Tail-risk protection: Machine Learning meets modern Econometrics

Bruno Spilak [*], Wolfgang Karl Härdle [†]

October 8, 2020

Abstract

Tail risk protection is in the focus of the financial industry and requires solid mathematical and statistical tools, especially when a trading strategy is derived. Recent hype driven by machine learning (ML) mechanisms has raised the necessity to display and understand the functionality of ML tools. In this paper, we present a dynamic tail risk protection strategy that targets a maximum predefined level of risk measured by Value-At-Risk while controlling for participation in bull market regimes. We propose different weak classifiers, parametric and non-parametric, that estimate the exceedance probability of the risk level from which we derive trading signals in order to hedge tail events. We then compare the different approaches both with statistical and trading strategy performance, finally we propose an ensemble classifier that produces a meta tail risk protection strategy improving both generalization and trading performance.

[*] corresponding author, IRTG 1792, School of Business and Economics, Humboldt-Universität zu Berlin, Dorotheenstr. 1, 10117 Berlin, Germany. Email: bruno.spilak@hu-berlin.de

[†] W.I.S.E. - Wang Yanan Institute for Studies in Economics, Xiamen University, Xiamen, 361005, Fujian, China
C.A.S.E. - Center for applied Statistics and Economics, Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany
Singapore Management University, 50 Stamford Road, 178899 Singapore, Singapore
Department of Mathematics and Physics, Charles University Prague, Ke Karlovu 2027/3, 12116 Praha 2, Czech
Email: haerdle@wiwi.hu-berlin.de

1 Introduction

"Black swan events" are occurring from time to time. After 2008, investors in the DAX had to wait 5 years before recovering their loss. The Nikkei stock index still has not yet recovered, even 25 years after the bubble burst. The cryptocurrency market has plunged drastically at the end of 2017. What choices are left to investors in these situations ?

Indeed, it is one of the main challenges of quantitative finance to build models and develop tools able to protect investments from such extreme events (Bollerslev and Todorov; 2011). The first risk management strategy addressing the latter is diversification but Longin and Solnik (2001) show that when financial markets exhibit huge downturn periods, correlation significantly increase, hence are counteracting the effect of portfolio diversification. Another strategy for tail risk protection is option based, where the risk manager is long in put options. Yet, loss aversion leads to high prices of put options (Kozhan et al. (2013), Packham et al. (2017)). A solution to escape this dilemma is to build a strategy with dynamic asset allocation. The risk manager creates an asymmetric risk profile with participation in upside market and protection against severe loss. This trading strategy is often called tail risk protection strategy by practitioners and is much more cost efficient and flexible than the options strategy (Franke et al.; 2019). In order to pursue such a strategy, one needs to predict distributional properties of the portfolio. This, of course, is obviously impossible since not only distributions but also their parameters change in time. This motivates our paper where we hint on comparing Econometrics tools, Machine learning (ML) and modern Local Parametric Approach (LPA). We present a dynamic tail risk protection strategy that targets a maximum predefined level of risk measured by Value-At-Risk while controlling for participation in bull market regimes. We propose different weak classifiers, parametric, based on GARCH tool, and non-parametric using ML, that estimate the exceedance probability of the risk level from which we derive trading signals in order to hedge tail events.

The GARCH tool with normally distributed innovations allows us to catch volatility clusters and yields good volatility forecasts. The application of Extreme Value Theory (EVT) to GARCH residuals provides insight into the tail probabilities, that is, the likeliness of an extreme loss. Nevertheless, econometric models rely on strong assumptions and cannot deal with structural breaks that happen in financial time series.

ML has been successful in many applications during the last years thanks to its generalization power on large datasets, but, in quantitative finance, in particular for financial returns forecasting, ML did not prove its superiority in comparison with its application to text analysis or image recognition, where Deep Learning (DL) tools became state-of-the-art. Moreover, the black box effect of DL makes the industry reluctant to invest in such models. In theory, DL can extract information from non-linear relations in high dimensional space, so for quantitative finance practitioners, it should be natural to use the machine learning tool box. Nevertheless, DL models suffer from overfitting and can be very difficult to train. If they offer similar prediction accuracy than econometrics, it might be preferable to use the latter with regards to its interpretability, lower complexity and cheaper computational costs.

The cryptocurrency market has experienced an exponential growth during 2017, where BTC peaked at \$19 783.06 on December 17th 2017 and dropped below \$14 000 24 hours later, losing one third of its value. One year later, on December 7th 2018, BTC price briefly dipped below

$3 300, a 76% drop from the previous year and a 15-month low. At the time of writing, the cryptocurrency market is experiencing a new large uptrend as swathes of institutional investors are gaining new interest because of the launch of new futures contracts on both regulated and unregulated cryptocurrency exchanges. Central banks also reacted to private initiatives of launching digital currencies, such as Facebook's Libra, as they may have the potential to dilute the main power of central banks - to control the supply of money to the economy. With high volatility and new interest in the cryptocurrency market, the BTC ecosystem is a perfect environment to show the effectiveness of tail risk strategy. Indeed, its high volatility promises high gains in upward movement of the price. Our goal is then to protect investors from large downturns of the market through an accurate prediction of tail risk.

Since Basel committee on Banking Supervision Amendment to incorporate market risk (1996), regulators imposed the use of certain metrics to measure the risk of investments, such as Value-At-Risk (VaR) and Expected-Shortfall (ES). In order to meet both investors' and regulators' will, we build a risk protection strategy that controls the VaR of our portfolio by ensuring that it is below a certain level, denoted as target VaR.

As high volatility segments often precede market swings, the GARCH model is a natural tool for tail risk protection, since GARCH catches these volatility clusters. Nevertheless, a volatility based risk management strategy forces us to divest in such period, reducing alpha possibilities in case of positive movement. Since financial returns have heavy tails, the EVTGARCH (McNeil and Frey; 2000) allows us to improve our forecast of the tail event direction. Indeed, seeking alpha, excess return over a benchmark, has become more challenging for banks and institutional investors which follow strong regulations based on VaR estimates. We address this problem here by focusing on dynamic risk management based on econometrics, explaining stylized facts of financial time series with parametric approaches, and non- or local parametric methods, in particular the LPA from Spokoiny (2009a), Multilayer Perceptron (MLP) and Long Short-Term Memory (LSTM) neural networks, which catches non-linear features with memory.

Our contribution is a rigorous comparison of GARCH and ML based tools in the context of extreme loss prediction based on their forecast in a classification scheme. On top, we show how to use standard Ensemble method as a meta classifier that produces a hybrid trail risk protection strategy, improving both generalization and trading performance by taking advantage of each approach. Finally, we evaluate our strategy with a realistic backtest including trading fees, by comparing it with classical buy-and-hold benchmark and other recent machine learning oriented tail risk protection strategies, such as the constant target VaR from Rickenberg (2019) and the Varspread strategy from Packham et al. (2017).

The results showed here will certainly motivate practitioners to apply ML techniques in order to improve GARCH performance. We provide comparison metrics such as forecast error, classification metrics and backtest results of our tail risk strategies on the cryptocurrency market with BTC investment. We also provide robustness checks through cross-validation.

This paper is structured as follows. First we review the current literature of our subject, then we explain the trading strategy we aim to build. In the third and fourth part, we explain the theoretical models used to build our strategy. In the final section, we present our results.

2 Background & Literature review

Volatility as a risk measure ? A good risk measure must be tailored to the investor's preferences which are often unknown in practice. Starting from preferences, practitioners often make assumptions in order to build an "optimal risk measure", corresponding to an imagined investor's goal. A large part of the financial literature studies volatility as a risk measure, since it is nicely tied to Gaussian and LS techniques. For example, the VIX, referred to as the "investor fear gauge", is often taken as a sentiment indicator since volatility reflects investors' aversion to risk. The mean-variance portfolio, developed by Markowitz, is built under such assumptions where the weights of the risky assets included in the portfolio are derived from their volatility. Nevertheless, financial returns often have fat-tails and are not normally distributed, which Markowitz theory does not account for. Finally, volatility, being symmetric, is not realistic as a risk measure, since it does not take into consideration investors' loss aversion, weighting equally volatility associated with gains and losses. Most investors are more concerned about downside risk, or losses, rather than volatility (Bollerslev et al.; 2015). In this paper, our goal is to build a trading strategy avoiding large losses or tail risk which is better suited for loss averse investors.

Packham et al. (2017) showed that a trading strategy accounting for tail risk can outperform simple buy-and-hold and traditional portfolio protection strategies. By using models such as GARCH with normally distributed innovations and GARCH with innovations following a Generalized Pareto Distribution (GPD), they built a new criterion for riskiness defined as the evolution of the estimated Value-At-Risk (VaR) spread between the two models. Thus, in period of increasing tail risk, this spread is significantly different from 0 which allows the trader to take adequate decisions. We denote this strategy as the Varspread strategy. Rickenberg (2019) compares different risk measure such as volatility, VaR and Conditional-Value-at-Risk (CVaR), also named Expected Shortfall (ES), in order to build dynamic trading strategies and find that downside risk measures outperform volatility in terms of a higher Sharpe Ratio, better drawdown protection and higher utility gains for mean-variance and loss-averse investors. Happersberger et al. (2019) also focus ES and VaR forecasts in order to manage dynamic tail risk protection strategies.

All the papers mentioned above and in general the literature of tail risk protection, focus on predicting risk measures in a regression manner and study the total distribution of the returns where the goal is to minimize Mean Squared Error (MSE) type measures of fit. Our argument is that it is not necessary to predict the total distribution of the return, whether it is crucial to correctly forecast the direction of the tail, in particular since in our case big profits or losses are at stake (Jordà and Taylor; 2011).

Investors' preferences Rickenberg (2019) developed a tail risk protection strategy, denoted target VaR strategy, where the trading signals are calibrated so the VaR of the strategy is constant over time for a predefined significance level α. The weights of the risky asset in the simple portfolio consisting of two assets, one risky and one riskyless asset, is a function of a constant level of risk defined as a VaR level denoted, target VaR, depending on investors' risk aversion measured by α. The weights are calibrated so the VaR of portfolio is constant, equals to the target VaR. The target VaR strategy has the main advantage to be better interpretable for investors who can prescribe their acceptable loss limit to the trader. Nevertheless, it is assumed that investors' preferences are static since the target VaR is fixed and does not depend on t. Such strategy is based on the standard financial theory assumption that investors are rational and have invariant risk preferences.

However, with the development of Behavioral Finance, numerous studies draw attention on the fact that investors are often irrational and their preferences change with different situations. The early paper of Kahneman and Tversky (1979) shows that investors are more risk averse with gains, but less with losses. More recently, Wen et al. (2014) study the characteristics of investors' risk preferences with different states of gains and losses and show that investors' risk preferences are time-varying with them. Indeed, the degree of risk aversion rises with the increasing gains and that of risk seeking improves with the increasing losses.

Taking into consideration investors' time-varying preferences, a constant target VaR strategy is inadequate. Indeed, we should decrease our target VaR in period of gains and increase it in periods of losses to allow more conservative trades in period of gains and more aggressive ones in case of losses.

Dynamic tail risk protection strategy Our goal is to build a dynamic tail risk protection with an adaptive predefined level or risk, adapting to the benchmark's performance which influences investors' preferences, Wen et al. (2014). At constant significance level α, when the benchmark is experiencing a period of good performance where the gains are increasing, its VaR is decreasing, thus the investors give more attention to smaller losses and tend to be more risk averse. As follows, the trader should adapt its strategy and aim for a smaller target VaR. In the inverse situation, during a period of losses, the tail risk of the benchmark is increasing and the investor only pay attention to large losses, tending to seek more risk. For example, one could use a threshold as a function of volatility in order to build a dynamic threshold labeling function as in de Prado (2018), chapter 3.3. In the next section, we will explain how to build such strategy.

3 Trading strategy

3.1 Tail risk as a risk measure

Tail Risk Definition Throughout this paper, we consider a risky asset, for example one stock, with price process p_t, where $t \in [0, ..., T]$ with T the final period time step. As usual, we define the one period log-return, where one period is here one hour, as $r_t = \log \frac{P_t}{P_{t-1}}$ and the loss series $-r_t$. As in Packham et al. (2017), we define the Value-At-Risk for a *risk level* α, denoted VaR_t^α, as the α-quantile of the distribution of the loss $-r_t$:

$$\mathrm{VaR}_t^\alpha = \inf\left\{l \in \mathbb{R} : \mathrm{P}\left(-r_t > l\right) \leq 1 - \alpha\right\} = \inf\left\{l \in \mathbb{R} : \mathrm{P}\left(-r_t \leq l\right) \geq \alpha\right\}$$

If r_t follows an absolutely continuous loss distribution, then the Value-At-Risk can be defined as an unlikely and severe loss which satisfies:

$$\mathrm{P}(-r_t \leq \mathrm{VaR}_t^\alpha) = 1 - \alpha$$

where α is small corresponding to investors' risk aversion. In practice, the risk level often takes the value of 0.01, 0.025 or 0.05. The Value-At-Risk characterizes the far right tail of the distribution of the loss, thus we use it as a measure of tail risk. For example, a $\mathrm{VaR} = u_t^\alpha = 0.05$ for $\alpha = 0.01$ means that returns below -5% only happen 1% of the time. Moreover, for any threshold u, $\mathrm{P}(-r_t \geq u)$ is called the exceedance probability over u.

3.1.1 Tail risk protection strategy

Now, let us consider a simple buy/sell trading strategy where the trader decides at time t either to enter the market with a full position (he invests the totality of his available capital at time t into BTC) or to stay out of the market (he sells all available BTC or does not invest capital at time t in order to have all his capital in a risk free asset) based on the information available at time t denoted $\mathcal{F}_t$. The trader's decision can be represented as a binary variable, or trading signal, $s_t \in \{0, 1\}$ where 1 corresponds to the decision to stay out of the market and 0 to the decision to fully invest the capital at time t. The return of such strategy is defined for all $t \in [1, \dots, T]$ as $R_t = (1 - s_{t-1})r_t$ and the excess return $R_t - r_t$.

Target VaR The goal of tail risk protection is to maximize the expected economic utility of a risk averse investor which can be represented by the risk-adjusted return of the strategy, where risk is characterized by the tail behavior of the portfolio. In other words, our goal is to lower the probability of tail risk, that is, at constant level α, to have a lower portfolio VaR than the buy-and-hold benchmark strategy which buys the asset at the beginning of the period and sell it at time T, the end of the period of investment. In parallel, we have two choices to maximize the return, either we maximize the expected return, following the portfolio selection criterion from Markowitz (1952), or we can maximize the total return at T, as suggested by Kelly (1956). Since in our setting, we are optimizing the strategy for multiple periods (Hakansson; 1971) we choose the latter. We can write the following optimization program:

$$\begin{aligned} \max \quad & \sum_{t=1}^{T} (R_t - r_t) \\ \text{s.t. } & \mathrm{VaR}_t^{\alpha} \leq \mathrm{TVaR}_t^{\alpha} \end{aligned} \tag{1}$$

where VaR^{α} is the portfolio VaR and TVaR_t^{α} is the target VaR for level α given by the investor.

We have the following equivalence, $\forall t \in [1, \dots, T]$:

$$\begin{aligned} & \mathrm{VaR}_t^{\alpha} \leq \mathrm{TVaR}_t^{\alpha} \\ & \mathrm{P}(-R_t \leq \mathrm{TVaR}_t^{\alpha}) \geq 1 - \alpha \end{aligned} \tag{2}$$

Thus, one relaxes the constant constraint from Rickenberg (2019) and aims at constructing a trading mechanism with signal $s = \{s_1, \dots, s_T\}$ that has a maximum TVaR^{α}. From now on, we think of fixed α and we will write TVaR_t^{α} as TVaR_t for the benchmark Value-At-Risk and VaR_t^{α} as VaR_t for the portfolio.

How to define the signals s_t ? We define for all $t \in [0, T-1]$

$$s_t = \mathbf{I}_{-r_{t+1} \geq \mathrm{TVaR}_{t+1}} \tag{3}$$

and by construction $R_t = (1 - \mathbf{I}_{-r_t \geq \mathrm{TVaR}_t})r_t$. Since TVaR_t is strictly positive for financial assets loss series, we have $\mathrm{P}(-R_t \leq \mathrm{TVaR}_t) = 1$, thus (2) is verified.

Since we do not know the true distribution of r_t and, obviously, at time t, we do not observe r_{t+1}, we do not know s_t and we must build estimates based on the observation available in order to make a trading decision. We estimate the following conditional probability, $\mathrm{P}(-r_{t+1} \geq \mathrm{TVaR}_{t+1} | \mathcal{F}_t)$, which is the exceedance probability over the threshold TVaR_{t+1}, and decide whether the trader must close his position $s_t = 1$ or stay in the market $s_t = 0$.

How to define TVaR^{α} The tail risk target is defined by the investors' preferences. Here, our goal is not to study investors' preferences, but to show how to build trading signals based on estimates of the exceedance probability over a given risk level. This problem has been well studied in the literature, for various applications, such as seismic risk assessment (Honegger and Wijewickreme;

2013), risk assessment for decision making with application to terrorism (Kunreuther; 2002) or again floods, earthquakes, drought and hurricanes risk management (Lambert et al.; 1994), (Mason et al.; 2007). For trading strategies, Christoffersen and Diebold (2006) and Linton and Whang (2007) considered exceedances above 0, whereas (Chung and Hong; 2007) focused on non-zero thresholds. In particular, Taylor and Yu (2016) studied thresholds that are not close to 0, as this is of greater relevance for risk management. This is also our interest but we consider extreme thresholds which are time varying and can be used as tail risk measure.

For illustration, we focus in this paper on different tail risk targets derived from the data itself. That is we use sample quantiles for different level α computed on the historical losses in a rolling window manner for different window size. In other words, to describe the investors' preferences for the next trading period $t+1$, we use the historical VaR of the loss series as tail risk measure defined for $w \in [1, T]$ as:

$$\forall t \in [w, T], \quad \forall i \in [t-w+1, t], \mathrm{TVaR}_{t+1} = \text{hist-VaR}_t^{\alpha,w} = \inf\left\{l \in \mathbb{R} : \mathrm{P}\left(-r_i > l\right) \leq 1-\alpha\right\} \quad (4)$$

The main advantage of this risk target is its computational simplicity allowing us to easily build training labels for our classifiers that we will develop in the next sections. On top, it is adaptable to investors' preferences, since using a small rolling window would suit an investor with varying preferences where the risk target quickly adapts itself to changes in the true return distribution, while large rolling windows give more stable risk target, corresponding to an investor with static preferences. In this paper, we will use three window sizes: 24 (one day), 2880 (four months) and 4320 (six months), since we are using hourly data, which are plotted on Figure 1 for $\alpha = 0.01$. We give more details about the dataset in the next section. Our choice is motivated by our goal to study how the tools used here react to different levels of noise in the target variable.

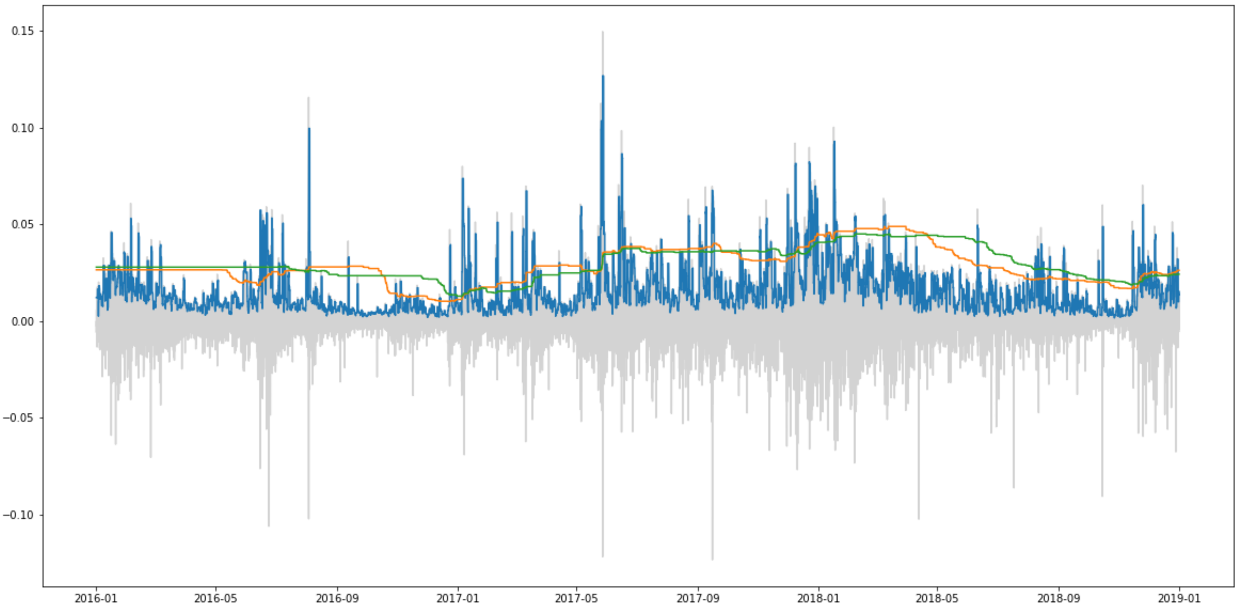

Figure 1: Hourly btc losses and $TVaR_t^{0.01,w}$ for different window size, $w = 24$ (one day), $w = 2880$ (four months), $w = 4320$ (six months)

As we can see on Figure 1 the $\mathrm{TVaR}_t^{0.01,24}$ is much more conservative than the $\mathrm{TVaR}_t^{0.01,w}$ with larger w, with much more losses exceeding the risk target. Indeed, we have respectively 0.045, 0.012 and 0.011 exceedance for the $\text{TVaR}_t^{0.01,w}$ with w is 24, 2880 and 4320, where the exceedance is defined as $N_e = \frac{1}{T}\sum_{t=1}^{T} -r_t \geq \text{TVaR}_t^{\alpha}$, that is the proportion of data exceeding the threshold. A strategy with $\text{TVaR}_t^{0.01,24}$ as tail risk target should be much more conservative and hedges much more losses than the two other risk targets. .

Trader decision How to decide whether we should hedge our position for the next trading period ? We could build an estimation of the return at the next period, r_{t+1}, in a regression manner, nevertheless, we are only interested in the distribution of the tail. Thus, in this paper, we directly approximate the conditional probability classes of the variable s_t, that is p_t, defined $\forall t \in [0, \ldots, T-1]$ as:

$$p_t = \text{P}(s_t = 1|\mathcal{F}_t) = \text{P}(-r_{t+1} \geq \text{TVaR}_{t+1}^{\alpha}|\mathcal{F}_t) \tag{5}$$

From p_t, we can make the trading decision $\hat{s}_t$ defined for a threshold $u \in \mathbb{R}$:

$$\hat{s}_t = \begin{cases} 1, & \text{if } p_t \geq u \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

We can now write the realized return of our strategy, for all $t \in [1, T]$:

$$R_t = (1 - \hat{s}_{t-1})r_t \tag{7}$$

4 Machine learning trader

Let $X_t \in \mathbb{R}^p$ be some input feature vector of dimension p which summarize $\mathcal{F}_t$. The aim is to approximate the true unknown conditional probability $\text{P}(s_t|X_t)$ with a learning algorithm $\mathcal{M}$ for different risk targets. We consider four different α: 1%, 2.5%, 5% and 10%, and three rolling windows to compute TVAR: 24, 2880, and 4320. In the next two sections, we present different models $\mathcal{M}$ we use as estimator for each risk target. We first present the dataset we use and then we explain the different models considered.

4.1 Data

We apply the proposed strategy on BTC. We collected 36193 close prices from Poloniex exchange, using its API, from 2016-01-01 to 2020-02-16 on a hourly basis. We look at intraday frequency, since it is not rare to observe severe loss at such frequency, due to a relatively higher volatility in the cryptocurrency market compared to traditional assets.

We split the dataset into two sets, train, from 2016-01-01 to 2019-01-01 00:00:00, and test sets from 2019-01-01 01:00:00 to 2020-02-16 22:00:00. We keep a relatively large test set in order to produce a final robust estimation of the out-of-sample trading strategy performance with a large backtesting period of 9486 observations.

4.2 Neural networks

We first use two different neural network architectures for the trading signals prediction, the Multi-Layer Perceptron (MLP) and the Long short-term memory neural network (LSTM) architecture from Hochreiter and Schmidhuber (1997). Neural networks are non-parametric, which allows us to avoid making strong assumptions on the data that are not met in reality and that is often the case for financial time series, for example with the normality assumption of financial returns. On top, neural networks are universal approximators (Leshno et al.; 1993) which means that, in theory, neural networks can arbitrarily closely approximate the true process of r_t. In particular, LSTM neural networks are state-of-the-art for many applications such as speech recognition, text extraction, translation or handwriting recognition, since plain recurrent neural networks are not suited to non-stationary time series modeling. In finance, Deep learning has been used in various research, in particular by Franke (1999) and Zhang et al. (2020) for portfolio management or by (Kim and Won; 2018) for volatility forecasting with LSTM in comparison with GARCH. Nevertheless, one has yet to prove the superiority of neural networks compared to simpler parametric models in their application.

4.2.1 Training

Output In order to train the neural networks, we need to build the training input-output pairs. We could directly use the trading signal s_t from (3) as output labels, nevertheless, since $\text{TVaR}_t^{\alpha,w}$ belongs to the tail of the loss series, s_t suffers from sever class imbalance since $\frac{\sum_{t=w+1}^{T} \mathbf{I}_{-r_t > \text{TVaR}_t^{\alpha,w}}}{T} \ll 0.5$, which is difficult to handle for machine learning models (REFERENCE). On top, financial returns follow an asymmetric distribution and suffers from the leverage effect (Black; 1976), (Christie; 1982), commonly defined as volatility rising more rapidly when returns are negative than positive. To address that problem, we introduce a new category for r_t, when r_t belongs to the right tail of the distribution, which allow us to control the misclassification cost of the final model. Indeed, let us introduce the following output variable defined as:

$$Y_t = \begin{cases} 1, \text{ if } -r_t < \text{TVaR}_t^{1-\alpha,w} \\ 2, \text{ if } -r_t > \text{TVaR}_t^{\alpha,w} \\ 0, \text{ otherwise} \end{cases} \tag{8}$$

We can define the predicted strategy signals as a function of the predicted output variable $\hat{Y}$ as follows :

$$\hat{s}_t = \begin{cases} 1, \text{ if } \hat{Y}_t = 2 \\ 0, \text{ otherwise} \end{cases} \tag{9}$$

and we easily derive the unconditional misclassification costs matrix in terms of excess returns.

		Prediction		
		0	1	2
Outcome	0	0	0	$\bar{r}_0\bar{p}_0$
	1	0	0	$\bar{r}_1\bar{p}_1$
	2	0	0	$\bar{r}_2\bar{p}_2$

where $\bar{r}_i$ is the expected return on class i estimated with $\frac{1}{\sum_{t=w+1}^{T} \mathbf{I}_{Y_t=i}} \sum_{t=w+1}^{T} r_t \, \mathbf{I}_{Y_t=i}$ and $\bar{p}_i$ is the weight of class i defined as $\frac{\sum_{t=w+1}^{T} \mathbf{I}_{Y_t=i}}{T-w}$ which is the empirical estimate of the unconditional probability $\text{P}(Y_t = i)$.

As expected, when using the risk target defined in (4), we can see on table 1 that one has negative costs for the correct classification of class 2 since $\bar{r}_2 \leq 0$ and by definition, the conditional cost for predicting class 2 instead of 1 is much higher than the one for predicting class 2 instead

		Prediction		
α	w (hours)	0	1	2
0.01	24	0.01	0.15	-0.15
	2880	0.01	0.14	-0.14
	4320	0.01	0.13	-0.14
0.025	24	0.01	0.15	-0.15
	2880	0.01	0.14	-0.14
	4320	0.01	0.13	-0.14
0.05	24	0.01	0.15	-0.15
	2880	0.01	0.14	-0.14
	4320	0.01	0.13	-0.14
0.1	24	0.01	0.15	-0.15
	2880	0.01	0.14	-0.14
	4320	0.01	0.13	-0.14

Table 1: Class 2 misclassification costs for different $\mathrm{TVaR}^{\alpha,w}$

of 0, since in the former, we miss the opportunity to invest during an extremely positive return period. Thus, we clearly see that the classifier must correctly predict the sign of the tail events in order to maximize the strategy return. In order to do so, we use a 3 neurons output layer with softmax activation function corresponding to the output variable Y_t defined in (8).

Input layer As we are interested in understanding whether neural networks can, by themselves, learn valuable features to produce trading signals and feature selection is not our interest in this work, we use simple transformations of the returns, which can be seen as momentum features.

In particular, since price time series are not stationary, we use multiple returns instead, $X_t^p = r_{t-p}$ for different periods p, where $r_{t-p} = p_t/p_{t-1-p} - 1$ and $p \in \{0, 1, 2, 4, 6, 13\}$. We also use the normalized difference between the returns and the two class thresholds which should help identifying a risk-buildup situation, when the returns either explodes toward the upper class threshold $\mathrm{TVaR}_t^{1-\alpha,w}$ or severely drops toward the lower threshold $\mathrm{TVaR}_t^{\alpha,w}$, which are defined respectively as $U_t^{\alpha,w} = \frac{r_t - \mathrm{TVaR}_t^{\alpha,w}}{\mathrm{TVaR}_t^{\alpha,w}}$ and $D_t^{\alpha,w} = \frac{r_t - \mathrm{TVaR}_t^{1-\alpha,w}}{\mathrm{TVaR}_t^{1-\alpha,w}}$.

For the MLP, we then use the vector $X_t \in \mathbb{R}^8 = (X_t^0, X_t^1, X_t^2, X_t^4, X_t^6, X_t^{13}, D_t^{\alpha,w}, U_t^{\alpha,w})$ as input for each risk target. For the LSTM model, since it is a recurrent neural network, we take advantage of its ability to directly modelize sequential data and we use the same features as above on a certain historical window of length 24, using one historical day to make a prediction for the next hour. Thus the features become: $X_t^p = (r_{t-p-23}, \ldots, r_{t-p})$ for $p \in \{0, 1, 2, 4, 6, 13\}$ and $U_t^{\alpha,w} = (\frac{r_{t-23} - \mathrm{TVaR}_t^{\alpha,w}}{\mathrm{TVaR}_t^{\alpha,w}}, \ldots, \frac{r_t - \mathrm{TVaR}_t^{\alpha,w}}{\mathrm{TVaR}_t^{\alpha,w}})$ and $D_t^{\alpha,w} = (\frac{r_{t-23} - \mathrm{TVaR}_t^{1-\alpha,w}}{\mathrm{TVaR}_t^{1-\alpha}}, \ldots, \frac{r_t - \mathrm{TVaR}_t^{1-\alpha,w}}{\mathrm{TVaR}_t^{1-\alpha}})$. We use the same input vector as for the MLP, but now X_t is in $\mathbb{R}^{24,8}$. We explain the hidden layer architecture in the next section.

Finally, we train our classifier, using Keras python library, on the training data $D = \{(X_{w+1}, Y_{w+1}), \ldots, (X_T, Y_T)\}$ with Adam algorithm, which is a stochastic gradient descent method based on adaptive estimation of first-order and second-order moments of the gradient (Kingma and Ba; 2015) and 128 batch size.

4.3 Model selection

4.3.1 Hidden layers architecture

On the train set we performed 10 folds cross-validation for time series, described on figure 3, with one fold corresponding to one month, in order to test different hidden layers architectures for the MLP model. In this work, we keep the hidden layers relatively simple in order to show how a single architecture can effectively extract features for different risk target output. Based on the cross-validation performance evaluated with AUC score (area under the ROC curve), we choose the model with highest AUC score between class 1 and 2, while still controlling for robustness. That is we select the model with highest median AUC score between class 1 and class 2. After a small tuning, we keep 3 hidden layers, three fully connected layers with 16, 4 and 2 neurons respectively, as on the Figure 2

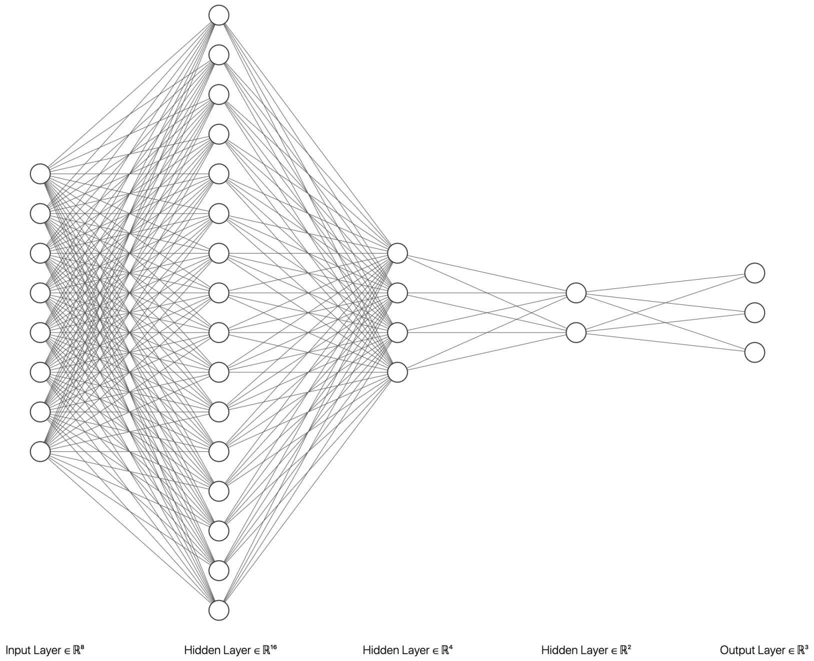
Input Layer ∈ $\mathbb{R}^8$
Hidden Layer ∈ $\mathbb{R}^{16}$
Hidden Layer ∈ $\mathbb{R}^4$
Hidden Layer ∈ $\mathbb{R}^2$
Output Layer ∈ $\mathbb{R}^3$

Figure 2: Final MLP architecture

As regularization technique, we use the state-of-the-art Dropout layer after the first and the second hidden layers with a dropout rate of 0.2, in order to prevent overfitting (Srivastava et al.; 2014).

As for the LSTM architectures, we simply change the first two fully connected layers to LSTM units in order to test whether LSTM can directly improve the performance of the MLP models. We use the hyperbolic tangent as activation function for the hidden layers.

4.3.2 Strategy parameters and final decision

Any classifier $\mathcal{M}$ trained on the dataset D gives for output the probability class p_t. Thus the trading strategy based on $\mathcal{M}$ has one parameter, the probability threshold u from which we can make a trading decision with respect to (6). The optimal parameter must solve the investor's goal define in (1) with respect to their risk target. We can now reformulate the optimization problem as it follows:

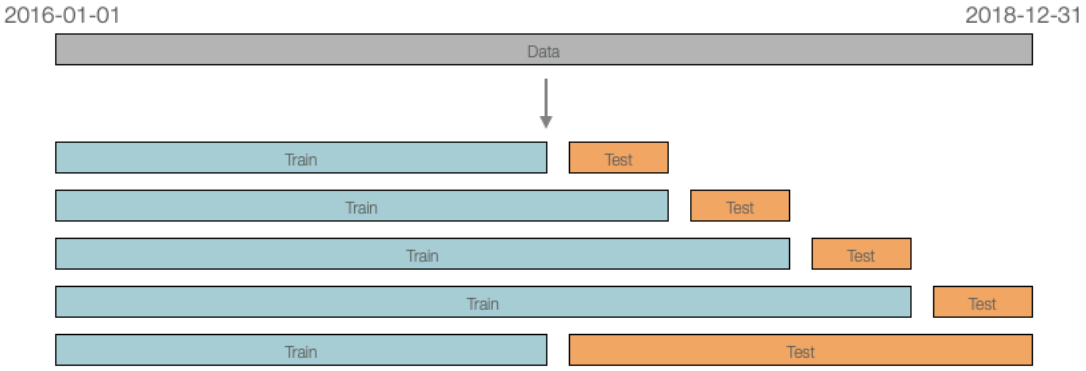

Figure 3: Time series cross-validation with four folds

$$
\begin{aligned}
&\max_{u,w} \sum_{t=w+1}^{T} (R_t - r_t) = \sum_{t=w+1}^{T} \{(1-\hat{s}_t) r_t\} \\
&\text{s.t. } \mathrm{VaR}_t^{\alpha} \leq \mathrm{TVaR}_t^{\alpha}
\end{aligned}
\tag{10}
$$

Probability threshold u Denote $\mathcal{U}_\alpha$ the set of u which are tail loss optimal, that is the set of u respecting the constraint formulated in (2). Let us define our strategy with the trading signals $\{\hat{s}_{w+1}, \dots, \hat{s}_T\}$ and corresponding return $S_{w,\alpha} = \{R_{w+1}, \dots, R_T\}$ for a predefined risk target with level α. We wish to define:

$$\mathcal{U}_\alpha = \{u \in \mathbb{R}, \quad \mathrm{P}(R_t \leq \mathrm{TVaR}_t^{\alpha}) \geq 1-\alpha\}$$

Let us introduce the conditional probability classes for an arbitrary threshold $u \in \mathbb{R}$:

		Prediction	
		0/Hold	1/Sell
Outcome	0/Hold	$\mathrm{TNR}(u) = \mathrm{P}(\hat{s}_t = 0 \mid s_t = 0)$	$\mathrm{FPR}(u) = \mathrm{P}(\hat{s}_t = 1 \mid s_t = 0)$
	1/Sell	$\mathrm{FNR}(u) = \mathrm{P}(\hat{s}_t = 0 \mid s_t = 1)$	$\mathrm{TPR}(u) = \mathrm{P}(\hat{s}_t = 1 \mid s_t = 1)$

where $\mathrm{TNR}(u)$, $\mathrm{FPR}(u)$, $\mathrm{FNR}(u)$ and $\mathrm{TPR}(u)$ stand for *True Negative*, *False Positive*, *False Negative* and *True Positive Rates* for the threshold u respectively. In statistical tests, $\mathrm{FPR}(u)$ and $\mathrm{FNR}(u)$ are often called type I and type II errors respectively. The optimal threshold is given by the optimal trade-off between FPR and FNR, but in order to respect the risk target, one cannot choose any trade-off, since, as one can easily see, $\mathrm{FNR}(u)$ corresponds to the exceedance probability of the portfolio over the risk limit. In practice, as it is pointed out in (Packham et al.; 2017), such a strategy cannot avoid tail events that occur "totally out of the blue" (type II error in our formulation). Also, given the empirical stylized fact that return data feature little or no autocorrelation implies that such a strategy may signal a sell order, when ex-post holding the position would have been optimal (type I error).

We can easily see that any threshold $u^* \in \mathcal{U}_\alpha = \{u \in \mathbb{R}, \quad \mathrm{TPR}_(u) \geq \frac{\mathrm{P}(-r_t > \mathrm{TVaR}_t^{\alpha}) - \alpha}{\mathrm{P}(-r_t > \mathrm{TVaR}_t^{\alpha})}\}$ is tail loss optimal (PROOF IN APPENDIX). Finally, to decide for the final probability class threshold

u^*, we simply choose the threshold that maximize the cumulative excess return on the train set and apply it on the corresponding test set.

For illustration, on Figure 4, we plotted the shape of the constraint on TPR with respect to the risk limit of the investor for different risk level α. As we can see, as the target risk for a level α diverge from the α-quantile of the true distribution of the returns, the constraint on the classifier performance strengthens.

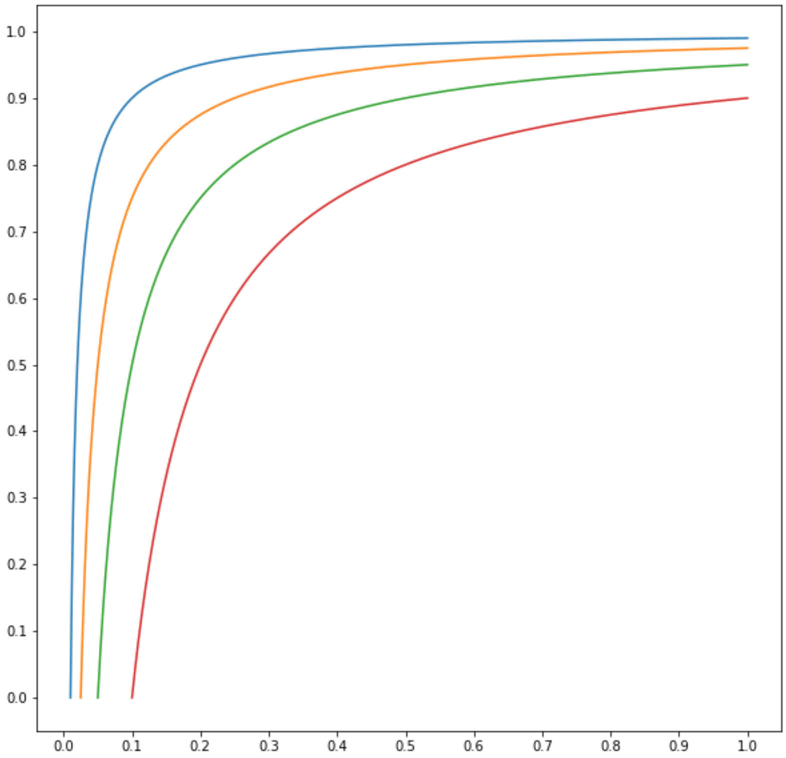

Figure 4: Expected exceedance and corresponding minimum TPR for significance levels 0.01, 0.025, 0.05, 0.1

As described in section 3, the risk targets are defined by Equation (4). In the Table 2, we show the different constraints from equation (2) on the classifier in order to achieve the goal of the tail loss strategy based on the hourly returns of btc on the train set.

α	w (hours)	exceedance	min TPR (%)
0.01	24	4.5	78
	2880	1.2	18
	4320	1.1	11
0.025	24	5.5	55
	2880	2.7	7
	4320	2.6	5
0.05	24	8.0	38
	2880	5.2	4
	4320	5.1	3
0.1	24	12.5	20
	2880	9.7	0
	4320	9.4	0

Table 2: Expected exceedance and corresponding minimum TPR for different VaR estimators with parameters (α, w), $w = 2880$ and $w = 4320$ correspond to 4 months and 6 months respectively

As we can see, using a small rolling window to compute the risk target is more conservative thus the classifier must have a large TPR in order to ensure a maximum risk target of the final strategy.

5 Auto-regressive models

Since neural networks are complex non-linear multi-factor models, it is justified to ask if a simple parametric model calibrated on the same training set can catch tail-events as good ML methods or better. In order to compare both approaches, we used three different parametric and a LPA technique from Spokoiny (2009a), that we present in this section.

5.1 ARMA-GARCH and ARMA-EVTGARCH

ARMA-GARCH As a benchmark, we first used an ARMA-GARCH model for BTC loss series. Indeed, GARCH, developed by Bollerslev (1986), catches well volatility clusters, which often appears in returns time series and ARMA model allows for a time-varying mean of the returns. The ARMA(P, Q)-GARCH(q, p) model is defined as:

$$\begin{aligned} r_t = {} & a_1 r_{t-1} + a_2 r_{t-2} + \ldots + a_P r_{t-P} \\ & + \varepsilon_t + b_1 \varepsilon_{t-1} + b_2 \varepsilon_{t-2} + \ldots + b_Q \varepsilon_{t-Q} \end{aligned}$$

with

$$\begin{aligned} &\varepsilon_t = Z_t \sigma_t \\ &Z_t \sim N(0,1) \\ &\sigma_t^2 = \omega + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2 + \sum_{j=1}^{q} \alpha_j \varepsilon_{t-j}^2 \end{aligned}$$

where $\omega > 0$, $\alpha_i \geq 0$, $\beta_i \geq 0$ and $\sum_{i=1}^{p} \beta_i + \sum_{j=1}^{q} \alpha_j < 1$.

ARMA-EVTGARCH Assuming normally distributed innovations, Z, is often not justified and, since in our case we are interested in modeling extreme events, we also fitted an ARMA-GARCH model with a Generalised Pareto distribution (GPD) for the innovations of the GARCH process. This approach has been studied numerous time in the literature, in particular in Packham et al. (2017) or Taylor and Yu (2016) for risk management in trading. We refer to that model as ARMA-EVTGARCH.

The distribution function of the GPD is given by:

$$G_{\xi,\beta}(x) = \begin{cases} 1 - (1 + \xi x/\beta)^{-1/\xi}, & \xi \neq 0 \\ 1 - \exp^{-x/\beta}, & \xi = 0 \end{cases}$$

where $\beta > 0$,$x \geq 0$ when $x \geq 0$ and $0 < x \leq -\beta/\xi$ when $\xi < 0$.

The GPD describes the tail of the data since only the extremes outcomes are included in the estimation with the method of threshold exceedances as we explain in the next paragraph.

Model selection, calibration and exceedance probability To find the proper lag orders, P and Q, for the mean, and p and q, for the volatility processes, we use the classical Box-Jenkins method and AIC criterion, see Chen et al. (2018) for a detailed explanation. We perform the model selection on the last 2280 observations (four months) of the train set. The final model selected is an ARMA$(3, 1)$-GARCH$(1, 2)$, we use the same orders ARMA-EVTGARCH.

We calibrate the parameters of ARMA-GARCH and ARMA-EVTGARCH simultaneously via quasi-maximum likelihood estimation (QMLE) (McNeil and Frey; 2000) using a rolling window of fixed length of four months (2880 observations). The size of the rolling window is relatively large for GARCH modelling, nevertheless, in this paper, we are using intraday data which justifies that choice. As a comparison Taylor and Yu (2016) used a rolling window of 2500 observations. The calibration of the GPD is based on the upper 5% of the residuals of the ARMA-GARCH, we denote the upper 5% threshold as g. For illustration, residual QQ-plots based on hourly BTC loss data are given in Figure 5.

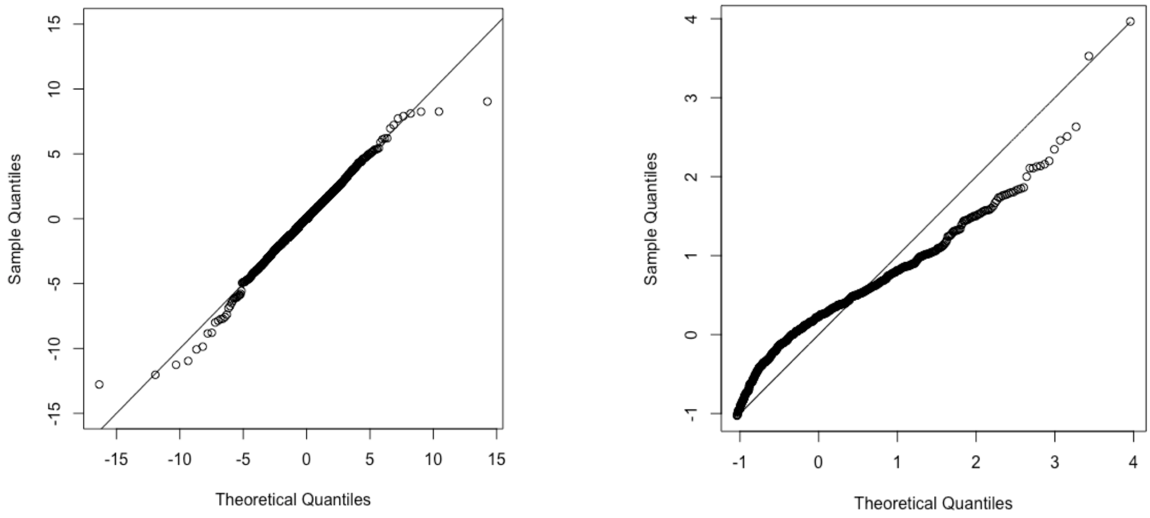

Figure 5: QQ plots of GARCH residuals. Left: normal distribution; right: GPD distribution (samples beyond upper 5% threshold, g).

The estimated probability class of for the category 2 of Y_t, that is $\mathrm{P}(-r_t > \mathrm{TVaR}_t^{\alpha})$, is then given by $p_t = 1 - F((\mathrm{TVaR}_t^{\alpha} - \hat{\mu}_{t+1})/\hat{\sigma}_{t+1})$ where $\hat{\mu}_{t+1}$ and $\hat{\sigma}_{t+1}$ are the models' mean and volatility forecasts and F is the normal distribution function for the ARMA-GARCH and GPD for the

ARMA-EVTGARCH model. For computing p_t of ARMA-EVTGARCH, the GPD is defined only for $x \geq g$, so if $(\text{TVaR}_t^{\alpha} - \hat{\mu}_{t+1})/\hat{\sigma}_{t+1} < g$, F is the normal distribution, since it is not necessary to use a GPD distribution for observations that do not belong to the tail. We recalibrate the parameters of the models every hour for each new observation in the test set.

5.2 CARL-vol model

The benefit of ARMA-GARCH type models is their effectiveness at catching stylized facts of financial time series and their simplicity, nonetheless, they do not directly model the exceedance probability p_t, but the total distribution of the return r_t. In order to use the advantages of GARCH models and to directly modelize the exceedance probability, we also fit a CARL-vol model from Taylor and Yu (2016) which specifies p_t for a constant threshold $Q \in \mathbb{R}$ as:

$$p_t = \frac{0.5}{1 + \exp(-x_t)} + 0.5\mathbf{I}(Q > 0)$$

where $x_t = \phi_0 + \phi_1 \sigma_t$ and we assume a GARCH$(1,1)$ for the return that is, $\sigma_t^2 = \omega + \beta_1 \sigma_{t-1}^2 + \alpha_1 (r_{t-1} - \mu)^2$. ϕ_i, α_i, β_i and μ are constant parameters.

Model calibration We calibrate the parameters of CARL-vol with Maximum Likelihood Estimation using a Bernoulli distribution with density specified as $f(r_t) = p_t^{\mathbf{I}(y_t \leqslant Q)}(1-p_t)^{1-\mathbf{I}(y_t<Q)}$. For the model calibration, we use the same rolling window as for the GARCH based models of the previous section, where Q is set to TVaR$_t$ of the last observation of the rolling window. Since CARL-vol assumes a fixed threshold Q and that, in our case, it is time-varying, we also recalibrate the model every hour, estimating new parameters for each observation in an online manner.

5.3 Local Parametric Approach (LPA)

For the calibration of the three previous models, we use a fixed rolling window, assuming time homogeneity on that interval, in the sense that the process r_t follows the same structural equation at each time point defined by ARMA-GARCH, ARMA-EVTGARCH and CARL-vol models. This approach, of course, does not take into consideration structural breaks (Bouri et al.; 2019). Ignoring breaks will increase the bias (Hillebrand; 2005). Though, in order to relax time homogeneity, we can either use a non-parametric model which allows for time-varying parameters, as we do with the neural network models presented in section 4, or we can use a Local Parametric Approach (LPA) thanks to the Local Change Point detection method suggested by Spokoiny (1998) and developed by Härdle et al. (2003), Mercurio and Spokoiny (2004) and Spokoiny (2009a) in the context of volatility modelling.

As in Cizek et al. (2009), we assume GARCH$(1,1)$, characterized by parameters $\boldsymbol{\theta}$, is a good candidate for the process r_t on a local scale. For estimating $\boldsymbol{\theta}$, we apply the QMLE approach using a Gaussian distribution for the innovations. The LPA allows us to find the largest interval of homogeneity, $\mathcal{I}(t)$, i.e. the longest interval $\mathcal{I}$ with the right-end point t, where data do not contradict the assumption that $\boldsymbol{\theta}$ is constant. To find the change points, we use the test procedure proposed by Klochkov et al. (2019) that uses the multiplier bootstrap method from Spokoiny and Zhilova (2015) to find the critical values for the likelihood ratio test, generating 100 bootstrap weights sequences for the test. The computational cost is high and our test set is quite large, thus, in practice, we choose a fixed decreasing size of 5 observations for the successive intervals tested and we only performed the LPA detection every 5 observations in the test set. This does not come without cost and automatically increases the bias of $\hat{\boldsymbol{\theta}}$, nevertheless, it decreases the bias in comparison of a fixed window size approach, as for the GARCH based models of the previous sections. We denote the model calibrated with LPA method as LCP-GARCH.

6 Applications

6.1 Meta strategy: stacking

In the previous section we presented six different classifiers to build trading signals for tail events hedging. Since the GARCH based approach is essentially linear, but MLP and LSTM are non-linear classifiers, we expect them to extract different features from the data. A direct question is then to know which model to use in which situation. To address that problem, we propose to combine all models output from the test set into a meta model with a stacking classifier. Stacking is an ensemble learning technique where the predictions of multiple classifiers (referred as level-0 classifiers) are used as new features to train a meta-classifier (level-1 classifier).

The meta-classifier is trained on the predictions made by level-0 classifiers on the out-of-sample data. That is, data which has not been used to train the level-0 classifiers is fed to them in order to get the predictions $p_t = (p_t^j)$, where $1 \leq j \leq 6$ corresponds to the level-0 models and t belongs to the test set indices. We then use the predictions to build input features and the corresponding labels Y_t to build the training input-output pairs for the meta classifier.

Our meta-classifier should be explainable in order to directly interpret which level-one classifier contributes the most to the final meta-prediction. For its simplicity and interpretability, we use logistic regression. We have for all $t \in [T-n-1, T]$ where n is the test set size:

$$s_t = \sum_{j=1}^{6} \beta^j P_t^j$$

where $P_t^j = \log \frac{p_t^j}{1-p_t^j}$ and p_t^j is the output of each one-level classifier that is the estimated probability of class 2 of Y_t. Moreover, since we need to interpret the coefficients β^j, we rescale p_t^j to $[0, 1]$ range.

When using stacking technique, the issue is the multicollinearity of the prediction of the level-one classifiers. In particular, we can expect ARMA-GARCH and ARMA-EVTGARCH models to present multicollinearity in their output. To address that problem, we use L2 regularization to improve robustness, performing a Ridge type logistic regression (Hoerl and Kennard; 1970), where the final estimator for the model coefficients is the solution of:

$$\min_{\beta_1 \cdots \beta_6} \sum_{i=T-n-1}^{T} \left(s_t - \sum_{j=1}^{6} \beta^j p_t^j \right)^2 + \lambda \sum_{j=1}^{6} {\beta^j}^2$$

where $\lambda \in \mathbb{R}^+$ is the regularization hyper-parameter. In practice, we must finetune this parameter by performing, for example, cross-validation. In our setup, we did not tune it since cross-validation would be too computationally intensive and fixed it to $\lambda = 1$. We will denote that model as Ensemble.

In the next sections, we present our results. They correspond to out-of-sample performance of the different models on the test set from 2019-01-01 to 2020-02-16. The neural network models are retrained every 824 observations corresponding to the size of the validation folds in order to reflect the validation performance, whereas the ARMA-GARCH, ARMA-EVTGARCH, CARL-vol and

Ensemble are recalibrated every new observation and LPA-GARCH every 5 observations as stated in the previous section.

Every hour, each model generate the trading decision $\hat{s}_t$ based on the data available until t, then it is applied to the return, r_{t+1}, on the next period $[t, t+1]$. The current set-up does not account for market frictions, such as bid-ask spreads, and other associated costs, but we included the trading fees from Binance exchange, which are, at most, 0.1% on each trade. Nonetheless, the cost associated to the fees should be very low since we only trade when a tail event is predicted.

6.2 Benchmarks

There are a few well documented tail risk protection strategies in the literature, such as the constant proportion portfolio insurance (CPPI) strategy (Black and Jones; 1987), the Dynamic Proportion Portfolio Insurance (DPPI) or Time-varying Proportion Portfolio Insurance (TPPI) strategy (Hamidi et al.; 2009), (Happersberger et al.; 2019) or the protective put strategy using options. In this paper, we compared our results with more recent and machine learning oriented work, in particular the Varspread and target VaR strategies. Finally, the last benchmark used is the simple buy-and-hold strategy, where an investor buys BTC at 2019-01-01 00:00:00 and sells it at the end of the test set at 2020-02-16 22:00:00.

6.3 Out-of-sample performance

6.3.1 Model risk evaluation

In order to assess the performance of each strategy, we first look at the model risk which comes from the statistical performance of the classifier. A familiar metric is the Area Under the ROC Curve (AUC), nevertheless, the classical AUC, while taking into consideration class imbalance, does not include the cost associated with each class. For our trading application, as we stated in section 4.2.1, we have strong cost imbalance between the outcomes. Thus, for our evaluation, we use the risk-adjusted or cost-adjusted AUC by Jordà and Taylor (2011).

The ROC curve is defined for all threshold u as:

$$\begin{aligned} \mathrm{ROC} \colon [0,1] &\to [0,1] \\ \mathrm{FPR}(u) &\mapsto \mathrm{TPR}(u) \end{aligned}$$

We incorporate the average costs associated with false positives ($c_F = \bar{r}_0 p_0 + \bar{r}_1 p_1$) and true positives ($c_T = -\bar{r}_2 p_2$) in order to get the risk-adjusted curve, denoted AROC and rescaled to $[0,1]$, defined for all threshold u as:

$$\begin{aligned} \mathrm{AROC} \colon [0,1] &\to [0,1] \\ c_F\, \mathrm{FPR}(u) &\mapsto c_T\, \mathrm{TPR}(u) \end{aligned}$$

The risk-adjusted AUC, denoted AAUC is then AAUC $= \int_0^1 \mathrm{AROC}(u) du$.

In Table 3 we show the risk-adjusted AUC score on the whole test period. We can see that the MLP and LSTM models have similar performance, we can make the same observation for the ARMA-GARCH, ARMA-EVTGARCH and LPA-GARCH approach model. Nonetheless, we observe that LPA-GARCH performs better than ARMA-GARCH and ARMA-EVTGARCH for stable risk target where $w = 2880$ or $w = 4320$. This indicates that using a model with less complexity fitted on a time-homogeneous window generalized better than more complex models fitted on large window. Finally, all parametric or local parametric models perform better than the neural network based models in terms of risk-adjusted classification, which is a major drawback to the deep learning approach in our setting.

On Table 4 we present the cross-validation performance on the test set. Indeed recall, that the results from the MLP and the LSTM models are obtained by retraining the models every 824 observations in the same approach as depicted in Figure 3. For comparison, we also evaluate the other classifiers with the same sampling method.

α (%)	w (hours)	MLP	LSTM	ARMA-GARCH	ARMA-EVTGARCH	LPA-GARCH	CARL-vol	Ensemble
	24	69	69	73	73	71	71	73
1.0	2880	62	67	69	72	72	69	74
	4320	70	69	70	72	72	74	74
	24	66	69	70	70	69	69	71
2.5	2880	72	71	74	74	76	72	77
	4320	71	73	74	74	75	73	77
	24	65	65	66	66	65	66	67
5.0	2880	71	70	74	74	75	73	76
	4320	70	71	74	73	75	73	76
	24	63	63	63	63	60	62	64
10	2880	67	66	70	70	71	71	72
	4320	66	68	69	70	70	71	71

Table 3: Risk-adjusted AUC on the test set

We can clearly see that ARMA-GARCH and ARMA-EVTARCH outperforms MLP and LSTM classifiers for small significance level and varying risk target ($\alpha = 0.01$ or $\alpha = 0.025$ and $w = 24$) both on average and in terms of robustness. This shows that, in our setup, GARCH based models still outperforms non-parametric models at catching volatility clusters. On the other hand, for larger significance level and stable risk target ($\alpha = 0.05$ or $\alpha = 0.1$ and $w = 2280$ or $w = 4320$), the non-parametric approaches do at least as good as the simple GARCH based models on average and in terms of robustness which indicates a higher generalization power.

α (%)	w (hours)	MLP	LSTM	ARMA-GARCH	ARMA-EVTGARCH	LPC-GARCH	CARL-vol	Ensemble
	24	70 (0.3)	70 (0.4)	74 (0.1)	74 (0.1)	72 (0.2)	72 (0.2)	73 (0.2)
1.0	2880	58 (3.3)	64 (3.9)	60 (0.9)	61 (1.3)	61 (1.4)	55 (1.0)	62 (1.8)
	4320	63 (3.7)	63 (3.1)	58 (1.2)	62 (1.0)	59 (1.3)	57 (0.8)	60 (1.6)
	24	68 (0.4)	70 (0.1)	71 (0.1)	71 (0.1)	69 (0.1)	70 (0.2)	71 (0.1)
2.5	2880	66 (0.9)	65 (1.3)	62 (1.3)	62 (1.2)	64 (1.2)	57 (1.3)	66 (1.1)
	4320	63 (0.6)	65 (2.0)	62 (1.4)	62 (1.6)	61 (1.2)	57 (1.3)	65 (1.6)
	24	67 (0.4)	67 (0.2)	67 (0.1)	67 (0.1)	65 (0.2)	66 (0.2)	68 (0.2)
5.0	2880	65 (0.7)	64 (0.6)	62 (0.8)	62 (1.0)	63 (0.8)	61 (1.1)	64 (1.0)
	4320	66 (0.5)	64 (0.7)	62 (0.9)	61 (1.0)	63 (0.9)	60 (0.8)	64 (1.0)
	24	64 (0.2)	64 (0.2)	64 (0.1)	64 (0.1)	61 (0.1)	63 (0.2)	65 (0.2)
10	2880	61 (0.7)	63 (0.8)	61 (0.8)	61 (0.8)	61 (0.7)	60 (0.7)	62 (0.8)
	4320	62 (0.7)	64 (0.7)	61 (0.7)	61 (0.7)	61 (0.8)	60 (0.6)	62 (0.9)

Table 4: Average risk-adjusted AUC and (variance risk-adjusted AUC) over the testing periods

From the two tables above, we can also conclude that we benefit from combining those classifiers into a stacking classifier, since the AAUC of the Ensemble classifier is always higher or equal to the maximum AAUC of the weak classifiers on the whole test period. Moreover, in terms of cross-validation performance, the Ensemble classifier always outperforms the average performance, except for the strategy with $\text{TVAR}^{0.01,4320}$ risk target. In particular, for larger risk level ($\alpha = 0.025$, $\alpha = 0.5$ and $\alpha = 0.1$) and small window ($w = 24$), the Ensemble classifier outperforms the best performing weak classifier. This result is a clear indicator that each classifier can extract information which is uncorrelated with some of the other classifiers for highly adaptive risk target. This is a sign that each classifier might be more adapted to specific market regimes, but that the Ensemble model is more general.

Overall, from those metrics, we can say that the non-parametric models present more risk than the parametric ones or the Ensemble classifier. Now, in the next section, we will relate the model risk to the economic risk via backtest performance of the associated trading strategies.

6.3.2 Backtest performance

Risk profile First, we must verify that the strategy associated with each classifier is verifying the target risk constraint. For this we compare, in Table 5, the exceedance of each strategy with the risk level α, if the exceedance is larger than α, then the strategy does not respect the investor's risk preferences.

α (%)	w (hours)	Buy& Hold	MLP	LSTM	ARMA- GARCH	ARMA- EVTGARCH	LPC- GARCH	CARL-vol	Ensemble	target VaR norm	target VaR evt	Varspread
1.0	24	4.3	0.3	0.4	0.5	0.4	0.9	1.2	1.0	1.4	1.1	3.6
	2880	1.0	0.6	0.7	0.8	0.6	0.9	0.4	0.7	0.5	0.5	0.7
	4320	0.9	0.7	0.5	0.7	0.5	0.8	0.4	0.7	0.4	0.4	0.7
2.5	24	5.2	1.5	1.8	1.2	1.0	2.1	2.0	1.8	1.8	1.9	4.3
	2880	2.5	2.1	1.6	1.6	1.3	1.2	1.2	1.7	1.2	1.3	2.1
	4320	2.2	2.0	1.3	0.9	0.8	0.8	1.1	1.7	1.0	1.0	2.0
5.0	24	7.9	3.6	3.6	2.8	3.1	4.5	3.4	2.5	2.6	3.6	5.9
	2880	4.9	4.0	3.2	2.3	2.4	2.3	2.5	3.5	2.3	2.8	3.4
	4320	4.5	3.8	3.1	1.4	1.5	3.6	2.3	3.3	2.2	2.5	3.3
10	24	12.4	8.4	8.1	4.0	4.0	8.3	6.6	8.8	4.4	7.2	9.7
	2880	9.6	7.9	7.2	5.6	5.6	5.1	5.7	7.5	4.4	6.1	7.2
	4320	9.4	7.4	7.0	3.3	3.2	4.4	6.0	7.1	4.5	5.7	7.2

Table 5: Exceedances of tail risk protection strategies in %

All classifiers achieve the risk targets, but both benchmark target VaR strategies, where VaR is estimated with ARMA-GARCH and ARMA-EVTGARCH, do not respect the constraint for $\text{TVaR}^{1\%,24}$. The Varspread strategy is not a target VaR strategy, but we also look at its exceedance and we see that it also fails for the same risk target, $\text{TVaR}^{2.5\%,24}$ and $\text{TVaR}_{5\%,24}$, indicating difficulties for those strategies to adapt to varying risk preferences.

Now, on Table 6, we compare the average return of the strategies associated with each classifier and the benchmarks on the whole test period. Achieving a larger average return, while having lower exceedances indicates a better risk-profile for a tail risk averse investor. First, the Ensemble strategy always have a larger average return than the average performance of the weak strategies, that is the strategies associated with the weak classifiers, except for the $\text{TVaR}^{1\%,2880}$ risk target, and has larger average return than the maximum of the weak strategies for the $\text{TVaR}^{1\%,24}$ and $\text{TVaR}^{1\%,4320}$. In general, the Ensemble strategy also outperforms the benchmark strategies, except for the target VaR strategies with $\text{TVaR}^{1\%,2880}$, $\text{TVaR}^{2.5\%,24}$, $\text{TVaR}^{5\%,24}$ and $\text{TVaR}^{10\%,24}$. Only these strategies are better suited than our offered Ensemble strategy for their respective investor's profile, as they offer larger average return, while respecting the corresponding risk target. This result shows that traders can benefit from combining classical econometrics and machine learning oriented approaches in order to offer better tail risk protection strategies to their investors.

α (%)	w (hours)	MLP	LSTM	ARMA- GARCH	ARMA- EVTGARCH	LPC- GARCH	CARL-vol	Ensemble	target VaR norm	target VaR evt	Varspread
1.0	24	4.9	5.1	2.3	1.6	6.5	-15.0	8.9	6.5	6.3	4.0
	2880	8.0	5.5	10.0	4.8	8.6	17.0	4.4	7.1	6.3	4.0
	4320	8.1	11.0	8.8	7.0	9.5	12.0	14.0	8.2	7.1	4.0
2.5	24	7.3	7.1	3.4	6.3	8.9	-0.7	6.8	7.0	7.5	3.7
	2880	6.9	9.9	3.6	1.4	2.2	12.0	8.5	6.4	6.3	3.7
	4320	5.8	10.0	5.3	4.5	6.3	12.0	8.3	7.4	7.5	3.7
5.0	24	7.2	9.2	5.8	6.7	9.3	2.9	7.1	7.8	8.3	2.7
	2880	6.8	13.0	0.8	1.4	2.7	13.0	8.7	5.9	6.2	2.7
	4320	8.5	12.0	1.5	2.1	5.1	16.0	11.0	6.5	7.2	2.7
10	24	5.7	9.6	5.5	6.1	2.8	6.9	6.9	5.9	7.8	2.4
	2880	10.0	7.5	1.6	1.8	5.5	14.0	6.8	5.5	6.3	2.4
	4320	14.0	9.8	3.8	4.1	5.5	12.0	8.1	5.8	7.3	2.4

Table 6: Average return in % (e-3). Buy&Hold: 10e-3 %

Historical performance On Figure 6, we show the historical performance on the test period and on Figure 7, the excess return over the BTC Buy&Hold benchmark in strong uptrend period. We can clearly see how Varspread strategy is too conservative in period of uptrend which severely affects its total return. While the Ensemble strategy significantly outperforms the other actively managed tail risk protection strategies in terms of total return for $\text{TVAR}^{1\%,24}$, $\text{TVAR}^{1\%,4320}$, $\text{TVAR}^{2.5\%,2880}$, $\text{TVAR}^{2.5\%,4320}$, $\text{TVAR}^{5\%,2880}$ and $\text{TVAR}^{5\%,4320}$. For the other risk targets, the historical performance is similar. In the meantime, only two strategies outperforms the simple Buy&Hold strategy on the whole test period, for $\text{TVAR}^{2.5\%,4320}$ and $\text{TVAR}^{5\%,4320}$ risk targets, managed by our proposed Ensemble model. This shows that for BTC asset, it is very hard to find the right trade-off between investment during strong uptrend and deinvestment in period of downtrend and that strategies based on simpler models, as Varspread or target VaR strategies, cannot beat the Buy&Hold benchmark in terms of total return, as they overshoots risk in strong uptrend periods.

Indeed, as you can see on Figure 7, in strong uptrend period, all strategies underperforms the Buy%Hold benchmark except for the Ensemble strategy with $\text{TVAR}^{1\%,4320}$ risk target.

Nevertheless, as we can see on Figure 8, that, in period of downtrend from 2019-06-26 to 2019-11-25 where BTC suffered a 60% drawdown, the Varspread strategy is very efficient at catching tail events achieving 50%, 48%, 64% and 64% excess return for $\alpha \in \{1\%, 2.5\%, 5\%, 10\%\}$ respectively even generating a positive total return of 4% for $\alpha \in \{5\%, 10\%\}$.

Switching strategy In order to benefit from the effective tail risk protection from the Varsrpead strategy in bear market and the well calibrated risk protection of the Ensemble strategy in bull market, we build a final switching strategy which either invest in the Varspread strategy or in the Ensemble strategy depending on the market regime. In order to decide when to switch strategy, we build a simple estimator of the trend by using a simple moving average of BTC price, defined for $n \in \mathbb{N}$ as $\text{MA}_{t-1,n} = \frac{1}{n}\sum_{i=1}^{n} P_{t-1}$. We define the following trend indicator δ_t as in Rickenberg (2019):

$$\delta_t = \begin{cases} 0, & \text{if } P_{t-1} \leqslant \text{MA}_{t-1,n} \\ 1, & \text{if } P_{t-1} > \text{MA}_{t-1,n} \end{cases}$$

Then we define the weights for each strategy as $w_t^{Ens} = \delta_t$ and $w_t^{Varspread} = 1 - \delta_t$. The historical performance of the final meta-strategy, denoted Switch strategy, is presented on Figure 9.

In Table 7, we present different backtest statistics for the benchmark, Ensemble and final Switch

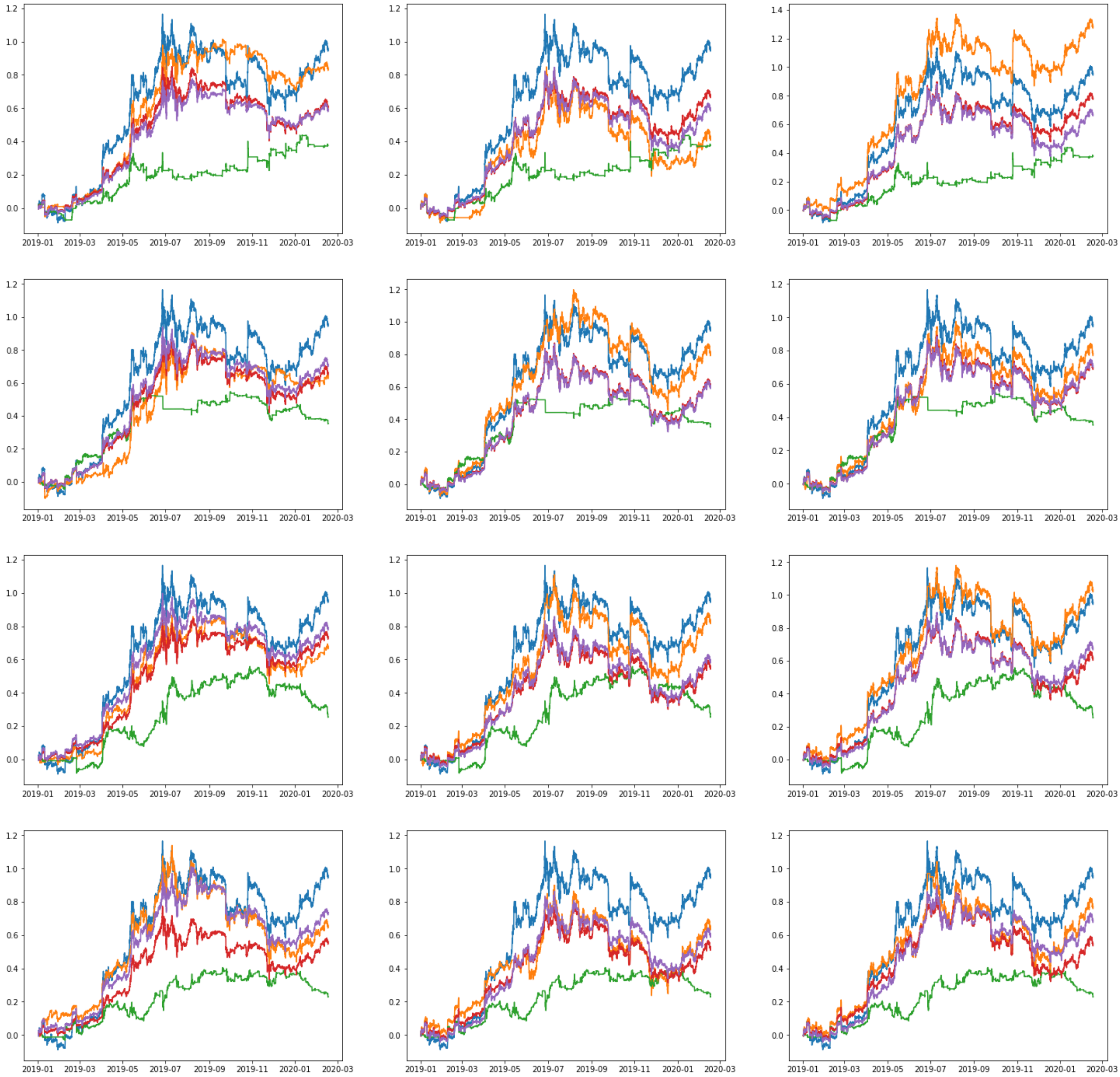

Figure 6: Ensemble strategy returns on test period for $\mathrm{TVAR}^{\alpha,w}$ with $\alpha \in \{1\%, 2.5\%, 5\%, 10\%\}$ from top to bottom and $w \in \{24, 2880, 4320\}$ from left to right with BTC Buy&Hold, Varspread, target VaR norm and target VaR evt benchmarks

strategies. First, we see that the Switch strategy has the highest total return on the testing period for relatively high risk target, but also for $\mathrm{TVAR}^{1\%,2880}$ and $\mathrm{TVAR}^{2.5\%,2880}$. Moreover, we can see that the Switch strategy provides an enhanced risk-return profile indicated by larger Sharpe ratio for all risk targets, than the average between the Varspread and the Ensemble strategies. For example, with $\mathrm{TVaR}^{5\%,2880}$, the Switch strategy increased the Sharpe Ratio by 62% and 219% for the Ensemble and Vaspread strategies respectively. On top, the switch strategy has the best risk-return profile for all risk targets, clearly outperforming the Buy&Hold and other benchmarks, except $\mathrm{TVAR}^{2.5\%,24}$ where the Varspread strategy has the largest Sharpe ratio. This indicates that the switch strategy is also better suited for Markowitz type optimisation where the objective

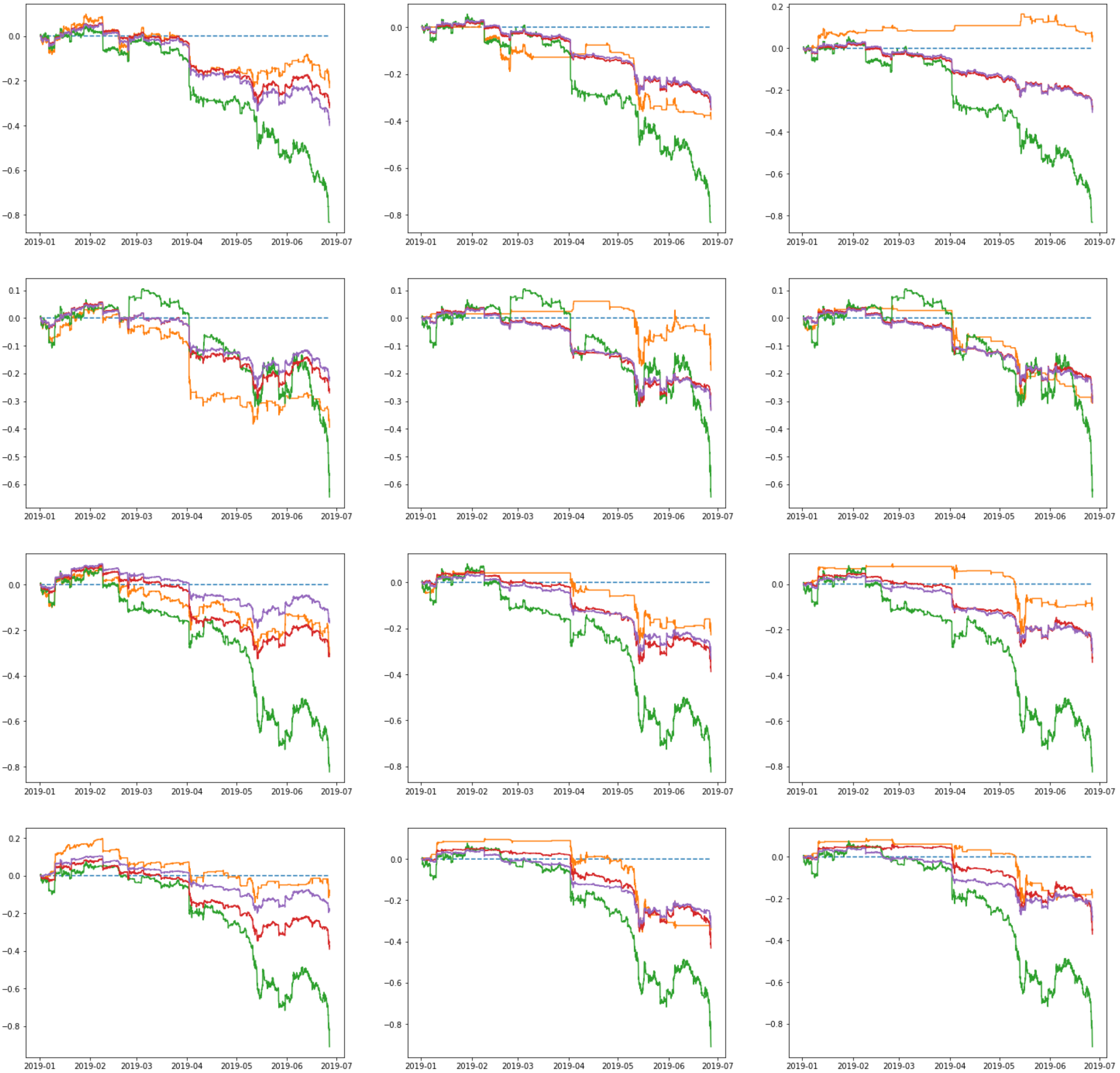

Figure 7: Ensemble strategy excess returns on uptrend period for TVAR$^{\alpha,w}$ with $\alpha \in \{1\%, 2.5\%, 5\%, 10\%\}$ from top to bottom and $w \in \{24, 2880, 4320\}$ from left to right with Var-spread, target VaR norm and target VaR evt benchmarks

is the mean-variance performance. For the Ensemble strategy itself the results are mixed with respect to mean-variance performance, since it has a relatively high volatility for all risk targets. Indeed, finding the right trade-off between participation in bull market regimes and hedging tail events comes with a cost in terms of volatility. Since in our case, we know that the Switch and Ensemble strategies are respecting the tail risk constraint, we are interested in the Sortino ratio, which penalizes only volatility associated with negative returns and is defined as $S = \frac{R}{\sigma^-}$, where R is the expected return and σ^- is the portfolio downside standard deviation. We can see that the Switch strategy clearly outperforms other benchmarks, except for conservative strategies with TVAR$^{1\%,24}$, TVAR$^{1\%,2880}$, TVAR$^{2.5\%,24}$ and TVAR$^{5\%,24}$.

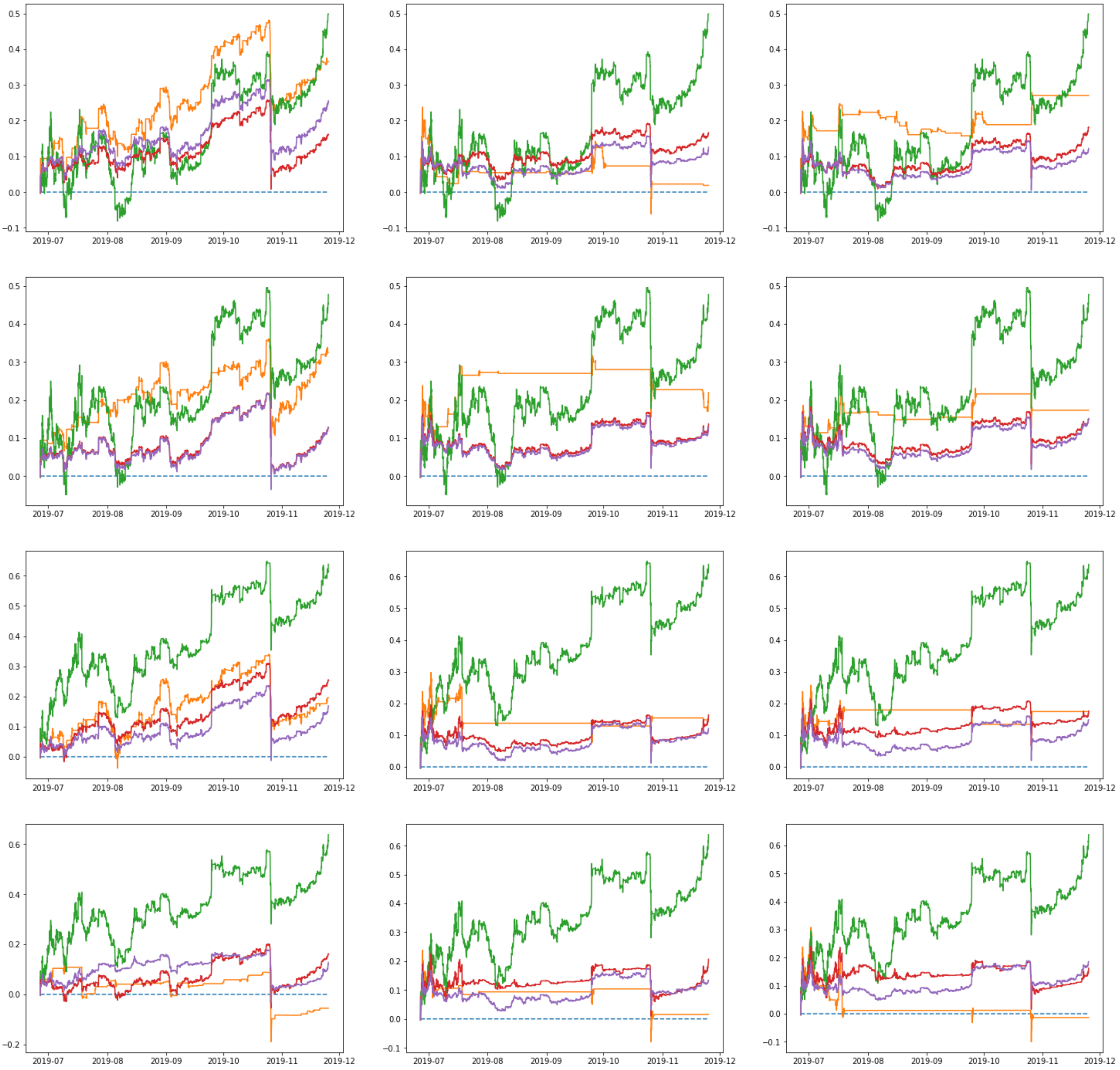

Figure 8: Ensemble strategy excess returns on downtrend period for TVAR$^{\alpha,w}$ with $\alpha \in \{1\%, 2.5\%, 5\%, 10\%\}$ from top to bottom and $w \in \{24, 2880, 4320\}$ from left to right with Var-spread, target VaR norm and target VaR evt benchmarks

7 Conclusion

We build a dynamic trading strategy which goals is to protect against severe losses by targeting a maximum Value-At-Risk defined for any significance level. For this, we model distributional properties of BTC returns in time with non-parametric models such as MLP and LSTM neural networks and compare their performance to local-parametric GARCH, parametric GARCH fitted with various distribution for the innovation and CARL-Vol models. In our approach, we directly take into account varying investor's risk preferences and optimize the strategy accordingly, allowing us to compare the models for multiple risk targets.

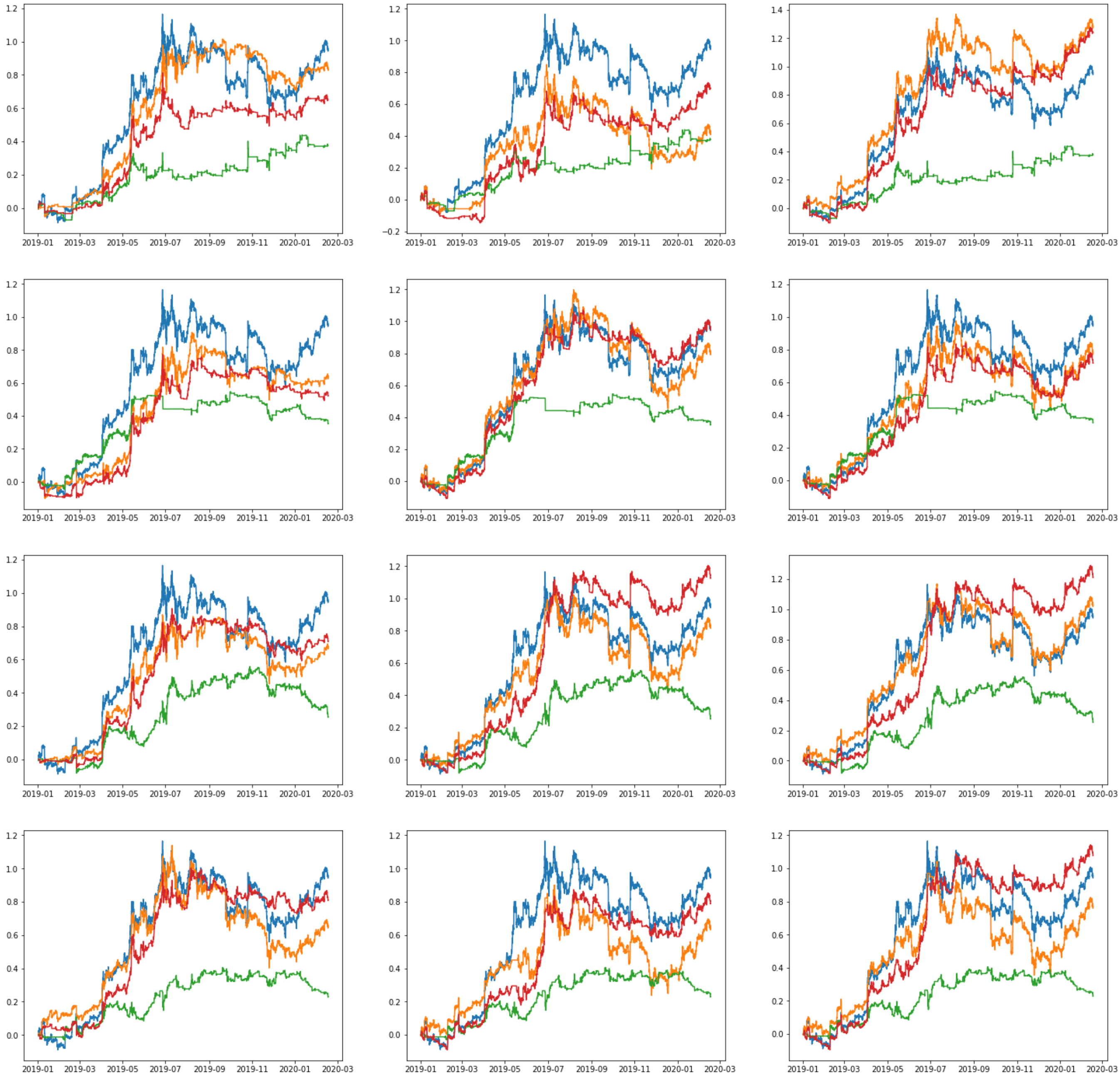

Figure 9: Historical performance on the test period of Ensemble, Varspread, Switch and Buy&Hold strategies

First, we can conclude that for relatively stable risk preferences, parametric and local parametric approaches have a better generalization than non-parametric methods, but for unstable preferences they work better, taking into account time varying risk targets and adapting to sudden changes in the market. On top, we show that, for BTC returns, a general model performing well enough in all market regimes is too hard to find. A tail loss protection strategy based on a single model, while often outperforming a simple Buy&Hold strategy, is still not optimal, either overshooting the risk target in periods of losses or being too conservative in bull market regimes. That is why, we show the benefits of combining multiple models in an Ensemble classifier, which improves generalization with a higher prediction accuracy and economic gains with an enhanced risk-return profile. Indeed, the Ensemble based strategy combines advantages of classical GARCH tools, which catches better

α (%)	w (hours)	Strategy	Total return	Sharpe ratio	Sortino ratio	MDD	volatility
		Buy&Hold	**86**	**1.37**	**2.62**	**52**	**0.69**
1.0	24	Ensemble	**76**	1.52	**2.52**	31	**0.54**
		Switch	59	**1.57**	2.29	30	0.41
		target VaR evt	54	1.28	2.13	33	0.46
		target VaR norm	55	1.24	2.10	**40**	0.49
		Varspread	**34**	**1.14**	**1.53**	**17**	**0.33**
	2880	Ensemble	38	**0.70**	**1.44**	**53**	**0.60**
		Switch	**63**	**1.50**	2.36	27	0.46
		target VaR evt	54	1.16	2.15	45	0.50
		target VaR norm	61	1.32	**2.37**	40	0.50
		Varspread	**34**	1.14	1.53	**17**	**0.33**
	4320	Ensemble	**115**	1.97	3.91	**43**	**0.64**
		Switch	111	**2.39**	**3.95**	23	0.51
		target VaR evt	60	1.31	2.37	**43**	0.50
		target VaR norm	71	1.54	2.75	39	0.50
		Varspread	**34**	**1.14**	**1.53**	**17**	**0.33**
2.5	24	Ensemble	58	**1.14**	**1.71**	36	**0.55**
		Switch	47	1.27	1.90	27	0.41
		target VaR evt	**64**	**1.40**	**2.37**	**42**	0.50
		target VaR norm	60	1.32	2.21	**42**	0.50
		Varspread	**34**	1.35	2.23	**18**	**0.26**
	2880	Ensemble	73	1.34	2.71	**55**	**0.59**
		Switch	**85**	**2.05**	**4.15**	32	0.46
		target VaR evt	54	**1.19**	**2.21**	45	0.50
		target VaR norm	55	1.23	2.30	45	0.49
		Varspread	**34**	1.35	2.23	**18**	**0.26**
	4320	Ensemble	**71**	**1.26**	2.54	**45**	**0.62**
		Switch	65	**1.50**	**2.91**	30	0.47
		target VaR evt	64	1.43	2.55	41	0.49
		target VaR norm	63	1.42	2.55	41	0.49
		Varspread	**32**	1.35	**2.23**	**18**	**0.26**
5.0	24	Ensemble	61	1.25	1.86	**39**	**0.53**
		Switch	64	**1.64**	2.69	29	0.43
		target VaR evt	**71**	1.58	**2.82**	39	0.49
		target VaR norm	66	1.53	2.67	34	0.47
		Varspread	**23**	**0.70**	**1.33**	**27**	**0.36**
	2880	Ensemble	75	1.38	2.60	**54**	**0.59**
		Switch	**101**	**2.23**	**5.08**	27	0.50
		target VaR evt	54	1.18	2.19	45	0.49
		target VaR norm	50	1.17	2.20	44	0.47
		Varspread	**23**	**0.70**	**1.33**	**27**	**0.36**
	4320	Ensemble	92	1.68	2.92	**49**	**0.61**
		Switch	**108**	**2.38**	**5.09**	23	0.50
		target VaR evt	61	1.38	2.48	42	0.49
		target VaR norm	56	1.29	2.39	42	0.47
		Varspread	**23**	**0.70**	**1.33**	**27**	**0.36**
10	24	Ensemble	59	1.07	1.94	**55**	**0.61**
		Switch	**73**	**1.70**	**3.03**	26	0.47
		target VaR evt	66	1.43	2.65	42	0.51
		target VaR norm	51	1.22	2.01	39	0.45
		Varspread	**21**	**0.74**	**1.18**	**17**	**0.30**
	2880	Ensemble	58	1.07	1.95	**52**	**0.60**
		Switch	**71**	**1.69**	**3.33**	27	0.46
		target VaR evt	54	1.21	2.29	44	0.49
		target VaR norm	47	1.18	2.16	40	0.44
		Varspread	**21**	**0.74**	**1.18**	**17**	**0.30**
	4320	Ensemble	70	1.32	2.33	**53**	**0.58**
		Switch	**97**	**2.27**	**4.50**	23	0.47
		target VaR evt	62	1.41	2.57	40	0.48
		target VaR norm	50	1.25	2.28	44	0.43
		Varspread	**21**	**0.74**	**1.18**	**17**	**0.30**

Table 7: Backtest performance of switch strategies: Total return (%), Sharpe ratio, Sortino ratio, Max Drawdown (MDD), Valut-at-Risk and Volatility of different strategies on the test period. The Total return, Sharpe and Sortino ratio are annualized. Legend: **winner** / **loser**

volatility clusters than more complex MLP or LSTM, and Deep learning tools which can adapt to time-varying properties of the data distribution. Further, we show that tail loss protection strategy also suits Markowitz type investors when we combine conservative strategies, such as Varspread and more aggressive ones, such as Ensemble strategy, into a switching strategy.

For the purpose of tail loss protection, our proposed Ensemble strategy can be improved. Here, we measure risk with Value-At-Risk, nonetheless, other risk metrics can be considered for optimization, in particular Expected Shortfall or Maximum Drawdown. On top, it would be interesting to open the Ensemble learning tool box and to study how the different classifiers interacts which other in time, to better understand when their corresponding tail loss protection strategy might be suited for the respective market regime.

References

Black, F. (1976). Studies of stock price volatility changes, *Proceedings of the 1976 Meetings of the American Statistical Association*, Business and Economics Statistics Section, pp. 177–189.

Black, F. and Jones, R. W. (1987). Simplifying portfolio insurance, **14**(1): 48–51.

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

Bollerslev, T. and Todorov, V. (2011). Tails, fears, and risk premia, *The Journal of Finance* **66**(6): 2165–2211.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.2011.01695.x*

Bollerslev, T., Todorov, V. and Xu, L. (2015). Tail risk premia and return predictability, *Journal of Financial Economics* **118**(1): 113–134.
URL: *https://EconPapers.repec.org/RePEc:eee:jfinec:v:118:y:2015:i:1:p:113-134*

Bouri, E., Gil-Alana, L., Gupta, R. and Roubaud, D. (2019). Modelling long memory volatility in the bitcoin market: Evidence of persistence and structural breaks, *International Journal of Finance & Economics* **24**(1): 412–426.

Chen, S., Chen, C. Y.-H., Härdle, W. K., Lee, T. and Ong, B. (2018). Econometric analysis of a cryptocurrency index for portfolio investment, *in* D. Lee Kuo Chuen and R. Deng (eds), *Handbook of Blockchain, Digital Finance, and Inclusion, Volume 1*, Academic Press, pp. 175 – 206.

Christie, A. (1982). The stochastic behavior of common stock variances: Value, leverage and interest rate effects, *Journal of Financial Economics* **10**(4): 407–432.

Christoffersen, P. F. and Diebold, F. X. (2006). Financial asset returns, direction-of-change forecasting, and volatility dynamics, *Management Science* **52**(8): 1273–1287.

Chung, J. and Hong, Y. (2007). Model-free evaluation of directional predictability in foreign exchange markets, *Journal of Applied Econometrics* **22**(5): 855–889.

Cizek, P., Härdle, W. K. and Spokoiny, V. (2009). Adaptive pointwise estimation in time-inhomogeneous conditional heteroscedasticity models, *The Econometrics Journal* **12**(2): 248–271.

de Prado, M. L. (2018). *Advances in Financial Machine Learning*, 1st edn, Wiley Publishing.

Franke, J. (1999). Nonlinear and nonparametric methods for analyzing financial time series, *in* P. Kall and H.-J. Lüthi (eds), *Operations Research Proceedings 1998*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 271–282.

Franke, J., Härdle, W. K. and Hafner, C. M. (2019). *Statistics of Financial Markets - An introduction*, 5th edn, Springer Verlag.

Hakansson, N. (1971). Capital growth and the mean-variance approach to portfolio selection, *Journal of Financial and Quantitative Analysis* **6**(1): 517–557.

Hamidi, B., Jurzenko, E. and Maillet, B. (2009). A caviar modelling for a simple time-varying proportion portfolio insurance strategy, *Bankers, Markets and Investors* **102**(4–21).

Happersberger, D., Lohre, H. and Nolte, I. (2019). Estimating portfolio risk for tail risk protection strategies, *European Financial Management* .

Härdle, W. K., Herwatz, H. and Spokoiny, V. (2003). Time inhomogeneous multiple volatility modelling, *Journal of Financial Econometrics* **1**: 55–99.

Hillebrand, E. (2005). Neglecting parameter changes in garch models, *Journal of Econometrics* **129**(1): 121 – 138. Modelling structural breaks.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory, *Neural Comput.* **9**(8): 1735–1780.
URL: *https://doi.org/10.1162/neco.1997.9.8.1735*

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems, *Technometrics* **12**(1): 55–67.

Honegger, D. and Wijewickreme, D. (2013). Seismic risk assessment for oil and gas pipelines, *in* S. Tesfamariam and K. Goda (eds), *Handbook of Seismic Risk Analysis and Management of Civil Infrastructure Systems*, Woodhead Publishing Series in Civil and Structural Engineering, Woodhead Publishing, pp. 682 – 715.

Jordà, O. and Taylor, A. M. (2011). Performance evaluation of zero net-investment strategies, *Working Paper 17150*, National Bureau of Economic Research.

Kahneman, D. and Tversky, A. (1979). Prospect theory: An analysis of decision under risk, *Econometrica* **47**(2): 263–91.
URL: *https://EconPapers.repec.org/RePEc:ecm:emetrp:v:47:y:1979:i:2:p:263-91*

Kelly, J. L. (1956). A new interpretation of information rate, *The Bell System Technical Journal* **35**(4): 917–926.

Kim, H. Y. and Won, C. H. (2018). Forecasting the volatility of stock price index: A hybrid model integrating lstm with multiple garch-type models, *Expert Systems with Applications* **103**: 25 – 37.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization, *CoRR* **abs/1412.6980**.

Klochkov, Y., Härdle, W. K. and Xu, X. (2019). Localizing multivariate CAViar, *SFB 649 Discussion Paper, submitted to Journal of Applied Econometrics* **2019**(007).

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.
URL: *https://doi.org/10.1093/rfs/hht039*

Kunreuther, H. (2002). Risk analysis and risk management in an uncertain world, *Risk Analysis* **22**(4): 655–664.

Lambert, J., Matalas, N., Ling, C., Haimes, Y. and Li, D. (1994). Selection of probability distributions in characterizing risk of extreme events, *Risk Analysis* **14**(5): 731–742.

Leshno, M., Lin, V. Y., Pinkus, A. and Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function, *Neural Networks* **6**(6): 861 – 867.

Linton, O. and Whang, Y.-J. (2007). The quantilogram: With an application to evaluating directional predictability, *Journal of Econometrics* **141**(1): 250–282.

Longin, F. and Solnik, B. (2001). Extreme correlation of international equity markets, *The Journal of Finance* **56**(2): 649–676.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1111/0022-1082.00340*

Markowitz, H. (1952). Portfolio selection, *Journal of Finance* **7**(1): 77–91.

Mason, S. J., Galpin, J. S., Goddard, L., Graham, N. E. and Rajartnam, B. (2007). Conditional Exceedance Probabilities, *Monthly Weather Review* **135**(2): 363–372.

McNeil, A. and Frey, R. (2000). Estimation of tail-related risk measures for heteroscedastic financial time series: an extreme value approach, *Journal of Empirical Finance* **7**(3): 271 – 300. Special issue on Risk Management.

Mercurio, D. and Spokoiny, V. (2004). Statistical inference for time-inhomogeneous volatility models, *The Annals of Statistics* **32**: 577–602.

Packham, N., Papenbrock, J., Schwendner, P. and Woebbeking, F. (2017). Tail-risk protection trading strategies, *Quantitative Finance* **17**(5): 729–744.
URL: *https://doi.org/10.1080/14697688.2016.1249512*

Rickenberg, L. (2019). Tail risk targeting: Target var and cvar strategies, *SSRN Electronic Journal* .

Spokoiny, V. (1998). Estimation of a function with discontinuities via local polynomial fit with an adaptive window choice, *The Annals of Statistics* **26**: 1356–78.

Spokoiny, V. (2009a). Multiscale local change point detection with applications to value-at-risk, *The Annals of Statistics* **37**: 1405–1436.

Spokoiny, V. and Zhilova, M. (2015). Bootstrap confidence sets under model misspecification, *The Annals of Statistics* **43**(1): 2653–2675.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting, *Journal of Machine Learning Research* **15**(56): 1929–1958.

Taylor, J. and Yu, K. (2016). Using auto-regressive logit models to forecast the exceedance probability for financial risk management, *Journal of the Royal Statistical Society: Series A (Statistics in Society)* **179**(4): 1069–1092.

Wen, F., He, Z., Chen, X. and Li, J. (2014). Investors' risk preference characteristics and conditional skewness, *Mathematical Problems in Engineering* .
URL: *https://doi.org/10.1155/2014/814965*

Zhang, Z., Zohren, S. and Roberts, S. (2020). Deep learning for portfolio optimization, *The Journal of Financial Data Science* .