal{F}_{t-1}$ is the information set up to the time $t - 1$. For $t = 1$, $\sigma_{t-1}^2 = \sigma_0^2$, is not known from the recursion. To initialize it, σ_0^2 is set to the unconditional variance $\sigma_0^2 = \sigma^2 = \frac{\omega}{1-\alpha-\beta}$. In practice, the recursion is rolled forward from the first training day so that the initial value of σ_0^2 has little effect when reaching the testing days the model is evaluated on. This is done for both GARCH models; however, as B-GARCH has a sampled posterior consisting of 8000 sets of parameter values, each set is used to roll forward the recursion.

The estimation of the parameters ω , α , and β varies between the two models. Traditionally, the parameters are estimated using maximum-likelihood estimation (MLE). This results in a single set of parameters that are the most likely to have generated the data (given the maximum is reached). This standard approach is utilized for the baseline GARCH model, via the `arch` Python package. Fitting is done using the default optimizer, which has a defined search space to ensure positive parameters and stationarity. For the optimizer to converge reliably, it is fitted on returns in percentage units, i.e., scaled by 100 before fitting. This additionally requires scaling back after forecasting the volatility, to have comparable levels between models. This scaling is also applied to B-GARCH. To see the estimated parameters, see Appendix A.1.

For the Bayesian alternative, the parameters are instead treated as random variables by imposing prior distributions and conditioning on the data to obtain the joint parameter posterior (Ardia, 2008). The parameter space is defined by $\Theta := \omega \times \alpha \times \beta$, and the MCMC method NUTS is used to approximate the posterior $p(\theta | D)$ for $\theta \in \Theta$ and the data D . With only three parameters, adaptive MCMC would likely also constitute a feasible choice; however, due to the native implementation of NUTS in PyMC, and that it comes with established convergence diagnostics, which are presented below, NUTS is the algorithm of choice.

NUTS makes use of the first-order gradient of the log-posterior, which creates an issue when imposing stationarity on the prior densities for α and β . The stationarity constraint is fulfilled for $\alpha + \beta < 1$ and can be imposed by setting the log-posterior to $-\infty$, resulting in an area of undefined gradient. When trying to sample from this region, NUTS diverges and results in parameter values near the cut-off getting underrepresented. Due to GARCH high-persistence tendencies (McMillan & Ruiz, 2009), a reparameterization of α and β to impose stationarity is used instead, where a prior is imposed on the total persistence $S = \alpha + \beta$ and the shock to persistence ratio $W = \alpha/S$. Imposing a prior from the family of Beta distributions naturally enforces positivity, $\alpha, \beta \geq 0$, and stationarity, $\alpha + \beta < 1$. Choosing the hyperparameters $\alpha_S, \alpha_W, \beta_S$, and β_W , for the prior densities $p(S) = \text{Beta}(S | \alpha_S, \beta_S)$ and $p(W) = \text{Beta}(W | \alpha_W, \beta_W)$, is done to be weakly informative, meaning the prior is intentionally weaker than the prior knowledge available (Gelman et al., 2013, p. 55). The high persistence of GARCH models is well established; therefore, a prior that favours moderate persistence is chosen. Specifically, the beta distribution hyperparameters are chosen as $\alpha_S = 6$ and $\beta_S = 2$. The ratio of α to persistence is in general lower and more varying; therefore, $\alpha_W = 2$ and $\beta_W = 3$ are chosen.

Choosing a prior for ω does not have any constraints relating to another parameter, and therefore does not require a reparameterization. Positivity is the only constraint present. Therefore, $p(\omega) = \text{HalfNormal}(\omega | 0.1)$ is a suitable choice when having scaled returns.

The prior choices above, reflect the prior knowledge of typical GARCH parameters, but still keep the distributions wide in order to let the data dominate the posterior through the likelihood. This is, however, a hypothesis and should be verified. Therefore, a prior sensitivity analysis is conducted inspired by the power-scaling methodology proposed by Kallioinen et al. (2023). The power-scalings are only applied to the prior, and not the likelihood, and are tested through manually refitting the posterior to different priors, as opposed to the appeal of the methodology, which only requires one fitted model. The original methodology requires either cumbersome implementation or using the language R, neither of which is motivated above this simpler approach. Fitting is done on three models per asset using three separate degrees of scaling, $\gamma \in \{\frac{1}{2}, 1, 2\}$, which scale the posterior according to

$$p(\theta | D)_\gamma \propto p(D | \theta)p(\theta)_\gamma \propto p(D | \theta)p(\theta)^\gamma, \quad (14)$$

where $p(\theta)_\gamma$ denotes the normalization of $p(\theta)^\gamma$ to a proper probability density, and $p(\theta | D)_\gamma$ the posterior it induces. The posterior for which the prior $p(\theta)_1$ is used are the baseline parameters that are being analyzed for sensitivity. For a Beta prior density $p(\theta) = \text{Beta}(\theta | \xi, \eta)$ it holds that

$$p(\theta) \propto \theta^{\xi-1}(1-\theta)^{\eta-1} \quad (15)$$

which, when scaled, yields

$$p(\theta)^\gamma \propto \theta^{\gamma(\xi-1)}(1-\theta)^{\gamma(\eta-1)}, \quad (16)$$

suggesting $p(\theta)_\gamma = \text{Beta}(\theta \mid \gamma(\xi - 1) + 1, \gamma(\eta - 1) + 1)$.

For a HalfNormal prior density $p(\theta) = \text{HalfNormal}(\theta \mid \sigma_0)$ it holds that

$$p(\theta)_\gamma \propto e^{-\gamma \frac{\theta^2}{2\sigma_0^2}} = e^{-\frac{\theta^2}{2(\sigma_0/\sqrt{\gamma})^2}} \quad (17)$$

suggesting $p(\theta)_\gamma = \text{HalfNormal}(\theta \mid \sigma_0/\sqrt{\gamma})$. Applying these power-scalings to our priors yields our set of priors

$$p(S)_{\frac{1}{2}} = \text{Beta}(S \mid 3.5, 1.5) \quad (18)$$

$$p(W)_{\frac{1}{2}} = \text{Beta}(W \mid 1.5, 2) \quad (19)$$

$$p(\omega)_{\frac{1}{2}} = \text{HalfNormal}(\omega \mid 0.1/\sqrt{\frac{1}{2}}) \quad (20)$$

$$p(S)_1 = \text{Beta}(S \mid 6, 2) \quad (21)$$

$$p(W)_1 = \text{Beta}(W \mid 2, 3) \quad (22)$$

$$p(\omega)_1 = \text{HalfNormal}(\omega \mid 0.1) \quad (23)$$

$$p(S)_2 = \text{Beta}(S \mid 11, 3) \quad (24)$$

$$p(W)_2 = \text{Beta}(W \mid 3, 5) \quad (25)$$

$$p(\omega)_2 = \text{HalfNormal}(\omega \mid 0.1/\sqrt{2}). \quad (26)$$

To evaluate whether the model is sensitive to the choice of prior, the posterior densities for each prior are plotted in the same graph, and if there is great overlap, the model is deemed not to be sensitive to the choice of prior, else it is reported that the model is sensitive to the choice of prior.

An additional important part of a Bayesian workflow is diagnostics of the MCMC algorithm to ensure that the samples proposed by the sampler are a fair representation of the posterior. Running four chains with 4000 draws, the initial 2000 of which are discarded as burn-in samples, generates 8000 posterior draws per asset, which are sampled with a **target accept** of 0.95. To verify that the parameter space has been mapped accurately, the rank plots, $\hat{R}$ -value, bulk-ESS (regular ESS), and tail-ESS are reported as outlined by Vehtari et al. (2021). The rank plots give a visual indication of how similar the parts of parameter space samples from each chain are. Ideally, chains should be indistinguishable. The $\hat{R}$ -value gives a numerical value to the mixing of chains instead by assigning a score based on the comparison of the variance between chains to that within chains. A value less than 1.01 is aimed for, following standard recommendations. The effective sample size (ESS) is a measure of how many independent samples the autocorrelated samples correspond to. Bulk-ESS measures this in the centre of the posterior, while tail-ESS measures it in the tails, computed as the minimum of the effective sample sizes at the 5% and 95% quantiles of the marginal posterior of each parameter. The tail-ESS is of special interest, as the epistemic uncertainty is estimated from the 5% and 95% quantiles of the posterior distribution of the VaR forecasts (see Section 2.4.4). Following Vehtari et al. (2021), at least 100 draws per chain are desired for both bulk- and tail-ESS, giving a threshold of a total of 400 for the four chains.

Another essential step in the Bayesian workflow is conducting a prior predictive check to assess whether the chosen priors generate reasonable data based on prior knowledge (Gabry et al., 2019). Drawing $N = 8000$ samples from the prior distribution and rolling the sampled parameter values through the recursion generates 8000 simulated return series of 1500 observations. Combining all simulated return distributions into one and comparing to the last 1500 observations of the data used for fitting for each asset shows whether the simulated returns have a similar distribution to the real returns. In addition, one of the series is plotted to verify it captures dynamics like volatility clustering. As the same prior is used for all assets, the check does not depend on the asset. As for the posterior predictive check, that follows the same procedure but instead uses the realized posterior draws, $\{\theta^{(i)}\}_{i=1}^N$, to simulate the return series; the checks are asset-specific, resulting in three separate checks. The procedure relies on evaluation through eye measure based on prior knowledge and does not follow a rigorous evaluation procedure.

After the diagnostics, checks, and analyses are conducted, the desired VaR quantiles can be computed. For $\theta \sim p(\theta | D)$ the resulting VaR quantiles are a function of θ according to, $\text{VaR}_\tau(\theta) = \sigma_t(\theta) \Phi^{-1}(\tau)$, for all quantiles $\tau \in Q$, and for all t in the test split, and where $\Phi^{-1}(\tau)$ is the inverse normal cumulative distribution function. As θ itself is a stochastic variable, so is $\text{VaR}_\tau(\theta)$. To retrieve a point estimate, the posterior mean over $\text{VaR}_\tau(\theta)$ is derived. Because the parameter posterior is intractable, the NUTS draws, $\theta^{(i)} \in \Theta$ for $i \in \{1, 2, \dots, N\}$, are used.

$$\widehat{\text{VaR}}_\tau = \mathbb{E}[\text{VaR}_\tau(\theta)] = \int_{\Theta} \text{VaR}_\tau(\theta) p(\theta | D) d\theta \approx \quad (27)$$

$$\frac{1}{N} \sum_{i=1}^N \text{VaR}_\tau(\theta^{(i)}) = \frac{1}{N} \sum_{i=1}^N \sigma_t(\theta^{(i)}) \Phi^{-1}(\tau) \quad (28)$$

To retrieve parameter uncertainty, a 90% credible interval is constructed. Letting $\text{VaR}_\tau^{(1)} \leq \dots \leq \text{VaR}_\tau^{(N)}$ denote the sorted quantiles, the credible interval is given by $R_\tau = [\text{VaR}_\tau^{(0.05N)}, \text{VaR}_\tau^{(0.95N)}]$.

This approach allows for the separation of the epistemic uncertainty through the credible interval. An alternate approach of retrieving the VaR quantiles would be to utilize the posterior predictive distribution $p(y^* | D)$ where y^* denotes the predicted return. This would require constructing the mixture density of the individual normal return distribution for each sample, and then finding the quantiles. This approach does not allow for extraction of epistemic uncertainty as it is already included in the point estimate and is not used for that reason.

2.3.2 LSTM

The Long Short-Term Memory network (LSTM), introduced by Hochreiter and Schmidhuber (1997), is a Recurrent Neural Network (RNN) designed to resolve the vanishing gradient problem (Bengio et al., 1994). An RNN is an extension

of the feedforward neural network that allows the processing of sequential data. This is done by maintaining a hidden state that gets updated at each timestep in the sequence and carries information from previous time steps. During training, when the weights are updated via backpropagation through time (BPTT), the gradient gets propagated back through the sequence and gets multiplied by a weight matrix, once for each time step back in the sequence. If the spectral norm is less than 1, then the gradient approaches 0, leading to difficulties in learning long-term dependencies. The LSTM introduces the cell state, often referred to as the long-term memory, to resolve this. The cell state is updated through scalar multiplication and addition and is augmented by the hidden state through the forget gate¹ which decides how much of the previous cell state to remember. Then the input gate decides how much of the input and previous hidden state should be added to the cell state. At last, the cell state is used through the output gate to update the hidden state and the output.

Having a long-term memory suits a model trying to capture volatility dynamics well, where clustering implies that current volatility levels depend on a long range of previous days.

The objective of the training is to predict one-day-ahead VaR at nine quantile levels. To achieve this, quantile loss is used as the loss function (Koenker & Bassett, 1978). For a target y , a prediction $\hat{y}$ and a level $\tau \in (0, 1)$, it is defined as

$$\rho_\tau(y, \hat{y}) = \begin{cases} \tau(y - \hat{y}), & y \geq \hat{y}, \\ (1 - \tau)(\hat{y} - y), & y < \hat{y}, \end{cases} \quad (29)$$

It penalizes the under- and overpredictions asymmetrically to be able to target a specific quantile. For $\tau = 0.01$, for example, an exceedance below the prediction is penalized 99 times harder than the same error above it. Effectively placing an optimal prediction far below the median return.

The network outputs all nine quantiles in Q simultaneously; in order to have a single training loss $\mathcal{L}$ to use in BPTT, they have to be combined. This is done through averaging over individual quantile loss values,

$$\mathcal{L} = \frac{1}{|Q|} \sum_{\tau \in Q} \rho_\tau(y, \hat{y}_\tau). \quad (30)$$

In order to train the LSTM, the dataset has to be windowed, as the LSTM expects input in the form of sequences. The sequence length is determined by random search discussed later. The windowing ensures that the days within the lookback window are used to predict today's target. The data is also standardized on the training data to avoid lookahead bias.

The LSTM model is implemented using the `torch` Python package. Training is done with early stopping, with a patience of 15 and a max epoch of 200, using the Adam optimizer (Kingma & Ba, 2015) with a weight decay of 10^{-4} . The

¹The forget gate was introduced by Gers et al. (2000); the formulation used here is the standard variant.

weight decay ensures the L_2 regularization is imposed, which is required for MC Dropout to be an approximation of VI (Gal & Ghahramani, 2016a).

As no clear set of hyperparameters is fit for all types of data, hyperparameter tuning is performed using random search, motivated by its simplicity over more sophisticated approaches like Bayesian optimization. For each hyperparameter in Table 3, the hyperparameter is chosen at random from the uniform distributions spanning the given intervals. In the case of the learning rate, it is drawn from a log-scaled uniform distribution, meaning the exponent p_r is drawn from the uniform distribution $\mathcal{U}(-4, -2)$, resulting in the learning rate $l_r = 10^{p_r}$, this is done to avoid almost entirely drawing from large learning rates near 10^{-3} to 10^{-2} .

Hyperparameter	Range / Set
Learning rate	10^{-4} to 10^{-2} (log-scale)
Batch size	{64, 128, 256}
Dropout rate	0.05 to 0.40
Lookback	20 to 80 days
Hidden size	16 to 64
Number of layers	{1, 2}

Table 3: Hyperparameter ranges

The random search is performed by drawing 20 samples of hyperparameters and training the LSTM with a fixed seed. Each validation loss is compared, and the one with the lowest is chosen as the main set of hyperparameters. For this model, re-seeding is performed for 4 additional training runs. This results in 5 models that only vary by initial seed; the median-performing model is chosen as the main model (see Appendix B.1 and B.2).

This main model is used for both the standard LSTM and its Bayesian counterpart that utilizes MC Dropout at inference, meaning both models share the same underlying architecture and weights. For the MC Dropout LSTM (MCD-LSTM), MC Dropout is applied during the test period by performing $J = 100$ forward passes with active dropout at each time step, effectively drawing a set of weights $\mathbf{w}_j \sim q_{\lambda^*}(\mathbf{w})$ for each pass. The dropout mask is held constant for each time step in its sequence, in correspondence with the variational scheme described in Section 2.2.3. Contrary to B-GARCH, no distributional assumptions are made, and quantiles are explicitly output from the model. Denoting the predicted VaR quantile as $\text{VaR}_{\tau}(\mathbf{w}) = f_{\tau}(x_t; \mathbf{w})$ for each $\tau \in Q$ and sequence data x_t , where f_{τ} denotes the corresponding output of the final dense layer, a point estimate is obtained by averaging over the forward passes at each quantile

level, retrieving the posterior mean over each quantile, as in B-GARCH.

$$\widehat{\text{VaR}}_\tau = \mathbb{E}[f_\tau(x_t; \mathbf{w})] = \int f_\tau(x_t; \mathbf{w}) q_{\lambda^*}(\mathbf{w}) d\mathbf{w} \approx \quad (31)$$

$$\frac{1}{J} \sum_{j=1}^J f_\tau(x_t; \mathbf{w}_j), \quad \mathbf{w}_j \sim q_{\lambda^*}(\mathbf{w}) \quad (32)$$

An issue that forecasted quantiles of the LSTM and the MCD-LSTM are subject to is that nothing guarantees that the quantiles follow a monotone ordering, as they are calculated independently. This is known as the quantile crossing problem (Koenker & Bassett, 1982), and occurs when the prediction of a lower quantile level exceeds the prediction of a higher one, which of course does not describe a valid distribution. Crossings are repaired by monotone rearrangement following Chernozhukov et al. (2010), who show that this procedure results in a curve closer to the true quantile curve. For the regular LSTM, rearrangement is done on the forward pass for each timestep, and for the MCD-LSTM, the rearrangement is applied to the aggregated forecast. The frequency of crossings before rearrangement is recorded and reported in Appendix B.3.

The epistemic uncertainty is again retrieved through a 90% credible interval around each quantile. However, this time under variational distribution $q_{\lambda^*}(\mathbf{w})$ and not the posterior. Letting $\text{VaR}_\tau^{(1)} \leq \dots \leq \text{VaR}_\tau^{(J)}$ denote the sorted quantiles, the credible interval is given by $R_\tau = [\text{VaR}_\tau^{(0.05J)}, \text{VaR}_\tau^{(0.95J)}]$.

2.4 Evaluation methods

In Section 2.4, the evaluation methods are described. Each subsection relates to a separate research question, apart from Sections 2.4.1 and 2.4.2, which both relate to RQ1.

2.4.1 CRPS and quantile loss

The Continuous Ranked Probability Score (CRPS) is a strictly proper scoring rule evaluating the precision of a predicted continuous probability distribution compared to the realized outcome (Gneiting & Raftery, 2007). A scoring rule $S(F, y)$ assigns a score to the predicted distribution F , given the realized outcome y . If the outcome y follows G , the expected score is given by $S(F, G) = \mathbb{E}_{y \sim G}[S(F, y)]$, for which it is said S is proper in case $S(G, G) \leq S(F, G)$. If equality holds only for $F = G$, then S is said to be strictly proper. This difference means that a proper scoring rule retrieves an optimal score for the true distribution alongside other distributions, whereas a strictly proper scoring rule only retrieves an optimal score by reporting the true distribution.

The CRPS is calculated by integrating over the squared difference between the predicted cumulative distribution function F and the realized outcome y as,

$$\text{CRPS}(F, y) = \int_{-\infty}^{\infty} (F(x) - \mathbb{1}\{x \geq y\})^2 dx, \quad (33)$$

where the realized outcome is represented with the step function

$$\mathbb{1}\{x \geq y\} = \begin{cases} 1, & x \geq y, \\ 0, & x < y. \end{cases} \quad (34)$$

This benefits both sharpness, the concentration of the forecast, and calibration, the statistical consistency of the forecasts, meaning, for instance, that returns are below $\text{VaR}_{0.05}$ 5% of the time.

The CRPS is, however, designed for continuous forecasts. This is the case for GARCH but not for the LSTM, which produces 9 quantiles; therefore, it is desired to find an approximation of the CRPS for discrete values. Laio and Tamea (2007) give an equivalent formulation using the quantile loss function,

$$\text{CRPS}(F, y) = 2 \int_0^1 \rho_\tau(y, F^{-1}(\tau)) d\tau. \quad (35)$$

Applying the trapezoidal rule, the discrete approximation follows as

$$\widehat{\text{CRPS}}(F, y) = 2 \sum_{k=1}^8 \frac{\tau_k - \tau_{k-1}}{2} (\rho_{\tau_k}(y, F^{-1}(\tau_k)) + \rho_{\tau_{k-1}}(y, F^{-1}(\tau_{k-1}))) \quad (36)$$

$$= \sum_{k=1}^8 (\tau_k - \tau_{k-1}) (\rho_{\tau_k}(y, \hat{y}_{\tau_k}) + \rho_{\tau_{k-1}}(y, \hat{y}_{\tau_{k-1}})) \quad (37)$$

and is the final formulation of the discretization of the CRPS. It is worth noting that this approximation does not capture the 1% tail mass at both ends, resulting in a smaller score. However, this affects models similarly, and the effects are considered negligible. Despite the GARCH models producing continuous forecasts, they are also evaluated on the same 9-point discretization of the CRPS, to ensure all models are compared on the same quantity. This serves as somewhat of a compromise for the GARCH models, as they do produce greater resolution of the returns, which is not accounted for.

It should also be noted that $\widehat{\text{CRPS}}$ is no longer strictly proper but instead only proper, as two forecasts can vary between the quantile levels and still yield the same score. However, due to the discretization, it might be more natural to view the score as a collection of point forecasts than a single distributional forecast. As the quantile loss function is a strictly consistent scoring function (Gneiting, 2011), which is the analogue of a strictly proper scoring rule for point forecasts, $\widehat{\text{CRPS}}$ inherits this property as it is the positive weighted sum over quantile loss scores. This does, however, only hold with respect to the nine quantile levels of Q , meaning that the expected score is uniquely minimized by reporting the true quantiles at these levels.

Rewriting the approximation as a weighted sum illustrates the similarity to the LSTM loss function $\mathcal{L}$ in Equation (30).

$$\widehat{\text{CRPS}}(F, y) = \sum_{k=0}^8 w_k \rho_{\tau_k}(y, \hat{y}_{\tau_k}), \quad (38)$$

$$w_k = \begin{cases} \tau_1 - \tau_0, & k = 0, \\ \tau_{k+1} - \tau_{k-1}, & k = 1, \dots, 7, \\ \tau_8 - \tau_7, & k = 8. \end{cases} \quad (39)$$

The loss $\mathcal{L}$ can be interpreted as a the case of $\widehat{\text{CRPS}}$ for which the weights $w_k = \frac{1}{9}$ are used. As the loss function can be constructed for other weights for each quantile loss function, one might argue that it would be convenient to train the LSTM with the same function as it is evaluated on, i.e., setting $\mathcal{L} = \widehat{\text{CRPS}}$. This would create steeper gradients for the centre quantiles and would hence make the calibration of the tail quantiles have a smaller effect on the training objective. As these models also will be evaluated on the 1% and 5% VaR levels through VaR backtesting and individual quantile loss performance, having a loss function that combines entire distributional calibration as well as the tail contributions is of interest, which motivates the current choice for $\mathcal{L}$.

To retrieve a definitive score, the approximate CRPS is averaged over all time steps

$$\widehat{\text{CRPS}} = \frac{1}{T} \sum_{t=1}^T \widehat{\text{CRPS}}(F_t, y_t). \quad (40)$$

This assigns a specific score for each model for each asset.

It is also of interest to examine the individual quantile loss scores of the 1% and 5% VaR levels. This is done as a complementary metric to the VaR backtests to test the models' predictive performance at these levels. They are denoted $\text{QL}_{0.01}$ and $\text{QL}_{0.05}$ respectively and are averaged over all time steps in the test period, the same as for $\widehat{\text{CRPS}}$.

2.4.2 Diebold-Mariano

The three scores per model presented in Section 2.4.1 can give an overview of each model's predictive performance. It cannot, however, distinguish whether any model performs better than another with any statistical significance. To correct for this, pairwise comparisons are conducted following the procedure of Diebold and Mariano (1995). The Diebold–Mariano (DM) test examines the difference between two competing forecast sequences, $\hat{y}_t^{(A)}$ and $\hat{y}_t^{(B)}$, under a given loss function, $g(y, \hat{y}_t)$. For time t , the difference is denoted $d_t = g(y, \hat{y}_t^{(A)}) - g(y, \hat{y}_t^{(B)})$. The null hypothesis of equal forecast accuracy is $H_0 : \mathbb{E}[d_t] = 0$, and under a two-sided test, the alternative hypothesis becomes $H_1 : \mathbb{E}[d_t] \neq 0$. The expected value of d_t is approximated using the series mean.

$$\bar{d} = \frac{1}{T} \sum_{t=1}^T d_t. \quad (41)$$

The loss differentials $\bar{d}$ are assumed covariance stationary, so that the autocovariance at lag difference j , is expressed as $\gamma_j = \text{Cov}(d_t, d_{t-j})$. The variance of

$\bar{d}$ can then be formulated as

$$\text{Var}(\bar{d}) = \text{Var}\left(\frac{1}{T} \sum_{t=1}^T d_t\right) = \quad (42)$$

$$\frac{1}{T^2} \sum_{t=1}^T \sum_{s=1}^T \text{Cov}(d_t, d_s) = \frac{1}{T} \left[\gamma_0 + 2 \sum_{j=1}^{T-1} \left(1 - \frac{j}{T}\right) \gamma_j \right]. \quad (43)$$

For large T , the weights $1 - \frac{j}{T}$ approach 1 for small j , and for larger j the contribution of γ_j is negligible provided the autocovariances are absolutely summable. Therefore, the expression within the parentheses in the last expression approaches the long-run variance $\mathcal{J}$ ²,

$$\mathcal{J} = \sum_{j=-\infty}^{\infty} \gamma_j = \gamma_0 + 2 \sum_{j=1}^{\infty} \gamma_j, \quad (44)$$

so that $\text{Var}(\bar{d}) \approx \mathcal{J}/T$. To approximate $\mathcal{J}$, a Heteroskedasticity and Autocorrelation Consistent (HAC) estimator is used, namely the Newey-West estimator (Newey & West, 1987), which limits the sum to a bandwidth L .

$$\hat{\mathcal{J}} = \hat{\gamma}_0 + 2 \sum_{j=1}^L w_j \hat{\gamma}_j, \quad w_j = 1 - \frac{j}{L+1}, \quad \hat{\gamma}_j = \frac{1}{T} \sum_{t=j+1}^T (d_t - \bar{d})(d_{t-j} - \bar{d}). \quad (45)$$

The bandwidth length L is chosen such that it has a slower growth rate than $T^{1/4}$ following from the derivation of Newey and West (1987). Therefore $L = \lfloor T^{1/4} \rfloor$ is chosen, giving the final test statistic

$$DM = \frac{\bar{d}}{\sqrt{\hat{\mathcal{J}}/T}} \stackrel{asy}{\sim} \mathcal{N}(0, 1), \quad (46)$$

for which the null hypothesis is rejected if $|DM| > 1.96$ in the two-sided test at the 5% level. For positive DM , it can be determined that model B performs better on the given loss function with statistical significance, and vice versa.

As the test is between two models, it could be examined in a total of 6 pairs for 4 models, but not all combinations are of interest. Of interest are testing GARCH against B-GARCH, LSTM against MCD-LSTM, and B-GARCH against MCD-LSTM. These three cases capture whether Bayesian approaches increase predictive performance and whether they significantly differ between the GARCH and LSTM families.

2.4.3 VaR backtesting

Testing the statistical consistency of VaR quantiles and the independence of violations is performed using the unconditional coverage test by Kupiec (1995)

²Diebold and Mariano (1995) express $\mathcal{J}$ as $2\pi f_d(0)$, the spectral density of the loss differential at frequency zero, which also equals the sum over autocovariances.

and the conditional coverage framework of Christoffersen (1998). Both methods rely on the indicator variable,

$$I_t = \begin{cases} 1, & y_t < \hat{y}_{\tau,t} \\ 0, & y_t \geq \hat{y}_{\tau,t} \end{cases}, \quad (47)$$

which indicates whether the realized outcome of the series at time t is less than or greater than the forecasted quantile. The sequence $\{I_t\}$ is said to have correct conditional coverage if $\{I_t\} \stackrel{iid}{\sim} \text{Bernoulli}(\tau)$. Thus, for a sequence with correct conditional coverage, the proportion of violations should follow the nominal level, i.e., $\mathbb{E}[I_t] = \tau$. In addition, it should also follow that violations are independent, i.e., $P(I_t = 1 \mid I_{t-1} = 1) = P(I_t = 1 \mid I_{t-1} = 0)$. Both criteria are inspected using likelihood ratio tests.

Firstly, the unconditional coverage is tested under the null hypothesis that the realized violation ratio $\hat{p} = \frac{1}{T} \sum_{t=1}^T I_t$ is equal to the nominal quantile τ , i.e., $H_0 : p = \tau$ and $H_1 : p \neq \tau$. For the number of violations k , the likelihood ratio score is

$$\text{LR}_{uc} = -2 \ln \frac{L(\tau)}{L(\hat{p})} = -2 \ln \frac{\tau^k (1-\tau)^{T-k}}{\hat{p}^k (1-\hat{p})^{T-k}} \stackrel{asy}{\sim} \chi_1^2. \quad (48)$$

As LR_{uc} approaches χ_1^2 asymptotically under H_0 , the null hypothesis is rejected in case LR_{uc} exceeds 3.841, corresponding to the 5% significance level.

Secondly, to construct the independence test, the sequence $\{I_t\}$ is first regarded as a first-order Markov chain with transition matrix

$$\Pi_1 = \begin{bmatrix} 1 - \pi_{01} & \pi_{01} \\ 1 - \pi_{11} & \pi_{11} \end{bmatrix} \quad (49)$$

where $\pi_{ij} = P(I_t = j \mid I_{t-1} = i)$ which is estimated using $n_{ij}/(n_{i0} + n_{i1})$, where n_{ij} denotes the number of occurrences value i is followed by value j . Under independence, π_{01} should equal π_{11} ; therefore, to represent this case, a second transition matrix is constructed.

$$\Pi_2 = \begin{bmatrix} 1 - \hat{\pi}_2 & \hat{\pi}_2 \\ 1 - \hat{\pi}_2 & \hat{\pi}_2 \end{bmatrix} \quad (50)$$

where $\hat{\pi}_2 = (n_{01} + n_{11})/(n_{00} + n_{01} + n_{10} + n_{11})$ is the violation rate estimated from the transitions under independence. The null hypothesis thus becomes $H_0 : \Pi_1 = \Pi_2$, and the alternative hypothesis $H_1 : \Pi_1 \neq \Pi_2$, and thus the test ratio,

$$\text{LR}_{ind} = -2 \ln \frac{L(\hat{\Pi}_2)}{L(\hat{\Pi}_1)} = -2 \ln \frac{\hat{\pi}_2^{n_{01}+n_{11}} (1-\hat{\pi}_2)^{n_{00}+n_{10}}}{\hat{\pi}_{01}^{n_{01}} (1-\hat{\pi}_{01})^{n_{00}} \hat{\pi}_{11}^{n_{11}} (1-\hat{\pi}_{11})^{n_{10}}} \stackrel{asy}{\sim} \chi_1^2. \quad (51)$$

Once again, the null hypothesis is rejected in case LR_{ind} exceeds 3.841.

To construct the full conditional coverage test, a last ratio is analyzed,

$$\text{LR}_{cc} = \text{LR}_{uc} + \text{LR}_{ind} \stackrel{asy}{\sim} \chi_2^2. \quad (52)$$

In this case, the null hypothesis can only be rejected if LR_{cc} exceeds 5.991, meaning a model can fail neither the unconditional coverage nor the independence test, and still fail the conditional coverage test. To exemplify, this could occur if both the unconditional coverage and independence tests yield respective test statistics of 3.0, neither being large enough to be rejected at the 5%, i.e., exceeding a value of 3.841. Their sum of 6.0 does, however, exceed 5.991, leading to the null hypothesis of the conditional coverage test being rejected at the 5% significance level.

2.4.4 Uncertainty analysis

This section presents the methods used to examine whether the extrapolated epistemic uncertainty from the models subject to Bayesian inference is informative about the reliability of the forecasts.

The estimation of epistemic uncertainty is made similarly to Kendall and Gal (2017). The per-quantile epistemic uncertainty is given by the difference in upper and lower bounds of the 90% credible range around each quantile forecast, defined at the end of Sections 2.3.1 and 2.3.2, as

$$U_{epi,\tau}^{(A)} = R_{\tau,upper}^{(A)} - R_{\tau,lower}^{(A)} \quad (53)$$

where A denotes the model. The total epistemic uncertainty is then given by the mean over quantiles,

$$U_{epi}^{(A)} = \frac{1}{|Q|} \sum_{\tau \in Q} U_{epi,\tau}^{(A)} \quad (54)$$

It is worth making a note on the implied epistemic uncertainty for the B-GARCH followed by this construction,

$$U_{epi,\tau} = \text{VaR}_{\tau}^{(0.95N)} - \text{VaR}_{\tau}^{(0.05N)} = |\Phi^{-1}(\tau)| \Delta \sigma_t \quad (55)$$

$$U_{epi} = \frac{1}{|Q|} \sum_{\tau \in Q} |\Phi^{-1}(\tau)| \Delta \sigma_t = c \cdot \Delta \sigma_t, \quad (56)$$

where

$$\Delta \sigma_t = \sigma_t^{(0.95N)} - \sigma_t^{(0.05N)}, \quad (57)$$

The main properties that follow from this construction are no epistemic uncertainty for the 0.5 quantile, which is also due to the assumption of zero mean. The epistemic level increases monotonically and symmetrically around the 0.5 quantile, and the individual quantiles do not bear information above that of the total level. Furthermore, the total level of epistemic uncertainty is a scaling of the difference of conditional volatility, essentially tracking the volatility mechanically, illustrating the limits of expressing model uncertainty for the GARCH model.

To visualize how the epistemic uncertainty behaves over time, a heatmap where the color is represented by $U_{epi,\tau}^{(A)}$, for each quantile and time step, is

used. It is worth noting that the color scale differs between models, and the level of epistemic uncertainty should not be compared between models. Both models use separate Bayesian inference methods, with MCMC being asymptotic and MC Dropout at best approximate. In addition, MC Dropout does not possess a well-calibrated weight decay and is therefore unreasonable to expect a comparable level of epistemic uncertainty to what is produced using MCMC.

To examine whether epistemic uncertainty is informative about the performance of the predictions, the test period is divided into quartiles based on epistemic level. For each quartile, the approximate CRPS, quantile loss, and violation rates for 1% and 5% VaR are reported. The p -value of the UC test is also reported for the 5% quantile. It is, however, not reported for the 1% quantile, as there are approximately 4 expected violations per quartile, yielding little power to the test. The small number of expected violations should also be considered when regarding the violation rates. These scores result in a table indicating whether predictive performance increases, stays the same, or decreases when the model signals uncertainty.

Additionally, a regression controlling for the 20-day rolling realized volatility (RV) is performed to assess whether the epistemic uncertainty is an independent signal of predictive accuracy, or whether it simply coincides with the RV, just telling how the model performs in different volatility regimes. Additionally, the loss-based scores of the discretized CRPS and quantile loss are trivially scaled by the volatility level. Therefore, controlling for the volatility ensures that results cannot be explained by the scaling. The regression is in the log-log form as,

$$\log(U_{epi,t}) = \beta_0 + \beta_1 \cdot \log(RV_t) + \varepsilon_t, \quad (58)$$

suggesting that the true relationship between the epistemic level and RV follows a ratio $\frac{U_{epi}}{RV^{\beta_1}} = e^{\beta_0}$. The parameter estimations $\hat{\beta}_0$ and $\hat{\beta}_1$ are estimated using OLS, leaving the residual estimations $\hat{\varepsilon}_t$ as the part of the epistemic uncertainty that is not explained by the RV. This regression provides three results. First are the estimated elasticities $\hat{\beta}_1$, telling how many percent change in epistemic level occurs for a one percent change in the realized volatility. Secondly is the coefficient of determination, R^2 , telling how much of the variance of the epistemic uncertainty is explained by RV. And lastly is the quartile table divided based on the residual level, providing the same table of predictive accuracy as previously discussed, but without RV as an explanatory variable.

To evaluate whether the regression is at risk of being spurious, which can arise due to unit root series (Granger & Newbold, 1974), stationarity is tested using the augmented Dickey–Fuller (ADF) test (Dickey & Fuller, 1979; Said & Dickey, 1984), and the KPSS test (Kwiatkowski et al., 1992). The opposite null hypothesis of the tests, unit root for ADF and stationarity for KPSS, gives a strong measure of stationarity if the tests agree. In case the series are not stationary, stationarity is tested on the first differences to see if the series integration orders are $I(1)$. In that case, they are tested for cointegration using the Engle–Granger test (Engle & Granger, 1987). Rejection of the null hypothesis of no cointegration indicates the regression is not spurious. Results are reported in Appendix C.

3 Results

Section 3 compiles the results of the evaluations of Section 2.4. To provide additional context to the results, Table 4 presents the mean realized volatility of each asset across data splits, giving insight into the variations in the results that might occur between assets. The volatility level increases from S&P 500 to Brent to TSLA, and all assets are tested on a more volatile period than what they are trained on, with TSLA differing the most.

Asset	Train	Val	Test	All
S&P 500	0.00978	0.00850	0.01072	0.00995
TSLA	0.02920	0.03240	0.03793	0.03320
Brent	0.01986	0.01801	0.02378	0.02103

Table 4: Mean realized volatility by split and asset.

3.1 Predictive accuracy and Diebold-Mariano

Calculating $\widehat{\text{CRPS}}$, $\text{QL}_{0.01}$, and $\text{QL}_{0.05}$ provides scores according to Table 5. The smallest value for each asset and evaluation metric is highlighted in bold text to emphasize the best score. For the S&P 500/ $\text{QL}_{0.01}$ pair, the best score is tied between GARCH and B-GARCH to the accuracy of the sixth decimal; more decimals would, however, disclose B-GARCH as the model with lesser loss. This is in accordance with the results in Table 6.

Asset	Model	CRPS	$\text{QL}_{0.01}$	$\text{QL}_{0.05}$
S&P 500	GARCH	0.005920	0.000434	0.001372
	B-GARCH	0.005919	0.000434	0.001371
	LSTM	0.005807	0.000442	0.001336
	MCD-LSTM	0.005812	0.000442	0.001335
TSLA	GARCH	0.020882	0.001508	0.004579
	B-GARCH	0.020879	0.001486	0.004569
	LSTM	0.020933	0.001338	0.004573
	MCD-LSTM	0.020936	0.001347	0.004583
Brent	GARCH	0.013066	0.001200	0.003207
	B-GARCH	0.013051	0.001196	0.003198
	LSTM	0.012965	0.001168	0.003208
	MCD-LSTM	0.012972	0.001175	0.003215

Table 5: CRPS and quantile loss (1% and 5% levels) by asset and model.

It can be observed that the relative magnitudes differ on a relatively small scale within asset-metric comparisons. However, whether the differences are

significant or not is determined by the Diebold-Mariano test statistic. Table 6 provides the specific statistic of each test. As the statistic asymptotically follows the standard normal distribution and the test is two-sided, a statistic of magnitude greater than 1.96 is significant at the 5% level; these values are highlighted in bold. A positive value indicates a lower average loss, i.e., better performance, for the second listed model of each pair and vice versa.

Pair	Asset	CRPS	QL _{0.01}	QL _{0.05}
GARCH vs. B-GARCH	S&P 500	2.253	1.032	1.464
	TSLA	0.666	1.980	1.082
	Brent	4.011	1.309	2.414
LSTM vs. MCD-LSTM	S&P 500	-2.529	0.104	0.338
	TSLA	-0.371	-1.244	-1.170
	Brent	-1.577	-1.584	-1.072
B-GARCH vs. MCD-LSTM	S&P 500	4.442	-0.198	1.428
	TSLA	-0.659	2.270	-0.194
	Brent	1.726	0.318	-0.253

Table 6: Diebold-Mariano test statistics for each metric, pair, and asset.

It can be observed that for each pair, all significant outcomes have a consistently better-performing model across all assets. In addition, B-GARCH has a lower average loss across all metrics and assets compared to GARCH, regardless of test significance. The same is true for LSTM against MCD-LSTM with the exception of QL_{0.01} and QL_{0.05} for S&P 500.

It can also be noted that the tests are performed at the 5% significance level, so that among comparisons with no true difference in expected loss, one in twenty tests is expected to be rejected. With 27 tests reported, conclusions should be drawn from the overall pattern rather than from individual statistics.

3.2 VaR backtesting

Performing the unconditional coverage, independence, and conditional coverage tests at the 1% and 5% quantile levels provides Tables 7 and 8. The p -value of each test is presented, and values below 0.05 are highlighted in bold, indicating that the null hypothesis is rejected at the 5% level for that test.

Model	Asset	Violation rate	UC	IND	CC
GARCH	S&P 500	0.0227	0.0000	0.7834	0.0001
	TSLA	0.0184	0.0022	0.2806	0.0052
	Brent	0.0227	0.0000	0.2591	0.0000
B-GARCH	S&P 500	0.0233	0.0000	0.8039	0.0000
	TSLA	0.0178	0.0043	0.2968	0.0099
	Brent	0.0233	0.0000	0.2844	0.0000
LSTM	S&P 500	0.0117	0.5092	0.2140	0.3716
	TSLA	0.0141	0.1150	0.4068	0.2048
	Brent	0.0165	0.0152	0.0779	0.0111
MCD-LSTM	S&P 500	0.0123	0.3710	0.0223	0.0492
	TSLA	0.0154	0.0441	0.3677	0.0878
	Brent	0.0172	0.0084	0.0902	0.0074

Table 7: VaR backtesting p -values at the 1% quantile level.

Model	Asset	Violation rate	UC	IND	CC
GARCH	S&P 500	0.0614	0.0408	0.5354	0.1019
	TSLA	0.0448	0.3313	0.1198	0.1860
	Brent	0.0527	0.6202	0.1090	0.2449
B-GARCH	S&P 500	0.0627	0.0240	0.4794	0.0609
	TSLA	0.0436	0.2271	0.0104	0.0180
	Brent	0.0558	0.2941	0.1857	0.2403
LSTM	S&P 500	0.0455	0.3931	0.6410	0.6229
	TSLA	0.0670	0.0028	0.0001	0.0000
	Brent	0.0564	0.2466	0.0969	0.1289
MCD-LSTM	S&P 500	0.0448	0.3313	0.6123	0.5486
	TSLA	0.0657	0.0054	0.0001	0.0000
	Brent	0.0558	0.2941	0.0860	0.1321

Table 8: VaR backtesting p -values at the 5% quantile level.

From Table 7, it can be observed that each violation rate exceeds the nominal level of 1%. In addition, both GARCH models do so with sufficient margin that they fail the UC test. Both also fail the CC test, but pass the IND test. The LSTM models exhibit closer violation rates to the nominal level across all assets, with some passing the UC test and some also the CC test. The violation rate is, in addition, closer to the nominal level for each asset for LSTM compared to its Bayesian counterpart, MCD-LSTM. The MCD-LSTM model is also the only model failing the IND test for any asset.

Observing the results from the 5% quantile level, more tests are passed

overall. The LSTM models do not pass any tests for TSLA, but do for the other assets. It can be observed this time that GARCH has violation rates closer to the nominal level than B-GARCH; despite this, the results for the UC test are the same.

As in Section 3.1, the tests are performed at the 5% significance level, and Tables 7 and 8 report 72 tests in total, so individual rejections should be interpreted with the corresponding caution.

3.3 Uncertainty analysis

Visualizing the epistemic uncertainty for both models in a heat map as described in Section 2.4.4 in Figure 1, some general behaviour is apparent. Firstly, B-GARCH expresses the same behaviour as discussed in Section 2.4.4, no model uncertainty for the 0.5 quantile, and the uncertainty increases monotonically and symmetrically around the 0.5 quantile.

MCD-LSTM does not have these characteristics. Instead, its uncertainty marginally favours the right tail asymmetrically. This is most apparent early in the test period.

Most notably, both models signal high uncertainty early in the test period, which coincides with the March 2020 COVID-19 pandemic market crash, and a few months into 2025, which coincides with the increased tariffs from the United States. An event that is unique for Brent is the blockade of the Strait of Hormuz beginning late in February 2026, corresponding to the last epistemic increase of the heatmap.

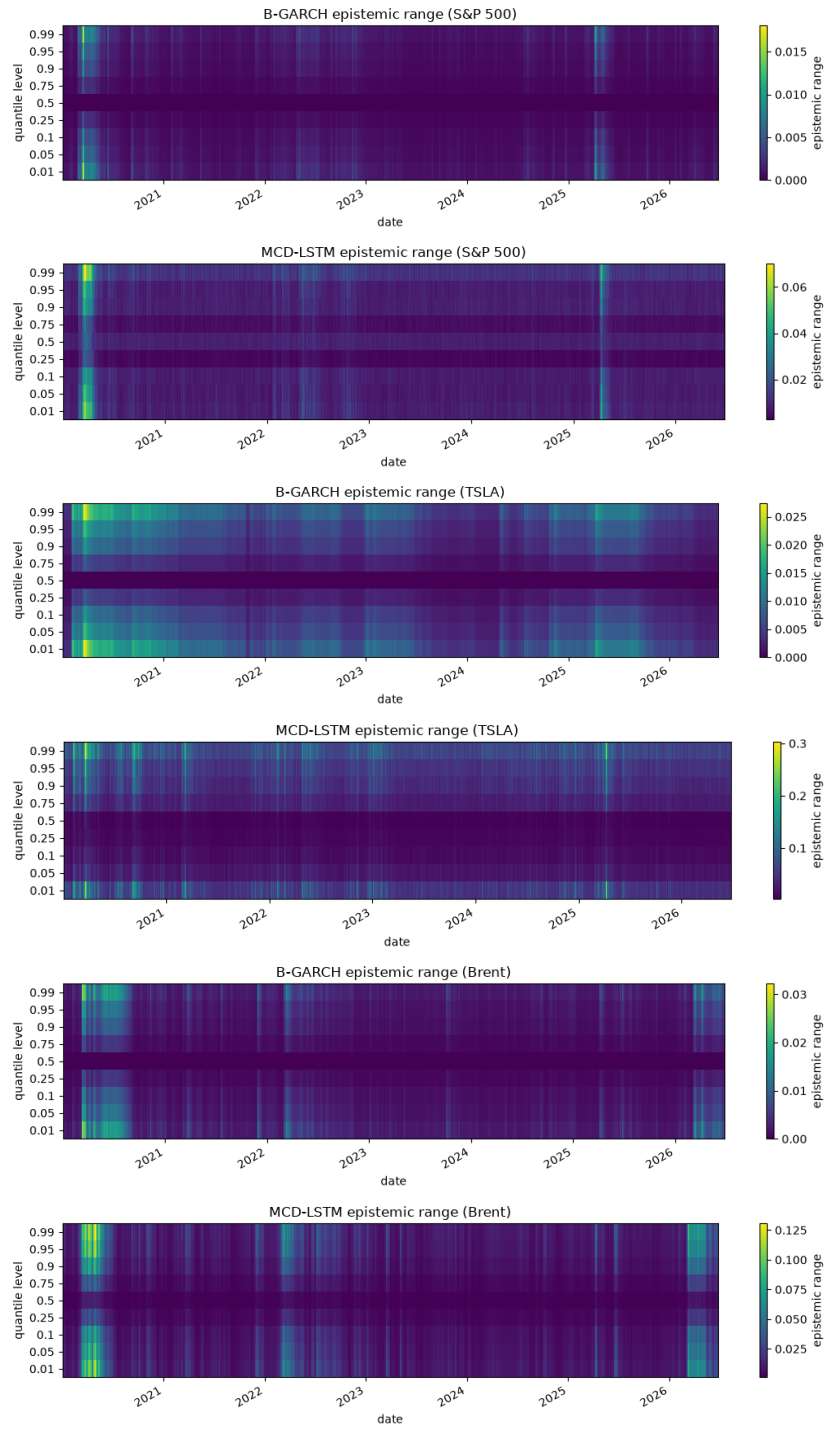


Figure 1: Epistemic level for each quantile over time

Dividing the results of predictive accuracy into four separate quartiles based on the level of epistemic uncertainty yields Tables 9 and 10. Q1 is the calmest quartile of the test period and Q4 the most uncertain. The UC column corresponds to unconditional coverage at the 5% level within each quartile. The VR_τ columns show the violation rate for the τ VaR level.

Asset	Quartile	CRPS	$QL_{0.01}$	$VR_{0.01}$	$QL_{0.05}$	$VR_{0.05}$	UC
S&P 500	Q1	0.00390	0.00033	0.0197	0.00095	0.0614	0.306
	Q2	0.00433	0.00036	0.0344	0.00107	0.0762	0.024
	Q3	0.00575	0.00050	0.0246	0.00135	0.0567	0.547
	Q4	0.00969	0.00055	0.0147	0.00212	0.0565	0.555
	Overall	0.00592	0.00043	0.0233	0.00137	0.0627	0.024
TSLA	Q1	0.01736	0.00125	0.0172	0.00362	0.0393	0.305
	Q2	0.02042	0.00141	0.0270	0.00458	0.0688	0.099
	Q3	0.01972	0.00106	0.0074	0.00380	0.0271	0.021
	Q4	0.02603	0.00223	0.0197	0.00628	0.0393	0.305
	Overall	0.02088	0.00149	0.0178	0.00457	0.0436	0.227
Brent	Q1	0.00991	0.00074	0.0245	0.00221	0.0613	0.312
	Q2	0.01042	0.00097	0.0270	0.00260	0.0662	0.152
	Q3	0.01119	0.00089	0.0172	0.00263	0.0418	0.433
	Q4	0.02071	0.00219	0.0245	0.00536	0.0539	0.720
	Overall	0.01305	0.00120	0.0233	0.00320	0.0558	0.294

Table 9: B-GARCH predictive performance by quartile of epistemic uncertainty.

Asset	Quartile	CRPS	QL _{0.01}	VR _{0.01}	QL _{0.05}	VR _{0.05}	UC
S&P 500	Q1	0.00486	0.00030	0.0049	0.00102	0.0393	0.305
	Q2	0.00458	0.00040	0.0123	0.00113	0.0319	0.074
	Q3	0.00448	0.00032	0.0099	0.00105	0.0493	0.945
	Q4	0.00932	0.00075	0.0221	0.00214	0.0590	0.419
	Overall	0.00581	0.00044	0.0123	0.00134	0.0448	0.331
TSLA	Q1	0.01661	0.00117	0.0147	0.00363	0.0590	0.419
	Q2	0.01768	0.00109	0.0074	0.00353	0.0590	0.419
	Q3	0.02226	0.00125	0.0099	0.00465	0.0714	0.062
	Q4	0.02721	0.00187	0.0295	0.00653	0.0737	0.040
	Overall	0.02094	0.00135	0.0154	0.00458	0.0657	0.005
Brent	Q1	0.00933	0.00064	0.0098	0.00213	0.0417	0.427
	Q2	0.00951	0.00060	0.0074	0.00209	0.0515	0.892
	Q3	0.01106	0.00097	0.0172	0.00278	0.0467	0.756
	Q4	0.02201	0.00249	0.0343	0.00587	0.0833	0.005
	Overall	0.01297	0.00118	0.0172	0.00322	0.0558	0.294

Table 10: MCD-LSTM predictive performance by quartile of epistemic uncertainty.

A consistent result is that both models across all three assets exhibit the largest CRPS, QL_{0.01}, and QL_{0.05} for Q4; some also monotonically increase from Q1 to Q4. The results for the frequency metrics, VR_{0.01}, VR_{0.05}, and UC, are only consistent in the same pattern as described above for MCD-LSTM, as for B-GARCH the values vary without a clear pattern between quartiles. To account for the observed consistencies caused by the RV, the elasticity and R^2 of the log-log regression are inspected in Table 11, and the evaluation metrics divided into quartiles based on residual level instead are inspected in Tables 12 and 13. All still exhibit the largest CRPS, QL_{0.01}, and QL_{0.05} for Q4, with the exception of the B-GARCH CRPS on TSLA, where Q3 is marginally higher. However, fewer are monotonically increasing from Q1 to Q4. MCD-LSTM exemplifies this well on the S&P 500 where the second-highest scores are registered for Q1. The ratio of predictive accuracy between Q4 and Q1 is still larger than one but less so. As for the frequency metrics, a similar pattern for the MCD-LSTM is still observable but less pronounced and consistent, and as for the B-GARCH, the values still vary without a clear pattern between quartiles.

Model	Asset	$\hat{\beta}_1$	R^2
B-GARCH	S&P 500	1.0442	0.7035
	TSLA	0.6658	0.2510
	Brent	0.9034	0.4302
MCD-LSTM	S&P 500	0.3863	0.4564
	TSLA	0.6002	0.5438
	Brent	1.1639	0.5944

Table 11: Elasticity and R^2 of epistemic uncertainty with respect to realized volatility. $T = 1627$ for S&P 500 and TSLA, $T = 1631$ for Brent.

Asset	Quartile	CRPS	QL _{0.01}	VR _{0.01}	QL _{0.05}	VR _{0.05}	UC
S&P 500	Q1	0.00508	0.00043	0.0147	0.00121	0.0811	0.008
	Q2	0.00581	0.00032	0.0172	0.00120	0.0418	0.433
	Q3	0.00537	0.00039	0.0222	0.00123	0.0591	0.412
	Q4	0.00743	0.00060	0.0393	0.00184	0.0688	0.099
	Overall	0.00592	0.00043	0.0233	0.00137	0.0627	0.024
TSLA	Q1	0.01890	0.00124	0.0197	0.00397	0.0565	0.555
	Q2	0.02115	0.00141	0.0221	0.00446	0.0516	0.883
	Q3	0.02198	0.00154	0.0123	0.00460	0.0345	0.129
	Q4	0.02149	0.00176	0.0172	0.00524	0.0319	0.074
	Overall	0.02088	0.00149	0.0178	0.00457	0.0436	0.227
Brent	Q1	0.01108	0.00067	0.0172	0.00240	0.0588	0.426
	Q2	0.01273	0.00112	0.0245	0.00291	0.0564	0.562
	Q3	0.01328	0.00138	0.0270	0.00357	0.0590	0.419
	Q4	0.01513	0.00161	0.0245	0.00392	0.0490	0.927
	Overall	0.01305	0.00120	0.0233	0.00320	0.0558	0.294

Table 12: B-GARCH predictive performance by quartile by residual level.

Asset	Quartile	CRPS	QL _{0.01}	VR _{0.01}	QL _{0.05}	VR _{0.05}	UC
S&P 500	Q1	0.00600	0.00046	0.0123	0.00133	0.0418	0.433
	Q2	0.00503	0.00032	0.0049	0.00109	0.0393	0.305
	Q3	0.00496	0.00028	0.0074	0.00107	0.0394	0.310
	Q4	0.00725	0.00071	0.0246	0.00185	0.0590	0.419
	Overall	0.00581	0.00044	0.0123	0.00134	0.0448	0.331
TSLA	Q1	0.01957	0.00116	0.0147	0.00395	0.0811	0.008
	Q2	0.01923	0.00112	0.0074	0.00364	0.0442	0.586
	Q3	0.02069	0.00140	0.0172	0.00486	0.0739	0.039
	Q4	0.02427	0.00171	0.0221	0.00589	0.0639	0.217
	Overall	0.02094	0.00135	0.0154	0.00458	0.0657	0.005
Brent	Q1	0.01026	0.00065	0.0098	0.00232	0.0515	0.892
	Q2	0.01029	0.00076	0.0098	0.00233	0.0319	0.073
	Q3	0.01384	0.00089	0.0098	0.00302	0.0565	0.555
	Q4	0.01752	0.00240	0.0392	0.00520	0.0833	0.005
	Overall	0.01297	0.00118	0.0172	0.00322	0.0558	0.294

Table 13: MCD-LSTM predictive performance by quartile by residual level.

4 Discussion

4.1 RQ1

The first question this thesis aims to answer is how the overall predictive performance compares between standard and Bayesian models, as well as between the two Bayesian models, evaluated through the discretized Continuous Ranked Probability Score and quantile loss functions at the 1% and 5% VaR levels, and whether any such differences are statistically significant. Following from Section 3.1, Tables 5 and 6 help answer these.

The comparisons do not yield a single answer to whether a Bayesian approach to VaR forecasting improves predictive performance, because the answer differs between the model families. Within the GARCH family, the predictive performance does improve. B-GARCH attains a lower mean loss than GARCH in all nine asset-metric combinations, and significantly so in four cases. Within the LSTM family, the sign of the Diebold-Mariano test is in general reversed, although more weakly, with the LSTM performing better in seven out of nine combinations and significantly so in one. These differing directions in which the Bayesian approach contributes to predictive performance are the central finding of this first research question.

In the remaining comparison between the two Bayesian models, the MCD-LSTM leads the B-GARCH in five of nine comparisons, significantly in two, and is never significantly worse. Studying just the raw scores can lead one to believe that the performance difference is far more significant than what it is, due to the magnitude of the mean loss difference between the two series, compared to that

produced by the Bayesian treatment, which can yield significance by differing in the fifth or sixth decimal. This can be explained by within-family comparisons sharing underlying prediction mechanisms: the volatility recursion (GARCH) and neural weights (LSTM), so the loss-difference series is consistent, and its long-run variance is small. Across families, the mean loss difference is larger, but the two loss series are far less correlated, increasing the long-run variance and leaving several tests insignificant. In practice, the Bayesian treatment offers a marginal but consistent improvement, whereas changing model family offers a larger but less reliable one.

The result of varying effect on predictive performance is perhaps not so surprising when considering the methods of Bayesian Inference being used. MCMC targets the exact posterior over three parameters, whereas MC Dropout targets a variational approximation of the posterior over a space of a far higher dimension.

The additional diagnostics of the GARCH models, provided in Appendix A.1, A.2 and A.3, may give insight into the origin of the better performance of B-GARCH. Considering the underlying parameter values of the models in Tables 14 and 16 reveals the consistent lesser persistence of the posterior mean compared to the ML-estimated parameters of the GARCH, and ω following the inverse relationship. Further observing the posterior distribution of the $\gamma = 1$ model in Figure 5 reveals the light skewness of the posteriors of said parameters. The skewness helps explain the separation of the models, as the GARCH is only concerned with the maximum of the likelihood, and B-GARCH with the entire posterior. As this separation caused B-GARCH to perform better, it is reasonable to believe that the additional parameter uncertainty of the posterior contributed to the better predictive performance.

The comparison between LSTM and MCD-LSTM is further interesting due to the nature of the shared underlying model. Both share the same underlying weights and architecture, only differing in the aggregation of the output at inference. The LSTM completes a deterministic forward pass, and the MCD-LSTM applies a dropout mask over the weights and averages over 100 such instances. Viewing the results confirms that the better-performing aggregation choice is the simple deterministic forward pass. The opposite of what the Bayesian treatment resulted in for the GARCH model. The difference could be understood from the difference in Bayesian inference. NUTS targets the exact posterior over three parameters, so averaging the forecasts over the posterior provides a better-informed quantity. For the MCD-LSTM, the corresponding shift is governed by an uncalibrated variational approximation over a weight space of far higher dimension, restricted to a Bernoulli family that is known not to recover posterior uncertainty faithfully (Le Folgoc et al., 2021). Therefore, whether propagating parameter uncertainty into the forecast improves it depends on the quality of the posterior approximation available, and the two families differ in that regard.

It is, however, not too much of a concern that the MCD-LSTM aggregate produces slightly less accurate results, as in practice, training the model once allows for using the standard forward pass for prediction, and additionally run-

ning the 100 dropout passes to utilize the epistemic uncertainty. This results in only one more pass at inference than for the MCD-LSTM and does not require any additional training.

4.2 RQ2

The second question being answered is whether the models produce reliable downside risk forecasts for 1% and 5% VaR levels, evaluated through the unconditional coverage, independence, and conditional coverage backtests. Observing Tables 7 and 8 of Section 3.2, the answer does seem to depend on the level. At the 1% level, every violation rate in model-asset combinations exceeds the nominal level, with none of the GARCH models being well calibrated according to the UC test. The LSTM family does, however, pass some UC tests while also exceeding the nominal level. Overall, the UC test is the limiting test for the models to perform reliably. For all except the MCD-LSTM on S&P 500, which fails the independence test and hence the conditional coverage test. If any model were deemed to produce reliable risk forecasts at the 1% level, it would be the LSTM, only failing two tests for the same asset. But still, the test result should speak to what extent it can be relied upon. The nominal level was exceeded for all assets, which is considerably worse than failing the UC test by being too conservative with the 1% VaR estimate. In essence, such a model could inform decisions that would expose market actors to unforeseen risks, potentially losing unexpected capital, whereas the alternative explanation to test-failure would rather cause actors to be more cautious, but potentially missing opportunities.

At the 5% level, the separation between the GARCH and LSTM families is less prevalent, and the test results improve across models. The UC tests that fail are S&P 500 for the GARCH models, and TSLA for the LSTM models. The LSTM models additionally fail the independence test with significance for TSLA, suggesting severe clustering and thereby not being reliable as VaR models for assets with the characteristics of TSLA. The performance is, however, not entirely deducible to be a function of characteristics of the asset, as the test at the 1% level did not present the same conclusions. A possible explanation for the LSTMs performing better at the 1% (and in general, for that matter) could be due to the statistical power of the tests. For approximately 1630 test observations per asset, 16 violations are expected; this affects the UC test such that a model can have at least 41% more violations and still pass (as for LSTM on TSLA). And more importantly, as the independence test only tests dependence of the first order, two violations have to occur in sequence to affect the test score negatively; with a total of 16 expected violations, there are a limited number of combinations in which this can occur. Therefore, models with limited reliability can pass at the 1% level but fail at the 5% level, as could be the case for TSLA.

As for the reliability of the GARCH models, three observations in particular are interesting. Firstly is the better consistency at the 5% level compared to the 1% level. Figure 3 of Appendix A.3 shows how the distributional assump-

tion due to the normal innovations of the GARCH models produces predictive return series unfit for financial data with leptokurtosis characteristics. The VaR backtests illustrate the same results on out-of-sample data, where the tails are not accurately modeled, resulting in excess violations. These are not new results, and therefore there exist several flavors of the GARCH models designed to resolve this issue, a common one being an alternative with Student-t innovations, where the degrees of freedom of the innovations act as an additional parameter of the model. This explanation only explains the excess violations above that of the LSTM, which itself also had excess violations despite no distributional assumptions. A candidate explanation affecting both models is the higher volatility profile of the test period compared to the train period in Table 4, in combination with the lack of refitting and retraining during testing, keeping fixed parameters and weights designed for different volatility dynamics. The second observation is that the standard GARCH models' violation rates are closer to the nominal level at the 5% level despite having larger quantile loss for that level. This illustrates how sharpness and calibration do not necessarily coincide, and choosing between the models is perhaps more nuanced than what the previous discussion suggests. Yet, it is important to acknowledge that no statistical test between the models has been performed regarding the violation rates and is therefore no more than speculation. The last observation that is of interest is how the GARCH model is the most reliable choice at the 5% level, only failing one test.

4.3 RQ3

The third question this thesis attempts to answer consists of three parts and only relates to the Bayesian models, B-GARCH and MCD-LSTM. How does the epistemic uncertainty produced by the two Bayesian models behave, and to what extent is it a distinct signal rather than a rescaling of the realized volatility level, and does this signal carry information about forecast reliability? To answer these, the results of Section 3.3 are used. From Figure 1 and the construction of the epistemic uncertainty of B-GARCH, it is evident that the model does not carry quantile-specific uncertainty, and that the signal is a simple scaling of the conditional volatility of the model. The MCD-LSTM does not follow any such mechanical behaviour, allowing for asymmetrical uncertainty, favouring the higher quantiles. The fact that the uncertainty is asymmetric suggests that the epistemic signal has distinct features apart from the RV, as that variable alone cannot favour any specific quantile level. It is also apparent from the figure that the uncertainty is higher for historic, high-volatility periods. But it cannot be deduced to what extent it is a rescaling of the volatility level, which the next part of this research question aims to answer.

Viewing Table 11, the log epistemic uncertainty of B-GARCH exhibits an elasticity of near 1.04 for S&P 500 and 0.90 for Brent relative to the log realized volatility, with respective R^2 values of near 0.70 and 0.43. The asset least explained by the RV is TSLA with an elasticity of 0.67 and an R^2 of 0.25. That the realized volatility would explain the epistemic level is anticipated due to the

construction of the epistemic uncertainty, following from Equation (56). It illustrates how this uncertainty signal is a scaled difference in conditional volatility between two separate posterior draws. During volatile periods, this difference grows bigger, and the model thus expresses more uncertainty. Following a similar behaviour to what is expected of the RV, also growing in volatile periods. Because of this, the relatively low $\hat{\beta}_1$ and R^2 observed for TSLA are more surprising. However, observing Table 4, the increasing mean RV level from S&P 500 to Brent to TSLA is the inverse pattern seen for the $\hat{\beta}_1$ and R^2 , providing a possible explanation. As the GARCH models are designed to capture the latent conditional volatility, and the realized volatility is simply an observable proxy on a 20-day window, the correlation between the two perhaps decreases for volatile series. Therefore, it seems that whether the epistemic signal of B-GARCH carries distinct information apart from the realized volatility is asset-specific, and is rather dependent on how well the RV is a proxy of the true volatility.

As for the MCD-LSTM, there is no clear mechanical explanation for a potential relationship; simultaneously, the RV stands as a reasonable alternative explanation of the epistemic level, as it is a feature of the model. The R^2 value never exceeds 0.6, and the elasticities are unstable across assets, with values of 0.39, 0.60, and 1.16. A signal that was a pure rescaling of the realized volatility would be expected to produce a stable elasticity near one; this one does not, suggesting that the epistemic signal carries distinct information apart from the realized volatility.

Finally, the epistemic signal carries information about forecast reliability, which is distinct from that explained by the level of realized volatility; however, the degree of reliability depends on the model and metric. The MCD-LSTM showed great score differences between Q1 and Q4 on the raw epistemic levels. This pattern persisted on the residual level but with lesser relative differences. The fact that the ratio between Q4 and Q1 decreased is not entirely unexpected considering the observed R^2 values of the regression, where a portion of the epistemic level could be explained by the RV, but not all. Furthermore, the residual quartiles disclosed how the model can be overconfident when the level of epistemic uncertainty is low relative to the realized volatility, as shown by the high test scores of Q1 on S&P 500. This pattern was, however, only visible for that asset and could not be generalized. Moreover, it should be stated that none of the observations are backed by statistical testing, and it cannot be determined if these observations are artifacts of coincidences or true patterns. The consistency of the patterns does, however, indicate that some underlying mechanism is being observed. The violation rates of the raw levels showed a less consistent pattern, which was further minimized on the residual levels. In most cases, the violation rate was the highest for Q4, but the violation rates can be unreliable both by exceeding and being less than the nominal level. Therefore, evaluating the reliability through the UC test did not show any clear pattern because of this. These signals indicate that the frequency-based forecasts do not carry clear information about the reliability of the forecasts.

As for the B-GARCH, the results are similar but less confident. Most notable is that the frequency-based metrics never expressed a clear signal of reliability

for the raw levels, and thereby not for the residual levels either. The scores did, however, show a persisting signal of worsening performance on the residual levels, perhaps not as strong as for MCD-LSTM.

5 Conclusion

This study investigated the usefulness of Bayesian approaches to VaR forecasting applied to the GARCH and LSTM models. This was done by comparing the models trained by traditional means with counterparts applying Bayesian inference, specifically NUTS and MC Dropout. As well as investigating whether the additional epistemic signal of Bayesian models is informative. This was done by testing forecasts of nine separate VaR quantiles on a one-day horizon, across S&P 500, TSLA, and Brent crude oil.

Evaluating the models’ predictive accuracy over the VaR quantiles, the Bayesian approach showed improvements when applied to GARCH, but not for the LSTM. The B-GARCH attains a lower mean loss than the GARCH in all nine asset-metric combinations, significantly so in four. The deterministic LSTM performs better than the MCD-LSTM in seven of nine combinations, significantly so once. Therefore, the method of Bayesian inference seems to matter, as they attain the parameter uncertainty differently. The comparison between the two Bayesian models is inconclusive, but favours the MCD-LSTM, attaining lower mean loss in five out of nine comparisons and significantly so in two. In practice, the Bayesian treatment offers marginal but consistent changes, whereas changing the model family offers a larger but less reliable one.

Reliability evaluated using Christoffersen’s conditional coverage framework of the downside forecasts depends on the quantile level. At the 1% level, every model exceeded the nominal violation rate, and both GARCH models failed the unconditional coverage test and hence the conditional coverage test throughout. The LSTM family was better calibrated there, with individual passing results. At the 5% level, the separation between families disappeared, and coverage improved overall, with the standard GARCH only failing one out of nine tests, and the other three out of nine.

The epistemic uncertainty of the two Bayesian models proved structurally different. That of the B-GARCH acted as a scaling of the conditional volatility, as well as a scaling of the realized volatility for the less volatile assets, and it differed between quantiles with a predetermined scaling factor. The MCD-LSTM gave separate signals per quantile and could not be deduced to be a scaling of realized volatility. Sorting the test period by epistemic uncertainty showed consistently degraded loss-based scores in the most uncertain quartile for both models and all three assets, and the pattern persisted after accounting for the level of realized volatility, though attenuated. The frequency-based metrics showed no clear pattern.

Taken together, the epistemic signal carries information about forecast reliability that the realized volatility level does not entirely account for, and the benefit of a Bayesian approach is tied to the inference method and model rather

than the use of the Bayesian framework itself.

A few limitations to this study are the normal innovations of the GARCH, the limited posterior approximation of MC Dropout, the lack of retraining during testing, and using RV estimated using a rolling window as an alternative explanation for the epistemic uncertainty. All of which could be addressed in future work by utilizing either Student-t innovations or other GARCH specifications, implementing a deep ensemble for a more reliable epistemic signal for the LSTM (Lakshminarayanan et al., 2017), partial or continuous retraining at inference, as well as using a more accurate volatility proxy and or additional market variables.

Appendix

A.1

Table 14 shows the estimated values of the GARCH parameters using MLE. Two additional columns are included, $\alpha + \beta$ and $\alpha/(\alpha + \beta)$, corresponding to the S and W parameters of the B-GARCH model. Table 15 shows the standard errors of the estimates.

Asset	ω	α	β	$\alpha + \beta$	$\alpha/(\alpha + \beta)$
S&P 500	0.0240	0.1217	0.8558	0.9775	0.1245
TSLA	0.0485	0.0160	0.9791	0.9951	0.0161
Brent	0.0276	0.0575	0.9378	0.9953	0.0578

Table 14: GARCH parameter estimates by asset.

Asset	SE(ω)	SE(α)	SE(β)
S&P 500	0.0056	0.0154	0.0159
TSLA	0.0314	0.0056	0.0075
Brent	0.0125	0.0097	0.0109

Table 15: Standard errors for the GARCH parameter estimates, by asset.

A.2

The four chains of 2000 post-burn-in samples for each asset using NUTS mix well and provide good diagnostics, as seen in Table 16. As mentioned in Section 2.3.1, well-performing sampling is characterized by $\hat{R} \leq 1.01$ and $\text{ESS} \geq 400$ (both bulk and tail) and zero divergences. All of which are met with margin. The last column labeled “Div.” refers to the number of divergent transitions, and the “Par.” label refers to the parameter being analyzed.

In Figure 2 it can be determined that the chains mix well enough. Ideally, all four chains should follow a straight line from 0 to 1 and should be indistinguishable from each other. However, all rank values within the grey shaded area are deemed good enough.

Asset	Par.	Mean	SD	$\hat{R}$	bulk-ESS	tail-ESS	Div.
S&P 500	S	0.9754	0.0051	1.00094	3605	3283	0
	W	0.1265	0.0114	1.00108	3882	4157	0
	ω	0.0253	0.0035	1.00057	3537	3904	0
TSLA	S	0.9944	0.0026	1.00199	2042	2089	0
	W	0.0194	0.0032	1.00223	3032	3153	0
	ω	0.0630	0.0240	1.00277	2021	2204	0
Brent	S	0.9928	0.0026	1.00102	2668	3216	0
	W	0.0617	0.0075	1.00037	3049	3572	0
	ω	0.0364	0.0097	1.00088	2411	2686	0

Table 16: NUTS sampling diagnostics for the B-GARCH posterior, by asset and parameter.

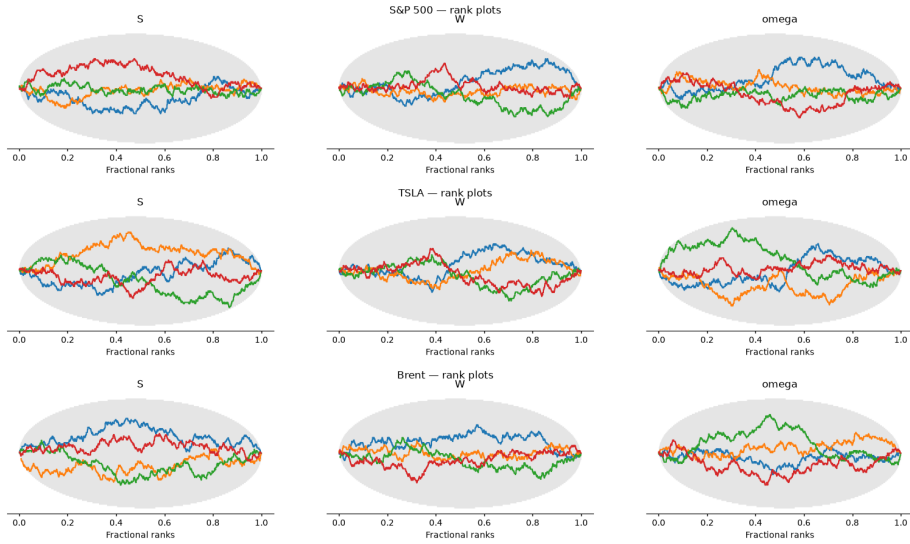


Figure 2: Rank plots over the parameters S , W and ω

A.3

The prior and posterior predictive checks can be evaluated by observing Figures 3 and 4. All realized returns are in-sample and are from the last 1500 observations of the training and validation splits, both of which are used for fitting. Firstly, looking at the prior predictive return distribution, the predictive returns from the prior have both a higher peak and less heavy tails compared to all asset realized returns, though out of the three assets, it seems decently suited for the S&P 500. Conditioning on the data increases the similarities to the realized

series; however, issues of the distributional assumptions of GARCH models become more apparent. For all assets, the peak and tails are underestimated, not capturing the leptokurtosis of the real data. This is an issue of model choice rather than prior choice. In the Bayesian workflow described by Gabry et al. (2019), this result could call for a change in the model; however, as the model choice is set, this limitation is accepted without any modifications.

Figure 4 is perhaps less informative, but illustrates reasonable behaviour. The prior predictive series show clear signs of volatility clustering, which is desired when modelling financial time series. However, the clustering seems to be at a magnitude greater than the actual series. This is, however, not of great concern, as the data shows itself informative and increases the similarities between the posterior predictive and realized returns. This evaluation is far from quantitative, but passes eye judgement.

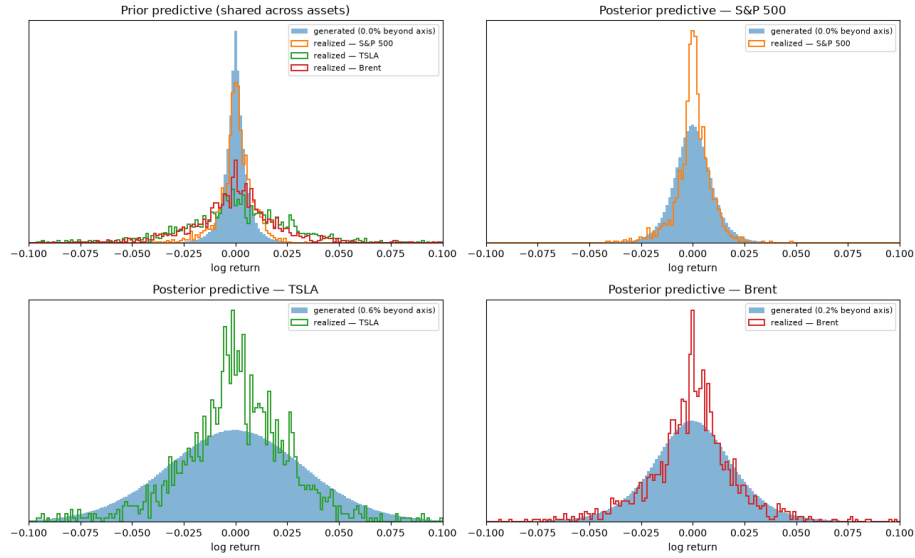


Figure 3: Prior and posterior predictive check of return distributions

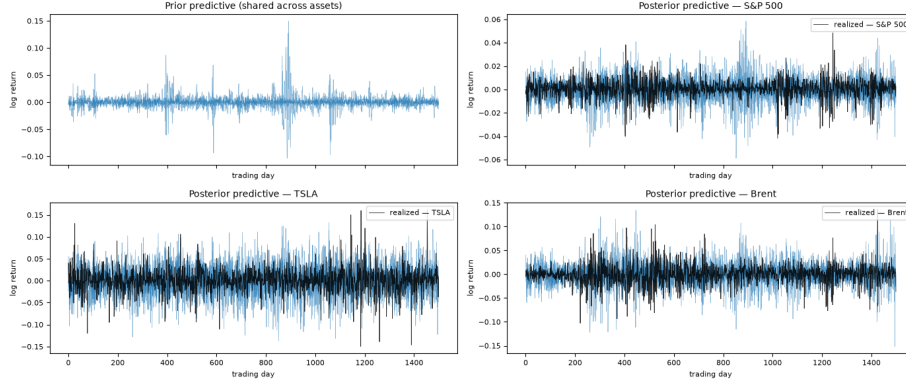


Figure 4: Prior and posterior predictive check of return series

A.4

Indications of informative likelihoods have been observed from Appendix A.3. To explicitly handle the sensitivity to the choice of prior, the baseline ($\gamma = 1$) posterior distributions are compared to those of the alternate power-scaled priors ($\gamma \in \{\frac{1}{2}, 2\}$). Power-scaling using $\gamma = 2$ represents more informative priors, making high-density regions of the prior denser, and $\gamma = \frac{1}{2}$ represents the opposite.

Figure 5 shows how the S&P 500 posterior is robust against the choice of prior, as all posteriors share great overlap. For TSLA and Brent, larger shifts in the posterior with some common behaviour are visible. The S parameter is generally shifted to the right, favouring higher persistence for the least informative prior $\gamma = \frac{1}{2}$, and gradually shifts to the left as the prior becomes more informative, which can be explained by the density distributions of the priors. The same effect can be observed for ω , but in the inverse direction. These changes indicate slight sensitivity to the choice of prior. However, it should also be noticed that the $\gamma = 1$ posterior visibly shows more overlap with the weaker $\gamma = \frac{1}{2}$ posterior than with the $\gamma = 2$ posterior, suggesting that the baseline is chosen to be weakly informative. Therefore, despite being slightly sensitive, these findings and those of Appendix A.3 suggest that the baseline prior is a fine choice.

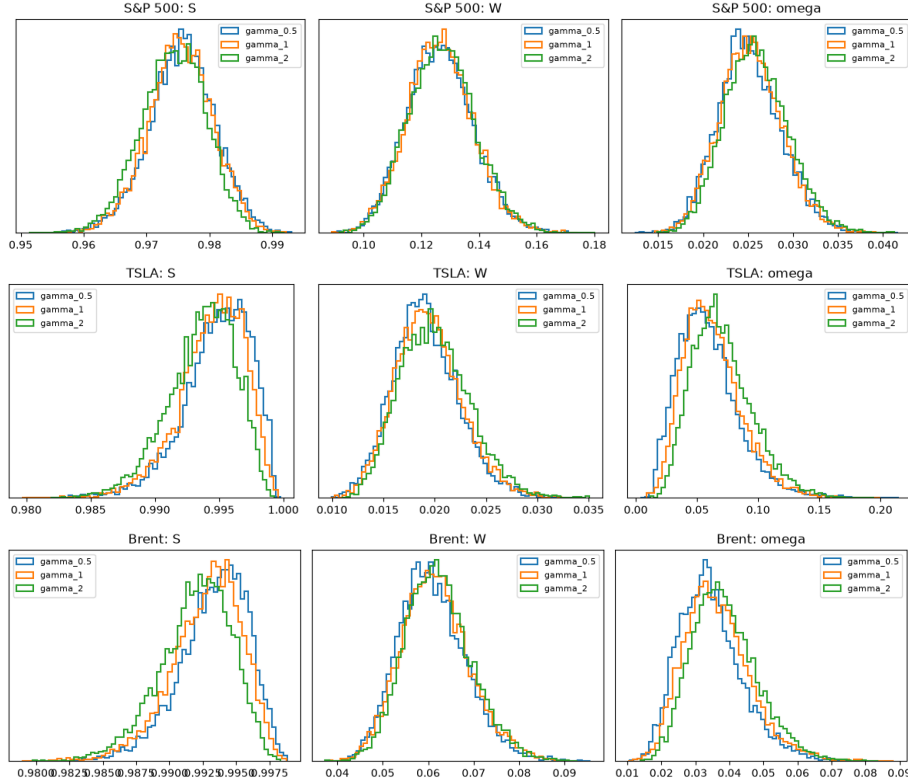


Figure 5: Prior sensitivity over the parameters S , W and ω , for power scaling of degrees $\gamma \in \{\frac{1}{2}, 1, 2\}$

B.1

Performing random search for 20 iterations for each asset yields the best-performing hyperparameter configurations on the validation loss, as seen in Table 17.

Asset	Learning rate	Batch	Dropout	Lookback	Hidden	Layers
S&P 500	3.96×10^{-4}	256	0.120	60	53	2
TSLA	2.35×10^{-3}	256	0.291	59	50	1
Brent	4.23×10^{-3}	128	0.262	75	63	2

Table 17: LSTM configuration of the selected model

The validation loss for each search iteration ranked from lowest to highest can be seen in Table 18. Rank 1 is the configuration reported in Table 17.

Rank	S&P 500	TSLA	Brent
1	0.001551	0.005980	0.003370
2	0.001554	0.005984	0.003372
3	0.001556	0.005988	0.003377
4	0.001557	0.005991	0.003377
5	0.001558	0.005991	0.003384
6	0.001558	0.005991	0.003385
7	0.001558	0.005993	0.003386
8	0.001559	0.005993	0.003386
9	0.001559	0.005994	0.003389
10	0.001561	0.005994	0.003390
11	0.001561	0.005995	0.003390
12	0.001561	0.005996	0.003397
13	0.001561	0.005997	0.003399
14	0.001563	0.005998	0.003401
15	0.001563	0.006001	0.003403
16	0.001565	0.006008	0.003405
17	0.001567	0.006013	0.003407
18	0.001572	0.006016	0.003415
19	0.001579	0.006018	0.003440
20	0.001588	0.006022	0.003454

Table 18: Validation loss of every random-search draw.

B.2

Running reseeding on the configuration with the lowest validation loss for each asset yields separate validation losses for the five models as seen in Table 19. The median-performing model was chosen as the main model for this thesis, and the range between lowest and highest validation loss is also reported on the last row for each asset.

Asset	Seed	Validation loss	Main model
S&P 500	0	0.001551	✓
	1	0.001567	
	2	0.001555	
	3	0.001558	
	4	0.001555	
	<i>range</i>	0.0000157	
TSLA	0	0.005980	✓
	1	0.005983	
	2	0.005990	
	3	0.005998	
	4	0.005979	
	<i>range</i>	0.0000187	
Brent	0	0.003370	✓
	1	0.003377	
	2	0.003374	
	3	0.003386	
	4	0.003387	
	<i>range</i>	0.0000168	

Table 19: Seed sensitivity and model selection.

Notable is how the seed with the highest validation loss would place itself at rank 9 or higher in Table 18. Meaning, the variation between the best and 8th configuration is less than that produced by random initialization of 5 models with the same configuration. Thus, suggesting that the chosen configuration is in fact less informed than what it might seem.

B.3

Quantile crossing before monotone rearrangement is reported in Tables 20 and 21. As both rearrange the quantiles in different manners, the tables report separate quantities. For the LSTM, the frequency of crossings across each pair is reported. As the LSTM only has one forward pass for each timestep, the value of 0.06 for S&P 500 means that for 0.06% of the forward passes the 0.05 and 0.10 quantiles did not produce predictions according to a valid distribution. For all other pairs of quantiles and assets, it did produce valid distributions. The last row, “Any pair”, specifics for what frequency of the days that there were any crossings in the prediction.

For the MCD-LSTM, two values are instead reported. First is “Passes”, which corresponds to the frequency with which crossings occur for that pair for the 100 forward passes over time. To exemplify, there are in total approximately $100 \times 1630 = 163000$ passes for each quantile pair. For the 0.95–0.99 pair, 5.71% of the 163 000 passes are in the wrong order. These large figures are, however,

not any issues as this is done before aggregating to a point estimate, and these crossings are therefore never repaired. These figures could rather be regarded as a way to illustrate the LSTM’s ability to produce proper distributions without active dropout at inference, and the “noise” that dropout introduces. Second is “Aggr.”, which reports the frequency with which the aggregated point-estimated VaR quantiles cross. These are the crossings being rearranged.

Quantile pair	S&P 500	TSLA	Brent
0.01 – 0.05	0.00	0.00	0.00
0.05 – 0.10	0.06	0.00	0.00
0.10 – 0.25	0.00	0.00	0.00
0.25 – 0.50	0.00	0.00	0.00
0.50 – 0.75	0.00	0.00	0.00
0.75 – 0.90	0.00	0.00	0.00
0.90 – 0.95	0.00	0.00	0.00
0.95 – 0.99	0.00	0.00	0.00
Any pair	0.06	0.00	0.00

Table 20: Quantile crossing rates (%) in the LSTM

Quantile pair	S&P 500		TSLA		Brent	
	Passes	Aggr.	Passes	Aggr.	Passes	Aggr.
0.01 – 0.05	0.00	0.00	2.82	0.00	0.00	0.00
0.05 – 0.10	6.64	0.06	0.54	0.00	0.00	0.00
0.10 – 0.25	0.69	0.00	0.01	0.00	0.00	0.00
0.25 – 0.50	3.07	0.00	0.00	0.00	0.00	0.00
0.50 – 0.75	8.88	0.00	3.64	0.00	0.00	0.00
0.75 – 0.90	0.59	0.00	4.36	0.00	0.00	0.00
0.90 – 0.95	7.26	0.00	16.88	0.00	0.01	0.00
0.95 – 0.99	5.71	0.00	1.92	0.00	0.00	0.00
Any pair	26.77	0.06	26.49	0.00	0.01	0.00

Table 21: Quantile crossing rates (%) in the MCD-LSTM.

C

Testing stationarity for the three series used in the regression shows relatively consistent results in Table 22. All series have conflicting results between the two null hypotheses of ADF and KPSS for the 5% significance level, indicated with a dash –, except for $\log(U_{epi}^{(B-GARCH)})$ on TSLA and $\log(RV)$ on Brent, whose p -values are not themselves far from showing conflicting results. That

$\log(U_{epi}^{(B-GARCH)})$ shows conflicting results is perhaps a little surprising considering that in Equation (56) it was shown that the epistemic uncertainty of the B-GARCH is a rescaling of the conditional volatility, which was imposed to be stationary through its prior distributions. However, observing the posterior densities of the persistence S of Figure 5, they are near a unit root, which could lead to the conflicting result, considering a finite sample size is used. Therefore, the result that the epistemic uncertainty of B-GARCH provides a conflicting result despite stationarity being imposed indicates the other series could in fact be stationary as well, but near a unit root and also yield conflicting results. The other series do, in addition, also possess higher p -values for the KPSS test and lower p -values for the ADF, compared per asset, except for the ADF on Brent, suggesting they might be further from a unit root. However, this is not conclusive as the majority of the tests still reject stationarity.

From Table 23, it is evident that the first differences are stationary. This provides two separate possibilities of interpretation. Either the log-level series are non-stationary, and the integration order is $I(1)$, and Engle-Granger’s (EG) cointegration test of Table 24 is valid and shows the regression is non-spurious. Alternatively, the log-levels possess a partial integration order of $I(d)$ for $d \in (0, 1)$, for which the high persistence and stationarity suggest slow mean reversion and coupled trending, and the regressions are therefore unlikely to be spurious.

Table 24 provides the result of the EG cointegration test, in which the null of no cointegration is tested.

Series	Asset	ADF	KPSS	Stationary
$\log(U_{epi}^{(B-GARCH)})$	S&P 500	0.0002	0.0100	—
	TSLA	0.0726	0.0100	×
	Brent	0.0000	0.0100	—
$\log(U_{epi}^{(MCD-LSTM)})$	S&P 500	0.0000	0.0245	—
	TSLA	0.0001	0.0100	—
	Brent	0.0001	0.0462	—
$\log(RV)$	S&P 500	0.0005	0.0100	—
	TSLA	0.0022	0.0187	—
	Brent	0.0005	0.0679	✓

Table 22: p -values of stationarity tests (ADF and KPSS) on the log-level series (not differenced) used as input to the Engle-Granger cointegration test. $T = 1627$ for S&P 500 and TSLA, $T = 1631$ for Brent. The p -values of the KPSS are truncated outside 0.01 and 0.1.

Series	Asset	ADF	KPSS	Stationary
$\Delta U_{epi}^{(B-GARCH)}$	S&P 500	4.81×10^{-28}	0.1	✓
	TSLA	2.04×10^{-25}	0.1	✓
	Brent	6.35×10^{-24}	0.1	✓
$\Delta U_{epi}^{(MCD-LSTM)}$	S&P 500	≈ 0	0.1	✓
	TSLA	5.25×10^{-19}	0.1	✓
	Brent	≈ 0	0.1	✓
ΔRV	S&P 500	3.61×10^{-21}	0.1	✓
	TSLA	4.03×10^{-23}	0.1	✓
	Brent	7.01×10^{-21}	0.1	✓

Table 23: p -values of stationarity tests (ADF and KPSS) on the first-order differences on the log-level series (denoted Δ). $T = 1627$ for S&P 500 and TSLA, $T = 1631$ for Brent. The p -values of the KPSS are truncated outside 0.01 and 0.1.

Model	Asset	EG	Cointegrated
B-GARCH	S&P 500	0.0000	✓
	TSLA	0.0311	✓
	Brent	0.0000	✓
MCD-LSTM	S&P 500	0.0000	✓
	TSLA	0.0000	✓
	Brent	0.0000	✓

Table 24: Engle–Granger cointegration p -values for the residual of the $\log(U_{epi})$ on $\log(RV)$ regression, by model and asset. $T = 1627$ for S&P 500 and TSLA, $T = 1631$ for Brent.

References

- Andersen, T. G., Bollerslev, T., Diebold, F. X., & Labys, P. (2003). Modeling and forecasting realized volatility. *Econometrica*, 71(2), 579–625.
- Andrieu, C., & Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4), 343–373.
- Ardia, D. (2008). *Financial risk management with Bayesian estimation of GARCH models*. Springer.
- Ardia, D., Kolly, J., & Trottier, D.-A. (2017). The impact of parameter and model uncertainty on market risk predictions from GARCH-type models. *Journal of Forecasting*, 36(7), 808–823.
- Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2), 157–166.
- Betancourt, M. (2017). *A conceptual introduction to Hamiltonian Monte Carlo*. arXiv: 1701.02434.
- Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518), 859–877.
- Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3), 307–327.
- Chernozhukov, V., Fernández-Val, I., & Galichon, A. (2010). Quantile and probability curves without crossing. *Econometrica*, 78(3), 1093–1125.
- Christoffersen, P. F. (1998). Evaluating interval forecasts. *International Economic Review*, 39(4), 841–862.
- Dickey, D. A., & Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366), 427–431.
- Diebold, F. X., & Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 13(3), 253–263.
- Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 50(4), 987–1007.
- Engle, R. F., & Granger, C. W. J. (1987). Co-integration and error correction: Representation, estimation, and testing. *Econometrica*, 55(2), 251–276.
- Engle, R. F., & Manganelli, S. (2004). CAViaR: Conditional autoregressive value at risk by regression quantiles. *Journal of Business & Economic Statistics*, 22(4), 367–381.
- Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., & Gelman, A. (2019). Visualization in Bayesian workflow. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 182(2), 389–402.
- Gal, Y., & Ghahramani, Z. (2016a). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In M. F. Balcan & K. Q. Weinberger (Eds.), *Proceedings of the 33rd international conference on machine learning* (pp. 1050–1059, Vol. 48). PMLR.

- Gal, Y., & Ghahramani, Z. (2016b). A theoretically grounded application of dropout in recurrent neural networks. In D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 29). Curran Associates, Inc.
- Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian data analysis* (Third) [(with errors fixed as of 20 February 2025)]. Chapman; Hall/CRC.
- Gers, F., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12, 2451–2471.
- Gneiting, T. (2011). Making and evaluating point forecasts. *Journal of the American Statistical Association*, 106(494), 746–762.
- Gneiting, T., & Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477), 359–378.
- Granger, C., & Newbold, P. (1974). Spurious regressions in econometrics. *Journal of Econometrics*, 2(2), 111–120.
- Haario, H., Saksman, E., & Tamminen, J. (2001). An adaptive metropolis algorithm. *Bernoulli*, 7(2), 223–242.
- Hansen, P. R., & Lunde, A. (2005). A forecast comparison of volatility models: Does anything beat a GARCH(1,1)? *Journal of Applied Econometrics*, 20(7), 873–889.
- Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1), 97–109.
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9, 1735–1780.
- Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(47), 1593–1623.
- Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37(2), 183–233.
- Kallioinen, N., Paananen, T., Bürkner, P.-C., & Vehtari, A. (2023). Detecting and diagnosing prior and likelihood sensitivity with power-scaling. *Statistics and Computing*, 34(57).
- Kendall, A., & Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc.
- Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations (ICLR)*.
- Koenker, R., & Bassett, G. (1978). Regression quantiles. *Econometrica*, 46(1), 33–50.
- Koenker, R., & Bassett, G. (1982). Robust tests for heteroscedasticity based on regression quantiles. *Econometrica*, 50(1), 43–61.

- Kupiec, P. H. (1995). Techniques for verifying the accuracy of risk measurement models. *The Journal of Derivatives*, 3(2), 73–84.
- Kwiatkowski, D., Phillips, P. C., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of Econometrics*, 54(1), 159–178.
- Laio, F., & Tamea, S. (2007). Verification tools for probabilistic forecasts of continuous hydrological variables. *Hydrology and Earth System Sciences*, 11(4), 1267–1277.
- Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc.
- Le Folgoc, L., Baltatzis, V., Desai, S., Devaraj, A., Ellis, S., Martinez Manzanera, O. E., Nair, A., Qiu, H., Schnabel, J., & Glocker, B. (2021). *Is MC dropout Bayesian?* arXiv: 2110.04286.
- Linsmeier, T. J., & Pearson, N. D. (1999, January). *Risk measurement: An introduction to value at risk* (Working Paper) (Original version July 1996). University of Illinois at Urbana-Champaign. https://www.casact.org/sites/default/files/old/specsem_99frmgt_pearson2.pdf
- Manganelli, S., & Engle, R. F. (2001, August). *Value at risk models in finance* (ECB Working Paper No. 75). European Central Bank. Frankfurt am Main. <https://www.ecb.europa.eu/pub/pdf/scpwps/ecbwp075.pdf>
- Martin, O. A., Kumar, R., & Lao, J. (2021, December). *Bayesian Modeling and Computation in Python*. Chapman Hall/CRC Press.
- McMillan, D. G., & Ruiz, I. (2009). Volatility persistence, long memory and time-varying unconditional mean: Evidence from 10 equity indices. *The Quarterly Review of Economics and Finance*, 49(2), 578–595.
- McNeil, A. J., & Frey, R. (2000). Estimation of tail-related risk measures for heteroscedastic financial time series: An extreme value approach [Special issue on Risk Management]. *Journal of Empirical Finance*, 7(3), 271–300.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6), 1087–1092.
- Neal, R. M. (2011). MCMC using Hamiltonian Dynamics. In S. Brooks, A. Gelman, G. L. Jones, & X.-L. Meng (Eds.), *Handbook of Markov Chain Monte Carlo* (pp. 113–162). CRC Press.
- Newey, W. K., & West, K. D. (1987). A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix. *Econometrica*, 55(3), 703–708.
- Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J. V., Lakshminarayanan, B., & Snoek, J. (2019). Can you trust your model’s uncertainty? Evaluating predictive uncertainty under dataset shift. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, &

- R. Garnett (Eds.), *Advances in neural information processing systems* (pp. 13991–14002, Vol. 32). Curran Associates, Inc.
- Parkinson, M. (1980). The extreme value method for estimating the variance of the rate of return. *The Journal of Business*, 53(1), 61–65.
- Said, S. E., & Dickey, D. A. (1984). Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*, 71(3), 599–607.
- Taylor, J. W. (2000). A quantile regression neural network approach to estimating the conditional density of multiperiod returns. *Journal of Forecasting*, 19(4), 299–311.
- Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., & Bürkner, P.-C. (2021). Rank-Normalization, Folding, and Localization: An Improved $\hat{R}$ for Assessing Convergence of MCMC (with Discussion). *Bayesian Analysis*, 16(2), 667–718.