Dynamic Asset Management Factor Timing Project

March 22, 2020

Time Series Momentum for Improved Factor Timing

Arijit Santra, Max Lamberti, Jon Kaplan, Sam Smith, Suki Yang

Abstract

We improve on quant factor portfolios by implementing *Deep Momentum Networks* - a neural network based approach for constructing optimized time series momentum factors. We find that the Deep Momentum Network implementation provides improved risk to reward metrics over a 25 year backtest period when compared to baseline time series momentum implementations as well traditional cross-sectional momentum. We demonstrate that the performance of the factor is sensitive to transaction costs and find that the factor signal has deteriorated significantly over the previous decade. Lastly, we incorporate the time series momentum strategy into our reference factor portfolios.

1 Introduction

We implement a factor timing approach for time series momentum (TSMOM). In particular, we improve on baseline implementations presented in Moskowitz et al. (2012) and Baltas & Kosowski (2013) by deploying the *Deep Momentum Network* (DMN) proposed by Lim et al. (2019). The aim is to time the factor exposures of an actively managed portfolio by using a time-series momentum factor to outperform a reference portfolio.

Momentum trading strategies are customarily implemented as cross-sectional trading strategies among individual stocks, but rather than focus on the relative returns of securities in the cross-section, we focus on TSMOM, defined in Gupta & Kelly (2018) as a performance persistence phenomenon by which an asset's own recent return predicts its future returns, and which provides a purer measure of expected factor returns than does the cross-sectional method.

The DMN approach optimizes exposure to an asset through a chosen portfolio metric (expected return, Sortino ratio, Sharpe ratio, etc.), making use of lagged asset performance and volatility measures when forming predictions. The DMN is different to other TSMOM implementations in that it is able to utilize non-linear relationships in lagged asset data. It furthermore differentiates itself from classical supervised machine learning approaches in that it doesn't optimize for an explicit target, but rather the portfolio performance metric itself.

We also examine using TSMOM as an additional 6th factor to traditional portfolios composed of the five Fama-French factors. The addition of TSMOM increases portfolio returns and decreases historical risk metrics. We believe, with the deterioration of some historical risk premiums after publications, that a model composed of several risk factors in addition to TSMOM may be more robust to future out of sample returns.

The rest of the paper is organized as follows. Section 2 considers the target investor for a TSMOM strategy, explains the benchmark portfolio against which we are measuring our strategy, and explores the potential motivation for adding time series momentum as a factor. Section 3 examines the different factors to which a quant hedge fund might have exposure. Section 4 introduces the strategy that we used and the methodology. Section 5 analyzes the results that the TSMOM model produces, incorporating the DMN to enhance performance. Section 6 concludes.

2 Target Investor & Motivation

Given the complexity of the strategy, we only consider it appropriate for a highly sophisticated investor that views their investments in terms of quant factors, and who might be looking to add a new factor to their considerations, so we view our target investors to be quant factor hedge funds (like AQR).

So why might we consider a time series momentum factor? While hardcore proponents of the efficient market hypothesis may take offense to the earlier statement that an asset's own recent return can predict its future returns, evidence suggests that this investment strategy isn't as futile as traditional market theory would lead one to believe. In fact, Moskowitz et al. (2012) show the *trend*-pricing anomaly exists across time as well as asset classes, documenting the effect in equity indexes, currency, commodity and bond futures over the course of 25 years. They report significant persistence in returns within 1-12 month lags and evidence of mean reversion across longer horizons.

To illustrate the effect, we partially replicate the results presented in Baltas & Kosowski (2013) by performing an AR(1) regression on our own cross-sector futures data. Similar to Baltas & Kosowski (2013), we recover the same short and long horizon trends for the regression coefficient, illustrated in Figure 1. The following equation specifies the AR model:

$$\frac{r_i(t_m, t_{m+1})}{\sigma_i(t_m)} = \alpha + \beta_i \frac{r(t_{m-\tau}, t_{m-\tau+1})}{\sigma_i(t_{m-\tau})}$$

Where r is the return of asset i, t indicates time period, σ is a volatility estimator for the asset returns, α is the regression constant and β is the regression coefficient. Note that in order to combine data across asset classes, we normalize returns by the previous month realized volatility. While Baltas & Kosowski (2013) use a variance-covariance estimator which is robust to cross-asset correlations, autocorrelation and heteroskedasticity, for simplicity, we only make use of a heteroskedasticity and autocorrelation robust estimator. This implies that our standard errors are underestimated, for which reason we don't cite confidence intervals. We mainly include the regression for illustrative purposes. Nevertheless, we would be remiss not to mention that our results do appear to be less significant than those reported in Baltas & Kosowski (2013), which could be a result of windowing – as we show later, the TSMOM anomaly seems to have been increasingly deteriorating after 2010.

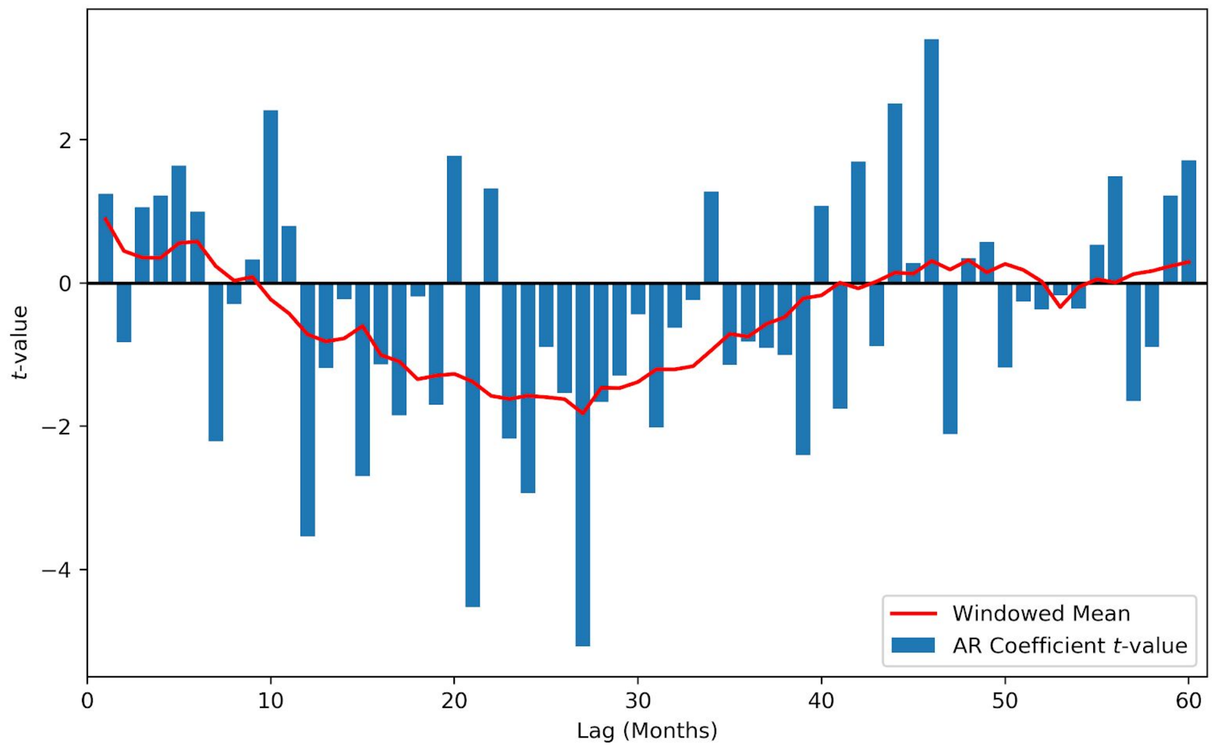

Figure 1: An illustration of the t-value of the AR coefficient of 60 separate regressions at different monthly lags. Bars show the coefficient t-value, the red line is a 12-month windowed mean of the t-values. The data is cross-asset-class monthly futures returns data normalized by asset volatility.

The t-statistic of AR beta at different monthly horizons is shown in Figure 1. We observe a positive bias for lags within a 1-12 month timeframe (trend-effect) and a negative bias for 1-3 year time frames (mean-reversion). For longer horizons the signal subsides. A positive AR-coefficient value implies a trend-effect, a negative value indicates mean reversion.

It should be noted that the more popular relative of TSMOM, cross-sectional momentum, is closely related to the TSMOM factor. Cross-sectional momentum is dynamically constructed based on the relative performance of similar securities to one another, i.e. buying winners and selling losers. Moskowitz et al. (2012) report an R^2 of 44% between an all-asset TSMOM factor and the all-asset cross-sectional factor constructed in Asness et al. (2010), showing that the two factors capture a significant part of each other's variance.

3 Our Factor Universe

In addition to the time series momentum factor we construct, we consider the following factors:

a. Market: Market Returns excess of risk-free rate
b. Size: Small minus Big (Fama-French SMB)
c. Value: High minus Low book value-to-market price ratio (Fama-French HML)
d. Operating Profit: Robust minus Weak (Fama-French RMW)
e. Investment: Investing Conservatively minus Aggressively (Fama-French CMA)

We use the 5 Fama-French factors from the mid-1960's to 2020 to determine what risk factors each portfolio is exposed to. Note this does not include momentum. The graph below shows historical portfolio value growth over time for each of these 5 factors (Market, Size, Value, Operating Profit, Investment).

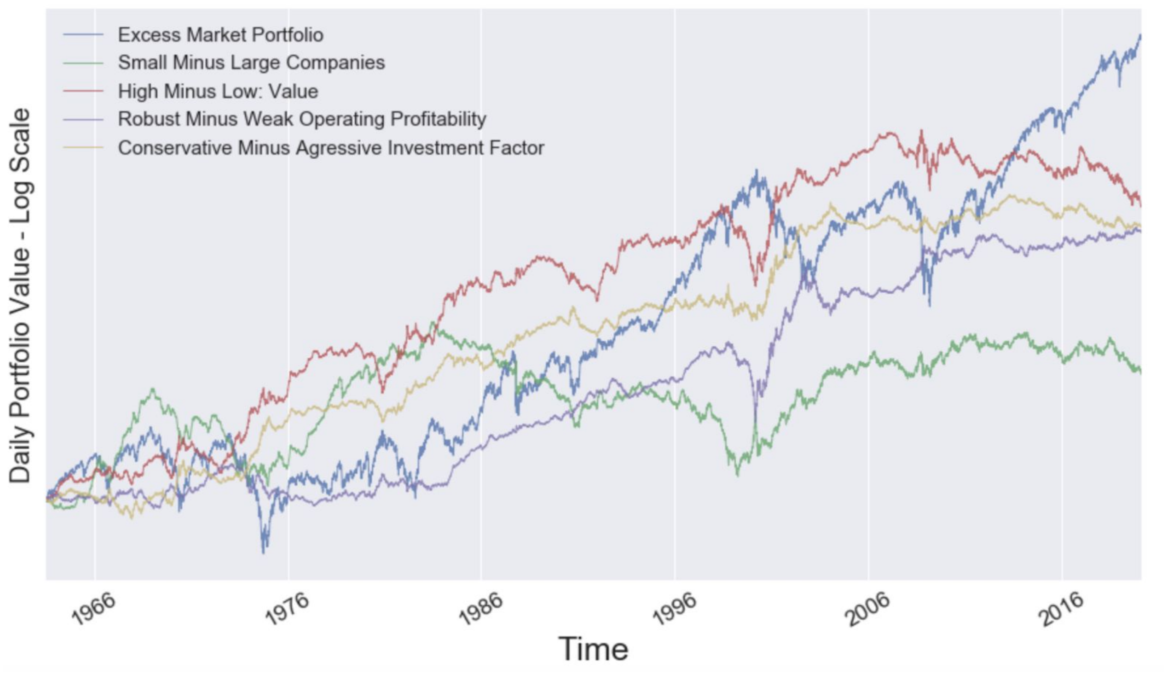

Figure 2: Historical portfolio value growth for each of the five Fama-French factors

These factors are all long/short with low risk/returns historically. Returns range from roughly 6% (excess market return) to only 1.5% (for size). Remember the sample period we have data for is from 1964 so some of the size premium addressed in literature is not captured in our sample.

4 Strategy & Methodology

100% Equity

Why not invest in 100% equities? Well 100% equities (assuming no costs) would return roughly 10% a year annualized (on a historical basis), but with a 15% average annual volatility and a max drawdown of 55%. This implies a multiple of 0.66 for the strategy and our factor exposure (using the FF 5-factor model) would clearly show that we have all market risk. We should be able to diversify these risks to produce better risk adjusted returns!

60/40 Portfolio

The next classic portfolio we should touch on is the 60/40 portfolio. We proxied this portfolio assuming we were invested in 60% the FF market portfolio and 40% the risk free rate with no transaction fees or holding costs. While this portfolio only produces an 8% per year average annualized return, it does so with a smaller vol of only 9% and a smaller max drawdown of 36% than being invested in 100% equities. Overall the multiple is 0.89 so in many ways the 60/40 does appear to be better diversified from the all equity portfolio. The factor exposures are still overwhelmingly correlated with the market risk (see factor exposures below).

Factor	Beta
Constant	0%
Excess Market Portfolio	60%
Small Minus Large Companies - Size	0%
High Minus Low - Value	0%
Robust Minus Weak - Operating Profitability	0%
Conservative Minus Aggressive - Investment Factor	0%

Table 1: Risk factor decomposition of the 60/40 portfolio

Equal Weight, Risk Parity, and Momentum on the Fama French 5 Factors

Next we wanted to achieve better risk adjusted returns by diversifying our factor exposures. The best way to do this was to directly invest in combinations of the Fama French risk factors themselves. We started with an equal weight of each of the 5 factors to achieve maximum factor diversification. The equal weight portfolio produces a multiple of 1.09 and a maximum drawdown of 17%, but does so at the cost of any real returns with an average annual return of 3.7%.

We also applied a risk parity approach investing more in FF factors with lower volatility in the last month. This tilted away from more volatile market risk (as one would anticipate) and produced a multiple of 1.15 but with only 3.1% annualized returns. The diversified factor exposures can be seen in the chart below.

Factor	Beta
Constant	0%
Excess Market Portfolio	8%
Small Minus Large Companies - Size	16%
High Minus Low - Value	13%
Robust Minus Weak - Operating Profitability	21%
Conservative Minus Aggressive - Investment Factor	25%

Table 2: Risk factor decomposition of the 5 factor risk parity portfolio

Lastly, to try to get higher returns we looked at a momentum strategy on top of the FF 5 factor model to tilt out weights more heavily on factors that had performed well in the past year (excluding the last month to avoid the short term reversal). This did increase returns but only to 3.8% annualized with a multiple of 1.1.

We have seen that 100% equity is too concentrated and is only exposed to one risk factor (the market). With the 60/40 portfolio we were able to achieve better risk adjusted return but with lower absolute returns and we still had only one risk factor. Explicit diversifying risk factors brought us better risk adjusted returns, but with much too low absolute returns. Moving on, our paper proceeds to focus on a time series momentum strategy that aims to increase absolute returns while minimizing risk factors.

TSMOM Futures Data

We implement the TSMOM strategy using purchased futures data from Pinnacle Data Corp CLC Database. The original data set consists of 98 contracts traded across FX, Fixed Income, Commodities and Equity. The data ranges from 1969 for the oldest contracts all the way to 2020. We only use the subset of the contracts that have complete data from 1990 to 2020, which reduces the data set to 48 contracts. The reason we chose this data set is because it was the data set of choice for Lim et al. (2019) when creating their deep momentum network. When working with models that are difficult to fit and tune, like neural networks, having reportedly successful data provides some hope that it is the ill-calibrated model that is failing and not the poor quality of the data, a very powerful motivator when debugging custom tensorflow loss functions at 3am.

Since the raw data is a series of different maturity futures contracts, some preformatting must be done to obtain a continuous price series for each asset. We use the *backwards ratio-adjusted method* to do the rolling for each asset. Compared to other methods (*first-of-month*, *last-trading-day*, *most-liquid*), the ratio-adjusted method does not produce jumps in the continuous price series, which is a desirable feature if we are working with returns data.

An important quality to consider about futures data is that the price not only has exposure to shocks in the asset's spot price but is also exposed to roll yield. Based on the movement of the contract along the term structure, the holder of a futures contract may make or lose money (depending if he is on the long / short leg and if the term structure is contango / backwardation). The returns of a futures contract may thus be decomposed into two components:

$$r_{futures} = r_{spot} + r_{roll}$$

While we didn't have the time to disentangle the two return streams for this project, it would be an interesting extension to model these two return streams separately. As an example, for E-mini futures S&P 500 we could subtract out the S&P 500 spot price to obtain the roll yield. We could then use term structure data to create a predictive model for the roll yield separate of the TSMOM we capture from the spot price. The combination of the two would give a more granular view of the return dynamics and hopefully a more powerful forecasting model.

Baseline TSMOM Strategy

As a baseline TSMOM factor we use the *sign*-strategy presented by Moskowitz et al. (2012). The strategy is described by the following formula:

$$r_{t,t+1}^{TSMOM} = SIGN(r_{t-12,t}^{s})\frac{\sigma_{tgt}}{\sigma_t^s} r_{t,t+1}^{s}$$

$$SIGN(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{otherwise} \end{cases}$$

Where *s* indicates the asset or contract being traded, *t* is the time period, σ_t is a backward looking volatility estimator for asset *s*, σ_{tgt} is the target volatility. In plain English, this strategy is the trading rule to go long in an asset if it has had a net positive return over the past 12 months and go short otherwise. The exposure or leveraged weighting to the asset is such that a target volatility σ_{tgt} is achieved. This weighting is highly reliable on the accuracy of the volatility estimator σ_t. Different choices of the volatility estimator and the estimator's effect on the TSMOM strategy are discussed in Baltas & Kosowski (2013). We later return to this discussion when inspecting deep momentum networks.

Since we are normalizing asset returns with volatility, we can create a cross-asset TSMOM portfolio by arithmetically combining the risk-weighted returns and leveraging it to a desired target volatility (essentially a risk-parity portfolio of single-asset TSMOM strategies). This is done by taking the sum over all assets *S*:

$$r_{t,t+1}^{TSMOM} = \frac{1}{N}\sum_{s=1}^{N} SIGN(r_{t-12,t}^{s})\frac{\sigma_{tgt}}{\sigma_t^s} r_{t,t+1}^{s}$$

Where *N* is the number of assets. To introduce a more consistent formalism, Lim et al. argue that there are two components to any TSMOM strategy. First, a prediction of the *trend*, i.e. the direction and/or magnitude of the next period returns. Second, given the trend estimate, the position *size* to take for the next period bet. According to this formalism, the sign-strategy described above would be encapsulated by:

$$Y = r_{t-12,t}$$
$$X = SIGN(Y)$$

While this basic strategy is effective at capturing TSMOM, it does lend itself to criticism. To touch on the most important shortcomings of the strategy:

1. **Position Sizing:** The position sizing is determined fully by the volatility estimator σ_t, which is backward looking. If the model had predictive ability, based on confidence and size of the move, it should be able to do more optimal position sizing, taking larger risk when greater return is expected.

2. **Trend Estimator:** The trend estimator Y is qualitatively designed based on observational data such as that in Figure 1. By creating a statistical model which looks at a richer feature space, it ought to be possible to develop a better trend-estimator.

3. **Non-optimal:** A last criticism is that this model isn't optimized for any return metric. Depending on our investment preferences, we would like to optimize the TSMOM factor to have certain characteristics, for example *maximum-return*, *minimum-risk-to-reward*, or *minimum-downturn-to-reward*. By optimizing with respect to these investment metrics, we can design a TSMOM factor suited to the investor's preferences.

Deep Momentum Networks

The *Deep Momentum Networks* (DMN) proposed by Lim et al. (2019) address the shortcomings of the basic sign-strategy. DMNs use a traditional neural network architecture to map lagged time series features to an optimal next-period trend or size prediction.

In a traditional *supervised* learning problem, we have a labelled data set, i.e. explicitly defined training targets. For our problem, this would mean that we have knowledge of the optimal position size Y at all previous time steps and optimize a metric such as *mean-squared-error* or *mean-absolute-error* to derive the optimal model. However, when forming a trading strategy, we don't have these labels. If we did, we would already have a model to determine optimal position size, so why fit a model on top? In practice, optimal position sizing is determined by the investor's utility and risk preferences. So, instead of optimizing on known targets and traditional loss functions, DMNs optimize their prediction on financial trading metrics via a custom loss function.

DMN Loss Function

We explore a loss function that optimizes Sharpe ratio. The implementation isn't limited to these loss functions, for example, if the investor is more concerned about downside risk with respect to target returns, he may choose to optimize on Sortino ratio or drawdown. We define the loss functions as follows:

$$\mathcal{L}_{Sharpe} = \frac{\sum R_{s,t}}{\sqrt{\sum R_{s,t}^2 - (\sum R_{s,t})^2}}$$

$$R_{s,t} = X_{s,t} \frac{\sigma_{tgt}}{\sigma_{s,t}} r^{s}_{t,t+1}$$

Where s indicates the asset or contract, t the time period, r is return, $\sigma_{s,t}$ is a backward-looking volatility estimator for asset s, σ_{tgt} is the target volatility. In the loss functions the input the DMN provides is the position size $X \in [-1, 1]$. Note that we thus let the network jointly predict trend and position size. Positive values constitute a long position, negative values short. By predicting and optimizing on position size, we aim for the DMN to capture opportune times of when to take more or less risk.

DMN Architecture & Hyperparameter Optimization

Since the main innovation of DMN is the use of custom loss functions, the black box which fits between input and output is arbitrary. Indeed, we might be better-off using a simpler machine learning model such as a support-vector-machine. But for the sake of the modelling challenge, we proceed with the neural network approach.

The network architectures explored in Lim et al. (2019) are the feedforward *Multilayer-Perceptron* (MLP), a convolution-based architecture, *WaveNet* (CNV), as well as a *Long-Short-Term-Memory* Network (LSTM). They find, perhaps unsurprisingly, that LSTM outperforms MLP and CNV. However, the simpler MLP beats CNV.

While LSTMs seem to be the better approach for a time series problem like this, they require more extensive data preprocessing and tuning. In the end we settled for the MLP architecture, which provides the best compromise between result quality, implementation complexity, and convergence speed.

The MLP network is implemented using *Keras*. Tunable hyperparameters include the dropout of the input layer, dropout of the hidden layers, number of hidden layers, number of neurons per hidden layer, activation function for the input layer, activation function for the hidden layers and the learning rate of the optimizer. We optimize the model hyperparameters using *hyperopt*, a framework for Bayesian optimization of model parameters. For the hyperparameter optimization we train each network on data from 1990-1995 and validate on data from 1995-2000. We choose the model architecture with the smallest loss on the validation set, which results in the architecture presented in Table 3 (parameters rounded).

Label	Layer Type	Output Shape	Num Params	Activation	Dropout
Input	Dense	9	90	RELU	0.000
Hidden	Dense	5	50	RELU	0.350
Output	Dense	1	6	TANH	0.000

Table 3: Summary of our optimal network architecture as determined through our hyperparameter optimization routine. The total number of trainable parameters is 146 and the Adam optimizer uses a learning rate of 0.008.

To constrain the prediction interval to [-1, 1] we use the hyperbolic tan function as the output layer activation. Furthermore, we use the *Adam* optimizer, which we found to converge better than *Stochastic Gradient Descent*. We selected batch size for maximum computational speed of our specific compute / GPU machine. In general, we found that the data required limited training epochs, usually less than 10 epochs seemed sufficient, most best-scoring models only required two or three epochs of training. This points to the fact that these networks really aren't that "Deep". While the DMN does have hidden layers, given the small model complexity, the naming "Deep Momentum Networks" seems somewhat sensational.

The nine input features used to fit the DMN model are outlined in Appendix A.

DMN Training Methodology & Backtesting

To eliminate look-ahead-bias, we use a rolling window optimization technique where we refit the DMN at the start of every year. In each training session the DMN is fit only on chronologically preceding data. We use the final year of the data as a validation set for early stopping, which guards against overfitting the test set. For example, one training session may be training the DMN on data from 01/01/1990 to 31/12/1998 and validating on 1999 data for early stopping. This model would then only be used for predictions and backtesting for dates after 01/01/2000.

With the fitted model we make predictions for all assets for the full year following the validation data set. The model predicts the optimal position size at each time step as well as whether the position is long or short. Using the same volatility weighting methodology as in the Sharpe loss function, we combine predictions into a single portfolio and arrive at backtest returns according to the following formula:

$$r_{t,t+1}^{DMN} = \frac{1}{N} \frac{\sigma_{tgt}}{\sigma_t^{DMN}} \sum_{s,t} \frac{X_{s,t}}{\sigma_t^s} r_{t,t+1}^s$$

Where X is the position size, t indicates the time period, s the asset or contract, σ is the volatility estimator for the asset, N is the number of assets in the portfolio. The σ related to DMN is an additional backward looking scaling factor to make sure that the combined portfolio has a desired target volatility.

Due to volatility scaling, the assets are commonly traded on leverage. Leverage is not a practical implementation concern for the strategy, as all traded contracts are futures, which are de facto traded in accounts commonly allowing 20x leverage for institutional traders. However, we do make the controversial assumption that we can serve the margin requirement in times of stress. Our backtest methodology does not include margin calls or deleveraging of the account. Incorporating this into the returns of the strategy would severely complicate our goal of creating a pure TSMOM factor by deteriorating the signal which we are trying to encapsulate in the first place. From an implementation perspective, this means that the investor or fund must have enough liquidity to stomach the downturns, even during prolonged losing streaks.

For institutional investors, trading futures contracts has very little cost involved when compared to other assets. There are no extra fees for being on the short leg of a contract and exchange commissions or brokerage fees are small. However, when implementing any strategy, particularly a quant factor strategy for a large fund, there could be significant implementation shortfall involved. This means that we execute at a lower than desired price, consistently eating into our returns. To be mindful to trading costs and market impact, we gauge cost relative to portfolio turnover, which, in the context of volatility weighted portfolios, is defined by Baltas & Kosowski (2015) as:

$$O_{t+1,t} = c\left(\frac{X_{t+1}}{\sigma_{t+1}} - \frac{X_t}{\sigma_t}\right)$$

Where O is the turnover, c is a constant for transaction cost, X is the position size at t, σ is the volatility estimator.

We don't consider potential interest accrual from excess cash in times when the TSMOM strategy has decreased exposure.

5 Results and Analysis

TSMOM Factor Results

We display the performance of the TSMOM factor with and without transaction costs in Figure 3. For comparison, we display performance of the AQR cross-sectional momentum factor as well as those of the baseline SIGN-TSMOM. The curve closely replicates the one produced by Lim et al. (2019), which confirms that we are capturing the same qualitative factor behaviour as they did with their DMN. Pure DMN-TSMOM easily outperforms the other strategies over the backtest period. However, there seems to be a structural change around 2005 when the DMN-TSMOM returns start deteriorating and volatility starts increasing. This effect exacerbates over time and the DMN-TSMOM strategy posts flat returns post 2012. The AQR momentum factor as well as SIGN-TSMOM post more consistent performance throughout the backtest period. A plus for TSMOM strategies in general however, is that they produce smaller downturns than the AQR cross-sectional momentum. When applying a transaction cost of 2 basis points to DMN-TSMOM we observe a significant reduction in performance. This is evident of the fact that momentum strategies, as produced by our DMN, have large turnover. To observe the effect of a range of different transaction costs on the DMN-TSMOM factor, please refer to Appendix B.

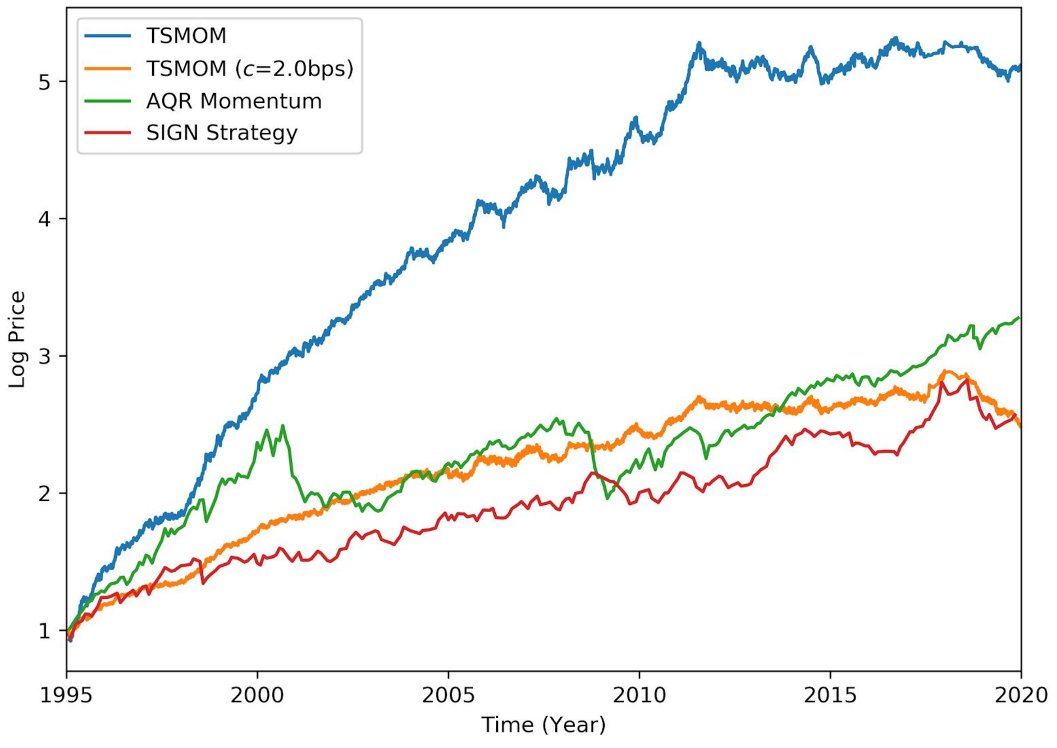

Figure 3: Performance of 4 momentum strategies - DMN-TSMOM, DMN-TSMOM including a transaction cost of 2 basis points, AQR cross-sectional momentum factor, the baseline SIGN-strategy. The y-axis is on log scale and all assets were normalized to the same origin and volatility.

Moving on to a quantitative evaluation, Table 4 shows relevant performance metrics for the four momentum strategies. For comparison, all strategies were normalized for a target vol of 15%. DTM-TSMOM has the best risk-to-reward as indicated by the superior Sharpe and Santino ratios (1.1 and 1.2 respectively). It also has the smallest negative skew out of all the strategies. Considering its simplicity, the SIGN-TSMOM does very well on maximum drawdown, which over 25 years of observation was 24.7%. Both AQR momentum as well as SIGN-TSMOM do very well on the percentage of positive returns across trading days, locking in above 60% on the backtest.

Strategy	TSMOM	TSMOM (2 bps)	Momentum	SIGN
E[Return]	**0.193***	0.066	0.106	0.088
E[Excess Return]	**0.164***	0.055	0.105	0.087
Vol.	0.150	0.150	0.150	0.150
Skew	**-0.043***	-0.298	-0.790	-0.316
Exc. Kurtosis	1.560	1.609	2.650	**0.735***
Sharpe	**1.094***	0.365	0.697	0.579
Sortino	**1.202***	-0.140	0.244	0.081
Max. Drawdown	-0.277	-0.343	-0.405	**-0.247***
Max. Dd. Duration (Years)	4.996	**2.934***	6.500	4.125
Downside Deviation	**0.087***	0.095	0.097	0.092
Positive Returns (%)	53.671	53.966	**66.848***	60.440
Avg. Profit / Avg. Loss	1.065	0.930	1.000	**1.163***

Table 4: Performance metrics of the DMN-TSMOM, DMN-TSMOM including a transaction cost of 2 basis points, AQR cross-sectional momentum factor, the baseline SIGN-TSMOM strategy. Sortino ratio and Downside Deviation metrics were computed using an annual target rate of return of 8%. For comparison purposes, all standard deviations were normalized to 15%. The best performing metrics in each row are highlighted.

Next we revisited our Market, 60/40, and Fama French 5-Factor portfolios but added TSMOM as a 6th factor to try to increase the absolute return. TSMOM significantly outperformed the market (100% equity portfolio) and the 60/40 portfolio in all metrics of risk/return that we chose to track. The time series of the returns can be seen in Figure 4 below.

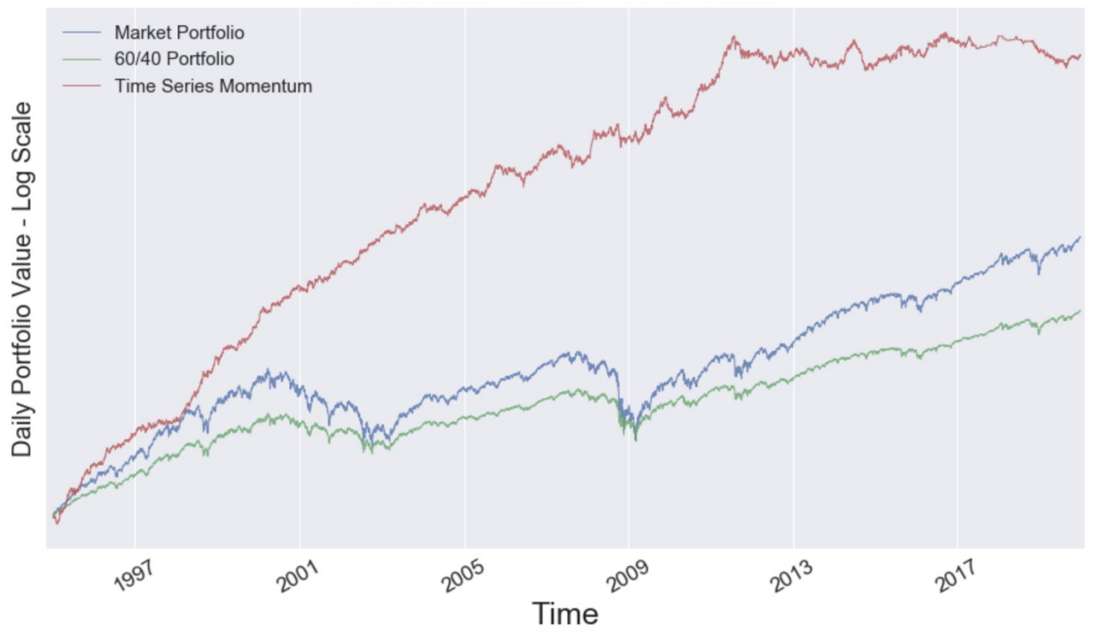

Figure 4: Shows the material outperformance of TSMOM over traditional benchmarks

The Fama-French portfolios all exhibit significantly superior returns and multiples when we added the TSMOM factor. Surprisingly, an equal weighting of each of the now 6 factors saw the best performance. The drastic difference with and without TSMOM can be seen in Figure 5 below.

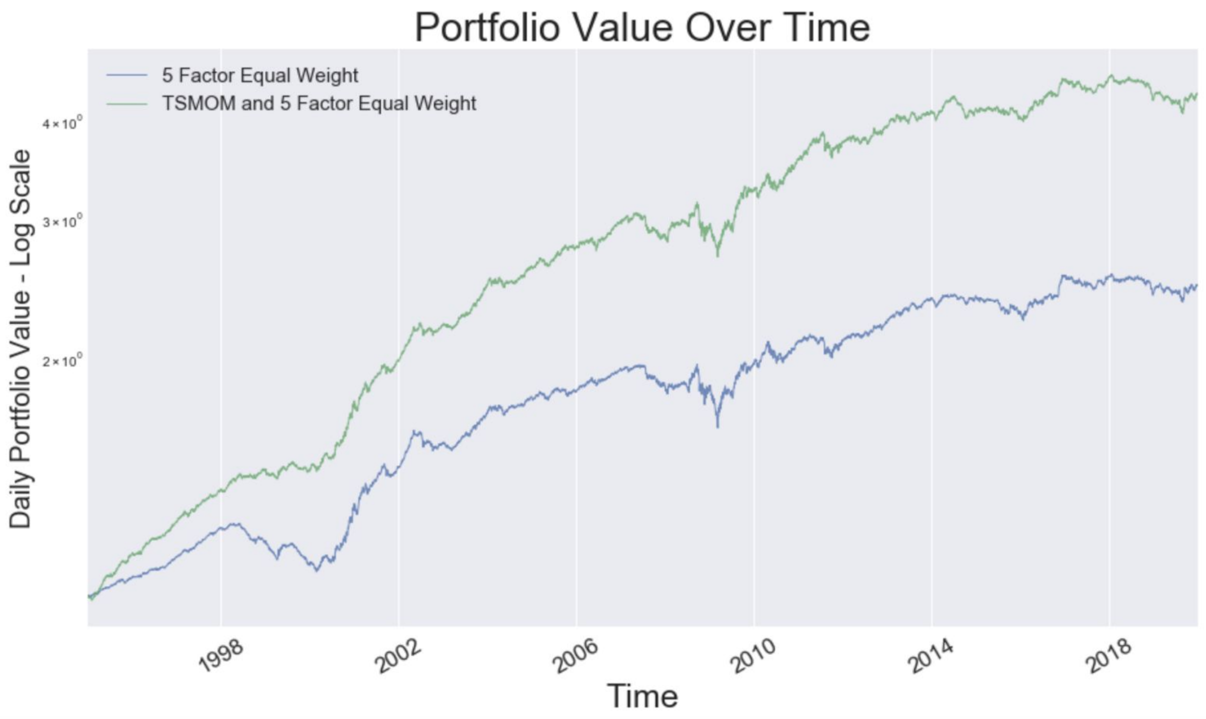

Figure 5: Shows the outperformance of Fama-French 5 factor equal wright portfolio when TSMOM is added.

An interesting observation in favor of TSMOM is that both the DTM and the SIGN strategies seem to have little-to-no sensitivity to the *dot-com* bust or the downturn of the *great financial crisis*. Baltas & Kosowski (2015) confirm this result for time series momentum. The strategy's robustness is most likely attributed to two factors. Firstly, the diversification effect of trading across asset classes (FX, fixed income, equity, commodities). Secondly, the ability to cheaply take short positions via futures contracts. The remarkable robustness of the factor to these two financial events increase its attractiveness of addition to a factor portfolio.

Given the outstanding risk adjusted returns, DTM-TSMOM appears a promising contender for inclusion in a quant factor portfolio. However, there are a couple of caveats to touch on. Firstly, the metrics don't hold once 2 basis points of transaction cost are applied to the strategy. However, we can't conclusively say if the transaction cost included DTM-TSMOM strategy is outperformed by AQR cross-sectional momentum or SIGN-TSMOM, as the latter don't include transaction cost. Further analysis would be required for such a statement. Secondly, a major concern with DTM-TSMOM is that its return profile seems to have been deteriorating over the past decade. It is possible that the trade has become crowded over time and the signal destroyed, which would make the strategy unprofitable going forward. This wouldn't be surprising as the inputs for this strategy are very simple and available to all investors. What we can say is that over the first 15 backtest years, DTM-TSMOM has had excellent performance.

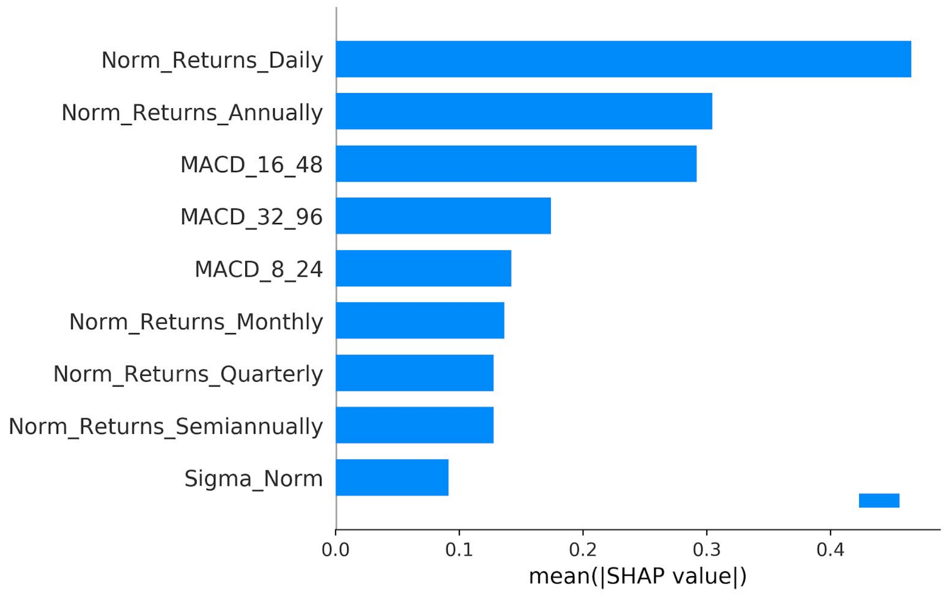

Figure 6: Shapley value based feature importance of the final Deep Momentum Network model. Impact is measured as the mean absolute Shapley value, which captures the average magnitude but not the direction of feature impact.

As a final point of discussion on DTM-TSMOM, some portfolio managers might be hesitant to oblige when a quantitative researcher presents them with a *black-box* model to trade on, and so models like DTM are easily dismissed. However, recent progress in the field of machine learning has particularly focused on explainability analysis. We here present two plots which we hope is a first step of turning the black-box trading strategy into white-box. Figure 6 shows the feature importance as determined by mean absolute *Shapley* value. In a nutshell, Shapley values are able to determine magnitude and direction of the predictive impact of a feature by comparing model output with and without the feature. We find that the most important feature is the past day normalized daily return. Lim et al. (2019) have made the same observation through feature elimination and running the backtest without the feature, which is an approximation to the Shapley method. Figure 7 shows the relationship between model output (position size) and normalized daily returns. The roughly positive linear relationship indicates that positive daily returns have a positive impact on the model output, promoting larger position size. The opposite is true for negative returns, where the model goes short. We therefore confirm that the model indeed performs a trend following strategy on average. Even though these strategies are termed time series momentum strategies, based on the loss function and strategy construction, the output of the model may as well have been mean reverting, if that indeed was the signal which reduces the loss function. Based on the observations from Figure 1, we have trained the model specifically for data within a 1-year horizon for which we know the momentum effect to be significant. If we were to incorporate the 2-3 year horizon data, we might add some sensitivity towards mean reversion into the model with respect to those features.

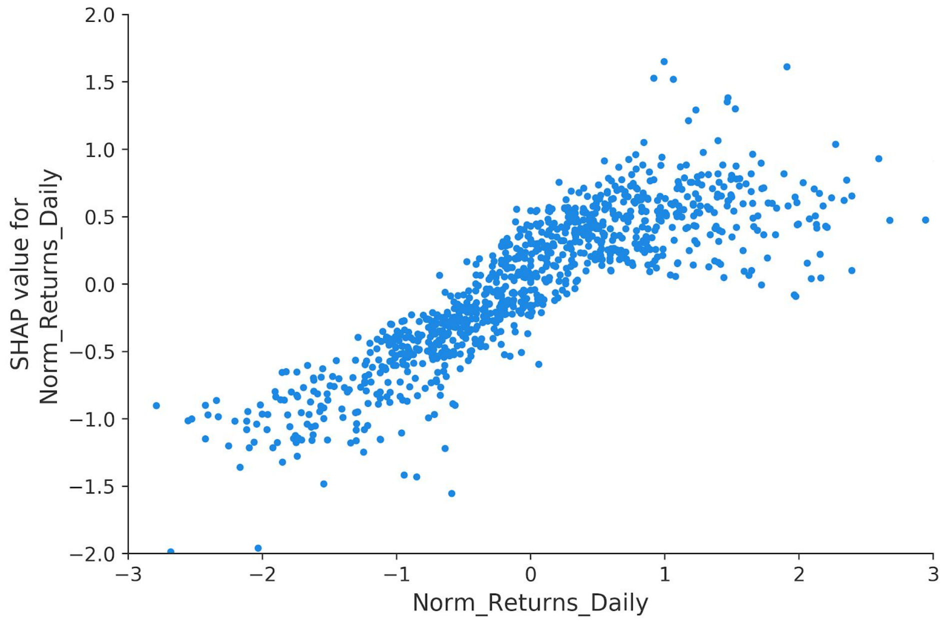

Figure 7: Impact of the daily normalized returns feature on predictions as measured by Shapley value. Strong positive returns indicate an increase in next-period model output, strong negative returns a decrease.

6 Conclusion

We've constructed a time series momentum factor using Deep Momentum Networks by combining nearly 50 futures contracts from several major asset classes into a strategy portfolio. The factor outperforms baseline strategies and cross-sectional momentum on risk and reward metrics such as Sharpe and Sortino ratio. It performs well in extreme periods such as the dot-com bubble and the great financial crisis, which makes it an attractive diversification to an equity factor portfolio. Adding time series momentum to our reference portfolios, each of the Fama-French factor-based portfolios were materially improved when we added the time series momentum factor.

However, since 2005 the performance of the time series momentum factor has been consistently deteriorating, and since 2012 has in fact been flat. The deterioration has manifested itself in the form of decreasing expected returns and increasing volatility. Accounting for transaction costs, the superior risk and return metrics of the strategy flaunder. This is a feature of the high turnover associated with implementing a momentum strategy.

While the factor has a proven track record and demonstrably adds value to the Fama-French factor portfolios, given the poor recent performance and high sensitivity to transaction costs, any investor considering incorporating this strategy should be aware that it comes with significant implementation risk.

References

Asness, C., Moskowitz, T.J., Pedersen, L.H.: 2010. Value and momentum everywhere. AQR Capital, University of Chicago, and National Bureau of Economic Research.

Baltas, AN. and Kosowski, R.: 2013, Improving Time-Series Momentum Strategies: The Role of Volatility Estimators and Trading Signals, *CME Working Paper*

Baltas, AN. and Kosowski, R.: 2015, Improving Time-Series Momentum Strategies: The Role of Volatility Estimators and Trading Signals (Revised), *CME Working Paper*

Cameron, A. C., Gelbach, J. B. and Miller, D. L.: 2011, Robust inference with multiway clustering, *Journal of Business and Economic Statistics 29(2), 238–249*

Gupta, T. and Kelly, B.: 2018, Factor Momentum Everywhere, *Yale ICF Working Paper No.2018-23*

Lim, B., Zohren, S. and Roberts, S.: 2019, Enhancing Time-Series Momentum Strategies Using Deep Neural Networks, *The Journal of Financial Data Science Fall 2019, 1 (4), 19-38*

Moskowitz, T., Ooi, Y. H. and Pedersen, L. H.: 2012, Time series momentum, *Journal of Financial Economics 104(2), 228 – 250*

Thompson, S. B.: 2011, Simple formulas for standard errors that cluster by both firm and time, *Journal of Financial Economics 99(1), 1–10*

Appendix A: DMN Feature Space

We use a mix of lagged returns, volatility estimates and technical momentum indicators as our deep momentum network feature space:

- Volatility normalized past returns:
    - Daily
    - Monthly
    - Quarterly
    - Semiannually
    - Annually
- Volatility normalized moving average convergence divergence (MACD) Indicators:
    - Short window: 8, 16, 32 days
    - Long Window: 24, 48, 96 days, respectively
- Normalized volatility estimator:
    - Exponentially weighted moving standard deviation with span factor of 3-months, normalized by the rolling 181 day average realized volatility for the asset.

Even though most features are already to some extent normalized, for convergence purposes, we perform additional standardization on the data set by transforming each feature to have normal mean and variance within the training set.

Appendix B: DMN-TSMOM Transaction Cost Impact

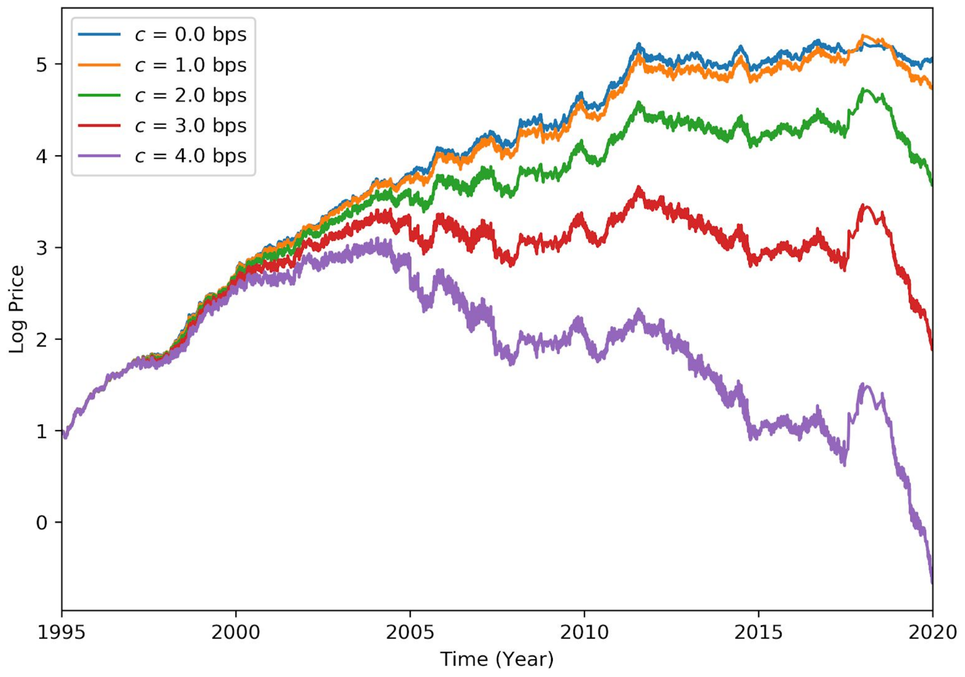

Figure 9: Performance of the DMN-TSMOM strategy with different levels of transaction cost. The y-axis is on log scale and the series have been normalized to start at the same origin.