Neural Networks and Value at Risk in Asset Management

A Preprint

Alexander Arimond
SDGLabs.ai
Sweden, EU
alexander.arimond@sdglabs.ai

Damian S. Borth
University of St. Gallen
Switzerland
damian.borth@unisg.ch

Sergio Garcia-Vega
SDGLabs.ai
Sweden, EU
sergio.garcia@sdglabs.ai

Maretno Harjoto
Pepperdine Graziadio Business School
Pepperdine University
California, USA
maretno.harjoto@pepperdine.edu

Andreas G.F. Hoepner*
University College Dublin
Ireland, EU
andreas.hoeopner@ucd.ie

Michael Klawunn
SDGLabs.ai
Sweden, EU
michael.klawunn@sdglabs.ai

Stefan Weisheit
SDGLabs.ai
Sweden, EU
stefan.weisheit@sdglabs.ai

July 13, 2025

*Acknowledgments: We thank Theodor Cojoianu, James Hodson, Juho Kanniainen, Qian Li, Yanan, Andrew Vivian, Xiaojun Zeng and participants at 2019 Financial Data Science Association conference in San Francisco and the International Conference on Fintech and Financial Data Science at University College Dublin (UCD) for their comments. We are also grateful to Warburg Invest AG for an early stage funding of this project. The views expressed in this manuscript are not necessarily shared by SDGLabs.ai, the Technical Expert Group of DG FISMA, or Warburg Invest AG. Authors are listed in alphabetical order, whereby Hoepner serves as the contact author (andreas.hoepner@ucd.ie). Any remaining errors are our own.

NEURAL NETWORKS AND VALUE AT RISK IN ASSET MANAGEMENT

ABSTRACT

This study investigates whether machine learning can be utilized to reduce the occurrence of breaching the Value at Risk (VaR) at 5% and 1% thresholds in a portfolio of equity and bond indices. We compare the occurrence and the likelihood of breaches from the Convolutional Neural Network (CNN), the Long Short-Term Memory (LSTM) recurrent neural networks, and the feed forward (FF) using the initiations from the traditional approach (mean/variance or classic) and the Hidden Markov Model (HMM). We perform Monte-Carlo simulations of asset returns to estimate the threshold breaches of VaR using the US, Euro area, UK, and World equity and bond market indices up to 1,343 weeks from January 1987 until mid-June 2020. We also investigate the number and the percentage of breaches under the incentive function that has been trained to take into account the bull and bear markets and the amount of historical data feed (2,000 versus 1,000 days). We find that the LSTM recurrent network initialized with the HMM and balanced incentive function can offer superior protection for asset managers against a downside risk through a reduction in VaR threshold breaches. However, such advancement relies on the availability of long historical data.

Keywords Neural Networks ; Value at Risk (VaR) ; Long Short-Term Memory (LSTM) ; Hidden Markov Model (HMM) ; Convolutional Neural Network (CNN)

JEL Classifications C45 C63 G32 G11 G12

1 Introduction

Researchers have demonstrated that neural networks consisting a set of powerful machine learning have the capability to solve complex relationships (Mamatzakis et al., 2024; Tsionas et al., 2024), especially to predict volatility (Fukuyama et al., 2024) and to evaluate efficiency (Johnes et al., 2025). Naturally, extant studies in finance have explored whether machine learning can help asset managers to predict the volatilities and Value at Risk (VaR) in equity and bond markets (Castillo et al., 2021; Sáenz et al., 2023). While leading papers on machine learning in asset pricing predominantly focus on returns and stochastic discount factors (Avramov et al., 2023; Azevedo and Hoegner, 2023; Chen et al., 2024; Gu et al., 2020; Pan et al., 2023; Sáenz et al., 2023), there has been growing studies that propose machine learning methods to estimate various asset pricing, risk management, and hedging strategies (Danielsson et al., 2022; Liu and Pun, 2022; Na and Wan, 2023; Novykov et al., 2025). Our study investigates whether machine learning methods can improve the estimates of Value at Risk (VaR) using international equity and bond markets, especially in estimating a catastrophic type of risk, i.e., tail risk (Gkillas and Vasiliadis, 2025; Mögel and Auer, 2018; Sehgal et al., 2025; Hossain et al., 2023; Wu et al., 2025), especially during the periods when risk estimation tasks have become more challenging[1].

Literature has demonstrated that machine learning can play a significant role in accounting and finance (Fedenia et al., 2024; Liaras et al., 2024; Mahmood et al., 2023). Gu et al. (2020) (GKX hereafter) apply the machine learning to improve measures for asset risk premia. They observe that machine learning improves the description of expected returns relative to traditional econometric forecasting methods based on (i) better out-of-sample R-squared and (ii) forecasts earning larger Sharpe ratios. More specifically, compared to four traditional econometric estimation methods (OLS, GLM, PCR/PCA, PLS) with regression trees (e.g. random forests), a simple feed forward neural network (FF) based on approximately 30,000 stocks over 720 months during 1957-2016, using 94 firm characteristics, 74 sectors and over 900 baseline signals, performs better. GKX demonstrate that FF perform best due to the flexibility of functional form and enhanced ability to prioritize vast sets of baseline signals.

GKX also indicate that neural network (NN) with 3 hidden layers (NN3) outperforms the neural networks with more than 3 hidden layers. They interpret this result as a consequence of a relatively much lower signal to noise ratio and much smaller data sets, which are typical in finance data. They find that outperformance of NN over the other four traditional econometric estimation methods widens at the portfolio level compared to the stock level. Hence, a greater understanding of the signal to noise ratio when using neural networks at the portfolio level is crucial in asset management practices.

[1]In other words, 'Machine Learning' has seemingly infinite practical applications in empirical finance and may leapfrog or even retire some classic econometric estimation methods. Hence, not exploring the potential benefits and risks of machine learning is not necessarily a progressive approach to academic research.

Studies on machine learning in finance have grown since the GKX study. Chen et al. (2024) (CPZ in the following) introduce more advanced (i.e. recurrent) neural networks and estimate a non-linear asset pricing model with regularized under no-arbitrage conditions operationalized via a stochastic discount factor while taking into account economic conditions. In particular they attribute the time varying dependency of the stochastic discount factor of about ten thousand US stocks to macroeconomic state processes via a recurrent Long Short-Term Memory (LSTM) network. In CPZ's view "it is essential to identify the dynamic pattern in macroeconomic time series before feeding them into a machine learning model" (Chen et al., 2024, p. 5).

Avramov et al. (2023) (ACM hereafter) replicate the approaches of GKX's, CPZ, and two conditional factor pricing models: Kelly et al.'s (2019) linear instrumented principal component analysis (IPCA) and Gu et al.'s (2021) nonlinear conditional autoencoder in the context of real-world economic restrictions. While they find strong positive Fama-French six-factor (FF6) adjusted returns in the original setting without real world economic constraints, these returns reduce by more than half when microcaps or firms without credit ratings are excluded. In fact, when ACM exclude distressed firms, all deep learning methods no longer generate significant (value-weighted) FF6-adjusted return at the 5% level. They confirm this finding by showing that the GKX and CPZ machine learning signals perform substantially weaker in economic conditions that limit stocks that are difficult-to-value and difficult-to-arbitrage (i.e. low market liquidity, high market volatility, high investor sentiment).

ACM also find that the only linear model they analyze from Kelly et al.'s (2019) IPCA, stands out as less sensitive to high limits-to-arbitrage market episodes. ACM's finding as well as the results from CPZ imply that economic conditions have to be explicitly accounted for when analyzing the abilities and performance of neural networks. Furthermore, ACM as well as GKX and CPZ make anecdotal observations that machine learning methods appear to reduce drawdowns[2]. Sáenz et al. (2023) find that machine learning method, i.e., Long Short-Term Memory (LSTM) with clustering techniques, could produce superior results in predicting prices and returns over the linear model. Wang et al. (2023) utilize random forest and LSTM from machine learning to forecast cryptocurrency volatility and find that machine learning methods are superior compared to traditional volatility model (i.e. GARCH).

While these studies focus on return predictability, we devote our work on the risk predictability in the context of market wide economic downturn conditions. The Covid-19 crisis as well as economic and financial crises in the previous three decades and unexpected risk from tradewars and geopolitical risk imply that catastrophic 'black swan' type risks occur more frequent than predicted by the symmetric economic distributions. Consequently, underestimating tail risks can bring catastrophic losses for asset managers and

[2]Further recent applications of machine learning in finance include the extensive recent works of Amini et al. (2021) on capital structure, Aubry et al. (2019) on real assets, Bianchi et al. (2021) on bond return predictability, Easley et al. (2021) on microstructure, Götze et al. (2020) on catastrophe bonds, Hunt et al. (2019) on earnings forecasts, Sadhwani et al. (2021) on mortgages, Verstyuk (2020) on macroeconomic forecasts, and Cao et al. (2021) on arbitrage-free option pricing. Studies on methodological work on dimension reduction techniques such as De Nard et al. (2020), Giglio and Xiu (2021), and Kozak et al. (2020) are also noteworthy, as well as efforts by Fallahgoul et al. (2021) and Horel and Giesecke 2019 to develop significant tests for neural networks.

institutional investors. Hence, the analysis of risks with the main goal is to avoid underestimating the tail risk deserves equivalent if not more attention among portfolio managers.

Studies that propose the use of machine learning for portfolio management have grown significantly (Lee et al., 2023; Novykov et al., 2025; Parisi and Manaog, 2025). Empirically, studies have found that equity market indices usually exhibit a negative skewness in its return payoff (Albuquerque, 2012; Kozhan et al., 2013). Hence, a symmetric distribution such as the Markowitz's Mean/Variance framework that implicitly assumes normality of asset returns is more likely to underestimate the tail risk for assets with negatively skewed payoffs (Agarwal and Naik, 2004). Consequently, in post Covid-19 world with substantial tail risk exposures from potential tradewars, geopolitical risk, cyber security breaches, etc., it is crucial that asset managers and institutional investors are equipped with tools that allow them to avoid the underestimation of tail risks. Naturally, neural networks with their unlimited flexibility in modelling non-linearities appear suitable candidates for such conservative tail risk modelling to avoid underestimations.

Therefore, the main objective of this study is to investigate whether more advanced neural networks can help asset management practices to reduce underestimation of the tail risk in the portfolio management setting. We operationalize the tail risk as Value at Risk (VaR) that is most often used as the tail risk measure in both commercial practice as well as academic literature (Billio et al., 2012; Billio and Pelizzon, 2000; Borer et al., 2023; Jorion, 2006; Nieto and Ruiz, 2016; Zhang et al., 2023; Wang et al., 2024). Specifically, we estimate VaR thresholds using classic method (i.e. Mean/Variance) and Hidden Markov Model (HMM) as well as machine learning methods (i.e. feed forward, convolutional, and recurrent) and we advance these methods via initialization of input parameter and regularization of incentive function. Recognizing the importance of economic conditions in machine learning studies (Avramov et al., 2023; Chen et al., 2024), we embed our analysis with a regime-based asset allocation setting to account for economic conditions.

We perform Monte-Carlo simulations of asset returns for VaR threshold estimation in a generative regime switching framework (Maringer and Ramtohul, 2012; Moody and Wu, 1997; Uyar et al., 2025). Using international equity and long-term bond with 7-10 years maturities in the US, Euro area, UK, and the World (Global) markets from various months and years starting points, we examine up to 1,343 weeks sample ending in mid-June 2020 and investigate three neural networks (FF, CNN, and LSTM) along three design steps relating (i) to the initialization of the neural network's input parameter, (ii) its incentive function according to which it has been trained to avoid extreme outputs if it is not regularized, and (iii) the amount of data feed we use.

First, we compare neural networks with random seeding with networks that are initialized via estimations from the Hidden Markov Model (HMM). We find that the HMM initiation outperforms the random seeding in terms of lower frequency of VaR breaches (i.e. the realized return falling short of the estimated VaR threshold). Comparing across asset classes (equity and bond), we find that the HMM initialization reduces the VaR breaches for the equity markets more than to the bond markets. Since volatility in equity markets is usually greater than in bond markets, adding the HMM initialization instead of random seed notably improves the performance of neural network models, especially the LSTM model, to reduce catastrophic losses from the equity markets. Beyond the considerable performance advances, HMM initialization removes the need to fix the random seed to a constant integer (i.e., 1) to ensure full replicability.

Second, we find that balancing the incentive structure of the loss function by adding a second objective to the training instructions so that the neural networks optimize for accuracy while also aiming to stay in empirically realistic economic regime distributions (i.e. bull vs. bear market frequencies) provide better outcomes. The HMM initiation and balancing incentive features leads to a better regularization substantially reduce extreme outcomes. In particular, these design features enable the long short-term memory (LSTM) to outperform any other neural network or established approach by statistically and economically significant levels. Thus, the choice of initiation and balancing incentive on the LSTM model could enhance the predictions for anomalies in the financial markets (Azevedo and Hoegner, 2023).

Third, we use a half our training data feeds and find our neural network models when fed with substantially less data are performing significantly worse which highlights a crucial weakness of neural networks in their dependence on large data sets. Overall, our findings imply that the initialization and balancing incentive are particularly critical for the LSTM since they highlight the feature of the LSTM model to perform better with the long time-series data while the feature of the convolutional neural network (CNN) is better for the cross-sectional data. The results also show that machine learning models with initiations and balancing incentive perform better in the equity markets than in the bond markets.

Thus, the contributions of our study are fivefold. First, we extend the current literature of machine learning in finance (Avramov et al., 2023; Chen et al., 2024; Gu et al., 2020; Pan et al., 2023; Sáenz et al., 2023; Wang et al., 2023) by emphasizing on estimating risk (VaR) thresholds for portfolio managers, which is different from those that examine the use of machine learning for estimating risk for asset management (Cosma et al., 2023; Nguyen et al., 2023b; Novykov et al., 2025; Parisi and Manaog, 2025). By assessing the advancements of machine learning to reduce the underestimation of tail risks, we offer valuable insights to asset managers such as pension funds to protect the retirement savings of their clients or members by reducing the tail risk exposures across asset classes (equity and bond markets).

Second, we advance the design of three types of neural networks by initializing their input parameter using the Hidden Markov Model (HMM). While initializations are a common research topic in core machine learnings fields such as image classification or machine translation (Glorot and Bengio, 2010; Zhang et al., 2019), our study advances international finance literature on systematic application of initialized neural networks on global portfolio management in equity and bond markets (Wang et al., 2024). Hence, demonstrating the statistical superiority of an initialized neural network over non-initialized neural network in global settings can be considered as a relevant contribution to the international finance community (Liu and Pun, 2022; Na and Wan, 2023; Sáenz et al., 2023; Wang et al., 2023).

Third, while CPZ regularize their neural networks using no arbitrage conditions, Sáenz et al. (2023) apply clustering in LSTM, and Wang et al. (2023) utilize artificial bee colony, we regularize our neural networks via balancing the incentive function based on two objectives (i.e. estimation accuracy and empirically realistic regime distributions). Balancing the incentive function prevents any single objective from leading to extreme outputs and hence balances the computational power of the trained neural network in desirable directions. In fact, our results show that amendments to the incentive function could be the strongest tool available to us in designing and implementing the neural networks.

Fourth, we also make a contribution to the current literature on Value at Risk (VaR) estimation (Borer et al., 2023; Pourkhanali et al., 2023; Wang et al., 2024). Our paper focuses on advancing machine learning techniques and is anchored in a regime-based portfolio asset allocation setting to account for time-varying economic states that has been considered as an essential element before feeding them into a machine learning model (Avramov et al., 2023; Chen et al., 2024). We believe that the nonlinearity and flexible form especially of recurrent neural networks (RNNs) draws interests in the broader VaR risk forecasting community (Lajili et al., 2024; Simlai, 2021).

Our final contribution lies in documenting the limitation of applying the neural networks to finance. While Avramov et al. (2023) point out the limitation of neural networks under economic conditions that limit arbitrage opportunity, we document how the neural network approach requires extensive historical data to advance the VaR thresholds estimation. Such long historical data may not always be available in practice when estimating VaR thresholds for asset management. Therefore, we expect established methods (e.g., Hidden Markov Model) and neural networks are more likely to be used in parallel for the foreseeable future.

The rest of this paper is organized as follow. In section two, we will describe our testing methodology including all five competing models (i.e. Mean/Variance, Hidden Markov Model, Feed Forward Neural Network, Convolutional Neural Network, Recurrent Neural Network). Section three describes data, model training, Monte Carlo simulations and baseline results. Section four then advances our neural networks via initialization and balancing the incentive functions and discusses the results of both features. Section five conducts robustness tests and sensitivity analyses before the conclusion in section six.

2 Background and Related Work

2.1 Classic Approach

When modelling financial time series related to investment decisions the asset log return $R_{t,p}$ of portfolio p at time t as defined in equation (1) below is the focal point of interest instead of asset price $P_{t,p}$, since investors earn on the difference between the prices at which they bought ($P_{t-1,p}$), and sold ($P_{t,p}$).

$$R_{t,p} = \ln\left(\frac{P_{t,p} - P_{t-1,p}}{P_{t-1,p}}\right) \tag{1}$$

Value-at-Risk (VaR) metrics are an important tool in many areas of risk management. Our particular focus on VaR measures as a means to perform risk budgeting in portfolio asset allocation (Migliavacca et al., 2023; Pearson, 2011). Asset managers in pension funds or insurance companies often incorporate VaR measures into their investment strategies (Jorion, 2006). Value at Risk (VaR) is defined as in equation (2) as the lower bound of a portfolio's return, which the portfolio return is not expected to fall below this lower bound with a certain probability α within the next period of allocation n.

$$Pr(R_{t+n,p} < VaR_{t,p}(n)) = \alpha \tag{2}$$

The standard practice in an investment fund is to indicate that, based on the composition of its portfolio and on current market conditions, there is a 95% probability (i.e., referring to 5% threshold) or 99% probability (i.e., referring to 1% threshold) that it will not lose more than a specified amount of assets over the next 5 trading days (Pearson, 2011). If the actual portfolio or asset return falls below this threshold, we refer to this as a *VaR breach* (i.e., either a breach of 5% threshold or a breach of 1% threshold).

Methods of estimating VaR could be divided into three categories: the parametric, the non-parametric, and the semi-parametric approaches, as outlined in Abad et al. (2014). The simplest of the non-parametric approaches would be historical simulation, where VaR is estimated directly from the empirical quantiles of the historical returns data, without any assumptions on the distribution of the (log) returns. The simplest of the parametric approaches is the Variance-Covariance method, where the (log) returns are assumed to follow a specific distribution function, for example a (multivariate) normal distribution. VaR thresholds can then be measured by estimating the mean and covariance (μ, Σ) of the asset log returns by calculating sample mean and sample covariance of the respective historical window. The 5% or 1% threshold of the resulting normal distribution will be an appropriate estimator of the 95% or 99% VaR probability.

As the baseline of our evaluation, we have chosen a semi-parametric approach since it is based on the normal distribution Variance-Covariance method, but we use the Monte-Carlo simulation to sample from the distribution with parameters estimated from the daily log return data[3]. We refer to this Mean/Variance way of estimating VaR thresholds as the "normal distribution Mean/Variance approach", or Classic approach. This Mean/Variance (Classic) approach, however, does not sufficiently reflect the skewness of real-world equity markets and the divergences of return distributions and dynamic trading across different economics regimes (i.e., bull and bear markets) (Fletcher, 2024). For this purpose, regime switching models have grown

[3]We have chosen this approach as a baseline since our neural network-based approaches are also based on Monte-Carlo simulation. Furthermore, Monte-Carlo simulation is a reliable way of estimating 5-day VaR from daily log returns, whilst getting the most out of the daily log return data. We consider the Historical Simulation approach unsuitable for estimation of 5-day VaR from daily data, as doing so would require a conversion from daily log return data to 5-daily, significantly reducing the size of the available data set.

in popularity well before machine learning entered finance (Billio and Pelizzon, 2000; Moody and Wu, 1997). In this study, we model financial markets inter alia using neural networks while accounting for shifts in economics regimes (Avramov et al., 2023; Chen et al., 2024). Due to the generative nature of these networks, they are able to perform Monte-Carlo simulation of future log returns, which could be beneficial for VaR estimation.

2.2 Hidden Markov Models

In risk budgeting, it is important for asset managers to know about which current market phase (which regime) and estimate the probability that the regime changes (Schmeding et al., 2019). The most common way of modelling market regimes is by distinguishing between bull markets and bear markets. Unfortunately, market regimes are not directly observable, but are rather derived indirectly from market data. The Hidden Markov Model (HMM) is an established tool for regime-swiching based financial modelling (MacLean and Zhao, 2022; Yousefi and Najand, 2022; Chen and Lin, 2022). The HMM is developed based on Markov chains that allow asset managers to analyse and to predict the characteristics of time series portfolio returns with a negative skewness and regime changes (Ang and Bekaert, 2002; Timmermann, 2000). We employ the HMM for the special case of two economic states called 'regimes' in the HMM context.

We model asset log returns $y_t \in R^n$ (we are looking at $n \geq 1$ assets) at time t to follow an n-dimensional Gaussian process with hidden states $S \in \{1,\ 2\}$ as shown in equation (3):

$$y_t \sim \mathcal{N}(\mu_{S_t}, \Sigma_{S_t}) \tag{3}$$

The returns are modelled to have state dependent expected log returns $\mu_{S_t} \in \mathbb{R}^n$ as well as covariance $\Sigma_{S_t} \in \mathbb{R}^{nxn}$. The dynamic of S_t is following a homogenous Markov chain with transition probability matrix as shown in equation (4):

$$A = \begin{pmatrix} p & 1-p \\ 1-q & q \end{pmatrix}, 0 \leq p, q \leq 1 \tag{4}$$

with $p = P(S_t = 1 \mid S_{t-1} = 1)$ and $q = P(S_t = 2 \mid S_{t-1} = 2)$. This definition describes if and how states are changing over time. It is also important to note in the 'Markov Property' that the probability of being in any state for the next point in time only depends on the present state, not the sequence of states that preceded it. Furthermore, the probability of being in a state at a certain point in time is given as $\pi_t = P(S_t = 1)$ and $(1 - \pi_t) = P(S_t = 2)$. This is also called the smoothed-state probability. By estimating the smoothed probability π_t of the last element of the historical window as the present regime probability, we can use the model to start from there and perform Monte-Carlo simulations of future asset log returns for the next l days[4]. This is outlined for the two-regimes case in the algorithm 1 below[5].

Algorithm 1 Hidden Markov Monte-Carlo simulation (for two regimes)

Estimate $\phi = (\pi_0, A, \mu, \Sigma)$ from history X_T
$p_0 \leftarrow \pi_T$ ▷ Compute smoothed (regime) probability
$s_0 \leftarrow S'_t \sim Bernoulli(p_0) + 1 \in \{1, 2\}$ ▷ draw first regime from Bernoulli distribution conditioned by p_0
for $i \in \{1, ..., l\}$ **do**
    $p_i \leftarrow A_{s_{i-1}0}$ ▷ Determine transition probability from previous regime
    $s_i \leftarrow S'_{t+i} \sim Bernoulli(p_i) + 1$ ▷ draw next regime
    $X'_{t+i} \sim \mathcal{N}(\mu_{s_i}, \Sigma_{s_i})$ ▷ draw return sample from regime's Gaussian
end for

2.3 Neural Network Models

Using a Mixture Density Network (MDN) to parametrize a Gaussian mixture predictive distribution (Bishop et al., 1995), Graves (2013) successfully uses a Long Short-Term Memory (LSTM) based recurrent neural network (RNN) to generate realistic time-series sequences of handwriting. Compared to standard feed forward neural networks (Multi-Layer Perceptron) as used by GKX (Gu et al., 2020), LSTM does not only predict the conditional average of the target variable as point estimate (in GKX's study is the expected risk premia), but it also estimates the conditional distribution of the target variable. Given the autoregressive nature of Graves' approach in LSTM, the output distributions are not assumed to be static over time, but dynamically conditioned on previous outputs, thus capturing the temporal context of the data. We consider both characteristics as being beneficial for modelling financial market returns, which generally exhibit a low signal to noise ratio in asset pricing problems, as highlighted by GKX, due to inherently high levels of intertemporal uncertainty.

The core of the proposed neural network regime switching framework is a swappable neural network architecture, which takes as input the historical sequence of daily asset log returns. At the output level, the framework computes regime probabilities and provides learnable Gaussian mixture distribution parameters, which can be used to sample new asset log returns for Monte-Carlo simulation. A multivariate Gaussian mixture model (GMM) is a weighted sum of k different components, each following a distinct multivariate normal distribution as shown in equations (5) and (6):

$$p_{GMM}(X) = \sum_{i=0}^{k} \phi_i \mathcal{N}(X \mid \mu_i, \Sigma_i) \tag{5}$$

[4]The parameter of the model $\phi = (\pi_0, A, \mu, \Sigma)$ can be estimated based on historical return data of some window size. This is done by using the Baum-Welch algorithm, which is an expectation-maximization (EM) algorithm.

[5]It is worth noting that for the one regime case, the algorithm estimates a standard multivariate Gaussian distribution and therefore mimics the classic Mean/Variance method.

$$\sum_{i=0}^{k} \phi_i = 1 \tag{6}$$

A GMM by its nature does not assume a single normal distribution, but naturally models a random variable as being the interleave of different (multivariate) normal distributions. In our model, we interpret k as the number of regimes and ϕ_i explains how much each regime contributes to the (current output). In other words, ϕ_i can be seen as the probability that we are in regime i. In this sense, the GMM output provides a suitable level of interpretability for the use case of regime-based modelling. With regard to the neural network regime switching model, we extend the notion of a Gaussian mixture by conditioning ϕ_i via a yet undefined neural network f on the historic asset log returns within a certain window of a certain size. We call this window *receptive field* and denote its size by r in equation (7):

$$\phi_i(t) = f(X^{t-r,t}) = p(\phi_i \mid X_{t-r}, ..., X_t) \tag{7}$$

This extension makes the Gaussian mixture weights dependent on the (recent) history of the time varying asset log returns. Note that we only condition ϕ on the historical log returns. The other parameters of the Gaussian mixture (μ_i, Σ_i), are modelled as unconditioned, yet optimizable parameters of the model. This basically means we assume the parameters of the Gaussians to be constant over time (per regime). This is in contrast to the standard MDN, where (μ_i, Σ_i) are also conditioned on X and therefore can change over time[6]. Keeping these remaining parameters unconditional is crucial to allow for a fair comparison between the neural networks and the HMM, which also exhibits time invariant parameters (μ_i, Σ_i) in its regime shift probabilities. Following Graves (2013), we define the probability given by the network and the corresponding sequence as shown in equations (8) and (9), respectively:

$$P(X) = \prod_{t=1}^{T} P(X_{t+1} \mid \phi(t), \mu, \Sigma) \tag{8}$$

$$\mathcal{L}(X) = -\sum_{t=1}^{T} \log P(X_{t+1} \mid \phi(t), \mu, \Sigma) \tag{9}$$

Since financial markets operate in weekly cycles with many investors shying away from exposure to substantial leverage during the illiquid weekend period, we are not surprised to observe that model training in machine learning is more stable when choosing the predictive distribution to not only be responsible for the next day, but for the next 5 days (Hann and Steurer, 1996). We call this forward looking window the *lookahead*. This

[6]We found this structure to be hard to interpret from a regime modelling point of view: If the regime distributions change at each point of time, it is hard to infer whether a regime describes a bullish or bearish market in general. With fixed optimizable distribution parameters we can interpret the distribution as belonging to either a bull market regime (positive expected return, low volatility) or bear market regime (negative expected return, high volatility).

is also practically aligned with the overall investment process, in which we want to appropriately model the upcoming allocation period, which usually spans multiple days. It also fits with the intuition that regimes do not switch daily but have a stability for at least a week. The extended sequence probability and sequence loss are denoted accordingly in equations (10) and (11):

$$P(X) = \prod_{t=1}^{T} \prod_{j=0}^{5} P(X_{t+j} \mid \phi(t), \mu, \Sigma) \tag{10}$$

$$\mathcal{L}(X) = -\sum_{t=1}^{T} \sum_{j=0}^{5} \log P(X_{t+j} \mid \phi(t), \mu, \Sigma) \tag{11}$$

An important feature of the neural network regime model is how it simulates future log returns. We follow Graves' approach and conduct sequential sampling from the network. When we want to simulate a path of log returns for the next N business days, we do this according to algorithm 2.

Algorithm 2 Neural Network Regime Switching Model - Monte-Carlo simulation (2-regime case)

```
Train model on history X_T
for i ∈ {0, ..., l − 1} do
    φ(t + i) ← φ(X^{t−r+i,t+i})                              ▷ Apply model on receptive field
    r_t ← R_t ∼ Bernoulli(φ_0(t)) + 1 ∈ {1, 2}    ▷ draw sample regime as conditioned by a Bernoulli distribution
    X'_{t+i+1} ← X_{t+i+1} ∼ N(μ_{r_i}, Σ_{r_i})             ▷ draw asset return sample from regime's Gaussian
    Append X'_{t+1} to X
end for
```

We display our research design in Figure 1 'Neural Network Regime Switching Model'. The network takes as input the most recent days (of receptive field size r). The Temporal Neural Network block is interchangeable (e.g. simple feed forward, convolutional neural network or CNN, LSTM). The network conditions are the regime probabilities of the Gaussian Mixture Model (GMM). The residual parameters are unconditioned, learnable parameters of the model. The network is trained by targeting the next 5 days. This is called $lookahead$. As building block for the temporal neural network part of the model we choose three different neural network architectures, which we introduce in the following sections.

2.3.1 Feed Forward Neural Networks (FF)

Following GKX (Gu et al., 2020), first, we conduct analysis using the traditional "feed-forward neural networks" (FF) before using more sophisticated neural network architectures for time series analysis within the neural network regime model. The traditional model of neural networks, also called Multi-Layer Perceptron, consists of an "input layer" which contains the raw input predictors and one or more "hidden layers" that combine input signals in a nonlinear way to produce an "output layer" that aggregates the output of the hidden layers into a final predictive signal. The nonlinearity of the hidden layers arises from

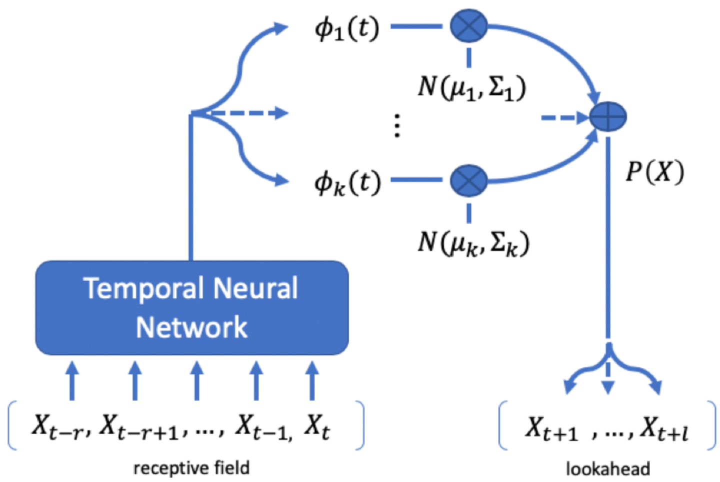


Figure 1: Neural Network Regime Switching Model

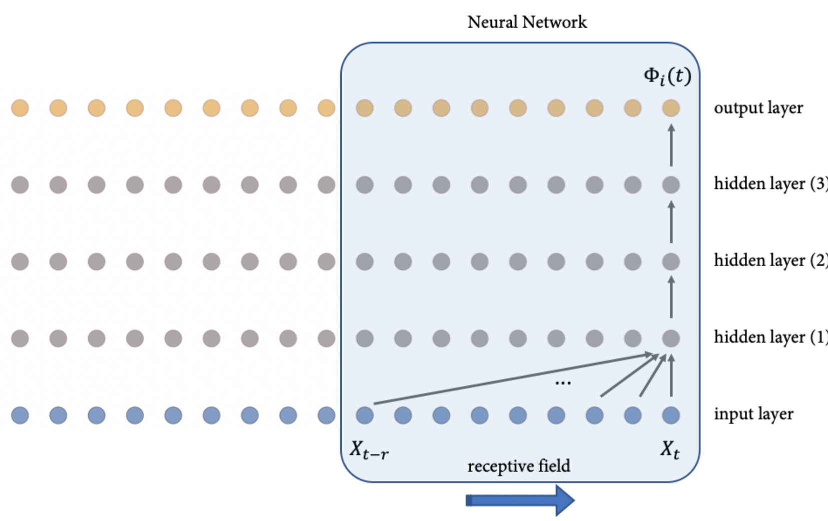


Figure 2: Feed Forward Neural Network

The figure depicts how the networks reads the data as a sequence from left to right. At a single point in time, the network takes as input the asset log returns of the last $N = 10$ days. In each layer, there is fixed number of hidden units (32, 16, 8) which are not visualized here. In between layers it uses tanh as activation function. The output layer aggregates the hidden layer output via SoftMax into regime probabilities.

the application of nonlinear "activation functions" on the combined signals. We visualize the traditional feed-forward neural network and its input layers in Figure 2.

2.3.2 Convolutional Neural Network (CNN)

The convolution neural network (CNN) model is a machine learning model that was originally developed for *an image recognition* (Oord et al., 2016). The CNN uses multiple layers, i.e., a small matrix of learnable parameters that slides over an image to extract features of the image by performing convolution operations,

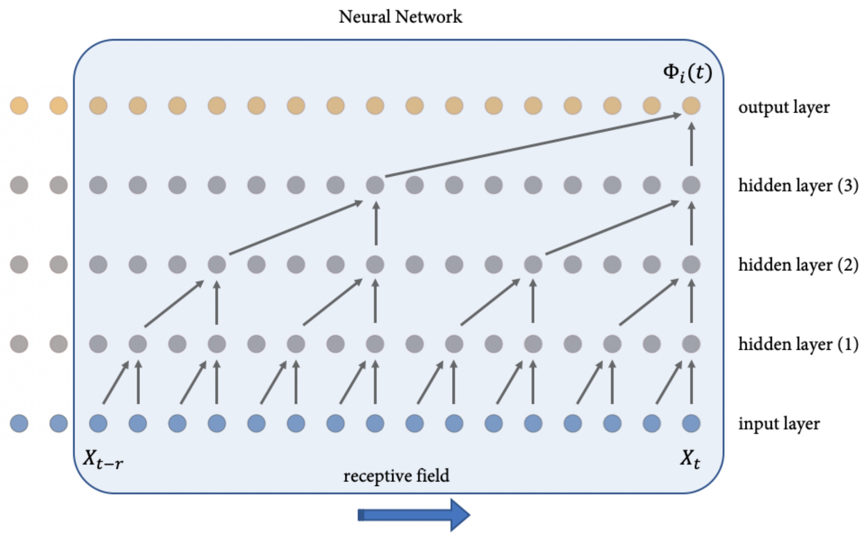


Figure 3: Visualization of Causal and Dilated Convolution as in Wavenet

The Convolutional Neural Network (CNN), due to its structure of stacked dilated convolutions, has a much greater receptive field than the simple feed forward network and needs much less weights to be trained. We restricted the number of hidden layers to 3 to illustrate the idea. Our network structure has 7 hidden layers. Each hidden layer furthermore exhibits a number of channels, which are not visualized here.

i.e., sliding a filter (kernel) across an image or data, multiplying the kernel's values with the corresponding data values, and summing the results to create a new value.

Convolutional Neural Network (CNN) can also be applied within the proposed neural network regime switching model. Recently, CNN gained popularity for time series analysis it was successfully applied convolutional neural networks on time series data for generating audio waveforms, the state-of-the-art text-to-speech and music generation by Oord et al. (2016). Their adaption of Convolutional Neural Networks - called WaveNet which takes a raw signal as an input and synthesises to produce an output one sample at a time. The WaveNet has been able to capture long ranging dependencies on sequences very well. In essence, a WaveNet consists of multiple layers of stacked convolutions along the time axis.

Crucial features of these convolutions are that they have to be causal and dilated. Causal means that the output of a convolution only depends on past elements of the input sequence. Dilated convolutions are ones that exhibit "holes" in their respective kernel, which effectively means that its filter size increases while being dilated with zeros in between. WaveNet typically is constructed with increasing dilation factor (doubling in size) in each hidden layer. By doing so, the model is capable of capturing an exponentially growing number of elements from the input sequence depending on the number of hidden convolutional layers in the network. The number of captured sequence elements is called the receptive field of the network (and in this sense is equal to the receptive field defined for the neural network regime model)[7]. Figure 3 illustrates the networks basic structure as a combination of stacked causal convolutions with a dilation factor of $D = 2$.

[7]Van den Oord et al. 2016 further describe activation functions and skip connections for WaveNet. For the interested reader, we kindly refer to their paper for an in-depth description.

Four features importance of the CNN are, (1) the use of kernels that detect cross-sectional patterns in an image, (2) identifying which cross-sectional parts of an image that CNN considers important in each layer, (3) assigning the degree of importance in each cross-sectional part of input images, and (4) masking cross-sectional parts of input image and examining how the predictive power change with the change in the input image.

In asset risk management setting, CNN can be used to analyze historical financial data to improve risk management decision-making by detecting volatility and liquidity and macroeconomic shocks that have happened in the past and to identify market regimes. CNNs can process historical price movements, sector correlations, and macroeconomic signals to determine which factors drive portfolio risk. CNN can analyze correlations between different asset classes (stocks, bonds, commodities). CNN enhances portfolio risk management by providing insights into risk drivers, asset correlations, market conditions, and anomalies. However, since CNN was originally developed to identify cross-sectional images, its features are most effective when it is applied to the cross-sectional data to identify which assets or sectors contribute the most to portfolio variance.

2.3.3 Long Short-Term Memory (LSTM)

Originally introduced by Hochreiter and Schmidhuber (1997), a main characteristic of Long Short-Term Memory (LSTM) - which are a subclass of Recurrent Neural Networks (RNN) is its purpose-built memory cells, which allows it to capture long range dependencies in the data. Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN). The RNN is an artificial intelligence neural network designed to process sequential (i.e., time-series) data where the order of elements matters allowing it to learn and predict the future based on the past inputs. LSTM was developed to address the vanishing gradient problem and its feature to enable better learning depends on long-term sequential (time-series) data feed (Hochreiter and Schmidhuber, 1997).

From a model perspective, LSTMs differ from other neural network architectures in that they are applied recurrently (see Figure 4). The output from a previous sequence of the network function serves - in combination with the next sequence element - as input for the next application of the network function. In this sense, the LSTM can be interpreted as being similar to an HMM, in that there is a hidden state which conditions the output distribution. However, the LSTM hidden state not only depends on its previous states, but it also captures long-term sequence dependencies through its recurrent nature. Most notably, the receptive field size of an LSTM is not bound architecture wise as in case of simple feed forward network and CNN. Instead, the LSTM's receptive field depends solely on the LSTMs ability to memorize the past input. The potential of LSTMs was noted by CPZ who note that "LSTMs are designed to find patterns in time series data and ... are among the most successful commercial AIs" (Chen et al., 2024, p. 7).

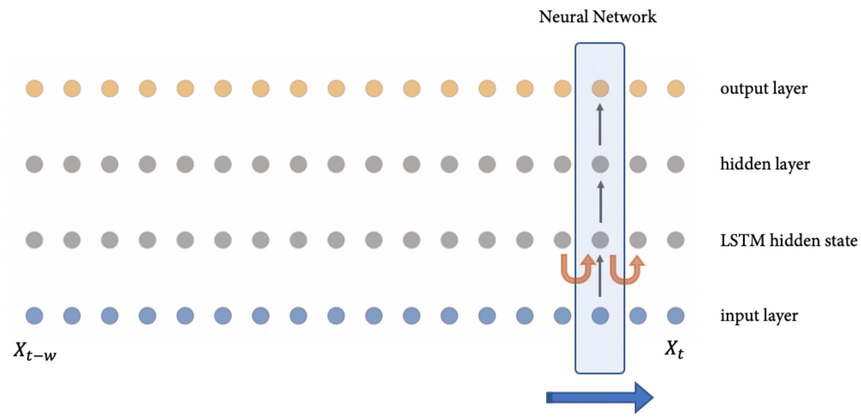


Figure 4: Dynamic of a Recurrent Neural Network

The Recurrent Neural Network (RNN) is characterized by its feedback loop: the output of a previous iteration of the function is used additionally as input when reading the next sequence element. Long Short-Term Memory (LSTM), which are a special variant of RNN, are able to capture long range sequence dependencies in this way.

Since LSTM excels at processing time-series (sequential) data, LSTM is suitable for predicting future stock prices or returns based on historical data. This enables portfolio managers to anticipate potential market movements and adjust their strategies accordingly. Using LSTM-predicted returns and risks, portfolio managers can optimize asset allocation to achieve a desired risk-return profile by dynamically rebalancing the portfolio based on changing market regimes. However, LSTM's capability to predict the future depends on the availability of long-term sequential data feed.

3 Experimental Design

3.1 Data

We obtain daily price data for World (global) stock and bond indices and three major global markets (i.e. EU, UK, US) to study the presented regime based neural network approaches on a variety of stock markets and long-term bond (maturity 7 to 10 years) markets. For each stock market, we focus on one major stock index. The markets in scope are (1) USA is represented by S&P 500 and US Treasury Bonds, (2) Europe (EU) is represented by EURO STOXX 50 and German Bundesanleihen, as well as (3) United Kingdom (UK) is represented vy FTSE 100 and UK government bonds. Furthermore, we also examine the World (Global) market using the MSCI World Index in cross-section to US Treasuries as being the most globally important government bonds. Each model is trained using equities, short-term bonds with maturities 1-year to 3-year (SB1-3y) and long-term bonds with maturities 7-year to 10-year (LB7-10y) as data feeds (i.e. $X_t \in \mathbb{R}^3$).

We use the data as early as January 1987 until mid-June 2020, when both bond and equity markets have bounced back from the Covid-19 crisis, which covers 30+ years of market development. Hence, the data also accounts for several major financial crises such as the dot-com bubble in the early 2000s as well as the 2008 global financial crisis and the Covid-19 crisis of 2020. These crises are especially important for

Panel A: Daily log returns

Region		US			EU			UK			Global	
Type	Equity	SB1-3y	LB7-10y	Equity	SB1-3y	LB7-10y	Equity	SB1-3y	LB7-10y	Equity	SB1-3y	LB7-10y
Start Date	05/01/88	05/01/88	05/01/88	02/01/90	02/01/90	02/01/90	02/01/89	02/01/89	02/01/89	02/01/87	02/01/87	02/01/87
End Date	16/06/20	16/06/20	16/06/20	16/06/20	16/06/20	16/06/20	16/06/20	16/06/20	16/06/20	16/06/20	16/06/20	16/06/20
Observations	8466	8466	8466	7946	7946	7946	8207	8207	8207	8615	8615	8615
Mean	0.0004	0.0002	0.0003	0.0002	0.0001	0.0002	0.0003	0.0002	0.0003	0.0003	0.0002	0.0003
Std. Dev.	0.0112	0.0009	0.0038	0.0133	0.0007	0.0029	0.0109	0.0010	0.0034	0.0095	0.0009	0.0039
Skewness	-0.4479	0.1663	-0.1299	-0.2650	0.0243	-0.4357	-0.2962	1.9675	0.1296	-0.7312	0.2948	-0.0336
Minimum	-0.1276	-0.0089	-0.0246	-0.1322	-0.0061	-0.0228	-0.1145	-0.0067	-0.0198	-0.1042	-0.0089	-0.0246
1%	-0.0313	-0.0022	-0.0102	-0.0395	-0.0018	-0.0081	-0.0311	-0.0025	-0.0089	-0.0271	-0.0022	-0.0103
10%	-0.0111	-0.0007	-0.0043	-0.0142	-0.0006	-0.0031	-0.0113	-0.0008	-0.0037	-0.0095	-0.0008	-0.0044
25%	-0.0040	-0.0003	-0.0019	-0.0057	-0.0002	-0.0013	-0.0049	-0.0003	-0.0016	-0.0038	-0.0003	-0.0019
5%	-0.0169	-0.0011	-0.0059	-0.0208	-0.0009	-0.0045	-0.0161	-0.0012	-0.0053	-0.0141	-0.0011	-0.0060
50%	0.0004	0.0001	0.0002	0.0005	0.0001	0.0003	0.0003	0.0002	0.0003	0.0006	0.0001	0.0003
75%	0.0054	0.0006	0.0025	0.0065	0.0005	0.0019	0.0059	0.0006	0.0022	0.0048	0.0006	0.0025
90%	0.0114	0.0011	0.0047	0.0138	0.0009	0.0035	0.0115	0.0012	0.0042	0.0096	0.0011	0.0047
95%	0.0160	0.0016	0.0061	0.0199	0.0013	0.0047	0.0163	0.0016	0.0056	0.0134	0.0016	0.0062
99%	0.0309	0.0027	0.0101	0.0349	0.0021	0.0072	0.0285	0.0029	0.0085	0.0245	0.0027	0.0103
Maximum	0.1096	0.0080	0.0346	0.1044	0.0065	0.0179	0.0938	0.0227	0.0306	0.0910	0.0082	0.0358

Panel B: Weekly log returns

Region		US			EU			UK			Global	
Type	Equity	SB1-3y	LB7-10y	Equity	SB1-3y	LB7-10y	Equity	SB1-3y	LB7-10y	Equity	SB1-3y	LB7-10y
Start Date	15/01/88	15/01/88	15/01/88	12/01/90	12/01/90	12/01/90	06/01/89	06/01/89	06/01/89	09/01/87	09/01/87	09/01/87
End Date	12/06/20	12/06/20	12/06/20	12/06/20	12/06/20	12/06/20	12/06/20	12/06/20	12/06/20	12/06/20	12/06/20	12/06/20
Observations	1692	1692	1692	1588	1588	1588	1641	1641	1641	1745	1745	1745
Mean	0.0019	0.0008	0.0013	0.0012	0.0007	0.0012	0.0015	0.0010	0.0014	0.0015	0.0008	0.0013
Std. Dev.	0.0232	0.0020	0.0085	0.0287	0.0017	0.0067	0.0233	0.0024	0.0079	0.0224	0.0021	0.0087
Skewness	-0.8836	0.2346	-0.3531	-0.9460	0.7340	-0.4425	-1.0267	1.0377	-0.0073	-1.2037	0.7458	-0.0786
Minimum	-0.2002	-0.0099	-0.0387	-0.2510	-0.0059	-0.0380	-0.2359	-0.0096	-0.0331	-0.2233	-0.0099	-0.0387
1%	-0.0686	-0.0043	-0.0226	-0.0745	-0.0039	-0.0166	-0.0603	-0.0049	-0.0193	-0.0629	-0.0043	-0.0225
10%	-0.0230	-0.0013	-0.0092	-0.0327	-0.0011	-0.0075	-0.0242	-0.0014	-0.0084	-0.0227	-0.0014	-0.0095
25%	-0.0097	-0.0003	-0.0038	-0.0137	-0.0002	-0.0026	-0.0110	-0.0002	-0.0032	-0.0094	-0.0003	-0.0039
5%	-0.0351	-0.0021	-0.0132	-0.0455	-0.0019	-0.0107	-0.0332	-0.0025	-0.0118	-0.0322	-0.0022	-0.0136
50%	0.0032	0.0006	0.0014	0.0033	0.0005	0.0016	0.0028	0.0008	0.0015	0.0029	0.0006	0.0014
75%	0.0143	0.0019	0.0070	0.0174	0.0016	0.0055	0.0141	0.0022	0.0061	0.0132	0.0019	0.0070
90%	0.0264	0.0033	0.0113	0.0319	0.0028	0.0090	0.0259	0.0036	0.0105	0.0246	0.0033	0.0113
95%	0.0353	0.0042	0.0140	0.0416	0.0036	0.0114	0.0344	0.0045	0.0131	0.0329	0.0043	0.0139
99%	0.0565	0.0063	0.0200	0.0715	0.0055	0.0164	0.0613	0.0078	0.0215	0.0527	0.0063	0.0205
Maximum	0.1146	0.0106	0.0357	0.1359	0.0160	0.0261	0.1262	0.0213	0.0436	0.1170	0.0215	0.0684

Table 1: Descriptive statistics of test assets

testing the advantages of regime-based approaches. The price indices are given as total return indices (i.e. dividends treated as being reinvested) to properly reflect market development. All data is collected from LSEG Datastream.

Descriptive statistics of data feeds are displayed in Table 1, whereby Panel A displays a daily frequency and Panel B a weekly frequency. Mean returns for equities exceed the mean returns for bonds whereby the returns for bonds with longer maturities exceed the returns of short-term bonds. Equities have naturally a much higher standard deviation and a far worse minimum return. In fact, equity returns in all four regions lose substantially more money than bond return even at the 25th percentile, which highlights that the holy grail of asset allocation depends on the ability to predict equity market drawdowns. Furthermore, equity markets tend to be quite negatively skewed as expected while short-term bonds experience a positive skewness, which reflects previous findings (Albuquerque, 2012; Kozhan et al., 2013) and the inherent differential in the riskiness of both asset's payoffs (Mishra et al., 2022).

3.2 Parameter Settings

The back testing is done on a weekly basis via a moving window approach. At each point in time, the respective model is fitted by providing the last 2,000 days (which is roughly 8 years) as training data. We choose this long-range window, because neural networks are known to need big datasets as inputs and it is reasonable to assume that over eight years include simultaneously times of at least a relative crisis and times of a market growth. The training sample covers both bull and bear markets, which is crucial to allow the models to "learn" these types of regimes. For all our models we set the number of regimes to $k = 2$.

After training the model for a specific point in time, we start a Monte Carlo simulation of asset returns for the next 5 days (one week - Monday to Friday). For the purpose of calculating statistically solid quantiles of the resulting distribution, we simulate 100,000 paths for each model. We do this for at the least 1,186 (EU), and at the most 1,343 (World) weeks within the back-test history window. As soon as we have simulated all return paths, we calculate a total (weekly) return for each path. The generated weekly returns follow a non-trivial distribution, which arises from the respective model and its underlying temporal dynamics. Based on the simulations we compute quantiles for Value at Risk estimations. For example, the 0.01 and 0.05 percentile of the resulting distribution represent the 99% and 95% - 5-day - VaR metric, respectively.

Additionally, we set the lookahead parameter for the neural network models under consideration (FF, CNN, and LSTM) to $l = 5$, as we back test an allocation strategy with a weekly re-allocation. We configured the back-testing dates to consistently align with the end of the business week (i.e., Fridays). The neural network models under considerations also include a data normalization step, as this is required to learn network weights that lead to meaningful regime probabilities and distribution parameters. The normalization step is also considered good practice for neural network training (Bishop et al., 1995). Here, we normalize the input data as follows: $X' = (X - mean(X))/var(X)$. This process demeans the input data and scales them by their variance while preserving the interactions between assets.

Finally, the neural network models under consideration employ the AdaMax optimization algorithm (Kingma and Ba, 2014) while simultaneously applying weight decay[8] to mitigate overfitting (Krogh and Hertz, 1992). The learning rate and the number of training epochs[9] vary depending on the model and were heuristically adjusted to achieve optimal performance. Detailed specifications of these and other parameters for the five considered methods (Classic Approach, Hidden Markov Models, FF, CNN, and LSTM) are provided below.

1. **Classic (Mean/Variance) Approach.** This method requires no configuration, as model fitting is equivalent to computing the sample mean and sample covariance of asset returns within the respective window.

[8]Weight decay is a regularization method that reduces overfitting by penalizing large weights through the loss function.
[9]An epoch is a hyperparameter defining one full pass over the training data, during which model parameters are updated.

2. **Hidden Markov Model.** This method also requires no additional configuration, as the Baum-Welch algorithm is guaranteed to converge the parameters to a local optimum with respect to the likelihood function (Baum et al., 1970).

3. **FF.** We configure our network architecture in accordance with GKX's highest-performing neural network, 'NN3' (Gu et al., 2020), which consists of three hidden layers. Specifically, our network comprises three hidden layers with a decreasing number of units (32, 16, and 8). To capture the temporal dependencies in our time series data, we condition the network output on a receptive field of at least 10 days. Although the receptive field is relatively small in this case, the network's dense structure results in a high number of parameters—1698 in total, including those associated with the Gaussian Mixture Model (GMM). We employ the $tanh$ activation function between layers. The learning rate is set to 0.1, and the network is trained for 3000 epochs.

4. **CNN.** The backing model presented in this investigation is inspired by WaveNet (Oord et al., 2016). We restrict the model to the basic layout, using causal structure and increasing dilation between layers. The output layer comprises the regime predictive distributions by applying a SoftMax function to the hidden layers' outputs. Our network consists of seven hidden layers, each containing three channels. The convolutions each have a kernel size of three. In total, the network has 242 weights, including those associated with the GMM parameters, and the receptive field spans 255 days. The learning rate is set to 0.02, and the network is trained for 3000 epochs.

5. **LSTM.** As Graves (2013) successfully applied Long Short-Term Memory (LSTM) networks for sequence generation, we also adopt this approach for the neural network regime-switching model. In our architecture, we use a single LSTM layer with a hidden state size of 5. In total, the model has 236 parameters, including those associated with the GMM parameters. The learning rate is set to 0.001, and the network is trained for 1000 epochs.

Table 2 summarizes the configuration of the considered methods. The parameters were fine-tuned based on the preceding analysis to ensure optimal performance, with adjustments made to accommodate the specific characteristics and requirements of each method.

Parameter	**Classic**	**HMM**	**FF**	**CNN**	**LSTM**
Regimes	-	2	2	2	2
Lookahead	-	-	5	5	5
Optimization Algorithm	-	-	*AdaMax*	*AdaMax*	*AdaMax*
Hidden Layers	-	-	3	7	1
Learning Rate	-	-	0.1	0.02	0.001
Epochs	-	-	3000	3000	1000

Table 2: Parameter setting of compared methods.

3.3 Model Training

In general, estimating parameters of a neural network model is a non-convex optimization problem. Thus, the optimization algorithm might become stuck in an infeasible local optimum. In order to mitigate this problem, it is common practice to repeat the training multiple times, starting off having different (usually randomly chosen) parameter initializations, and then averaging over the resulting models or picking the best in terms of loss. In this paper, we follow a best-out-of-five approach, that means each training is done five times with varying initialization and the best one is selected for simulation. The initialization strategy, which we will show in section 4.1, further mitigates this problem by starting off from an economically reasonable parameter set.

We observe that the in-sample regime probabilities learned by the neural network regime switching models generally show comparable results in terms of distribution and temporal dynamics as compared to those estimated by the HMM based regime switching model. When we set $k = 2$ and the model fits two regimes with nearly invariably one having a positive corresponding equity means and low volatility, and the other experiencing a low or negative equity mean and high volatility. These regimes can be interpreted as bull and bear market, respectively. The respective in-sample regime probabilities over time also show strong alignment with growth and drawdown phases. This holds true for the vast majority of seeds and hence indicates that the neural network regime model is a valid practical alternative for regime modelling when compared to a Hidden Markov Model.

3.4 Simulation

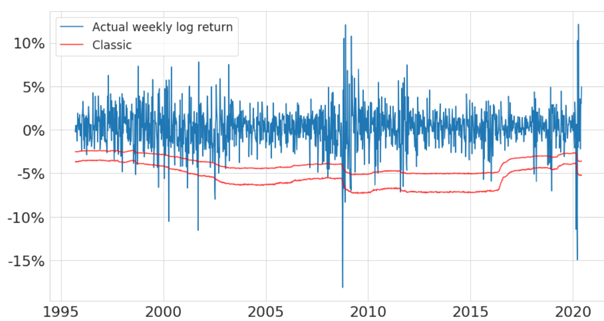
We evaluate the quality of our Value at Risk estimations by counting the number of breaches of the asset returns. When the actual return is below the estimated VaR threshold, we count this as a breach. Assuming an average performing model, it is e.g. reasonable to expect 5% breaches for a 95% VaR measurement. We compared the breaches of all models with each other. We classify a model as being superior to another model, if the number of VaR breaches is less than those from the compared model. A value comparison or comp = 1.0 or = 0.0 indicates that the row model is superior or inferior to the column model. We performed significance tests by applying paired t-tests on the value comparison (comp). We further evaluated a dominance value which is defined as shown in equation 12:

$$\mathrm{dom}(\mathrm{model}_1, \mathrm{model}_2) = \frac{|\mathrm{breaches}(\mathrm{model}_1) \cap \mathrm{breaches}(\mathrm{model}_2)|}{|\mathrm{breaches}(\mathrm{model}_1)|} - 1 \qquad (12)$$

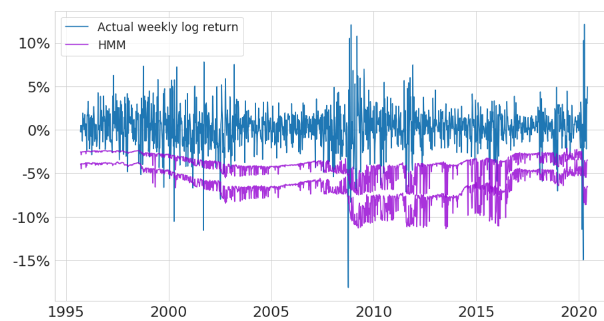
A value dominance or dom = 0.0 means that any VaR breach that occurs with model_1 also occurs with model_2. A negative value of dominance (dom) means that model_1 exhibits VaR breaches that do not occur with model_2. In that sense, model_2 is dominating model_1 for this particular breach. The lower the dom value the more cases in which model_1 has a breach that model_2 does not share. If the dom value is zero and model_1

wins the comparison against model_2 (i.e. comp = 1.0), then model_1 fully dominates model_2, as it has less breaches and every time it breaches, model_2 does too.

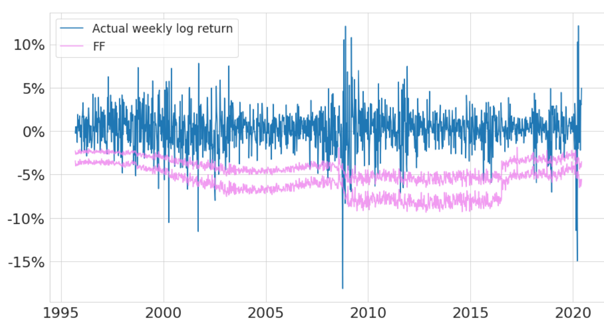
In Figure 5 we plot the VaR calculated using different methods (the Classic, HMM, FF, CNN, and LSTM methods) compared against the actual log returns of the S&P 500 index. Figure 5 shows some evidence that other methods (HMM, FF, CNN and LSTM) have less breaches than the classic (Mean/Variance) model. In Figures 6, 7 and 8 we similarly plot VaR for the EU and UK regions, and World VaR, compared against Eurostoxx 50, FTSE 100, and MSCI World respectively and also show some evidence that other methods have less breaches than the classic model.



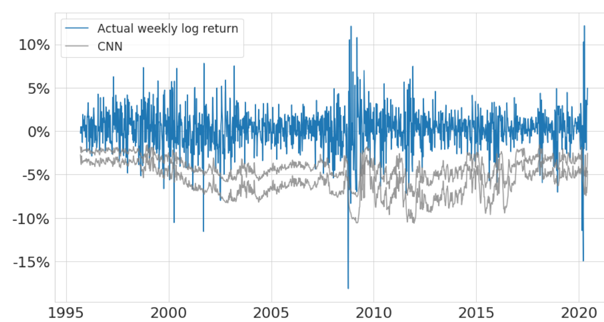
(a) Classic method VaR.



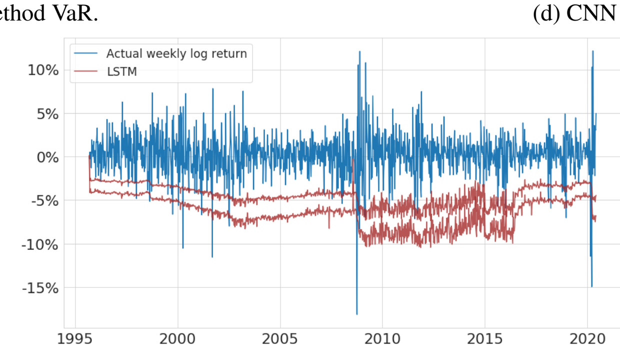
(b) HMM method VaR.



(c) FF method VaR.

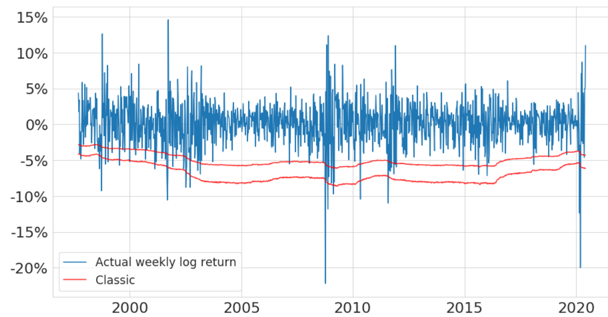


(d) CNN method VaR.

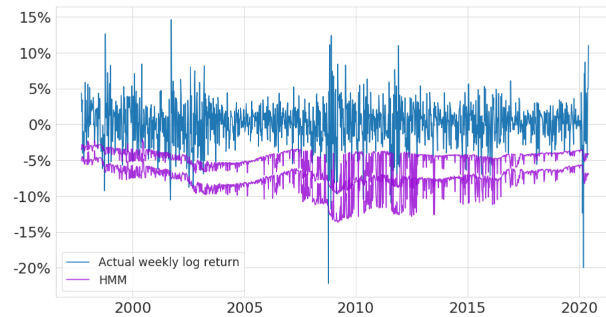


(e) LSTM method VaR.

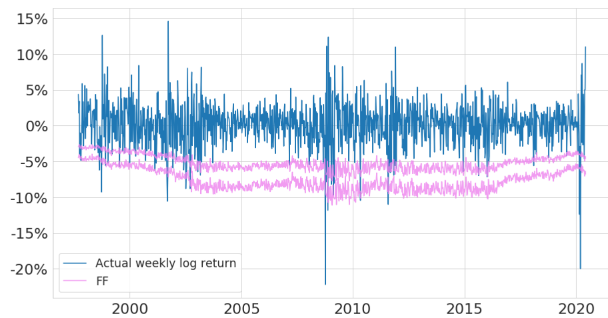
Figure 5: VaR for the US region, comparing the VaR predicted by the respective models when trained on a combined dataset of US data on Long Bonds, Short Bonds, and Equity to the actual log returns of S&P 500.



(a) Classic method VaR.

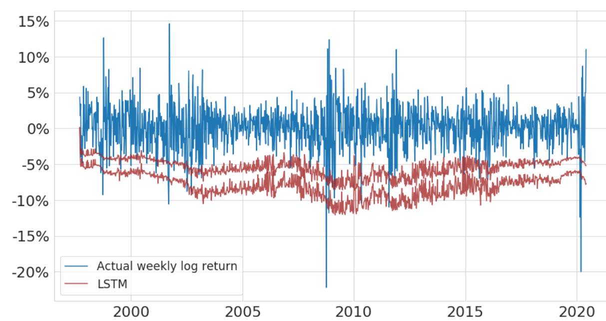


(b) HMM method VaR.



(c) FF method VaR.

(d) CNN method VaR.



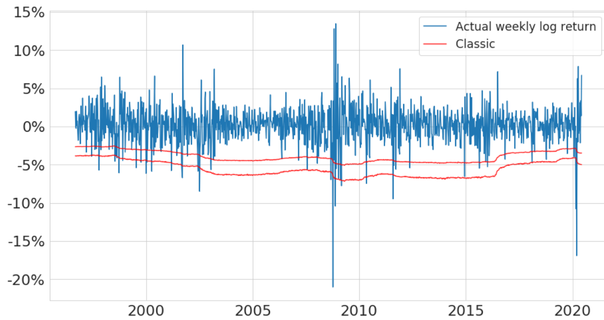
(e) LSTM method VaR.

Figure 6: VaR predicted by different models trained on an EU data set, compared to the log returns of EuroStoxx 50.

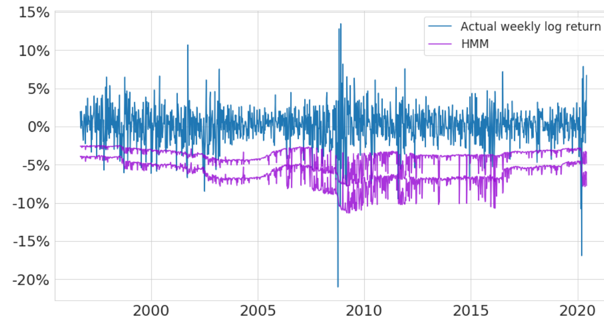
4 Discussion of Results by Design Feature

The three most crucial design features of neural networks in finance, where the sheer number of hidden layers appear less helpful due to the low signal to noise ratio (Gu et al., 2020) are: (i) amount of input data, (ii) initializing information, and (iii) incentive function.

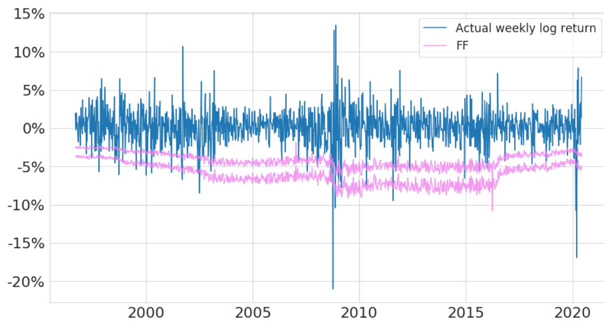
Big input data is important for neural networks, as they need to consume sufficient evidence also of rarer empirical features to ensure that their nonlinear abilities in fitting virtually any functional form are used in a relevant instead of an exotic manner. Similarly, the initialization of input parameters should be based on empirically established estimates to ensure that the gradient descent inside the neural network takes off from a suitable point of departure, thereby substantially reducing the risks that a neural network confuses itself into irrelevant local minima.



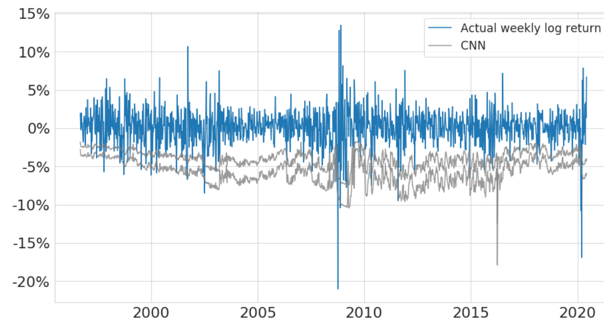
(a) Classic method VaR.



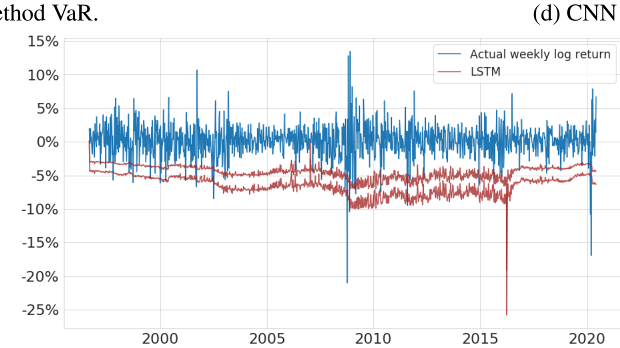
(b) HMM method VaR.



(c) FF method VaR.



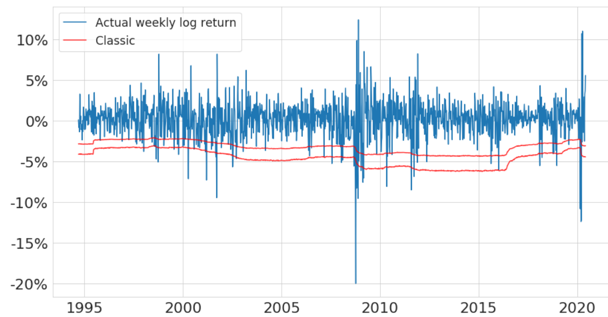
(d) CNN method VaR.



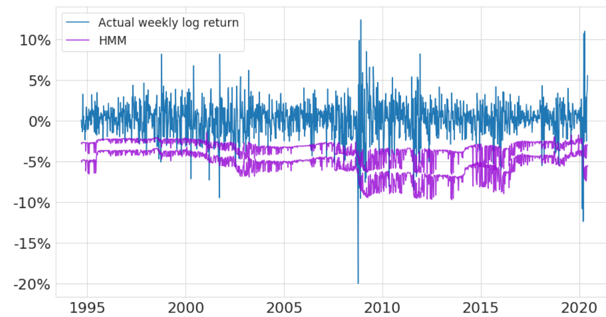
(e) LSTM method VaR.

Figure 7: VaR predicted by different models trained on a UK data set, compared to the log returns of FTSE 100.

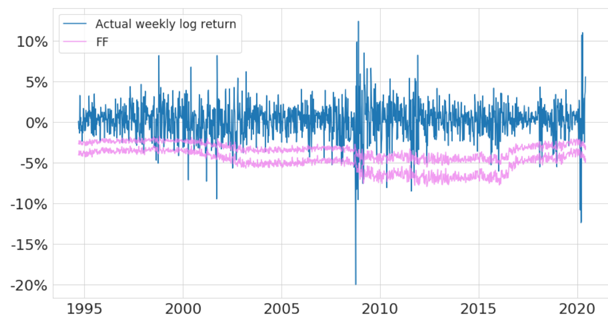
On the output side, every neural network is trained according to an incentive or a loss function. It is this particular loss function which determines the direction of travel for the neural network, which has no other ambitions than to minimize its loss best possible. Hence, if the loss function only represents one of several practically relevant parameters, the neural network may come to results with bizarre outcomes for those parameters not included in its incentive function. In our case, for instance, the baseline incentive is just estimation accuracy which could lead to forecasts dominated by a single regime (a bull or a bear) than realistically observed in practice. In other words, after a long bull market, the neural network could "conclude" that bear markets do not exist. Metaphorically speaking, a unidimensional loss function in a neural network has little decency (Marcus, 2018). Commencing with the initialization and the incentive functions, we will assess our three neural networks in the following vis a vis classic and HMM approach,



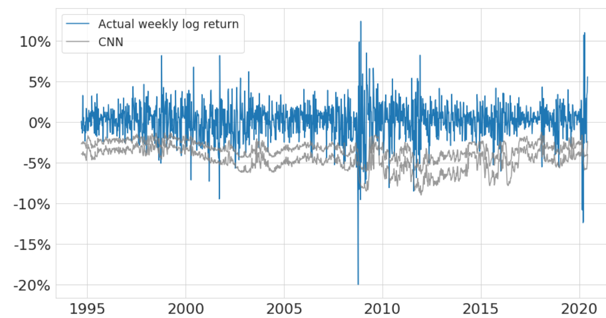
(a) Classic method VaR.



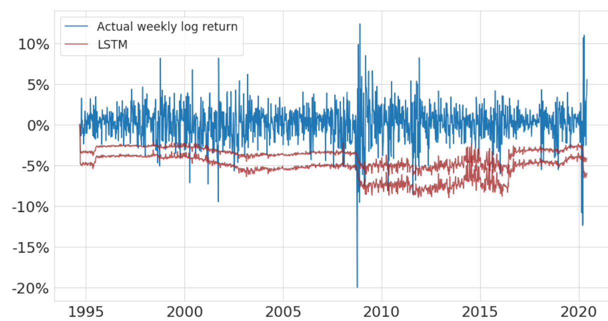
(b) HMM method VaR.



(c) FF method VaR.



(d) CNN method VaR.



(e) LSTM method VaR.

Figure 8: VaR predicted by different models trained on a global data set, compared to the log returns of MSCI World.

where each of the three networks is once displayed with an advanced design feature and once with a naive design feature.

4.1 Initialization

Under no specific initialization strategy, the initiation in the neural networks occurs entirely random via a computer-generated random number. Where established econometric approaches use naive priors (i.e. mean), neural networks originally relied on brute force computing power and a bit of luck. Hence, it is unsurprising that initializations are a common research topic in core machine learning fields such as image classification or machine translation (Glorot and Bengio, 2010; Zhang et al., 2019). However, we are not aware of any systematic application of initialized neural networks in the field of finance. Hence, we compare naive neural networks where initialization is completely random with neural networks that have been initialized with the best available prior. In our case, the best available prior for μ_i, Σ_i is the equivalent HMM estimation based on the same window [10]. Such initialization is feasible, since the structure of the Neural Network, due to its similarity with respect to μ_i, Σ_i - is broadly comparable with the HMM. In other words, we make use of already trained parameters from HMM training as starting parameters for the Neural Network training. In this sense, initialized neural networks are not only flexible in their functional form, they are also adaptable to "learn" from the best-established model in the field if suitably supervised by the human data scientists. metaphorically speaking, our neural networks can stand on the shoulders of the HMM giant for regime-based estimations.

Table 3 presents the results by comparing breaches between the two classic approaches (Mean/Variance and HMM) and the non-initialized and HMM initialized neural networks across all four regions. Panel A and B display the 1% VaR threshold for equities and long bonds, respectively, while Panels C and D show the equivalent comparison for 5% VaR threshold[11]. Note that for model training we apply a best-out-of-5 strategy as described in section 3.3. That means we repeat the training five times, starting off with random parameter initializations each time. In case of the presented HMM initialized model, we apply the same strategy, with the exception that μ_i, Σ_i of the model are initialized with HMM for each of the five neural networks iterations.

We observe three findings. First, not a single VaR threshold estimation process in a single region and in equity and bond asset classes was able to uphold its promise in that an estimated 1% VaR threshold should be breached no more than 1% of the time, except for Eurostoxx 50 under HMM (12 breaches out of 1186). This is quite alarming for institutional investors such as pension funds and insurance companies since it implies that all approaches - the established classic model and machine learning models - fail to sufficiently capture downside tail risks and hence underestimate 1% VaR thresholds.

Second, when inspecting the ability of eight methods to estimate 5% VaR thresholds, the result remains not as good as expected. The Mean/Variance approach, the HMM, and the LSTM with HMM initialization for equity markets display some cases where their VaR thresholds were breaches in less than the expected 5%. The Mean/Variance and HMM approach make their thresholds in 3 out of 8 cases and the HMM initialized LSTM in 2 out of 4 cases in equity markets. Overall, this is still a disappointing performance, especially for the LSTM with no initiation, the FF, and the CNN.

Third, comparing the initialized with the non-initialized neural networks, the non-initialized neural networks always perform worse. More importantly, LSTM results improved significantly with HMM initialization.

[10] Even though we initialize μ_i, Σ_i from HMM parameters, we still have weights to be initialized arising from the temporal Neural Network part of the model. We do this on a per layer level by sampling uniformly as $w \leftarrow \mathcal{U}(-\frac{1}{\sqrt{i}}, \frac{1}{\sqrt{i}})$ where i is the number of input units for this layer.

[11] We focus our discussion of results on the equities and long bonds since these have more variation, lower skewness and hence risk. Results for the short bonds are available upon request from the contact author.

Panel A: 1% VaR thresholds for Equity								
Index Model	S&P 500 breaches out of 1290		EuroStoxx 50 breaches out of 1186		FTSE 100 breaches out of 1239		MSCI World breaches out of 1343	
Classic	26	2.0%	20	1.7%	25	2.0%	40	3.0%
HMM	17	1.3%	12	1.0%	19	1.5%	22	1.6%
FF (no hmm)	31	2.4%	22	1.9%	29	2.3%	44	3.3%
CNN (no hmm)	33	2.6%	24	2.0%	32	2.6%	49	3.6%
LSTM (no hmm)	518	40.2%	475	40.1%	513	41.4%	539	40.1%
FF (hmm init)	20	1.6%	18	1.5%	23	1.9%	35	2.6%
CNN (hmm init)	26	2.0%	19	1.6%	24	1.9%	35	2.6%
LSTM (hmm init)	20	1.6%	17	1.4%	22	1.8%	28	2.1%
Panel B: 1% VaR thresholds for Long Bonds								
Index Model	US Long Bonds breaches out of 1290		German Long Bonds breaches out of 1186		UK Long Bonds breaches out of 1239		Global Long Bonds breaches out of 1343	
Classic	30	2.3%	26	2.2%	24	1.9%	30	2.2%
HMM	20	1.6%	19	1.6%	16	1.3%	22	1.6%
FF (no hmm)	51	4.0%	46	3.9%	44	3.6%	44	3.3%
CNN (no hmm)	60	4.7%	38	3.2%	41	3.3%	64	4.8%
LSTM (no hmm)	496	38.4%	388	32.7%	478	38.6%	563	41.9%
FF (hmm init)	32	2.5%	28	2.4%	27	2.2%	41	3.1%
CNN (hmm init)	34	2.6%	35	3.0%	31	2.5%	42	3.1%
LSTM (hmm init)	27	2.1%	21	1.8%	21	1.7%	32	2.4%
Panel C: 5% VaR thresholds for Equity								
Index Model	S&P 500 breaches out of 1290		EuroStoxx 50 breaches out of 1186		FTSE 100 breaches out of 1239		MSCI World breaches out of 1343	
Classic	64	5.0%	60	5.1%	58	4.7%	85	6.3%
HMM	61	4.7%	63	5.3%	54	4.4%	75	5.6%
FF (no hmm)	71	5.5%	62	5.2%	65	5.2%	94	7.0%
CNN (no hmm)	76	5.9%	71	6.0%	67	5.4%	102	7.6%
LSTM (no hmm)	534	41.4%	490	41.3%	535	43.2%	562	41.8%
FF (hmm init)	65	5.0%	62	5.2%	55	4.4%	84	6.3%
CNN (hmm init)	69	5.3%	77	6.5%	63	5.1%	96	7.1%
LSTM (hmm init)	62	4.8%	60	5.1%	52	4.2%	81	6.0%
Panel D: 5% VaR thresholds for Long Bonds								
Index Model	US Long Bonds breaches out of 1290		German Long Bonds breaches out of 1186		UK Long Bonds breaches out of 1239		Global Long Bonds breaches out of 1343	
Classic	71	5.5%	78	6.6%	69	5.6%	71	5.3%
HMM	75	5.8%	85	7.2%	72	5.8%	75	5.6%
FF (no hmm)	103	8.0%	116	9.8%	106	8.6%	98	7.3%
CNN (no hmm)	114	8.8%	98	8.3%	96	7.7%	118	8.8%
LSTM (no hmm)	550	42.6%	428	36.1%	519	41.9%	591	44.0%
FF (hmm init)	80	6.2%	83	7.0%	79	6.4%	88	6.6%
CNN (hmm init)	85	6.6%	104	8.8%	88	7.1%	96	7.1%
LSTM (hmm init)	81	6.3%	91	7.7%	73	5.9%	79	5.9%

Table 3: VaR breaches compared with and without initialization. Note that for the Global long bonds (panels B and D) we have considered US issued long bonds, and do so throughout.

When comparing across all eight approaches, the HMM appears most competitive which means that we either have to further advance the design of our neural networks or their marginal value add beyond classic econometric approaches appears inexistent. When comparing across asset classes, we find that the HMM initialization provides better outcomes for the equity markets compared to the bond markets in terms of lower percentages of breaches. Since the volatility of equity markets is usually greater than bond markets (see negative skewness of equity markets in Table 1), this finding implies that adding the HMM initialization instead of random feed improves the performance of neural network models, especially the LSTM model, to reduce catastrophic risks. Beyond the performace advantage of HMM initialization, it has a procedural advantage as it removes the need to fix the random seed to a constant integer (i.e., one) to ensure full replicability. We further advance the design of our neural networks with the aim of balancing its utility function to avoid extreme unrealistic results possible in the univariate case in the next section.

4.2 Balancing Incentive Functions

Whereas CPZ (Chen et al., 2024) regularize their neural networks via no arbitrage conditions, we regularize via balancing the incentive function of our neural networks on multiple objectives. Specifically, we extend the loss function to not only focus on accuracy of point estimates but also give some weight to eventually achieving empirically realistic regime distributions (i.e. in our data sample across all four regions no regimes display more than 60% frequency on a weekly basis). This balanced extension of the loss function prevents the neural networks from arriving at bizarre outcomes such as the conclusion that bear markets (or even bull markets) barely exist.

Technically, such bizarre outcomes result from cases where the regime probabilities $\phi_i(t)$ tend to converge globally either into 0 or 1 for all t, which basically means the neural network only recognizes one-regime. To balance the incentive function of the neural network and facilitate balancing between regime contributions, we introduced an additional regularization (reg(x)) into the loss function which penalizes unbalanced regime probabilities. The regularization term (reg(x)) is displayed in equation (13) below. If bear and bull market have equivalent regime probabilities the term (reg(x)) converges to 0.5. If there is imbalance between bull and bear, then it converges towards 1.

$$reg(x) = \sum_{i=1}^{k} \left(\frac{1}{T} \sum_{t=1}^{T} \phi_i(t) \right)^2 = \sum_{i=1}^{k} \bar{\phi}_i(t)^2 \tag{13}$$

Substituting equation (13) into our loss function of equation (11), leads to equation (14) below, which doubles the point estimation based standard loss function in case of total regime balance inaccuracy, but adds only 50% of the original loss function in case of full balance. Conditioning the extension of the loss function on its origin is important to avoid biases due to diverging scales. Setting the additional incentive function to initially have half the marginal weight of the original function also seems appropriate for comparability.

Panel A: 1% VaR thresholds for Equity

Index Model	S&P 500 breaches out of 1290		EuroStoxx 50 breaches out of 1186		FTSE 100 breaches out of 1239		MSCI World breaches out of 1343	
Classic	26	2.0%	20	1.7%	26	2.1%	39	2.9%
HMM	17	1.3%	12	1.0%	19	1.5%	22	1.6%
FF (hmm init)	20	1.6%	18	1.5%	23	1.9%	35	2.6%
CNN (hmm init)	26	2.0%	19	1.6%	24	1.9%	35	2.6%
LSTM (hmm init)	20	1.6%	17	1.4%	22	1.8%	28	2.1%
FF (hmm init + reg)	22	1.7%	18	1.5%	19	1.5%	30	2.2%
CNN (hmm init + reg)	21	1.6%	17	1.4%	21	1.7%	32	2.4%
LSTM (hmm init + reg)	18	1.4%	12	1.0%	15	1.2%	23	1.7%

Panel B: 1% VaR thresholds for Long Bonds

Index Model	US Long Bonds breaches out of 1290		German Long Bonds breaches out of 1186		UK Long Bonds breaches out of 1239		Global Long Bonds breaches out of 1343	
Classic	30	2.3%	26	2.2%	23	1.9%	31	2.3%
HMM	20	1.6%	19	1.6%	16	1.3%	22	1.6%
FF (hmm init)	32	2.5%	28	2.4%	27	2.2%	41	3.1%
CNN (hmm init)	34	2.6%	35	3.0%	31	2.5%	42	3.1%
LSTM (hmm init)	27	2.1%	21	1.8%	21	1.7%	32	2.4%
FF (hmm init + reg)	29	2.2%	24	2.0%	24	1.9%	37	2.8%
CNN (hmm init + reg)	29	2.2%	26	2.2%	29	2.3%	33	2.5%
LSTM (hmm init + reg)	14	1.1%	18	1.5%	16	1.3%	17	1.3%

Panel C: 5% VaR thresholds for Equity

Index Model	S&P 500 breaches out of 1290		EuroStoxx 50 breaches out of 1186		FTSE 100 breaches out of 1239		MSCI World breaches out of 1343	
Classic	63	4.9%	61	5.1%	60	4.8%	84	6.3%
HMM	61	4.7%	63	5.3%	54	4.4%	75	5.6%
FF (hmm init)	65	5.0%	62	5.2%	55	4.4%	84	6.3%
CNN (hmm init)	69	5.3%	77	6.5%	63	5.1%	96	7.1%
LSTM (hmm init)	62	4.8%	60	5.1%	52	4.2%	81	6.0%
FF (hmm init + reg)	63	4.9%	59	5.0%	58	4.7%	80	6.0%
CNN (hmm init + reg)	60	4.7%	54	4.6%	58	4.7%	90	6.7%
LSTM (hmm init + reg)	49	3.8%	44	3.7%	43	3.5%	63	4.7%

Panel D: 5% VaR thresholds for Long Bonds

Index Model	US Long Bonds breaches out of 1290		German Long Bonds breaches out of 1186		UK Long Bonds breaches out of 1239		Global Long Bonds breaches out of 1343	
Classic	72	5.6%	78	6.6%	70	5.6%	71	5.3%
HMM	75	5.8%	85	7.2%	72	5.8%	75	5.6%
FF (hmm init)	80	6.2%	83	7.0%	79	6.4%	88	6.6%
CNN (hmm init)	85	6.6%	104	8.8%	88	7.1%	96	7.1%
LSTM (hmm init)	81	6.3%	91	7.7%	73	5.9%	79	5.9%
FF (hmm init + reg)	79	6.1%	76	6.4%	70	5.6%	83	6.2%
CNN (hmm init + reg)	78	6.0%	98	8.3%	82	6.6%	83	6.2%
LSTM (hmm init + reg)	58	4.5%	64	5.4%	51	4.1%	65	4.8%

Table 4: VaR breaches compared with and without balanced incentive function (i.e. regularized)

$$\tilde{\mathcal{L}}(X) = (1 + reg(x))\mathcal{L}(x) \tag{14}$$

The results of neural networks with balancing the incentive functions (i.e., regularized) are displayed in Table 4, Panels A-D. The regularized LSTM is better than the non-regularized LSTM in all cases and performs better than HMM, especially in Long Bond. For the 5% VaR threshold, regularized LSTM reaches realized occurrences of breaches for less than 5% in all cases in Equity and three cases out of four cases in Long Bond. This implies that across asset classes, the regularized LSTM can produce lower breaches for the equity markets than the bond markets when asset managers utilize 5% VaR threshold . The regularized LSTM and

HMM also sets a new record for the 1% VaR thresholds with exactly 1% breaches for EuroStoxx 50 but all eight approaches remain to underestimate the downside tail risk with their VaR threshold estimations. The standard feed forward neural network also enhances its performance following the incentive balancing regularization in nearly all cases while the CNN regularization delivers a more mixed picture.

Since our main objective is to avoid significant losses from the tail risk because the tail risk can bring catastrophic losses for asset managers and institutional investors, we conclude that the LSTM with HMM initialization and balancing incentive functions (i.e., regularized) offers superior results from a conservative risk management in terms of lower number and probability of breaches in VaR thresholds compared to the classic Mean/Variance and other neural networks models i.e., FF and CNN. This presents a promising picture for asset managers that a machine learning model with proper initialization and balancing incentive functions can reduce the likelihood of significant losses especially during the periods when risk estimation tasks have become more difficult.

Tables 5 - 8 display the direction comparisons between the eight approaches for all four regions with the 1% (5%) VaR threshold results being displayed for Equities and Long Bonds in Tables 5 and 6 (Tables 7 and 8), respectively. Note that comp = 1.0 [= 0.0] and green color [red color] indicate that the row models are superior [inferior] compared to the column model. Dom = 0.0 [dom = negative] and green color [red color] imply that the row models VaR breaches also always occur [do not always occur] in the column model. Black color in comp and 0.5 in comp indicate that the row models are not superior nor inferior compared to the column model and black color in dom indicates that the column model is at the low level of dominance over the row models.

Focusing on the comparison (comp) from the last column of Table 5 for the regularized LSTM column shows that the regularized LSTM is more superior compared to almost all other approaches (comp = 0 and red color) and is statistically significant (p-value 0.1 or less), except for HMM in the US and Global (GL) Equity indices. Focusing on the dominance (dom), the regularized LSTM also substantially dominates especially the classic mean/variance approach, the non-regularized neural networks, regularized CNN, regularized feed forward (FF) neural networks and HMM for UK Equity indicated by negative and red numbers. Regularized LSTM only dominates HMM in the US and Global (GL) Equity indices and non-regularized LSTM in the US as indicated by black. Table 6 shows the results in the Bond markets at 1% threshold and we find that regularized LSTM are superior and more dominant compared to the other approaches. Tables 7 and 8 present both Equity and Bond markets at 5% threshold and these tables shows that regularized LSTM is superior and more dominant compared to the other approaches. These results imply that regularized LSTM can add real value to the risk management process for asset managers and institutional investors such as pension funds and insurance companies.

model	peer / region	Classic comp	pvalue	dom	HMM comp	pvalue	dom	FF (hmm init) comp	pvalue	dom	CNN (hmm init) comp	pvalue	dom	LSTM (hmm init) comp	pvalue	dom	FF (hmm init + reg) comp	pvalue	dom	CNN (hmm init + reg) comp	pvalue	dom	LSTM (hmm init + reg) comp	pvalue	dom
Classic	US	-	-	-	0.0	**0.00**	-0.35	0.0	**0.01**	-0.23	0.5	1.00	-0.35	0.0	**0.03**	-0.27	0.0	**0.05**	-0.15	0.0	0.13	-0.31	0.0	**0.01**	-0.35
	EU	-	-	-	0.0	**0.00**	-0.40	0.0	0.16	-0.10	0.0	0.74	-0.25	0.0	0.37	-0.35	0.0	0.32	-0.15	0.0	0.26	-0.25	0.0	**0.01**	-0.45
	UK	-	-	-	0.0	**0.01**	-0.24	0.0	0.32	-0.12	0.0	0.76	-0.24	0.0	0.32	-0.24	0.0	**0.01**	-0.24	0.0	0.21	-0.28	0.0	**0.00**	-0.40
	GL	-	-	-	0.0	**0.00**	-0.46	0.0	0.16	-0.15	0.0	0.43	-0.38	0.0	**0.00**	-0.33	0.0	**0.01**	-0.26	0.0	0.13	-0.36	0.0	**0.00**	-0.41
HMM	US	1.0	**0.00**	0.00	-	-	-	1.0	0.08	0.00	1.0	**0.01**	-0.12	1.0	0.08	0.00	1.0	**0.03**	0.00	1.0	**0.05**	0.00	1.0	0.56	-0.06
	EU	1.0	**0.00**	0.00	-	-	-	1.0	**0.01**	0.00	1.0	**0.01**	0.00	1.0	**0.03**	0.00	1.0	**0.01**	0.00	1.0	**0.03**	0.00	0.5	1.00	-0.17
	UK	1.0	**0.01**	0.00	-	-	-	1.0	0.10	-0.05	1.0	0.10	-0.11	1.0	0.26	-0.11	0.5	1.00	-0.05	1.0	0.41	-0.11	0.0	**0.05**	-0.21
	GL	1.0	**0.00**	-0.05	-	-	-	1.0	**0.00**	-0.05	1.0	**0.00**	-0.14	1.0	**0.01**	0.00	1.0	**0.02**	-0.09	1.0	**0.00**	-0.05	1.0	0.71	-0.14
FF (hmm init)	US	1.0	**0.01**	0.00	0.0	0.08	-0.15	-	-	-	1.0	0.13	-0.25	0.5	1.00	-0.05	1.0	0.32	-0.05	1.0	0.71	-0.15	0.0	0.32	-0.15
	EU	1.0	0.16	0.00	0.0	**0.01**	-0.33	-	-	-	1.0	0.74	-0.22	0.0	0.74	-0.28	0.5	1.00	-0.11	0.0	0.65	-0.17	0.0	**0.03**	-0.39
	UK	1.0	0.32	-0.04	0.0	0.10	-0.22	-	-	-	1.0	0.76	-0.22	0.0	0.74	-0.22	0.0	**0.05**	-0.17	0.0	0.48	-0.22	0.0	**0.00**	-0.35
	GL	1.0	0.16	-0.06	0.0	**0.00**	-0.40	-	-	-	0.5	1.00	-0.29	0.0	**0.03**	-0.26	0.0	0.06	-0.17	0.0	0.44	-0.26	0.0	**0.00**	-0.37
CNN (hmm init)	US	0.5	1.00	-0.35	0.0	**0.01**	-0.42	0.0	0.13	-0.42	-	-	-	0.0	0.13	-0.42	0.0	0.35	-0.42	0.0	0.10	-0.27	0.0	**0.05**	-0.46
	EU	1.0	0.74	-0.21	0.0	**0.01**	-0.37	0.0	0.74	-0.26	-	-	-	0.0	0.48	-0.26	0.0	0.76	-0.32	0.0	0.41	-0.21	0.0	**0.02**	-0.42
	UK	1.0	0.76	-0.21	0.0	0.10	-0.29	0.0	0.76	-0.25	-	-	-	0.0	0.53	-0.25	0.0	0.13	-0.33	0.0	0.32	-0.25	0.0	**0.01**	-0.46
	GL	1.0	0.43	-0.31	0.0	**0.00**	-0.46	0.5	1.00	-0.29	-	-	-	0.0	0.13	-0.40	0.0	0.25	-0.34	0.0	0.26	-0.14	0.0	**0.01**	-0.46
LSTM (hmm init)	US	1.0	**0.03**	-0.05	0.0	0.08	-0.15	0.5	1.00	-0.05	1.0	0.13	-0.25	-	-	-	1.0	0.41	-0.10	1.0	0.71	-0.15	0.0	0.16	-0.10
	EU	1.0	0.37	-0.24	0.0	**0.03**	-0.29	1.0	0.74	-0.24	1.0	0.48	-0.18	-	-	-	1.0	0.74	-0.24	0.5	1.00	-0.18	0.0	0.06	-0.35
	UK	1.0	0.32	-0.14	0.0	0.26	-0.23	1.0	0.74	-0.18	1.0	0.53	-0.18	-	-	-	0.0	0.18	-0.18	0.0	0.74	-0.23	0.0	**0.01**	-0.32
	GL	1.0	**0.00**	-0.07	0.0	**0.01**	-0.21	1.0	**0.03**	-0.07	1.0	0.13	-0.25	-	-	-	1.0	0.53	-0.14	1.0	0.29	-0.18	0.0	0.10	-0.25
FF (hmm init + reg)	US	1.0	**0.05**	0.00	0.0	**0.03**	-0.23	0.0	0.32	-0.14	1.0	0.35	-0.32	0.0	0.41	-0.18	-	-	-	0.0	0.74	-0.23	0.0	0.10	-0.23
	EU	1.0	0.32	-0.06	0.0	**0.01**	-0.33	0.5	1.00	-0.11	1.0	0.76	-0.28	0.0	0.74	-0.28	-	-	-	0.0	0.71	-0.22	0.0	**0.03**	-0.39
	UK	1.0	**0.01**	0.00	0.5	1.00	-0.05	1.0	**0.05**	0.00	1.0	0.13	-0.16	1.0	0.18	-0.05	-	-	-	1.0	0.41	-0.11	0.0	**0.05**	-0.21
	GL	1.0	**0.01**	-0.03	0.0	**0.02**	-0.33	1.0	0.06	-0.03	1.0	0.25	-0.23	0.0	0.53	-0.20	-	-	-	1.0	0.59	-0.20	0.0	**0.03**	-0.30
CNN (hmm init + reg)	US	1.0	0.13	-0.14	0.0	**0.05**	-0.19	0.0	0.71	-0.19	1.0	0.10	-0.10	0.0	0.71	-0.19	1.0	0.74	-0.19	-	-	-	0.0	0.26	-0.24
	EU	1.0	0.26	-0.12	0.0	**0.03**	-0.29	1.0	0.65	-0.12	1.0	0.41	-0.12	0.5	1.00	-0.18	1.0	0.71	-0.18	-	-	-	0.0	**0.03**	-0.29
	UK	1.0	0.21	-0.14	0.0	0.41	-0.19	1.0	0.48	-0.14	1.0	0.32	-0.14	1.0	0.74	-0.19	0.0	0.41	-0.19	-	-	-	0.0	**0.03**	-0.33
	GL	1.0	0.13	-0.22	0.0	**0.00**	-0.34	1.0	0.44	-0.19	1.0	0.26	-0.06	0.0	0.29	-0.28	0.0	0.59	-0.25	-	-	-	0.0	**0.01**	-0.34
LSTM (hmm init + reg)	US	1.0	**0.01**	-0.06	0.0	0.56	-0.11	1.0	0.32	-0.06	1.0	**0.05**	-0.22	1.0	0.16	0.00	1.0	0.10	-0.06	1.0	0.26	-0.11	-	-	-
	EU	1.0	**0.01**	-0.08	0.5	1.00	-0.17	1.0	**0.03**	-0.08	1.0	**0.02**	-0.08	1.0	0.06	-0.08	1.0	**0.03**	-0.08	1.0	**0.03**	0.00	-	-	-
	UK	1.0	**0.00**	0.00	1.0	**0.05**	0.00	1.0	**0.00**	0.00	1.0	**0.01**	-0.13	1.0	**0.01**	0.00	1.0	**0.05**	0.00	1.0	**0.03**	-0.07	-	-	-
	GL	1.0	**0.00**	0.00	0.0	0.71	-0.17	1.0	**0.00**	-0.04	1.0	**0.01**	-0.17	1.0	0.10	-0.09	1.0	**0.03**	-0.09	1.0	**0.01**	-0.09	-	-	-

Table 5: Comparison of estimated 1% VaR of equities with and without balanced incentive function (i.e. regularized)

	peer	Classic			HMM			FF (hmm init)			CNN (hmm init)			LSTM (hmm init)			FF (hmm init + reg)			CNN (hmm init + reg)			LSTM (hmm init + reg)		
model	region	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom
Classic	US	-	-	-	0.0	**0.00**	-0.37	1.0	0.48	-0.10	1.0	0.29	-0.17	0.0	0.37	-0.23	0.0	0.65	-0.10	0.0	0.71	-0.13	0.0	**0.00**	-0.57
	EU	-	-	-	0.0	0.05	-0.38	1.0	0.41	-0.08	1.0	0.06	-0.27	0.0	0.17	-0.35	0.0	0.41	-0.15	0.5	1.00	-0.27	0.0	**0.02**	-0.38
	UK	-	-	-	0.0	**0.01**	-0.30	1.0	0.16	-0.09	1.0	**0.02**	-0.09	0.0	0.41	-0.17	1.0	0.65	-0.09	1.0	0.06	-0.09	0.0	**0.02**	-0.35
	GL	-	-	-	0.0	**0.01**	-0.32	1.0	**0.00**	0.00	1.0	**0.00**	-0.06	1.0	0.76	-0.16	1.0	0.06	-0.06	1.0	0.53	-0.13	0.0	**0.00**	-0.48
HMM	US	1.0	**0.00**	-0.05	-	-	-	1.0	**0.00**	-0.05	1.0	**0.00**	-0.05	1.0	**0.03**	-0.10	1.0	**0.01**	-0.05	1.0	**0.01**	-0.05	0.0	0.08	-0.45
	EU	1.0	0.05	-0.16	-	-	-	1.0	**0.02**	-0.16	1.0	**0.00**	-0.21	1.0	0.48	-0.16	1.0	0.17	-0.21	1.0	0.05	-0.16	0.0	0.74	-0.26
	UK	1.0	**0.01**	0.00	-	-	-	1.0	**0.00**	0.00	1.0	**0.00**	0.00	1.0	**0.03**	0.00	1.0	**0.01**	-0.06	1.0	**0.00**	0.00	0.5	1.00	-0.12
	GL	1.0	**0.01**	-0.05	-	-	-	1.0	**0.00**	0.00	1.0	**0.00**	-0.05	1.0	**0.00**	-0.05	1.0	**0.00**	-0.05	1.0	**0.00**	-0.05	0.0	0.17	-0.41
FF (hmm init)	US	0.0	0.48	-0.16	0.0	**0.00**	-0.41	-	-	-	1.0	0.64	-0.25	0.0	0.20	-0.31	0.0	0.37	-0.22	0.0	0.37	-0.22	0.0	**0.00**	-0.59
	EU	0.0	0.41	-0.14	0.0	**0.02**	-0.43	-	-	-	1.0	0.11	-0.21	0.0	0.05	-0.36	0.0	0.10	-0.18	0.0	0.59	-0.29	0.0	**0.00**	-0.39
	UK	0.0	0.16	-0.22	0.0	**0.00**	-0.41	-	-	-	1.0	0.10	-0.04	0.0	**0.03**	-0.26	0.0	0.26	-0.19	1.0	0.48	-0.11	0.0	**0.00**	-0.44
	GL	0.0	**0.00**	-0.24	0.0	**0.00**	-0.46	-	-	-	1.0	0.82	-0.22	0.0	**0.01**	-0.27	0.0	0.21	-0.17	0.0	**0.05**	-0.29	0.0	**0.00**	-0.61
CNN (hmm init)	US	0.0	0.29	-0.26	0.0	**0.00**	-0.44	0.0	0.64	-0.29	-	-	-	0.0	0.09	-0.35	0.0	0.13	-0.24	0.0	0.06	-0.18	0.0	**0.00**	-0.65
	EU	0.0	0.06	-0.46	0.0	**0.00**	-0.57	0.0	0.11	-0.37	-	-	-	0.0	**0.00**	-0.46	0.0	**0.02**	-0.46	0.0	**0.01**	-0.31	0.0	**0.00**	-0.60
	UK	0.0	**0.02**	-0.32	0.0	**0.00**	-0.48	0.0	0.10	-0.16	-	-	-	0.0	**0.00**	-0.32	0.0	**0.02**	-0.26	0.0	0.32	-0.10	0.0	**0.00**	-0.52
	GL	0.0	**0.00**	-0.31	0.0	**0.00**	-0.50	0.0	0.82	-0.24	-	-	-	0.0	**0.02**	-0.33	0.0	0.25	-0.29	0.0	**0.00**	-0.21	0.0	**0.00**	-0.62
LSTM (hmm init)	US	1.0	0.37	-0.15	0.0	**0.03**	-0.33	1.0	0.20	-0.19	1.0	0.09	-0.19	-	-	-	1.0	0.53	-0.15	1.0	0.59	-0.22	0.0	**0.00**	-0.48
	EU	1.0	0.17	-0.19	0.0	0.48	-0.24	1.0	0.05	-0.14	1.0	**0.00**	-0.10	-	-	-	1.0	0.44	-0.29	1.0	**0.03**	0.00	0.0	0.26	-0.24
	UK	1.0	0.41	-0.10	0.0	**0.03**	-0.24	1.0	**0.03**	-0.05	1.0	**0.00**	0.00	-	-	-	1.0	0.32	-0.14	1.0	**0.01**	-0.05	0.0	0.06	-0.29
	GL	0.0	0.76	-0.19	0.0	**0.00**	-0.34	1.0	**0.01**	-0.06	1.0	**0.02**	-0.12	-	-	-	1.0	0.17	-0.12	1.0	0.80	-0.22	0.0	**0.00**	-0.47
FF (hmm init + reg)	US	1.0	0.65	-0.07	0.0	**0.01**	-0.34	1.0	0.37	-0.14	1.0	0.13	-0.10	0.0	0.53	-0.21	-	-	-	0.5	1.00	-0.10	0.0	**0.00**	-0.55
	EU	1.0	0.41	-0.08	0.0	0.17	-0.38	1.0	0.10	-0.04	1.0	**0.02**	-0.21	0.0	0.44	-0.38	-	-	-	1.0	0.59	-0.25	0.0	0.08	-0.38
	UK	0.0	0.65	-0.12	0.0	**0.01**	-0.38	1.0	0.26	-0.08	1.0	**0.02**	-0.04	0.0	0.32	-0.25	-	-	-	1.0	0.06	-0.04	0.0	**0.01**	-0.38
	GL	0.0	0.06	-0.22	0.0	**0.00**	-0.43	1.0	0.21	-0.08	1.0	0.25	-0.19	0.0	0.17	-0.24	-	-	-	0.0	0.32	-0.27	0.0	**0.00**	-0.57
CNN (hmm init + reg)	US	1.0	0.71	-0.10	0.0	**0.01**	-0.34	1.0	0.37	-0.14	1.0	0.06	-0.03	0.0	0.59	-0.28	0.5	1.00	-0.10	-	-	-	0.0	**0.00**	-0.55
	EU	0.5	1.00	-0.27	0.0	0.05	-0.38	1.0	0.59	-0.23	1.0	**0.01**	-0.08	0.0	**0.03**	-0.19	0.0	0.59	-0.31	-	-	-	0.0	**0.02**	-0.38
	UK	0.0	0.06	-0.28	0.0	**0.00**	-0.45	0.0	0.48	-0.17	1.0	0.32	-0.03	0.0	**0.01**	-0.31	0.0	0.06	-0.21	-	-	-	0.0	**0.00**	-0.52
	GL	0.0	0.53	-0.18	0.0	**0.00**	-0.36	1.0	**0.05**	-0.12	1.0	**0.00**	0.00	0.0	0.80	-0.24	1.0	0.32	-0.18	-	-	-	0.0	**0.00**	-0.52
LSTM (hmm init + reg)	US	1.0	**0.00**	-0.07	1.0	0.08	-0.21	1.0	**0.00**	-0.07	1.0	**0.00**	-0.14	1.0	**0.00**	0.00	1.0	**0.00**	-0.07	1.0	**0.00**	-0.07	-	-	-
	EU	1.0	**0.02**	-0.11	1.0	0.74	-0.22	1.0	**0.00**	-0.06	1.0	**0.00**	-0.22	1.0	0.26	-0.11	1.0	0.08	-0.17	1.0	**0.02**	-0.11	-	-	-
	UK	1.0	**0.02**	-0.06	0.5	1.00	-0.12	1.0	**0.00**	-0.06	1.0	**0.00**	-0.06	1.0	0.06	-0.06	1.0	**0.01**	-0.06	1.0	**0.00**	-0.12	-	-	-
	GL	1.0	**0.00**	-0.06	1.0	0.17	-0.24	1.0	**0.00**	-0.06	1.0	**0.00**	-0.06	1.0	**0.00**	0.00	1.0	**0.00**	-0.06	1.0	**0.00**	-0.06	-	-	-

Table 6: Comparison of estimated 1% VaR of long bonds with and without balanced incentive function (i.e. regularized)

model	peer region	Classic			HMM			FF (hmm init)			CNN (hmm init)			LSTM (hmm init)			FF (hmm init + reg)			CNN (hmm init + reg)			LSTM (hmm init + reg)		
		comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom
Classic	US	-	-	-	0.0	0.53	-0.10	1.0	0.56	-0.08	1.0	0.33	-0.25	0.0	0.83	-0.17	0.5	1.00	-0.10	0.0	0.58	-0.25	0.0	**0.00**	-0.24
	EU	-	-	-	1.0	0.70	-0.20	1.0	0.78	-0.10	1.0	**0.02**	-0.25	0.0	0.87	-0.31	0.0	0.59	-0.13	0.0	0.21	-0.31	0.0	**0.00**	-0.28
	UK	-	-	-	0.0	0.17	-0.15	0.0	0.21	-0.12	1.0	0.43	-0.19	0.0	**0.03**	-0.15	0.0	0.65	-0.05	0.0	0.83	-0.20	0.0	**0.00**	-0.27
	GL	-	-	-	0.0	**0.02**	-0.16	0.0	0.78	-0.08	1.0	0.08	-0.16	0.0	0.37	-0.14	0.0	0.13	-0.09	1.0	0.38	-0.16	0.0	**0.00**	-0.27
HMM	US	1.0	0.53	-0.07	-	-	-	1.0	0.32	-0.10	1.0	0.17	-0.21	1.0	0.82	-0.15	1.0	0.64	-0.13	0.0	0.84	-0.21	0.0	**0.00**	-0.25
	EU	0.0	0.70	-0.22	-	-	-	0.0	0.87	-0.29	1.0	**0.02**	-0.16	0.0	0.53	-0.21	0.0	0.48	-0.29	0.0	0.11	-0.32	0.0	**0.00**	-0.35
	UK	1.0	0.17	-0.07	-	-	-	1.0	0.81	-0.15	1.0	**0.05**	-0.11	0.0	0.56	-0.13	1.0	0.29	-0.09	1.0	0.25	-0.07	0.0	**0.01**	-0.26
	GL	1.0	**0.02**	-0.05	-	-	-	1.0	**0.05**	-0.08	1.0	**0.00**	-0.08	1.0	0.20	-0.11	1.0	0.30	-0.12	1.0	**0.00**	-0.08	0.0	**0.00**	-0.19
FF (hmm init)	US	0.0	0.56	-0.11	0.0	0.32	-0.15	-	-	-	1.0	0.52	-0.26	0.0	0.55	-0.22	0.0	0.53	-0.09	0.0	0.35	-0.26	0.0	**0.00**	-0.28
	EU	0.0	0.78	-0.11	1.0	0.87	-0.27	-	-	-	1.0	**0.03**	-0.26	0.0	0.76	-0.35	0.0	0.41	-0.13	0.0	0.16	-0.32	0.0	**0.00**	-0.32
	UK	1.0	0.21	-0.05	0.0	0.81	-0.16	-	-	-	1.0	0.13	-0.18	0.0	0.41	-0.15	1.0	0.37	-0.07	1.0	0.51	-0.16	0.0	**0.01**	-0.29
	GL	1.0	0.78	-0.07	0.0	**0.05**	-0.18	-	-	-	1.0	**0.04**	-0.13	0.0	0.51	-0.14	0.0	0.29	-0.11	1.0	0.29	-0.15	0.0	**0.00**	-0.27
CNN (hmm init)	US	0.0	0.33	-0.32	0.0	0.17	-0.30	0.0	0.52	-0.30	-	-	-	0.0	0.24	-0.30	0.0	0.34	-0.33	0.0	**0.03**	-0.19	0.0	**0.00**	-0.45
	EU	0.0	**0.02**	-0.40	0.0	**0.02**	-0.31	0.0	**0.03**	-0.40	-	-	-	0.0	**0.00**	-0.34	0.0	**0.01**	-0.40	0.0	**0.00**	-0.32	0.0	**0.00**	-0.51
	UK	0.0	0.43	-0.24	0.0	**0.05**	-0.24	0.0	0.13	-0.29	-	-	-	0.0	**0.03**	-0.29	0.0	0.30	-0.22	0.0	0.13	-0.13	0.0	**0.00**	-0.40
	GL	0.0	0.08	-0.26	0.0	**0.00**	-0.28	0.0	**0.04**	-0.24	-	-	-	0.0	**0.01**	-0.24	0.0	**0.01**	-0.30	0.0	0.16	-0.12	0.0	**0.00**	-0.38
LSTM (hmm init)	US	1.0	0.83	-0.16	0.0	0.82	-0.16	1.0	0.55	-0.18	1.0	0.24	-0.23	-	-	-	1.0	0.84	-0.19	0.0	0.68	-0.21	0.0	**0.01**	-0.29
	EU	1.0	0.87	-0.30	1.0	0.53	-0.17	1.0	0.76	-0.33	1.0	**0.00**	-0.15	-	-	-	0.0	0.87	-0.32	0.0	0.22	-0.25	0.0	**0.00**	-0.38
	UK	1.0	**0.03**	-0.04	1.0	0.56	-0.10	1.0	0.41	-0.10	1.0	**0.03**	-0.13	-	-	-	1.0	0.08	-0.06	1.0	0.18	-0.13	0.0	**0.01**	-0.21
	GL	1.0	0.37	-0.10	0.0	0.20	-0.17	1.0	0.51	-0.11	1.0	**0.01**	-0.10	-	-	-	0.0	0.82	-0.12	1.0	0.08	-0.11	0.0	**0.00**	-0.22
FF (hmm init + reg)	US	0.5	1.00	-0.10	0.0	0.64	-0.16	1.0	0.53	-0.06	1.0	0.34	-0.27	0.0	0.84	-0.21	-	-	-	0.0	0.56	-0.24	0.0	**0.00**	-0.29
	EU	1.0	0.59	-0.10	1.0	0.48	-0.24	1.0	0.41	-0.08	1.0	**0.01**	-0.22	1.0	0.87	-0.31	-	-	-	0.0	0.35	-0.29	0.0	**0.00**	-0.31
	UK	1.0	0.65	-0.03	0.0	0.29	-0.16	0.0	0.37	-0.12	1.0	0.30	-0.16	0.0	0.08	-0.16	-	-	-	0.5	1.00	-0.17	0.0	**0.00**	-0.28
	GL	1.0	0.13	-0.04	0.0	0.30	-0.18	1.0	0.29	-0.06	1.0	**0.01**	-0.16	1.0	0.82	-0.11	-	-	-	1.0	0.09	-0.15	0.0	**0.00**	-0.22
CNN (hmm init + reg)	US	1.0	0.58	-0.22	1.0	0.84	-0.20	1.0	0.35	-0.20	1.0	**0.03**	-0.07	1.0	0.68	-0.18	1.0	0.56	-0.20	-	-	-	0.0	**0.04**	-0.33
	EU	1.0	0.21	-0.22	1.0	0.11	-0.20	1.0	0.16	-0.22	1.0	**0.00**	-0.04	1.0	0.22	-0.17	1.0	0.35	-0.22	-	-	-	0.0	**0.03**	-0.30
	UK	1.0	0.83	-0.19	0.0	0.25	-0.14	0.0	0.51	-0.21	1.0	0.13	-0.05	0.0	0.18	-0.22	0.5	1.00	-0.17	-	-	-	0.0	**0.00**	-0.34
	GL	0.0	0.38	-0.21	0.0	**0.00**	-0.23	0.0	0.29	-0.21	1.0	0.16	-0.07	0.0	0.08	-0.20	0.0	0.09	-0.24	-	-	-	0.0	**0.00**	-0.31
LSTM (hmm init + reg)	US	1.0	**0.00**	-0.02	1.0	**0.00**	-0.06	1.0	**0.00**	-0.04	1.0	**0.00**	-0.22	1.0	**0.01**	-0.10	1.0	**0.00**	-0.08	1.0	**0.04**	-0.18	-	-	-
	EU	1.0	**0.00**	0.00	1.0	**0.00**	-0.07	1.0	**0.00**	-0.05	1.0	**0.00**	-0.14	1.0	**0.00**	-0.16	1.0	**0.00**	-0.07	1.0	**0.03**	-0.14	-	-	-
	UK	1.0	**0.00**	0.00	1.0	**0.01**	-0.07	1.0	**0.01**	-0.09	1.0	**0.00**	-0.12	1.0	**0.01**	-0.05	1.0	**0.00**	-0.02	1.0	**0.00**	-0.12	-	-	-
	GL	1.0	**0.00**	-0.02	1.0	**0.00**	-0.03	1.0	**0.00**	-0.03	1.0	**0.00**	-0.05	1.0	**0.00**	0.00	1.0	**0.00**	-0.02	1.0	**0.00**	-0.02	-	-	-

Table 7: Comparison of estimated 5% VaR of equities with and without balanced incentive function (i.e. regularized)

	peer	Classic			HMM			FF (hmm init)			CNN (hmm init)			LSTM (hmm init)			FF (hmm init + reg)			CNN (hmm init + reg)			LSTM (hmm init + reg)		
model	region	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom
Classic	US	-	-	-	1.0	0.32	-0.08	1.0	**0.01**	-0.01	1.0	**0.01**	-0.08	1.0	**0.02**	-0.06	1.0	**0.02**	-0.03	1.0	0.11	-0.08	0.0	**0.00**	-0.21
	EU	-	-	-	1.0	0.12	-0.12	1.0	0.16	-0.08	1.0	**0.00**	-0.05	1.0	**0.01**	-0.09	0.0	0.80	-0.10	1.0	**0.00**	-0.09	0.0	**0.01**	-0.23
	UK	-	-	-	1.0	0.56	-0.07	1.0	**0.01**	-0.01	1.0	**0.00**	-0.13	1.0	0.44	-0.09	0.5	1.00	-0.06	1.0	**0.02**	-0.11	0.0	**0.00**	-0.27
	GL	-	-	-	1.0	0.35	-0.10	1.0	**0.00**	0.00	1.0	**0.00**	-0.07	1.0	**0.05**	-0.06	1.0	**0.00**	-0.01	1.0	**0.01**	-0.06	0.0	0.08	-0.13
HMM	US	0.0	0.32	-0.13	-	-	-	1.0	0.23	-0.08	1.0	**0.03**	-0.07	1.0	0.13	-0.07	1.0	0.35	-0.09	1.0	0.49	-0.11	0.0	**0.00**	-0.28
	EU	0.0	0.12	-0.20	-	-	-	0.0	0.72	-0.19	1.0	**0.00**	-0.07	1.0	0.18	-0.08	0.0	0.08	-0.21	1.0	**0.02**	-0.09	0.0	**0.00**	-0.28
	UK	0.0	0.56	-0.10	-	-	-	1.0	0.09	-0.07	1.0	**0.00**	-0.11	1.0	0.78	-0.08	0.0	0.59	-0.11	1.0	**0.04**	-0.10	0.0	**0.00**	-0.29
	GL	0.0	0.35	-0.15	-	-	-	1.0	**0.00**	-0.05	1.0	**0.00**	-0.04	1.0	0.21	-0.04	1.0	0.06	-0.07	1.0	0.07	-0.08	0.0	**0.02**	-0.19
FF (hmm init)	US	0.0	**0.01**	-0.12	0.0	0.23	-0.14	-	-	-	1.0	0.30	-0.11	1.0	0.82	-0.11	0.0	0.56	-0.03	0.0	0.65	-0.14	0.0	**0.00**	-0.30
	EU	0.0	0.16	-0.14	1.0	0.72	-0.17	-	-	-	1.0	**0.00**	-0.11	1.0	0.17	-0.16	0.0	0.11	-0.16	1.0	**0.01**	-0.12	0.0	**0.00**	-0.30
	UK	0.0	**0.01**	-0.13	0.0	0.09	-0.15	-	-	-	1.0	0.08	-0.11	0.0	0.20	-0.18	0.0	**0.00**	-0.11	1.0	0.53	-0.13	0.0	**0.00**	-0.35
	GL	0.0	**0.00**	-0.19	0.0	**0.00**	-0.19	-	-	-	1.0	0.07	-0.07	0.0	0.06	-0.18	0.0	0.06	-0.07	0.0	0.23	-0.12	0.0	**0.00**	-0.28
CNN (hmm init)	US	0.0	**0.01**	-0.24	0.0	**0.03**	-0.18	0.0	0.30	-0.16	-	-	-	0.0	0.41	-0.16	0.0	0.22	-0.18	0.0	0.05	-0.12	0.0	**0.00**	-0.39
	EU	0.0	**0.00**	-0.30	0.0	**0.00**	-0.24	0.0	**0.00**	-0.29	-	-	-	0.0	**0.02**	-0.22	0.0	**0.00**	-0.32	0.0	0.18	-0.12	0.0	**0.00**	-0.41
	UK	0.0	**0.00**	-0.31	0.0	**0.00**	-0.27	0.0	0.08	-0.20	-	-	-	0.0	**0.01**	-0.28	0.0	**0.00**	-0.30	0.0	0.13	-0.12	0.0	**0.00**	-0.48
	GL	0.0	**0.00**	-0.31	0.0	**0.00**	-0.25	0.0	0.07	-0.15	-	-	-	0.0	**0.00**	-0.24	0.0	**0.01**	-0.19	0.0	**0.00**	-0.15	0.0	**0.00**	-0.36
LSTM (hmm init)	US	0.0	**0.02**	-0.17	0.0	0.13	-0.14	0.0	0.82	-0.12	1.0	0.41	-0.12	-	-	-	0.0	0.65	-0.14	0.0	0.55	-0.17	0.0	**0.00**	-0.32
	EU	0.0	**0.01**	-0.23	0.0	0.18	-0.14	0.0	0.17	-0.23	1.0	**0.02**	-0.11	-	-	-	0.0	**0.01**	-0.25	1.0	0.21	-0.13	0.0	**0.00**	-0.33
	UK	0.0	0.44	-0.12	0.0	0.78	-0.10	1.0	0.20	-0.11	1.0	**0.01**	-0.14	-	-	-	0.0	0.47	-0.14	1.0	0.11	-0.15	0.0	**0.00**	-0.30
	GL	0.0	**0.05**	-0.15	0.0	0.21	-0.09	1.0	0.06	-0.09	1.0	**0.00**	-0.08	-	-	-	1.0	0.37	-0.10	1.0	0.39	-0.11	0.0	**0.00**	-0.18
FF (hmm init + reg)	US	0.0	**0.02**	-0.13	0.0	0.35	-0.14	1.0	0.56	-0.01	1.0	0.22	-0.11	1.0	0.65	-0.11	-	-	-	0.0	0.83	-0.14	0.0	**0.00**	-0.29
	EU	1.0	0.80	-0.09	1.0	0.08	-0.12	1.0	0.11	-0.08	1.0	**0.00**	-0.07	1.0	**0.01**	-0.11	-	-	-	1.0	**0.00**	-0.08	0.0	**0.02**	-0.26
	UK	0.5	1.00	-0.06	1.0	0.59	-0.09	1.0	**0.00**	0.00	1.0	**0.00**	-0.11	1.0	0.47	-0.10	-	-	-	1.0	**0.01**	-0.09	0.0	**0.00**	-0.30
	GL	0.0	**0.00**	-0.16	0.0	0.06	-0.16	1.0	0.06	-0.01	1.0	**0.01**	-0.06	0.0	0.37	-0.14	-	-	-	0.5	1.00	-0.08	0.0	**0.00**	-0.25
CNN (hmm init + reg)	US	0.0	0.11	-0.17	0.0	0.49	-0.14	1.0	0.65	-0.12	1.0	0.05	-0.04	1.0	0.55	-0.14	1.0	0.83	-0.13	-	-	-	0.0	**0.00**	-0.32
	EU	0.0	**0.00**	-0.29	0.0	**0.02**	-0.21	0.0	**0.01**	-0.26	1.0	0.18	-0.07	0.0	0.21	-0.19	0.0	**0.00**	-0.29	-	-	-	0.0	**0.00**	-0.38
	UK	0.0	**0.02**	-0.24	0.0	**0.04**	-0.21	0.0	0.53	-0.16	1.0	0.13	-0.06	0.0	0.11	-0.24	0.0	**0.01**	-0.22	-	-	-	0.0	**0.00**	-0.41
	GL	0.0	**0.01**	-0.19	0.0	0.07	-0.17	1.0	0.23	-0.07	1.0	**0.00**	-0.01	0.0	0.39	-0.16	0.5	1.00	-0.08	-	-	-	0.0	**0.00**	-0.27
LSTM (hmm init + reg)	US	1.0	**0.00**	-0.03	1.0	**0.00**	-0.07	1.0	**0.00**	-0.03	1.0	**0.00**	-0.10	1.0	**0.00**	-0.05	1.0	**0.00**	-0.03	1.0	**0.00**	-0.09	-	-	-
	EU	1.0	**0.01**	-0.08	1.0	**0.00**	-0.05	1.0	**0.00**	-0.09	1.0	**0.00**	-0.05	1.0	**0.00**	-0.05	1.0	**0.02**	-0.12	1.0	**0.00**	-0.05	-	-	-
	UK	1.0	**0.00**	0.00	1.0	**0.00**	0.00	1.0	**0.00**	0.00	1.0	**0.00**	-0.10	1.0	**0.00**	0.00	1.0	**0.00**	-0.04	1.0	**0.00**	-0.06	-	-	-
	GL	1.0	0.08	-0.05	1.0	**0.02**	-0.06	1.0	**0.00**	-0.03	1.0	**0.00**	-0.06	1.0	**0.00**	0.00	1.0	**0.00**	-0.05	1.0	**0.00**	-0.06	-	-	-

Table 8: Comparison of estimated 5% VaR of long bonds with and without balanced incentive function (i.e. regularized)

Panel A: Monetary costs of 5% VaR breaches in Equity								
Region	S&P 500		EuroStoxx 50		FTSE 100		MSCI World	
Model	acc. loss per year	avg loss per breach	acc. loss per year	avg loss per breach	acc. loss per year	avg loss per breach	acc. loss per year	avg loss per breach
Classic	-5.49%	-1.11%	-6.21%	-0.97%	-5.10%	-1.36%	-6.35%	-1.18%
HMM	-4.94%	-0.85%	-5.41%	-0.65%	-4.73%	-1.07%	-5.45%	-0.82%
FF (hmm init)	-5.11%	-0.99%	-6.04%	-0.91%	-4.97%	-1.25%	-6.30%	-1.16%
CNN (hmm init)	-5.52%	-1.14%	-6.02%	-0.83%	-5.14%	-1.38%	-6.34%	-1.23%
LSTM (hmm init)	-4.93%	-0.92%	-5.41%	-0.57%	-4.79%	-1.15%	-6.00%	-1.08%
FF (hmm init + reg)	-5.19%	-0.94%	-5.71%	-0.76%	-4.77%	-1.21%	-6.04%	-1.07%
CNN (hmm init + reg)	-4.87%	-1.00%	-5.06%	-0.53%	-4.68%	-1.27%	-5.93%	-1.13%
LSTM (hmm init + reg)	-4.15%	-0.51%	-4.41%	-0.14%	-3.93%	-0.75%	-4.95%	-0.68%
Panel B: Monetary costs of 5% VaR breaches in Long Bonds								
Region	US		EU		UK		Global	
Model	acc. loss per year	avg loss per breach	acc. loss per year	avg loss per breach	acc. loss per year	avg loss per breach	acc. loss per year	avg loss per breach
Classic	-1.71%	-0.37%	-1.31%	-0.18%	-1.23%	-0.24%	-1.69%	-0.37%
HMM	-1.68%	-0.38%	-1.33%	-0.18%	-1.23%	-0.25%	-1.66%	-0.37%
FF (hmm init)	-1.96%	-0.45%	-1.39%	-0.20%	-1.40%	-0.30%	-2.09%	-0.48%
CNN (hmm init)	-2.11%	-0.49%	-1.71%	-0.28%	-1.66%	-0.37%	-2.24%	-0.54%
LSTM (hmm init)	-1.84%	-0.42%	-1.41%	-0.21%	-1.32%	-0.26%	-1.81%	-0.42%
FF (hmm init + reg)	-1.83%	-0.41%	-1.31%	-0.18%	-1.29%	-0.26%	-1.91%	-0.44%
CNN (hmm init + reg)	-1.85%	-0.42%	-1.51%	-0.24%	-1.46%	-0.32%	-1.99%	-0.46%
LSTM (hmm init + reg)	-1.20%	-0.18%	-1.05%	-0.09%	-0.89%	-0.07%	-1.33%	-0.23%

Table 9: Monetary cost of VaR breaches in Equity and Long Bonds. This table displays all losses that exceed the estimated VaR threshold of a model. Two ways of aggregation are given: accumulated loss per year and average loss per breach as calculated from given backtests per region.

To measure how much value the regularized LSTM can add compared to alternative approaches, we compute the annual accumulated costs of breaches as well as the average cost per breach. The results from including the accumulated costs of breaches and the average cost per breach across asset classes (equity and bond) are displayed in Table 9 for the 5% VaR threshold. The regularized LSTM after considering the accumulated costs of breaches or the average cost per breach performs better than the classic approaches (Mean/Variance and HMM) and the other machine learning approaches (CNN and FF).

We find that the magnitudes of differences between regularized LSTM and other approaches are economically significant. For equities the regularized LSTM results in annual accumulated costs of 117-180 basis points less than the classic (Mean/Variance) approach, at which on average, the regularized LSTM produces over US$1 billion less losses per annum for a US$100 billion equity portfolios of large pension fund such as CalPERS that held US$353 billion during fiscal year end 2023 (CalPers, 2024) or PGGM that held €258.7 billion during 2023 (PGGM, 2024). Compared to the HMM approach, the regularized LSTM avoids annual accumulated costs of 37-90 basis points, which is still a substantial amount of money for the vast majority of asset owners. Examining the long bonds, where total returns are naturally lower, the regularized LSTM produces lower annual costs by 25-51 basis points compared to the Mean/Variance and the HMM, which is economically significance for bond investments.

4.3 Size of Input Data

These statistically and economically attractive results have been achieved, however, based on 2,000 days of training data feed. Such "big" amounts of data may not always be available for newer investment strategies. Hence, it is natural to ask if the performance of the regularized neural networks drop when fed with just half the data (i.e. 1,000 days). Apart from reducing statistical power, a period of over 4 years also may comprise less information on downside tail risks. Indeed, the results displayed in Table 10 show that in all context of VaR thresholds and asset classes, the regularized networks trained on 2,000 days substantially outperform and usually dominate their equivalently designed neural networks with half (1,000 days) the training data feed. Hence, the attractive risk management features for HMM initialized and balanced incentive (regularized) LSTMs are likely only available for established discretionary investment strategies where sufficient historical data is available or for entirely rules-based approaches whose history can be replicated ex-post with sufficient confidence.

5 Robustness Tests and Sensitivity Analysis

We further conduct an array of robustness tests and sensitivity analysis to test the robustness of our results and the applicability of neural network regime switching models. As first robustness test, we extend the regularization in a manner that the balancing incentive function of equation (13) has the same marginal weight than the original loss function instead of just half the marginal weight. As shown in Table 11, the performance of both types of regularized LSTMs is essentially equivalent.

Second, we study higher VaR thresholds such as 10% and find the results to be very comparable to the 5% VaR results. Third, we estimate monthly instead of weekly VaR. Accounting for the loss of statistical power in comparison tests due to the lower number of observations, Table 12 shows that monthly VaR estimation has equivalent performance as compared to weekly VaR estimation.

We conduct two sensitivity analysis. First, we set up our neural networks to be generalized by two balancing incentive functions but without HMM initialization. Table 13 displays a comparison of this configuration with the previously shown approaches. The results show that the regularization enhances performance compared to the naive non-regularized and non-initialized models but that both design features are needed to achieve the full performance. In other words, initialization and regularization need to coexist in the design features to increase neural network performance.

Second, we run analytical approaches with $K > 2$ regimes. Adding a third or even fourth regime when asset prices only know two directions leads to substantial instability in the neural networks. The untabulated results tend to depreciate the quality of our results.

Panel A: Equity 1% VaR prediction comparison									
Name	FF (2k vs 1k)			CNN (2k vs 1k)			LSTM (2k vs 1k)		
Region	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom
EuroStoxx 50	1.0	0.4388	-0.33	1.0	0.2484	-0.24	1.0	0.0956	-0.17
MSCI World	1.0	0.0163	-0.17	1.0	0.0389	-0.16	1.0	0.1088	-0.17
FTSE 100	1.0	0.1088	-0.21	1.0	0.4388	-0.29	1.0	0.4388	-0.40
S&P 500	1.0	0.0593	-0.23	1.0	0.0184	-0.19	1.0	0.4671	-0.39
Panel B: Long Bonds 1% VaR prediction comparison									
Name	FF (2k vs 1k)			CNN (2k vs 1k)			LSTM (2k vs 1k)		
Region	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom
EU	1.0	0.6137	-0.29	1.0	0000.2	-0.12	1.0	1.0000	-0.22
Global	1.0	0.0075	-0.05	1.0	0.0495	-0.18	1.0	0.0522	-0.18
UK	1.0	0.2060	-0.12	1.0	0.0125	-0.07	1.0	0.4144	-0.25
US	1.0	0.0116	-0.14	1.0	0.0124	-0.10	1.0	0.0956	-0.14
Panel C: Equity 5% VaR prediction comparison									
Name	FF (2k vs 1k)			CNN (2k vs 1k)			LSTM (2k vs 1k)		
Region	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom
EuroStoxx 50	1.0	0.2395	-0.17	1.0	0.0001	-0.19	1.0	0.0947	-0.23
MSCI World	1.0	0.0555	-0.14	1.0	0.7390	-0.21	1.0	0.1798	-0.16
FTSE 100	1.0	0.2735	-0.21	1.0	0.0588	-0.16	1.0	0.1798	-0.16
S&P 500	1.0	0.5639	-0.19	1.0	0.0455	-0.20	1.0	0.5777	-0.33
Panel D: Long Bonds 5% VaR prediction comparison									
Name	FF (2k vs 1k)			CNN (2k vs 1k)			LSTM (2k vs 1k)		
Region	comp	pvalue	dom	comp	pvalue	dom	comp	pvalue	dom
EU	1.0	0.3361	-0.14	1.0	0.1574	-0.12	1.0	0.4915	-0.12
Global	1.0	0.0001	-0.04	1.0	0.0000	-0.04	1.0	0.7964	-0.11
UK	1.0	0.0017	-0.04	1.0	0.0718	-0.10	1.0	0.6173	-0.14
US	1.0	0.0290	-0.05	1.0	0.0588	-0.12	1.0	0.0736	-0.10

Table 10: Comparison of the performance of the three types of neural network (FF, CNN, and LSTM) with 1000 vs 2000 training days, in each of the four regions (EU,UK,US, and Global). We compare both predictions of 1% VaR and 5% VaR. The comp and dom values are computed equivalently to those in Tables 5 - 8.

6 Conclusions and Future Studies

This study investigates whether and how neural networks can be utilized to estimate the Value at Risk (VaR), especially to estimate the downside (tail) risks in asset management. We introduce a framework architectures which allows neural networks for learning of regime switching with the Hidden Markov Model (HMM) instead of a random seeding (initialization). Furthermore, we also apply the balancing incentive function to account for both bull and bear markets probabilities. By doing so, we are able to apply and evaluate state-of-the-art temporal neural network models (CNN and LSTM) in the domain of regime switching to reduce breaches of VaR at 1% and 5% thresholds. Employing equity markets and long-term (7-10 years) bonds as test of financial assets in international settings in the US, Euro area, UK, and World (Global)

Panel A: 1% VaR thresholds for Equity

Index Model	S&P 500 breaches out of 1290		EuroStoxx 50 breaches out of 1186		FTSE 100 breaches out of 1239		MSCI World breaches out of 1343	
FF (hmm init + reg)	22	1.7%	18	1.5%	19	1.5%	30	2.2%
CNN (hmm init + reg)	21	1.6%	17	1.4%	21	1.7%	32	2.4%
LSTM (hmm init + reg)	18	1.4%	12	1.0%	15	1.2%	23	1.7%
FF (hmm init + reg01)	21	1.6%	19	1.6%	24	1.9%	28	2.1%
CNN (hmm init + reg01)	21	1.6%	17	1.4%	22	1.8%	32	2.4%
LSTM (hmm init + reg01)	20	1.6%	13	1.1%	16	1.3%	25	1.9%

Panel B: 1% VaR thresholds for Long Bonds

Index Model	US Long Bonds breaches out of 1290		German Long Bonds breaches out of 1186		UK Long Bonds breaches out of 1239		Global Long Bonds breaches out of 1343	
FF (hmm init + reg)	29	2.2%	24	2.0%	24	1.9%	37	2.8%
CNN (hmm init + reg)	29	2.2%	26	2.2%	29	2.3%	33	2.5%
LSTM (hmm init + reg)	14	1.1%	18	1.5%	16	1.3%	17	1.3%
FF (hmm init + reg01)	31	2.4%	26	2.2%	23	1.9%	37	2.8%
CNN (hmm init + reg01)	31	2.4%	26	2.2%	30	2.4%	35	2.6%
LSTM (hmm init + reg01)	14	1.1%	16	1.3%	14	1.1%	20	1.5%

Panel C: 5% VaR thresholds for Equity

Index Model	S&P 500 breaches out of 1290		EuroStoxx 50 breaches out of 1186		FTSE 100 breaches out of 1239		MSCI World breaches out of 1343	
FF (hmm init + reg)	63	4.9%	59	5.0%	58	4.7%	80	6.0%
CNN (hmm init + reg)	60	4.7%	54	4.6%	58	4.7%	90	6.7%
LSTM (hmm init + reg)	49	3.8%	44	3.7%	43	3.5%	63	4.7%
FF (hmm init + reg01)	64	5.0%	57	4.8%	56	4.5%	82	6.1%
CNN (hmm init + reg01)	59	4.6%	51	4.3%	59	4.8%	91	6.8%
LSTM (hmm init + reg01)	56	4.3%	45	3.8%	43	3.5%	69	5.1%

Panel D: 5% VaR thresholds for Long Bonds

Index Model	US Long Bonds breaches out of 1290		German Long Bonds breaches out of 1186		UK Long Bonds breaches out of 1239		Global Long Bonds breaches out of 1343	
FF (hmm init + reg)	79	6.1%	76	6.4%	70	5.6%	83	6.2%
CNN (hmm init + reg)	78	6.0%	98	8.3%	82	6.6%	83	6.2%
LSTM (hmm init + reg)	58	4.5%	64	5.4%	51	4.1%	65	4.8%
FF (hmm init + reg01)	79	6.1%	79	6.7%	71	5.7%	83	6.2%
CNN (hmm init + reg01)	81	6.3%	96	8.1%	86	6.9%	86	6.4%
LSTM (hmm init + reg01)	59	4.6%	65	5.5%	51	4.1%	62	4.6%

Table 11: VaR breaches compared when using basic regularization versus [0,1]-normalized regularization term.

markets up to 1,343 weeks sample horizon ending in mid-June 2020, we investigate the performance of neural networks to predict VaR breaches across three design steps.

First, we compare neural networks with random seeding with networks that are initialized via estimations from the Hidden Markov Model (HMM) and find the latter to outperform in terms of lower frequency of VaR breaches (i.e. the realized return falling short of the estimated VaR threshold), especially for equity markets which generally have greater tail risks than the bond markets. This is due to the fact that the regime switching model from HMM provide a critical initializiation for the neural networks models to predict the potential extreme outcomes that ensure a full replicability, especially more prevalent for the equity markets than the bond markets. In contrast, based on our experiments, the random initiation has a chance of getting stuck in a local minimum and therefore produces worse results and results that are not fully replicable.

Second, we balance the incentive structure of the loss function of our networks by adding a second objective to the training instructions so that the neural networks optimize for accuracy while also aiming to stay in

Panel A: 1% VaR thresholds for Equity								
Index Model	S&P 500 breaches out of 321		EuroStoxx 50 breaches out of 295		FTSE 100 breaches out of 309		MSCI World breaches out of 335	
FF (hmm init + reg, monthly)	9	2.8% (1.7%)	10	3.4% (1.5%)	6	1.9% (1.5%)	7	2.1% (2.2%)
CNN (hmm init + reg, monthly)	5	1.6% (1.6%)	8	2.7% (1.4%)	5	1.6% (1.7%)	10	3.0% (2.4%)
LSTM (hmm init + reg, monthly)	5	1.6% (1.4%)	7	2.4% (1.0%)	5	1.6% (1.2%)	7	2.1% (1.7%)
Panel B: 1% VaR thresholds for Long Bonds								
Index Model	US Long Bonds breaches out of 1290		German Long Bonds breaches out of 1186		UK Long Bonds breaches out of 1239		Global Long Bonds breaches out of 1343	
FF (hmm init + reg, monthly)	3	0.9% (2.2%)	3	1.0% (2.0%)	3	1.0% (1.9%)	4	1.2% (2.8%)
CNN (hmm init + reg, monthly)	3	0.9% (2.2%)	7	2.4% (2.2%)	4	1.3% (2.3%)	4	1.2% (2.5%)
LSTM (hmm init + reg, monthly)	2	0.6% (2.2%)	4	1.4% (1.5%)	2	0.6% (1.3%)	3	0.9% (1.3%)
Panel C: 5% VaR thresholds for Equity								
Index Model	S&P 500 breaches out of 321		EuroStoxx 50 breaches out of 295		FTSE 100 breaches out of 309		MSCI World breaches out of 335	
FF (hmm init + reg, monthly)	17	5.3% (4.9%)	16	5.4% (5.0%)	13	4.2% (4.7%)	22	6.6% (6.0%)
CNN (hmm init + reg, monthly)	15	4.7% (4.7%)	15	5.1% (4.6%)	15	4.9% (4.7%)	19	5.7% (6.7%)
LSTM (hmm init + reg, monthly)	15	4.7% (3.8%)	17	5.8% (3.7%)	7	2.3% (3.5%)	17	5.1% (4.7%)
Panel D: 5% VaR thresholds for Long Bonds								
Index Model	US Long Bonds breaches out of 1290		German Long Bonds breaches out of 1186		UK Long Bonds breaches out of 1239		Global Long Bonds breaches out of 1343	
FF (hmm init + reg, monthly)	19	5.9% (6.1%)	20	6.8% (6.4%)	9	2.9% (5.6%)	17	5.1% (6.2%)
CNN (hmm init + reg, monthly)	18	5.6% (6.0%)	24	8.1% (8.3%)	16	5.2% (6.6%)	19	5.7% (6.2%)
LSTM (hmm init + reg, monthly)	11	3.4% (4.5%)	17	5.8% (5.4%)	9	2.9% (4.1%)	11	3.3% (4.8%)

Table 12: VaR breaches when estimating on a monthly basis. The number in brackets shows the weekly percentage for comparison.

empirically realistic economic regime distributions (i.e. bull vs. bear market frequencies). In particular this design feature enables the balanced incentive recurrent neural network (RNN), specifically the Long Short-Term Memory (LSTM) to outperform the other neural network or established approaches (Mean/Variance and HMM only) by statistically and economically significant levels.

Third, we employ a half of our training data set of 2,000 days (i.e. 1,000 days). We find our neural networks when fed with substantially less data perform significantly worse. This highlights a crucial weakness of neural networks because their ability to predict the VaR breaches depends on large data sets and a longer time frame (Cohen, 2023).

Our study contributes to the literature and is pertinent for asset managers and institutional investors by providing evidence that machine learning, specifically LSTM with HMM initialization and balanced incentive, can provide better VaR estimates to guide asset managers or insurance companies to manage their portfolios against the tail risk. However, such advancement depends on the amount of available historical data feeds. Future work can include the investigation of asset classes beyond equity and bond markets (i.e., commodity, derivatives, real estates, etc.) and may also investigate the question if the presented approach is applicable to Value at Risk estimation for financial institutions (i.e., banks) assets and to predict corporate bankruptcy (Nguyen et al., 2023a; Charalambous et al., 2022). Technically, our study aims to further optimize and stabilize the neural network parametrization in estimating all types of tail risks. We present evidence on how

Panel A: 1% VaR thresholds for Equity

Index Model	S&P 500 breaches out of 1290		EuroStoxx 50 breaches out of 1186		FTSE 100 breaches out of 1239		MSCI World breaches out of 1343	
Classic	26	2.0%	20	1.7%	26	2.1%	39	2.9%
HMM	17	1.3%	12	1.0%	19	1.5%	22	1.6%
FF (no hmm)	31	2.4%	22	1.9%	29	2.3%	44	3.3%
CNN (no hmm)	33	2.6%	24	2.0%	32	2.6%	49	3.6%
LSTM (no hmm)	518	40.2%	475	40.1%	513	41.4%	539	40.1%
FF (hmm init)	20	1.6%	18	1.5%	23	1.9%	35	2.6%
CNN (hmm init)	26	2.0%	19	1.6%	24	1.9%	35	2.6%
LSTM (hmm init)	20	1.6%	17	1.4%	22	1.8%	28	2.1%
FF (hmm init + reg)	22	1.7%	18	1.5%	19	1.5%	30	2.2%
CNN (hmm init + reg)	21	1.6%	17	1.4%	21	1.7%	32	2.4%
LSTM (hmm init + reg)	18	1.4%	12	1.0%	15	1.2%	23	1.7%
FF (no hmm + reg)	26	2.0%	19	1.6%	32	2.6%	41	3.1%
CNN (no hmm + reg)	33	2.6%	20	1.7%	31	2.5%	50	3.7%
LSTM (no hmm + reg)	524	40.6%	494	41.7%	538	43.4%	552	41.1%

Panel B: 1% VaR thresholds for Long Bonds

Index Model	US Long Bonds breaches out of 1290		German Long Bonds breaches out of 1186		UK Long Bonds breaches out of 1239		Global Long Bonds breaches out of 1343	
Classic	30	2.3%	26	2.2%	23	1.9%	31	2.3%
HMM	20	1.6%	19	1.6%	16	1.3%	22	1.6%
FF (no hmm)	51	4.0%	46	3.9%	44	3.6%	44	3.3%
CNN (no hmm)	60	4.7%	38	3.2%	41	3.3%	64	4.8%
LSTM (no hmm)	496	38.4%	388	32.7%	478	38.6%	563	41.9%
FF (hmm init)	32	2.5%	28	2.4%	27	2.2%	41	3.1%
CNN (hmm init)	34	2.6%	35	3.0%	31	2.5%	42	3.1%
LSTM (hmm init)	27	2.1%	21	1.8%	21	1.7%	32	2.4%
FF (hmm init + reg)	29	2.2%	24	2.0%	24	1.9%	37	2.8%
CNN (hmm init + reg)	29	2.2%	26	2.2%	29	2.3%	33	2.5%
LSTM (hmm init + reg)	14	1.1%	18	1.5%	16	1.3%	17	1.3%
FF (no hmm + reg)	34	2.6%	29	2.4%	27	2.2%	49	3.6%
CNN (no hmm + reg)	53	4.1%	38	3.2%	41	3.3%	61	4.5%
LSTM (no hmm + reg)	521	40.4%	410	34.6%	511	41.2%	573	42.7%

Panel C: 5% VaR thresholds for Equity

Index Model	S&P 500 breaches out of 1290		EuroStoxx 50 breaches out of 1186		FTSE 100 breaches out of 1239		MSCI World breaches out of 1343	
Classic	63	4.9%	61	5.1%	60	4.8%	84	6.3%
HMM	61	4.7%	63	5.3%	54	4.4%	75	5.6%
FF (no hmm)	71	5.5%	62	5.2%	65	5.2%	94	7.0%
CNN (no hmm)	76	5.9%	71	6.0%	67	5.4%	102	7.6%
LSTM (no hmm)	534	41.4%	490	41.3%	535	43.2%	562	41.8%
FF (hmm init)	65	5.0%	62	5.2%	55	4.4%	84	6.3%
CNN (hmm init)	69	5.3%	77	6.5%	63	5.1%	96	7.1%
LSTM (hmm init)	62	4.8%	60	5.1%	52	4.2%	81	6.0%
FF (hmm init + reg)	63	4.9%	59	5.0%	58	4.7%	80	6.0%
CNN (hmm init + reg)	60	4.7%	54	4.6%	58	4.7%	90	6.7%
LSTM (hmm init + reg)	49	3.8%	44	3.7%	43	3.5%	63	4.7%
FF (no hmm + reg)	65	5.0%	60	5.1%	63	5.1%	96	7.1%
CNN (no hmm + reg)	73	5.7%	65	5.5%	76	6.1%	102	7.6%
LSTM (no hmm + reg)	546	42.3%	509	42.9%	554	44.7%	566	42.1%

Panel D: 5% VaR thresholds for Long Bonds

Index Model	US Long Bonds breaches out of 1290		German Long Bonds breaches out of 1186		UK Long Bonds breaches out of 1239		Global Long Bonds breaches out of 1343	
Classic	72	5.6%	78	6.6%	70	5.6%	71	5.3%
HMM	75	5.8%	85	7.2%	72	5.8%	75	5.6%
FF (no hmm)	103	8.0%	116	9.8%	106	8.6%	98	7.3%
CNN (no hmm)	114	8.8%	98	8.3%	96	7.7%	118	8.8%
LSTM (no hmm)	550	42.6%	428	36.1%	519	41.9%	591	44.0%
FF (hmm init)	80	6.2%	83	7.0%	79	6.4%	88	6.6%
CNN (hmm init)	85	6.6%	104	8.8%	88	7.1%	96	7.1%
LSTM (hmm init)	81	6.3%	91	7.7%	73	5.9%	79	5.9%
FF (hmm init + reg)	79	6.1%	76	6.4%	70	5.6%	83	6.2%
CNN (hmm init + reg)	78	6.0%	98	8.3%	82	6.6%	83	6.2%
LSTM (hmm init + reg)	58	4.5%	64	5.4%	51	4.1%	65	4.8%
FF (no hmm + reg)	83	6.4%	84	7.1%	84	6.8%	101	7.5%
CNN (no hmm + reg)	107	8.3%	109	9.2%	106	8.6%	116	8.6%
LSTM (no hmm + reg)	563	43.6%	454	38.3%	543	43.8%	604	45.0%

Table 13: Sensitiviy analysis for VaR breaches: comparison of all possible configurations

to best initialize the weights of the network and the impact of the amount of data feed needed to achieve more stable and better results to predict the downside risk.

Conflict of Interest Statement

As of 2025, the author(s) declare no conflict of interest (financial and non-financial) with respect to the presented analysis. This study does not use Generative AI and AI-assisted technologies in the writing process of this manuscript. Study is conducted without human subject. Therefore, there is no informed consent needed.

References

Abad, P., Benito, S., and López, C. (2014). A comprehensive review of value at risk methodologies. The Spanish Review of Financial Economics, 12(1):15–32.

Agarwal, V. and Naik, N. Y. (2004). Risks and portfolio decisions involving hedge funds. The Review of Financial Studies, 17(1):63–98.

Albuquerque, R. (2012). Skewness in stock returns: Reconciling the evidence on firm versus aggregate returns. The Review of Financial Studies, 25(5):1630–1673.

Amini, S., Elmore, R., Öztekin, Ö., and Strauss, J. (2021). Can machines learn capital structure dynamics? Journal of Corporate Finance, 70:102073.

Ang, A. and Bekaert, G. (2002). International asset allocation with regime shifts. The Review of Financial Studies, 15(4):1137–1187.

Aubry, M., Kräussl, R., Manso, G., and Spaenjers, C. (2019). Machine learning, human experts, and the valuation of real assets. HEC Paris Research Paper No. FIN-2019-1332.

Avramov, D., Cheng, S., and Metzker, L. (2023). Machine learning vs. economic restrictions: Evidence from stock return predictability. Management Science, 69(5):2587–2619.

Azevedo, V. and Hoegner, C. (2023). Enhancing stock market anomalies with machine learning. Review of Quantitative Finance and Accounting, 60(1):195–230.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. The annals of mathematical statistics, 41(1):164–171.

Bianchi, D., Büchner, M., and Tamoni, A. (2021). Bond risk premiums with machine learning. The Review of Financial Studies, 34(2):1046–1089.

Billio, M., Getmansky, M., Lo, A. W., and Pelizzon, L. (2012). Econometric measures of connectedness and systemic risk in the finance and insurance sectors. Journal of financial economics, 104(3):535–559.

Billio, M. and Pelizzon, L. (2000). Value-at-risk: a multivariate switching regime approach. Journal of Empirical Finance, 7(5):531–554.

Bishop, C. M. et al. (1995). Neural networks for pattern recognition. Oxford university press.

Borer, D., Perera, D., Fauzi, F., and Chau, T. N. (2023). Identifying systemic risk of assets during international financial crises using value at risk elasticities. International Review of Financial Analysis, 90:102832.

CalPers (2024). 2022-23 Annual Comprehensive Financial Report. Available at https://www.sco.ca.gov/Files-ARD/ACFR/acfr23web.pdf.

Cao, Y., Liu, X., and Zhai, J. (2021). Option valuation under no-arbitrage constraints with neural networks. European Journal of Operational Research, 293(1):361–374.

Castillo, B., León, Á., and Ñíguez, T.-M. (2021). Backtesting var under the covid-19 sudden changes in volatility. Finance Research Letters, 43:102024.

Charalambous, C., Martzoukos, S. H., and Taoushianis, Z. (2022). Estimating corporate bankruptcy forecasting models by maximizing discriminatory power. Review of Quantitative Finance and Accounting, 58(1):297–328.

Chen, C.-N. and Lin, C.-H. (2022). Optimal carry trade portfolio choice under regime shifts. Review of Quantitative Finance and Accounting, 59(2):483–506.

Chen, L., Pelger, M., and Zhu, J. (2024). Deep learning in asset pricing. Management Science, 70(2):714–750.

Cohen, G. (2023). Intraday algorithmic trading strategies for cryptocurrencies. Review of Quantitative Finance and Accounting, 61(1):395–409.

Cosma, S., Rimo, G., and Torluccio, G. (2023). Knowledge mapping of model risk in banking. International Review of Financial Analysis, page 102800.

Danielsson, J., Macrae, R., and Uthemann, A. (2022). Artificial intelligence and systemic risk. Journal of Banking & Finance, 140:106290.

De Nard, G., Hediger, S., and Leippold, M. (2020). Subsampled factor models for asset pricing: The rise of vasa. Available at SSRN 3557957.

Easley, D., López de Prado, M., O'Hara, M., and Zhang, Z. (2021). Microstructure in the machine age. The Review of Financial Studies, 34(7):3316–3363.

Fallahgoul, H., Franstianto, V., and Lin, X. (2021). Asset pricing with neural networks: A variable significant test. Available at SSRN 3889777.

Fedenia, M., Ronen, T., and Nam, S. (2024). Machine learning and trade direction classification: insights from the corporate bond market. Review of Quantitative Finance and Accounting, 63(1):1–36.

Fletcher, J. (2024). An examination of linear factor models in uk stock returns in the presence of dynamic trading. Review of Quantitative Finance and Accounting, 63(3):1121–1147.

Fukuyama, H., Tsionas, M., and Tan, Y. (2024). Bayesian artificial neural networks for frontier efficiency analysis. Annals of Operations Research, 2342:1865–1904.

Giglio, S. and Xiu, D. (2021). Asset pricing with omitted factors. Journal of Political Economy, 129(7):1947–1990.

Gkillas, K. and Vasiliadis, L. (2025). How does the brexit vote affect tail risk? an extreme value approach for the uk financial markets. Review of Quantitative Finance Accounting, Forthcoming:https://doi.org/10.1007/s11156–025–01422–4.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, pages 249–256.

Götze, T., Gürtler, M., and Witowski, E. (2020). Improving cat bond pricing models via machine learning. Journal of Asset Management, 21(5):428–446.

Graves, A. (2013). Generating sequences with recurrent neural networks. arXiv springrint arXiv:1308.0850.

Gu, S., Kelly, B., and Xiu, D. (2020). Empirical asset pricing via machine learning. The Review of Financial Studies, 33(5):2223–2273.

Gu, S., Kelly, B., and Xiu, D. (2021). Autoencoder asset pricing models. Journal of Econometrics, 222(1):429–450.

Hann, T. H. and Steurer, E. (1996). Much ado about nothing? Exchange rate forecasting: Neural networks vs. linear models using monthly and weekly data. Neurocomputing, 10(4):323–339.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8):1735–1780.

Horel, E. and Giesecke, K. (2019). Towards explainable AI: Significance tests for neural networks. arXiv springrint arXiv:1902.06021.

Hossain, M., Lobo, G. J., and Mitra, S. (2023). Firm-level political risk and corporate tax avoidance. Review of Quantitative Finance and Accounting, 60(1):295–327.

Hunt, J., Myers, J., and Myers, L. (2019). Improving earnings predictions with machine learning. Working paper.

Johnes, G., Tsionas, M., and Izzeldin, M. (2025). Multi-stage stochastic frontier analysis for simple networks. International Transactions in Operational Research, 32(5):2497–2522.

Jorion, P. (2006). Value at Risk. McGraw-Hill Education, 3rd edition.

Kelly, B. T., Pruitt, S., and Su, Y. (2019). Characteristics are covariances: A unified model of risk and return. Journal of Financial Economics, 134(3):501–524.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv springrint arXiv:1412.6980.

Kozak, S., Nagel, S., and Santosh, S. (2020). Shrinking the cross-section. Journal of Financial Economics, 135(2):271–292.

Kozhan, R., Neuberger, A., and Schneider, P. (2013). The skew risk premium in the equity index market. The Review of Financial Studies, 26(9):2174–2203.

Krogh, A. and Hertz, J. A. (1992). A simple weight decay can improve generalization. In Advances in Neural Information Processing Systems, pages 950–957.

Lajili, K., Mohsni, S., Polizzi, S., and Scannella, E. (2024). Value relevance of bank credit risk disclosure:“interventionist” and “non-interventionist” supervisory regimes. Review of Quantitative Finance and Accounting, pages 1–34.

Lee, Y., Thompson, J. R., Kim, J. H., Kim, W. C., and Fabozzi, F. A. (2023). An overview of machine learning for asset management. Journal of Portfolio Management, 49(9).

Liaras, E., Nerantzidis, M., and Alexandridis, A. (2024). Machine learning in accounting and finance research: a literature review. Review of Quantitative Finance and Accounting, 63(4):1431–1471.

Liu, R. and Pun, C. S. (2022). Machine-learning-enhanced systemic risk measure: A two-step supervised learning approach. Journal of Banking & Finance, 136:106416.

MacLean, L. and Zhao, Y. (2022). Kelly investing with downside risk control in a regime-switching market. Quantitative Finance, 22(1):75–94.

Mahmood, F., Ahmed, Z., Hussain, N., and Ben-Zaied, Y. (2023). Working capital financing and firm performance: A machine learning approach. Review of Quantitative Finance and Accounting, pages 1–36.

Mamatzakis, E., Patel, P., and Tsionas, M. (2024). A bayesian learning model of hedge fund performance. Annals of Operations Research, 333:201–238.

Marcus, G. (2018). Deep learning: A critical appraisal. arXiv preprint arXiv:1801.00631.

Maringer, D. and Ramtohul, T. (2012). Regime-switching recurrent reinforcement learning for investment decision making. Computational Management Science, 9:89–107.

Migliavacca, M., Goodell, J. W., and Paltrinieri, A. (2023). A bibliometric review of portfolio diversification literature. International Review of Financial Analysis, page 102836.

Mishra, A., Srivastava, P., and Chakrabarti, A. S. (2022). 'too central to fail'firms in bi-layered financial networks: linkages in the us corporate bond and stock markets. Quantitative Finance, 22(5):943–971.

Mögel, B. and Auer, B. R. (2018). How accurate are modern value-at-risk estimators derived from extreme value theory? Review of Quantitative Finance and Accounting, 50:979–1030.

Moody, J. and Wu, L. (1997). Optimization of trading systems and portfolios. In Proceedings of the IEEE/IAFE 1997 computational intelligence for financial engineering (CIFEr), pages 300–307. IEEE.

Na, A. and Wan, J. (2023). Efficient pricing and hedging of high dimensional american options using recurrent networks. arXiv preprint arXiv:2301.08232.

Nguyen, H. H., Viviani, J.-L., and Ben Jabeur, S. (2023a). Bankruptcy prediction using machine learning and shapley additive explanations. Review of Quantitative Finance and Accounting, pages 1–42.

Nguyen, Q., Diaz-Rainey, I., Kuruppuarachchi, D., McCarten, M., and Tan, E. K. (2023b). Climate transition risk in us loan portfolios: Are all banks the same? International Review of Financial Analysis, 85:102401.

Nieto, M. R. and Ruiz, E. (2016). Frontiers in VaR forecasting and backtesting. International Journal of Forecasting, 32(2):475–501.

Novykov, V., Bilson, C., Gepp, A., Harris, G., and Vanstone, B. J. (2025). Deep learning applications in investment portfolio management: a systematic literature review. Journal of Accounting Literature, 47(2):245–276.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499.

Pan, S., Long, S. C., Wang, Y., and Xie, Y. (2023). Nonlinear asset pricing in chinese stock market: A deep learning approach. International Review of Financial Analysis, 87:102627.

Parisi, L. and Manaog, M. L. (2025). Optimal machine learning-and deep learning-driven algorithms for predicting the future value of investments: A systematic review and meta-analysis. Engineering Applications of Artificial Intelligence, 142:109924.

Pearson, N. D. (2011). Risk budgeting: portfolio problem solving with value-at-risk. John Wiley & Sons.

PGGM (2024). Annual Report of PGGM N.V. Available at https://www.pggm.nl/media/ydmhtls5/pggm-nv-annual-report-2023.pdf.

Pourkhanali, A., Tafakori, L., and Bee, M. (2023). Forecasting value-at-risk using functional volatility incorporating an exogenous effect. International Review of Financial Analysis, 89:102803.

Sadhwani, A., Giesecke, K., and Sirignano, J. (2021). Deep learning for mortgage risk. Journal of Financial Econometrics, 19(2):313–368.

Sáenz, J. V., Quiroga, F. M., and Bariviera, A. F. (2023). Data vs. information: Using clustering techniques to enhance stock returns forecasting. International Review of Financial Analysis, 88:102657.

Schmeding, C., Klawunn, M., and Weisheit, S. (2019). Maschinelles lernen bei der entwicklung von wertsicherungsstrategien. Zeitschrift für das gesamte Kreditwesen, 72(2).

Sehgal, S., Agrawal, T. J., and Deisting, F. (2025). The tale of two tails and stock returns for two major emerging markets. Review of Quantitative Finance and Accounting, 64(1):163–189.

Simlai, P. (2021). Accrual mispricing, value-at-risk, and expected stock returns. Review of Quantitative Finance and Accounting, 57(4):1487–1517.

Timmermann, A. (2000). Moments of Markov switching models. Journal of Econometrics, 96(1):75–111.

Tsionas, M., Parmeter, C., and Zelenyuk, V. (2024). Bayesian artificial neural networks for frontier efficiency analysis. Journal of Econometrics, 238(2):105491.

Uyar, A., Al-Shaer, H., Kusey, C., and Karaman, A. (2025). Board characteristics and aggressive csr engagement: Do csr committees provoke or restrain? Review of Quantitative Finance Accounting, forthcoming:https://doi.org/10.1007/s11156–025–01403–7.

Verstyuk, S. (2020). Modeling multivariate time series in economics: From auto-regressions to recurrent neural networks. Available at SSRN 3589337.

Wang, Y., Andreeva, G., and Martin-Barragan, B. (2023). Machine learning approaches to forecasting cryptocurrency volatility: Considering internal and external determinants. International Review of Financial Analysis, 90:102914.

Wang, Z., He, T., Ren, X., and Huynh, L. D. T. (2024). Robust portfolio strategies based on reference points for personal experience and upward pacesetters. Review of Quantitative Finance and Accounting, 63(3):863–887.

Wu, C.-C., Chen, W.-P., and Korsakul, N. (2025). Extreme risk spillover in the equity markets: Evidence from the us–china trade war. Review of Quantitative Finance and Accounting, pages 1–26.

Yousefi, H. and Najand, M. (2022). Geographical diversification using etfs: Multinational evidence from covid-19 pandemic. International Review of Financial Analysis, 83:102261.

Zhang, H., Dauphin, Y. N., and Ma, T. (2019). Fixup initialization: Residual learning without normalization. arXiv preprint arXiv:1901.09321.

Zhang, N., Su, X., and Qi, S. (2023). An empirical investigation of multiperiod tail risk forecasting models. International Review of Financial Analysis, 86:102498.