

# Trading Signals In VIX Futures\*

M. Avellaneda<sup>†</sup>, T. N. Li<sup>‡</sup>, A. Papanicolaou<sup>§</sup>, G. Wáng<sup>¶</sup>

March 9, 2021

## Abstract

We propose a new approach for trading VIX futures. We assume that the term structure of VIX futures follows a Markov model. The trading strategy selects a multi-tenor position by maximizing the expected utility for a day-ahead horizon given the current shape and level of the VIX futures term structure. Computationally, we model the functional dependence between the VIX futures curves, the VIX futures positions, and the expected utility as a deep neural network with five hidden layers. Out-of-sample backtests of the VIX futures trading strategy suggest that this approach gives rise to reasonable portfolio performance, and to positions in which the investor can be either long or short VIX futures contracts depending on the market environment.

**Keywords:** VIX Futures, Trading Signals, Contango, Deep Learning, Feedforward Neural Networks, Cross Validation.

**AMS Subject Codes:** 62P05, 68T05, 91B28.

## 1 Introduction

The shape of the VIX futures curves is informative if it shows a shape that is likely to persist for only a short period of time. In this situation, there may be a simple VIX futures trade that will produce profits when the curve reverts to a more typical shape. For example, if the curve has a hump then there may be a long-short VIX futures position, or a calendar spread, with zero entry cost, which will pay a positive amount when the curve reverts to contango. Ideally, such a reversion will happen quickly so that the trade generates a profit with near certainty. In practice there is some risk because most trades involve non-zero probability of losses. Nevertheless, over long-term horizons with multiple trading opportunities, losses can be diminished if trading strategies are constructed to optimize the expected value of a suitable utility function. VIX futures are a good choice for such trading strategies because their curves have a propensity to quickly revert to contango, which allows for fast turnaround before the next trading opportunity.<sup>1</sup>

We use a stationary VIX futures curves model, as done in Avellaneda and Papanicolaou (2019), to generate day-ahead scenarios of VIX futures. Let  $U(\cdot)$  denote a chosen utility function. A trading signal is the optimal action maximizing the expected utility function under the probability distribution of the model,

$$\mathbf{a}(\mathbf{x}) = \arg \max_{\mathbf{a} \in \mathcal{A}} \mathbb{E}[U(\mathbf{R}_{t+1}(\mathbf{a})) | \mathbf{X}_t = \mathbf{x}], \quad (1.1)$$

---

\*The authors would like to thank Brian Healy and Xunyang Wu for their feedback and support.

<sup>†</sup>Department of Mathematics, New York University. 251 Mercer Street, New York, NY, 10012. [avellane@cims.nyu.edu](mailto:avellane@cims.nyu.edu)

<sup>‡</sup>Department of Mathematics, New York University. 251 Mercer Street, New York, NY, 10012. [thomli@cims.nyu.edu](mailto:thomli@cims.nyu.edu)

<sup>§</sup>Department of Mathematics, North Carolina State University. Campus Box 8205, Raleigh, NC 27695. [apapani@ncsu.edu](mailto:apapani@ncsu.edu). The author is partially supported by NSF grant DMS-1907518.

<sup>¶</sup>Department of Mathematics, Columbia University. 2990 Broadway New York, NY 10027. [gw2376@columbia.edu](mailto:gw2376@columbia.edu)

<sup>1</sup>Here, “quickly” means relative to other curves such as crude oil or treasuries.

where  $t$  denotes time,  $\mathbb{E}$  denotes expected value, and where

$$\begin{aligned}\mathbf{X}_t &= \text{VIX futures curves at time } t, \text{ vector valued,} \\ \mathcal{A} &= \text{a set of possible trades/actions } \mathbf{a}, \text{ vector valued,} \\ \mathbf{R}_{t+1}(\mathbf{a}) &= \text{change in position from time } t \text{ to } t+1 \text{ if action } \mathbf{a} \in \mathcal{A} \text{ is taken.}\end{aligned}$$

The action space  $\mathcal{A}$  consists of long or short positions in VIX futures, and  $\mathbf{R}_{t+1}(\mathbf{a})$  is a function of the action  $\mathbf{a}$  and the transition occurring in the VIX futures curves,

$$(\mathbf{X}_t, \mathbf{X}_{t+1}, \mathbf{a}) \mapsto \mathbf{R}_{t+1}(\mathbf{a}).$$

We take  $\mathcal{A}$  to be a finite set of trades that are predetermined, and we assume that the transition distribution for  $\mathbf{X}_t$  is also given. We estimate the expected value in equation (1.1) using a deep neural network, see Goodfellow et al. (2016). Historical VIX futures data are applied to estimate the parameters for the model of  $\mathbf{X}_t$ , and then the neural network is trained using simulated data generated by this estimated model. In our model, the most likely curve is a contango, and all other shapes of curves will revert toward this most likely state. To illustrate, Figure 1.1 shows a contango and a backwardation curve of VIX futures. We construct a trading signal by solving the optimization problem (1.1) with a set of trading actions  $\mathcal{A}$  consisting of various allocations in one month and five-month rolling VIX futures positions. For most contango curves, the trading signal's suggested action is to long the one-month and to short the five-month rolling positions. In backwardation, the suggested trade is to short the one month and 2× long the five month. In backtesting this trading signal we find that if transaction costs are not too high, then for a trading period of 200 days and more, there can be profits of double-digit percentage and Sharpe ratios significantly higher than one.

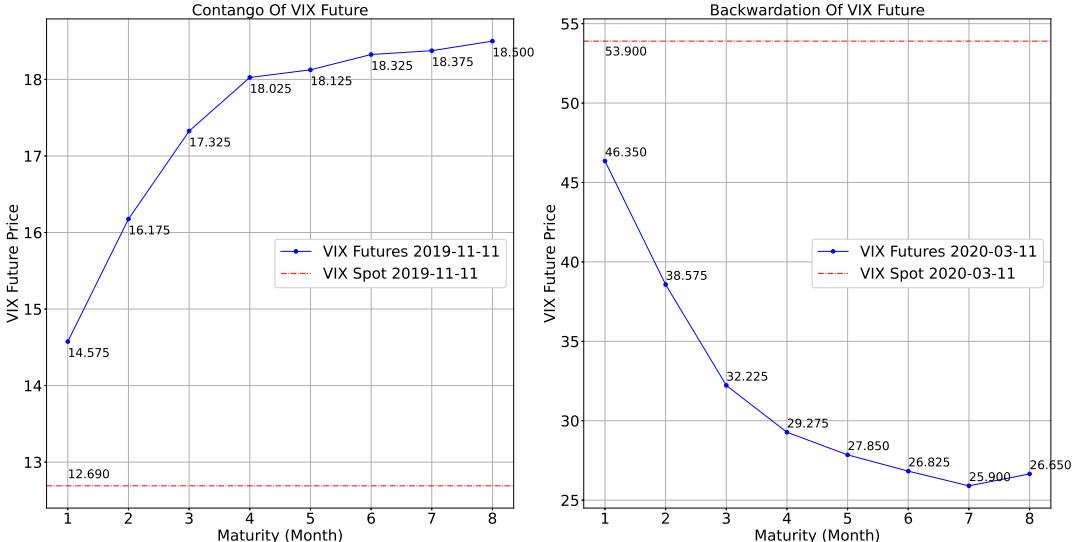


Figure 1.1: The VIX futures' contango curve seen on 2019-11-11 (left) and the backwardation seen on 2020-03-11 (right). A trading signal is constructed based on the value and shape of this curve.

## 1.1 Background Literature

The VIX has been the “fear gauge” for the financial markets of the United States since 1993, see Whaley (2000) and Whaley (2009). Since 2004, the market for VIX futures has made it possible to gain exposure to VIX, and the creation of exchange-traded notes (ETNs) has made it possible to

gain exposure with greater ease, see Alexander et al. (2015). The significance of mean reversion and contango in the VIX futures curves and ETNs is analyzed in Avellaneda and Papanicolaou (2019), and mean reversion is also the key assumption in the class of stochastic volatility models driven by stationary factor processes, see for instance Fouque et al. (2000). Historically, volatility models in finance have relied on the Markov property, but recently there has been a trend toward VIX pricing driven by fractional Brownian motion, which is the research subject of Bayer et al. (2016). The fractional approach is compatible with Markovian optimizations if the state is taken to be the entire VIX futures curves, with the reason being that, the appropriate Markovianization of the fractional-curve model is to consider the infinite-dimensional futures curve in its entirety as the Markovian state process, see Euch and Rosenbaum (2018).

Foundational concepts in machine learning such as convergence and deep learning extensions can be found in Mohri et al. (2018) and Sutton and Barto (2018). The implementation of high-dimensional learning is been made possible by recent developments in neural network software such as Tensor Flow and Pytorch. Example of note include the deep-Q neural network (DQN) algorithm, see Mnih et al. (2015) and Fan et al. (2020), and for applications to finance see Aldridge and Avellaneda (2020), Sirignano and Spiliopoulos (2017), and Casgrain et al. (2019). Studies on high-dimensional deep learning have highlighted the improvement in out-of-sample prediction when large neural networks are utilized, see Zhang et al. (2017), Belkin et al. (2018), and Hastie et al. (2019). Evaluation of out-of-sample performance is often done using cross-validation methods, but special care needs to be taken when applying these methods to financial data, see Arlot and Celisse (2010) and Arnott et al. (2019). In particular, in finance we often work with times series data with significant auto-correlations, yet cross-validation methods are still applicable so long as the times series are assumed to satisfy some basic assumptions such as zero auto-correlations in the noise process, see Burman and Nolan (1992), Bergmeir and Benítez (2012), and Bergmeir et al. (2018).

## 1.2 Main Results and Structure of the Paper

The focus of this paper is on a new method for trading VIX futures, wherein trading signals are the optimal action function given by (1.1). We implement this new approach on a variety of utility functions and utilize deep neural networks to estimate the objective in (1.1). We conduct cross-validation studies using a  $k$ -fold procedure. We use historical VIX futures data consisting of end-of-day VIX futures curves from January 2008 to February 2021. We find that portfolios constructed with deep neural networks have the potential to produce reasonable profits and Sharpe ratios in out-of-sample tests. These findings are an indication that VIX futures curves contain useful predictive information for trading, and that deep neural networks are able to filter and apply the relevant information from the curves.

The paper is organized as follows: Section 2 introduces the model, explains how parameters are estimated and describes the VIX futures positions that we optimize over; Section 3 presents cross-validation studies of the neural network method on historical VIX futures data – both with and without transaction costs; Section 4 concludes; Appendix A shows real-time backtest that we conducted with weekly re-training of the neural network from December 28th, 2020 through February 19th, 2021; Appendix A also provides a detailed account of how the output of the neural network maps to positions in VIX futures that can be directly executed by a trader; Appendix B provides portfolio metrics for various benchmarks for comparison.

## 2 A Model for Trading VIX Futures

Let  $d$  be an integer such that  $d + 1$  is the number of VIX futures contracts<sup>2</sup>, and let  $T_1 < T_2 < \dots < T_{d+1}$  denote the expiration dates. Let us denote

$$F_t^i := \text{VIX future expiring at time } T_i, \quad (2.1)$$

where  $t = 0, 1, 2, \dots, T_i$  is the current date. A term-structure of constant-maturity VIX futures (CMFs), each with horizon  $\theta_i$ -many months, for  $i = 1, 2, 3, \dots, d$ , are constructed as a linear interpolation of the VIX futures,

$$V_t^i := \omega_t^i F_t^i + (1 - \omega_t^i) F_t^{i+1}, \quad (2.2)$$

where  $t \leq T_i \leq t + \theta_i \leq T_{i+1}$ , and where  $\omega_t^i = \frac{T_{i+1} - t - \theta_i}{T_{i+1} - T_i}$ ; note now that  $V_t^i$  is defined for all  $t$ . By convention we denote  $V_t^0 = \text{VIX}_t$ . CMFs are preferable for statistical estimation because they do not have non-stationary effects caused by contract expiry.

### 2.1 Portfolios of Rolling VIX Futures

A portfolio of rolling VIX futures maintains the CMF weights of equation (2.2) for fixed maturity  $\theta_i$ . For each  $i$  we let  $I^i$  denote the value of the rolling VIX futures portfolio with horizon  $\theta_i$ , for which returns are given by

$$\frac{\Delta I_t^i}{I_t^i} := \frac{\omega_t^i \Delta F_t^i + (1 - \omega_t^i) \Delta F_t^{i+1}}{\omega_t^i F_t^i + (1 - \omega_t^i) F_t^{i+1}} + r \Delta t, \quad (2.3)$$

where  $\Delta I_t^i = I_{t+1}^i - I_t^i$ ,  $\Delta F_t^i = F_{t+1}^i - F_t^i$ ,  $r \geq 0$  is the interest rate, and  $\Delta t = \frac{1}{252}$ . Simple calculation leads to an equivalent expression to equation (2.3) in terms of the CMFs,

$$\frac{\Delta I_t^i}{I_t^i} = \left( r + \dot{\omega}_t^i \frac{F_{t+1}^{i+1} - F_{t+1}^i}{V_t^i} \right) \Delta t + \frac{\Delta V_t^i}{V_t^i}, \quad (2.4)$$

where  $\dot{\omega}_t^i = \frac{\omega_{t+1}^i - \omega_t^i}{\Delta t} < 0$  for all  $t < T_i$ . The drift term in equation (2.4) contains the quantity referred to as the roll yield,

$$\text{Roll}_{t+1}^i := \dot{\omega}_t^i \frac{F_{t+1}^{i+1} - F_{t+1}^i}{V_t^i},$$

which we use to re-write equation (2.4) as follows,

$$\frac{\Delta I_t^i}{I_t^i} = (r + \text{Roll}_{t+1}^i) \Delta t + \frac{\Delta V_t^i}{V_t^i}. \quad (2.5)$$

From equation (2.5) we see that if  $V_t^i$  is a stationary process then the return rate of the  $i^{th}$  rolling VIX futures portfolio has a most likely value equal to the risk-free rate plus the mode of  $\text{Roll}_{t+1}^i$ . As shown in Avellaneda and Papanicolaou (2019), the most likely VIX futures curves are contango and the most likely roll yields are negative, which explains why the value of the rolling VIX futures portfolios decay so rapidly.

The two main quantities we consider are the CMFs  $(V_t^i)_{i=0, 1, 2, \dots, d}$  and the roll yields  $(\text{Roll}_t^i)_{i=1, 2, 3, \dots, d}$ . As seen from equation (2.5), these quantities can be used to make short-term prediction of the rolling VIX futures portfolios. If the predictability is strong enough, then the magnitude of the roll yield and the anticipated direction of mean reversion could be the basis for a trading strategy that performs

---

<sup>2</sup>The VIX futures term structure is a collection of futures contracts with nine monthly maturities (and six weekly contracts that are not very liquid).

well over the long term. One trading idea is to utilize the Engle-Granger test to find co-integrated pairs among rolling VIX future portfolios, see Engle and Granger (1987). For the one-month rolling VIX futures portfolio ( $\theta = \text{one month}$ ) and the five-months rolling VIX futures portfolio ( $\theta = \text{five months}$ ), a simple linear regression of one portfolio on the other suggests that we should short the one-month portfolio and long  $0.9 \times$  five-months portfolio. However, this is not a good pair to trade because their residual is not stationary; for daily data between 2008 and 2020 the value of this position does not reject a unit root hypothesis. In addition, historical backtesting shows that this trade has large drawdowns and negative returns at the most inopportune times. Another possibility is to match volatility levels between the one-month and five-month rolling VIX futures portfolios, which suggests a position  $1 \times$  short and  $2 \times$  long, respectively. This was a popular trade during the decade of 2010, but also had large drawdowns. The conclusion is that allocations in these rolling VIX futures portfolios are useful but there needs to be a rule for deciding when to open and close the trade.

**Remark 2.1** (Exchange Traded Notes). *Rolling VIX futures portfolios represent the underlying redemption value for several VIX futures ETNs. Such notes are among the more liquid instruments for gaining exposure to VIX, see Alexander et al. (2015). Some of the more liquid ETNs include the iPath VXX (one-month long), the iPath VXZ (five-months long), the VelocityShares TVIX (one-month,  $2 \times$  long), and the iPath XIV (one-month short). Trading in these notes can be replicated with trades in the rolling VIX futures portfolios. However, in practice, replication is not entirely accurate. Firstly, the issuer of a note may have call-back features embedded that will terminate the note at any time. Secondly, the rolling VIX futures portfolio is technically just the redemption value and the notes are free to trade at market value, which means that there may be a slight discrepancy between an ETN's return and its respective rolling futures formula.*

## 2.2 Vector Autoregressive Model

For the  $t^{\text{th}}$  day of a given time period, returns on VIX futures are computable from the following state vector,

$$\mathbf{X}_t = \left[ \log \text{VIX}_t, \log V_t^1, \log V_t^2, \dots, \log V_t^d, \text{Roll}_t^1, \text{Roll}_t^2, \dots, \text{Roll}_t^d \right]^{\top},$$

where all entries of this vector are directly computable from  $(\text{VIX}_s, F_s^1, F_s^2, \dots, F_s^{d+1})_{s \leq t}$ . Given data at times  $t = 1, 2, \dots, T$ , let  $\mathbf{X}^*$  denote the mode,

$$\mathbf{X}^* = \underset{t \leq T}{\text{mode}}(\mathbf{X}_t),$$

that is,  $\mathbf{X}^*$  is the most likely curve, which is illustrated in Figure 2.1. The figure displays the mean given by  $\frac{1}{T} \sum_{t=1}^T \mathbf{X}_t$ , and the mode of the state. Statistical analysis in Avellaneda and Papanicolaou (2019) shows that  $\mathbf{X}_t$  is a stationary stochastic process whose historical time series exhibits a tendency to mean revert towards a contango curve. In its most likely state, the VIX future is around 12%-14%, the long-term VIX future is around 17%-20%, and all in-between CMFs lie on an upward sloping curve.

We take the state vector  $\mathbf{X}_t$  for  $t = 1, 2, \dots, T$ , we center it around the mode, and then place it in a larger matrix

$$\boldsymbol{\psi} = [\mathbf{X}_1 - \mathbf{X}^*, \mathbf{X}_2 - \mathbf{X}^*, \dots, \mathbf{X}_T - \mathbf{X}^*].$$

Note that we are centering around the mode rather than the mean, which we do for robustness because  $\mathbf{X}_t$  has heavy tails. The vector autoregressive (AR) model is the following,

$$\boldsymbol{\psi}_{t+1} = \boldsymbol{\mu} + \mathbf{A}\boldsymbol{\psi}_t + \mathbf{Z}_{t+1}, \quad (2.6)$$

where  $\mathbf{Z}_t$  is an independent and identically distributed Gaussian random vector with mean zero and covariance  $\boldsymbol{\Sigma}$ . The least-squares estimator of  $\mathbf{A}$  is given by

$$\widehat{\mathbf{A}} = \left[ \sum_{t=1}^{T-1} (\boldsymbol{\psi}_{t+1} - \overline{\boldsymbol{\psi}}) (\boldsymbol{\psi}_t - \overline{\boldsymbol{\psi}})^{\top} \right] \left[ \sum_{t=1}^{T-1} (\boldsymbol{\psi}_t - \overline{\boldsymbol{\psi}}) (\boldsymbol{\psi}_t - \overline{\boldsymbol{\psi}})^{\top} \right]^{-1}, \quad (2.7)$$

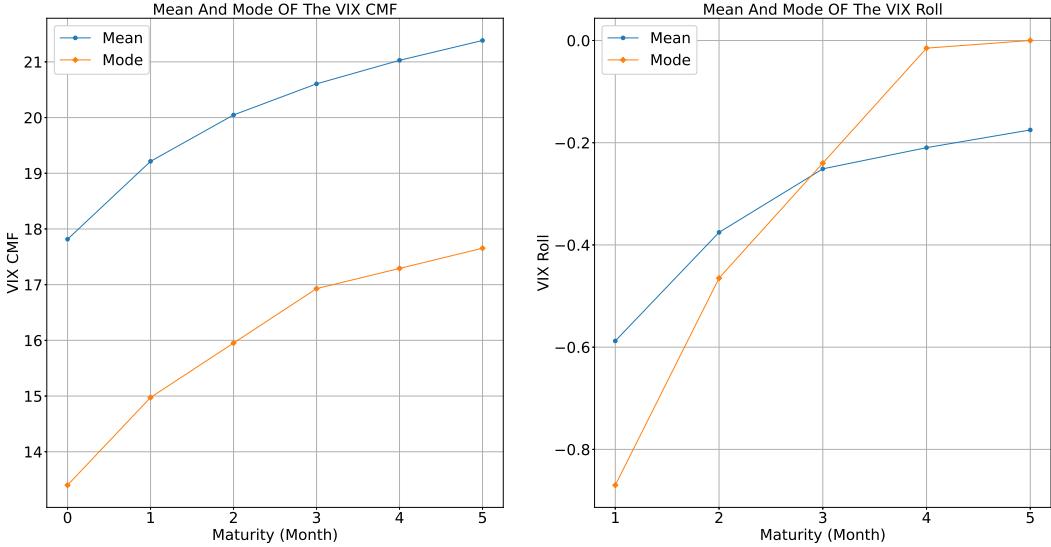


Figure 2.1: The mean and modal curves of VIX CMFs (left) and the mean and modal curves of the roll yields (right). The VIX futures curve is usually in contango, with the possibility of a volatility spike causing an upward skew in the futures' distribution. Therefore, the mean CMF curve is above the modal curve, and a similar relationship appears in the mean and modal curves of the negative roll yields.

$$\hat{\boldsymbol{\mu}} = (\mathbf{I} - \hat{\mathbf{A}}) \bar{\boldsymbol{\psi}}, \quad (2.8)$$

where  $\bar{\boldsymbol{\psi}} = \frac{1}{T} \sum_{t=1}^T \boldsymbol{\psi}_t$ . The matrix  $\Sigma$  can be estimated as

$$\hat{\Sigma} = \frac{1}{T} \sum_{t=1}^T \hat{\mathbf{Z}}_t \hat{\mathbf{Z}}_t^\top, \quad (2.9)$$

where  $\hat{\mathbf{Z}}_{t+1} = \boldsymbol{\psi}_{t+1} - \hat{\boldsymbol{\mu}} - \hat{\mathbf{A}}\boldsymbol{\psi}_t$ .

We can write the returns on the rolling VIX futures portfolios from equation (2.5) as

$$\frac{\Delta I_t^i}{I_t^i} = \left( r + \mathbf{X}_{t+1}^{d+i} \right) \Delta t + \frac{\exp(\mathbf{X}_{t+1}^i) - \exp(\mathbf{X}_t^i)}{\exp(\mathbf{X}_t^i)}, \quad \text{for } 1 \leq i \leq d, \quad (2.10)$$

which will be useful in the sequel where we draw samples from a distribution for  $\mathbf{X}_t$  and use to simulate trading returns. We use the vector AR model that is described by equation (2.6) to simulate  $\mathbf{X}_t$ , which we insert into equation (2.10) for computing returns on rolling VIX futures portfolios. Figure 2.2 shows simulations of the one-month and five-month rolling VIX futures portfolios, with each simulation including its respective historical portfolio value.

### 2.3 Trading-Signal Construction

We consider the following quantity

$$\mathbf{R}_{t+1}(\mathbf{a}) := \sum_i a^i \left( \frac{\Delta I_t^i}{I_t^i} - r \Delta t \right). \quad (2.11)$$

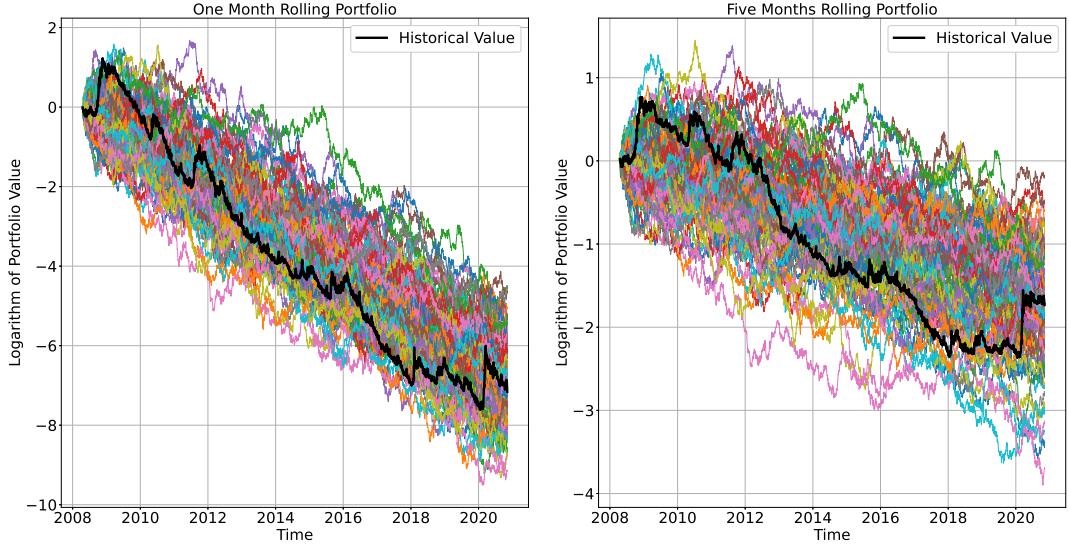


Figure 2.2: Simulations of the one-month rolling VIX futures portfolio and the five-month rolling VIX futures portfolio, generated from the vector AR model in equation (2.6). The dark line in each plot is the historical value of the respective portfolio. The declining value in these positions is studied in Avellaneda and Papanicolaou (2019).

This represents the profit or loss for a position in a portfolio of rolling VIX futures<sup>3</sup>. Let  $\mathcal{A}$  denote the space of admissible actions. An optimal action is determined by maximizing expected utility,

$$\max_{\mathbf{a} \in \mathcal{A}} \mathbb{E} \left[ U(\mathbf{R}_{t+1}(\mathbf{a})) \mid \mathbf{X}_t = \mathbf{x} \right], \quad (2.12)$$

where the action is decided by the trader at time  $t$  immediately before  $\mathbf{R}_{t+1}(\mathbf{a})$  is realized, and where  $U(\mathbf{R})$  is the utility function.

We denote by  $P_t$  the value of the trading-signal portfolio at time  $t$ , for which returns are computed as

$$\frac{\Delta P_t}{P_t} = \mathbf{R}_{t+1}(\mathbf{a}(\mathbf{X}_t)) + r\Delta t, \quad (2.13)$$

where  $\mathbf{a}(\mathbf{X}_t) = \arg \max_{\mathbf{a} \in \mathcal{A}} \mathbb{E} \left[ U(\mathbf{R}_{t+1}(\mathbf{a})) \mid \mathbf{X}_t \right]$ . In testing we use the time series of  $P_t$  to compute performance metrics, such as profit percentages and Sharpe ratios.

### 3 Computing the Trading Signals with Historical Data

We carried out the method described in Section 2 on historical VIX futures curves. The data we use were from April 14th of 2008 to November 6th of 2020 of one-month, two-months, three-months, four-months, five-months, and six-months VIX futures, in other words, in equation (2.1),  $i = 1, 2, \dots, 6$  and  $d = 5$ . The procedure for training and testing was straightforward: we divided the data into two blocks, with the first block designated for training, and the second block designated for testing. We took VIX futures curves from April 16th of 2008 to August 7th of 2019 for in-sample training, and then used the remaining curves from August 8th of 2019 to November 5th of 2020 for out-of-sample testing. The data that we used are downloadable from the VIX Central website.<sup>4</sup> Using these data, we constructed the time series of VIX CMFs and VIX rolls as given by the formulae of

<sup>3</sup>Of course, every portfolio of rolling VIX futures is equivalent to a portfolio of futures contract (see equation (2.3)).

<sup>4</sup>The VIX Central website: <http://vixcentral.com>

(2.2) and (2.4), respectively. We took the weights  $\omega^i$  appearing in (2.2) to be  $\omega^i \equiv \omega$  for all  $i$  such that there is 100% in the front-month contract as soon as the prior future matures, and then 0% in this front-month at the next maturity date. We analyzed the time series of portfolio value over this testing interval using the following performance metrics: annualized expected rate of return denoted by  $\mathbb{E}[\mathbf{R}_t(\mathbf{a}(\mathbf{X}_t))]$ , volatility denoted by  $\text{std}[\mathbf{R}_t(\mathbf{a}(\mathbf{X}_t))]$ , trading profit, Sharpe ratio, and maximum drawdown.

The out-of-sample test described in the previous paragraph was based on a single portfolio run, which means that good portfolio performance might be attributable to luck. Therefore, to make full usage of the data, we applied the method of the  $k$ -fold cross validation. We divided the data into  $k = 10$  folds, each with 316 or 317 days, and then used these folds to conduct ten separate in-sample and out-of-sample tests. Specifically, we trained on a configuration of nine folds, upon which we conducted an in-sample test, and then upon the remaining fold we conducted an out-of-sample test, see chapter four of Mohri et al. (2018) for details on  $k$ -fold cross validation. Table 3.1 gives the precise demarcation dates for the folds. When we pasted non-contiguous folds we excluded the so-called *pasting outlier* when estimating the vector AR model (2.6). For example, in order to out-of-sample test on fold #5 we needed to paste fold #4 to fold #6 for training, and in doing so we made sure to exclude the data point at the jump from fold #4 to #6.

This application of  $k$ -fold cross validation does not allow for truly out-of-sample testing because there are correlations between folds, see Arnott et al. (2019), but cross validation methods are still effective for auto-regressive models with uncorrelated noise, see Arlot and Celisse (2010), Bergmeir and Benítez (2012), Bergmeir et al. (2018), Burman and Nolan (1992), and Cerqueira et al. (2020). In the case of the VIX futures data, auto-correlations decay relatively quickly because of faster rates of reversion to the most likely curve, thereby reducing co-variation between folds.

Fold	Time Interval	Fold	Time Interval
0	2008-04-16 to 2009-07-17	5	2014-07-31 to 2015-10-29
1	2009-07-20 to 2010-10-19	6	2015-10-30 to 2017-02-01
2	2010-10-20 to 2012-01-23	7	2017-02-02 to 2018-05-04
3	2012-01-24 to 2013-04-29	8	2018-05-07 to 2019-08-07
4	2013-04-30 to 2014-07-30	9	2019-08-08 to 2020-11-05

Table 3.1: The start date and end date for each of the ten testing folds in the  $k$ -fold cross validation.

Our approach was to use the training data to estimate the parameters of the vector AR model (2.6) that was proposed in Section 2.2, and then to draw samples from the vector AR model for training of the neural network. The neural network is an approximation of the functional form of  $\mathbb{E}[U(\mathbf{R}_{t+1}(\mathbf{a}))|\mathbf{X}_t]$  (see Cybenko (1989) and Pinkus (1999)) for each action  $\mathbf{a}$  in the action space,

$$\mathcal{A} = \left\{ (0, 0), (-1, 1), (-1, 2), (1, -1), (1, -2) \right\}, \quad (3.1)$$

where the individual actions were

$$\begin{aligned} (0, 0) &= \text{no trade,} \\ (-1, 1) &= \text{short } I^1 \text{ and long } I^5, \\ (-1, 2) &= \text{short } I^1 \text{ and } 2 \times \text{long } I^5, \\ (1, -1) &= \text{long } I^1 \text{ and short } I^5, \\ (1, -2) &= \text{long } I^1 \text{ and } 2 \times \text{short } I^5, \end{aligned}$$

and where  $I^1$  and  $I^5$  denote the one month and five-month rolling VIX futures portfolios, respectively, as defined by (2.5) in Section 2.1.

### 3.1 Neural Network Approach

For general concave utility there is no explicit calculation for the expected utility  $\mathbb{E}[U(\mathbf{R}_{t+1}(\mathbf{a}))|\mathbf{X}_t]$ . Therefore, we utilize a neural network to find an approximating function. The neural-network architecture that we implement is a deep feed-forward neural network (DFN), as described in Goodfellow et al. (2016). For a discrete set of actions  $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_p\}$ , the universal approximation theorem, see Cybenko (1989) and Pinkus (1999), is a mathematical theorem to ensure that DFN is an effective way to estimate the nonlinear mapping

$$\mathbf{X}_t \mapsto [Q(\mathbf{X}_t, \mathbf{a}_1), Q(\mathbf{X}_t, \mathbf{a}_2), \dots, Q(\mathbf{X}_t, \mathbf{a}_p)]^\top,$$

where we denote  $Q(\mathbf{X}_t, \mathbf{a}_j) = \mathbb{E}[U(\mathbf{R}_{t+1}(\mathbf{a}_j))|\mathbf{X}_t = \mathbf{x}]$  is the state-action value function. Our approach is to sample  $\mathbf{X}_t$  from the vector AR model proposed in Section 2.2, and then use these samples to train the neural network, and then perform  $k$ -fold cross validation to test out-of-sample performance.

The DFN that we utilize has the specifications that are depicted in Figure 3.1. It has eleven neurons in the input layer, each of which represents an element  $\mathbf{X}_t^i$  for a given  $i$ . The number of neurons in the output layer is five, which represents the five actions in the action space  $\mathcal{A}$ . We set the number of the hidden layers to five, each of them containing  $J = 50 \times 11$  neurons. In training we ran the backpropagation for 15 epochs each having 160 steps, and we set the algorithm to withhold 20% for validation. We used a dense connective structure between layers, in other words, all layers had neurons fully connected with the neurons in the previous layer. The activation function  $f(x)$  was chosen to be the Parametric Rectified Linear Unit (PReLU) function,

$$f(x) = \begin{cases} x & \text{for } x \geq 0 \\ \alpha x & \text{for } x < 0, \end{cases} \quad (3.2)$$

where  $\alpha > 0$ , in our studies we set  $\alpha = 0.1$ . For all results that we present, we have taken the PReLU activation function for both the hidden layers and the output layer of the DFN. We did also repeat all tests using hyperbolic tangent activation function  $f(x) = \tanh(x)$  and linear activation function  $f(x) = wx + b$  for the output layer, but the results from PReLU were slightly better. Given the neurons, layers, and activation function, we assume the DFN is the underlying structure of  $Q : \mathbb{R}^{11} \rightarrow \mathbb{R}^5$  such that

$$\begin{pmatrix} Q(\mathbf{X}_t, \mathbf{a}_0) \\ Q(\mathbf{X}_t, \mathbf{a}_1) \\ Q(\mathbf{X}_t, \mathbf{a}_2) \\ Q(\mathbf{X}_t, \mathbf{a}_3) \\ Q(\mathbf{X}_t, \mathbf{a}_4) \end{pmatrix} = f\left(\mathbf{W}_6^\top f\left(\dots f\left(\mathbf{W}_2^\top f\left(\mathbf{W}_1^\top \mathbf{X}_t + \mathbf{b}_1\right) + \mathbf{b}_2\right) \dots\right) + \mathbf{b}_6\right), \quad (3.3)$$

where  $\mathbf{W}_\ell$ ,  $\ell = 1, 2, \dots, 6$  are the matrices of connection weights that connect the neurons on the  $(\ell - 1)^{th}$  layer to the  $\ell^{th}$  layer, and  $\mathbf{b}_\ell$ ,  $\ell = 1, 2, \dots, 6$  is a vector of some biasing values, here subscript 6 represents the terminal output layer.

We trained the DFN using samples that were drawn from the vector AR model of Section 2.2 with Gaussian noise. We drew independent and identically distributed samples  $\mathbf{X}_0^{(i)}$  for  $i = 1, 2, \dots, N$  from the stationary distribution of the vector AR model (2.6). We took  $N = 10^5$ . For each  $\mathbf{X}_0^{(i)}$ , we simulated a batch of one-step forward samples to approximate the conditional expected utility, which we label as  $\mathbf{R}_1^{(i, i')}(a)$  for  $i' = 1, 2, \dots, M$  for each  $a \in \mathcal{A}$ . We took  $M = 300$ . We then fitted the DFN to the sample averages by minimizing quadratic loss function with respect to the hyperparameters  $\mathbf{W}$  and  $\mathbf{b}$ ,

$$\min_{\mathbf{W}, \mathbf{b}} \sum_{a \in \mathcal{A}} \frac{1}{N} \sum_{i=1}^N \left( Q(\mathbf{X}_0^{(i)}, a) - \frac{1}{M} \sum_{i'=1}^M U\left(\mathbf{R}_1^{(i, i')}(a)\right) \right)^2. \quad (3.4)$$

After training, the optimally fitted neural network was utilized to compute the optimal trading actions, namely,  $\mathbf{a}(\mathbf{X}_t) = \arg \max_a Q(\mathbf{X}_t, \mathbf{a})$ .

## Neural Network Schematic Diagram

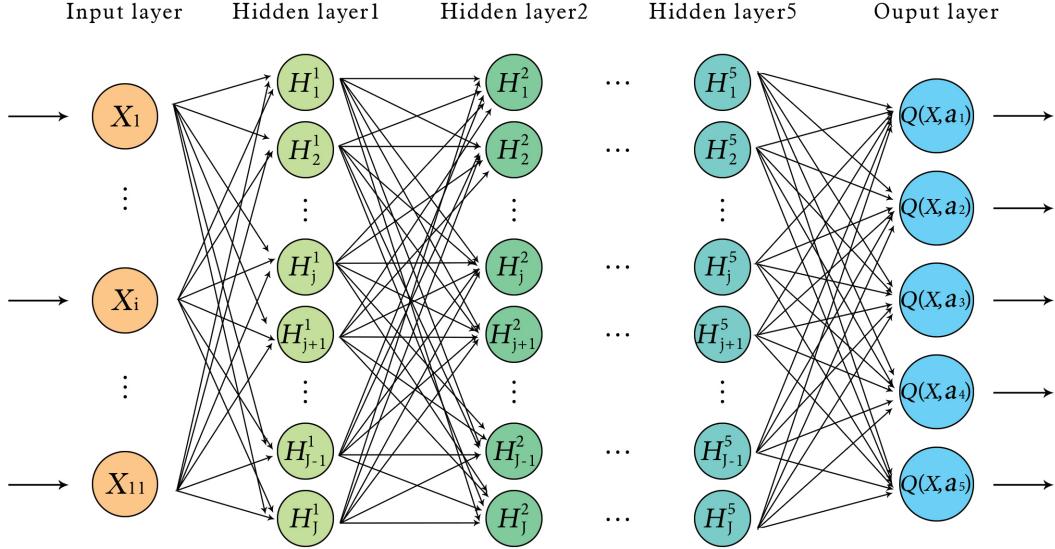


Figure 3.1: Schematic diagram of the deep neural network. In our studies we took  $J = 550$ , i.e., 550 neurons in each hidden layer.

### 3.2 Piece-Wise Linear and Exponential Utility Functions

We tested the trading signal constructed from a piece-wise linear utility function,

$$U(\mathbf{R}) = \max(\mathbf{R}, 0) + \gamma \min(\mathbf{R}, 0), \quad (3.5)$$

and exponential utility function,

$$U(\mathbf{R}) = -\frac{1}{\gamma} \exp(-\gamma \mathbf{R}), \quad (3.6)$$

where we took the risk aversion coefficient  $\gamma = 1.3$  for the piece-wise linear utilitiy function and  $\gamma = 3$  for the exponential utility function. We then fitted the DFN with respect to the piece-wise linear utility function (3.5) with the same quadratic loss given in (3.4), and fitted the DFN with respect to the exponential utility function (3.6) by minimizing the quadratic loss of the certainty equivalent,

$$\min_{\mathbf{W}, \mathbf{b}} \sum_{a \in \mathcal{A}} \frac{1}{N} \sum_{i=1}^N \left( Q(\mathbf{X}_0^{(i)}, \mathbf{a}) + U^{-1} \left( -\frac{1}{M} \sum_{i'=1}^M U(\mathbf{R}_1^{(i, i')}(\mathbf{a})) \right) \right)^2. \quad (3.7)$$

Figure 3.2 illustrates some trading-signal heat plots for the piece-wise linear utility function and the exponential utility function. The most obvious difference is that the piece-wise linear utility has states where the trading signal suggests to take position  $(0, 0)$ . Table 3.2 and Table 3.3 display the portfolio metrics for the  $k$ -fold cross validation of out-of-sample tests, and Figure 3.3 shows the time series of portfolio values, as given by (3.3). In these tables, strong portfolio performance can be concluded based on profit and Sharpe ratios, but it is important also to highlight the large drawdowns and the difficulty they would pose in practice.

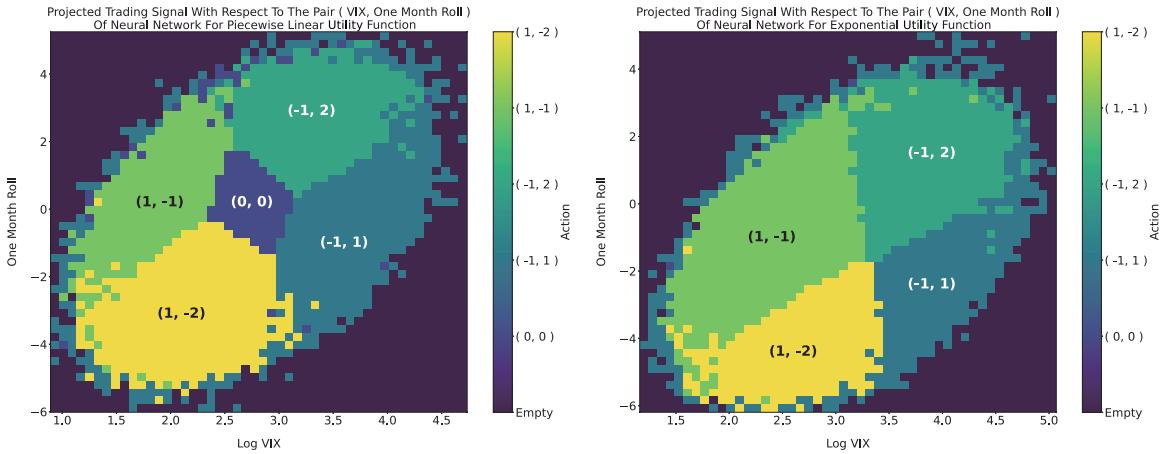


Figure 3.2: Heat plots showing the projection of the trading signals onto the two-dimensional space spanned by logarithm of VIX and the one-month roll, with the projected values being the most-common trading actions at these points. The left plot is the projection of the trading signal constructed with piece-wise linear utility function (3.5), the right is the projection of the trading signal constructed with exponential utility function (3.6). Both trading signals were constructed using the DFN approach of Section 3.1. The value “Empty” represents the values of logarithm of VIX and one-month roll that didn’t occur.

Statistics \ Fold	$\mathbb{E}[\mathbf{R}_t(\mathbf{a}(\mathbf{X}_t))]$	$\text{std}[\mathbf{R}_t(\mathbf{a}(\mathbf{X}_t))]$	Profit (%)	Sharpe Ratio	Maximum Drawdown
0	2.361	0.443	304.264	5.297	-0.196
1	1.195	0.368	145.924	3.215	-0.138
2	4.951	0.447	724.848	11.053	-0.117
3	2.835	0.410	384.387	6.878	-0.214
4	0.854	0.242	108.168	3.474	-0.128
5	1.129	0.361	137.044	3.093	-0.123
6	1.027	0.375	121.582	2.709	-0.156
7	1.415	0.754	130.578	1.862	-0.293
8	0.329	0.302	34.784	1.056	-0.180
9	3.284	0.491	429.191	6.661	-0.240

Table 3.2: Portfolio metrics for out-of-sample tests in  $k$ -fold cross validation on trading signal constructed with piece-wise linear utility function (3.5). These metrics are computed from the portfolio returns given by (2.13) with no transaction costs.

### 3.3 Transaction Costs

Finally, it remains to test if the trading signals from Section 3 can perform with transaction costs. Execution of this strategy is done using market orders, which means that market makers are providing liquidity, and therefore, we are crossing their bid-ask spread each time we complete a trade. The price data that we have used in our backtests are bid-ask midpoints. Thus, to simulate real-life trading of market orders, we should pay (at least)  $1/2$  the bid-ask spread each time we open or close a futures position.

VIX futures trade has a tick size of five cents,<sup>5</sup> which means that our backtests should always assume a bid-ask spread of at least five cents. In the simplest backtest, we hold the bid-ask spread

<sup>5</sup>Each VIX future traded on CBOE has a mutliplier of 1000, which means that the tick size is effectively \$50.

Statistics Fold	$\mathbb{E}[\mathbf{R}_t(\mathbf{a}(\mathbf{X}_t))]$	$\text{std}[\mathbf{R}_t(\mathbf{a}(\mathbf{X}_t))]$	Profit (%)	Sharpe Ratio	Maximum Drawdown
0	1.728	0.456	209.177	3.763	-0.239
1	1.610	0.412	198.810	3.880	-0.165
2	4.534	0.462	645.731	9.775	-0.175
3	3.138	0.463	418.006	6.751	-0.202
4	0.742	0.284	90.055	2.577	-0.190
5	0.683	0.388	74.484	1.731	-0.236
6	0.886	0.400	100.162	2.187	-0.189
7	1.239	0.785	103.799	1.564	-0.281
8	0.639	0.356	71.310	1.765	-0.171
9	2.518	0.564	294.004	4.443	-0.248

Table 3.3: Portfolio metrics for out-of-sample tests in  $k$ -fold cross validation with trading signal constructed with exponential utility function (3.6). These metrics are computed from the portfolio returns given by (2.13) with no transaction costs.

constant at five cents, which means we pay \$0.025 each time we open or close a VIX futures position. However, bid-ask spreads may widen, particularly when the VIX futures curves are in backwardation. With this widening in mind, a transaction-cost function for the  $i^{\text{th}}$  future is

$$\text{TC}_t^i = \frac{1}{2} \max(\varepsilon F_t^i, 0.05) , \quad (3.8)$$

where  $\varepsilon$  is a fixed basis points (bps) parameter, in other words,  $\varepsilon = 20\text{bps}$ ,  $30\text{bps}$ , or  $40\text{bps}$ . Using this notation for transaction costs, the returns on the value of the trading-signal portfolio are computed similarly to equation (2.13) except for an additional term for transaction costs,

$$\frac{\Delta P_t}{P_t} = \mathbf{R}_{t+1}(\mathbf{a}(\mathbf{X}_t)) + r\Delta t - \frac{1}{P_t} \sum_i \text{TC}_t^i |n_t^i - n_{t-1}^i| , \quad (3.9)$$

where  $n_t^i$  denotes the number of contracts in the  $i^{\text{th}}$  VIX future; computation of  $n_t^i$  is explained in Appendix A.

Table 3.5 and Table 3.4 display the metrics for portfolios computed with varying levels of transaction costs. Figure 3.4 illustrates the time series of portfolio value for trading signal constructed using piecewise linear utility function and using the portfolio value given by (3.5). In general, the portfolio can still perform well with transaction costs, but we do see a decline as we increase the basis points parameter in the transaction costs, in other words, as we increase  $\varepsilon$  in equation (3.8). Finally, as was the case in Table 3.2 and Table 3.3, drawdowns remain high when transaction costs are included.

## 4 Summary & Conclusion

We have proposed and analyzed a method for constructing VIX futures trading signals. The basis for the method is in the identification of certain trading opportunities by observing the shape of the VIX futures curves. The trading signal uses a neural network to determine the best action for day-ahead expectation of returns. We backtested this method and found it to perform well in out-of-sample tests, showing considerable profits and reasonable Sharpe ratios, but also showing levels of drawdown that would be difficult to manage in practice. When we included transaction costs we observed portfolio performance reduced to more pedestrian levels.

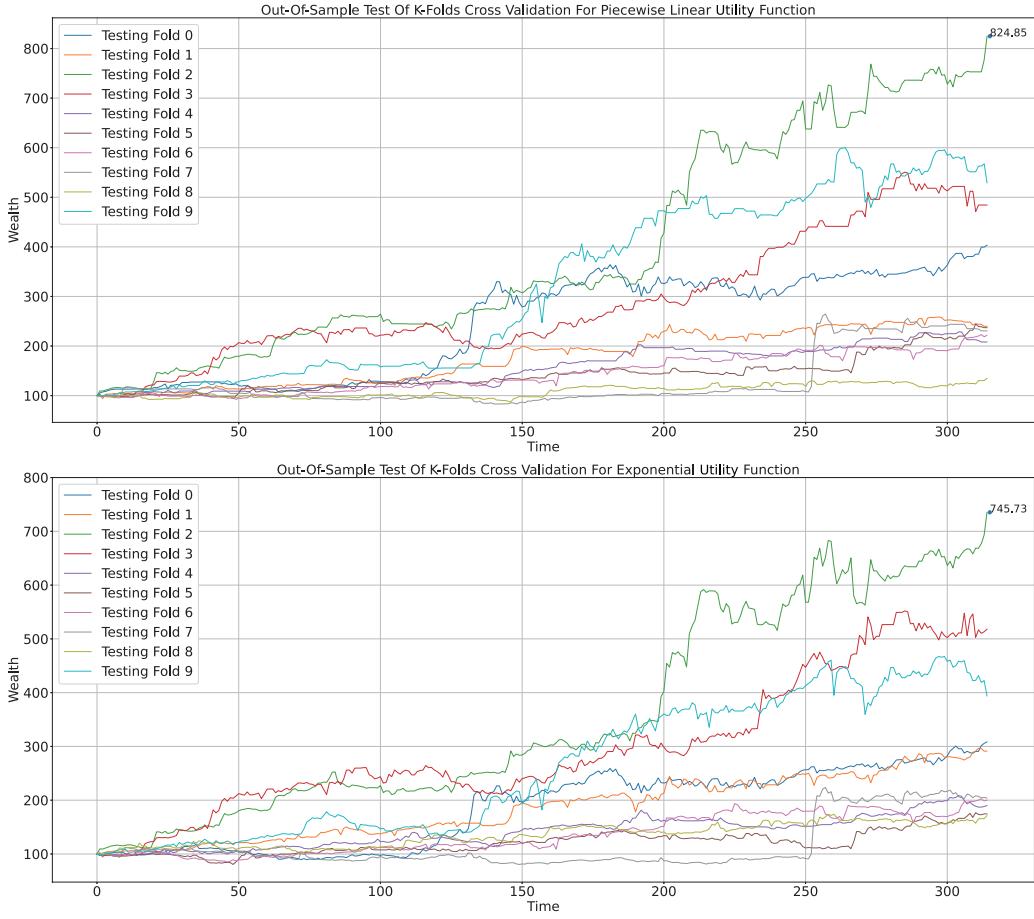


Figure 3.3: Time series of portfolio value for out-of-sample tests of  $k$ -fold cross validation on trading signals constructed with piece-wise linear utility function (top) and exponential utility function (bottom). The state-action value function  $Q(\mathbf{X}_t, \mathbf{a})$  is obtained by training DFN given by equation (3.3). The returns  $\mathbf{R}_t(\mathbf{a}(\mathbf{X}_t))$  were computed with the trading actions  $\mathbf{a}(\mathbf{X}_t) = \arg \max_{\mathbf{a} \in \mathcal{A}} Q(\mathbf{X}_t, \mathbf{a})$ , and the portfolio value is computed from (2.13) with no transaction costs.

## Appendix A Mapping Trading Signal to Futures Positions

The DFN model that is proposed in Section 3.1 provides optimal trading actions for the yields of rolling VIX futures portfolio  $I^i$  described in (2.11). For example, the trading signal occurring at the volatility spike on January 26th of 2021, the piece-wise linear utility function and the deep feed-forward neural network (3.3) produces an optimal action  $(-1, 1)$ . This represents a position with weight  $-1$  in  $I^1$  and weight of  $1$  in  $I^5$ . This appendix will give a translation of the neural network algorithm's output into the exact quantities that a real-life trader would use when setting up a position.

In expression (3.1) we defined the five actions considered in our analyses, which are  $\mathbf{a}_j$  for  $j = 0, 1, 2, 3, 4$ , with  $\mathbf{a}_0 = (0, 0)$ ,  $\mathbf{a}_1 = (-1, 1)$ ,  $\mathbf{a}_2 = (-1, 2)$ ,  $\mathbf{a}_3 = (1, -1)$ , and  $\mathbf{a}_4 = (1, -2)$ . Each action  $\mathbf{a}_i$  is a two-dimensional vector,

$$\mathbf{a}_j = (a_j^1, a_j^5)$$

where  $a_j^1$  is the portfolio weight in  $I^1$  and  $a_j^5$  is the weight in  $I^5$ . This action can be converted into the

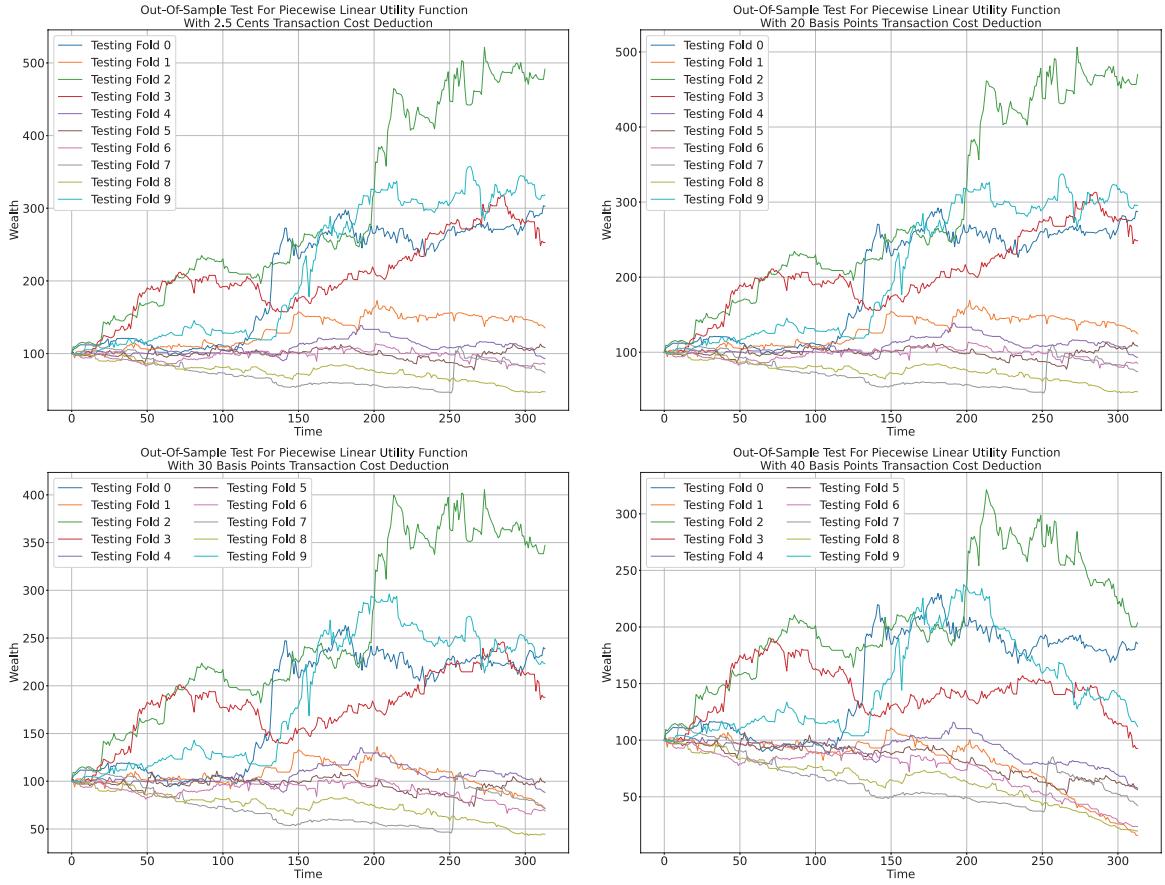


Figure 3.4: Time series of portfolio value, computed with transaction costs, for out-of-sample tests of  $k$ -fold cross validation on trading the signal constructed with piece-wise linear utility function (3.5). The state-action value function  $Q(\mathbf{X}_t, \mathbf{a})$  obtained by training DFN given by equation (3.3). The returns  $\mathbf{R}_t(\mathbf{a}(\mathbf{X}_t))$  were computed with the trading actions  $\mathbf{a}(\mathbf{X}_t) = \arg \max_{\mathbf{a} \in \mathcal{A}} Q(\mathbf{X}_t, \mathbf{a})$ , and the portfolio value with transaction cost deduction for the optimal action is  $P_t$  computed with equation (3.9).

actual number of contracts in VIX future  $F^i$  defined by equation (2.1). Letting  $n^i$  denote the number of contracts in  $F^i$  for  $i = 1, 2, 5, 6$ , the following are the conversions from a given  $a_j^i$  to the  $n^i$  for trading signals taking positions in  $I^1$  and  $I^5$ ,

$$n^1 = \frac{\omega a_j^1 P}{V^1}, \quad n^2 = \frac{(1 - \omega) a_j^1 P}{V^1}, \quad n^5 = \frac{\omega a_j^5 P}{V^5}, \quad n^6 = \frac{(1 - \omega) a_j^5 P}{V^5},$$

where  $P$  denotes the wealth of trading portfolio, given by either equation (2.13) or (3.9), and where we have taken the rolling weight  $\omega$  to be the same for all  $i$  as described in the beginning of Section 3. For example, if the optimal trading action is  $(-1, 1)$ , then we have  $a_j^1 = -1$  and  $a_j^5 = 1$ , and we apply accordingly the above equation for  $n^i$ . Table A.1 shows the positions in VIX futures  $F_t^i$  for a real-time run starting 2020-12-28 of trading signal constructed with piece-wise linear utility function. The portfolio value shown in the table includes 1/2 the bid-ask spread for each trade, and each position is rounded to the nearest whole number of contracts. Of note is the drop in  $P$  from January 26th to the 27th, which was the day of the GameStop trading freeze. The trade signal incurred a loss from the VIX spike caused by GameStop, but then recovered the losses in the following days.

Statistics \ Fold		0	1	2	3	4	5	6	7	8	9
$\varepsilon = 0$	Profit (%)	202.813	35.406	391.423	152.749	-6.926	8.136	-14.775	-26.051	-52.547	218.103
	Sharpe Ratio	3.756	0.956	6.643	3.099	-0.159	0.345	-0.175	-0.049	-1.403	3.768
	Maximum Drawdown	-0.213	-0.216	-0.163	-0.258	-0.329	-0.311	-0.306	-0.601	-0.542	-0.240
$\varepsilon = 20\text{bps}$	Profit (%)	187.321	24.631	370.000	148.617	-6.926	8.126	-14.780	-26.122	-52.550	195.975
	Sharpe Ratio	3.509	0.719	6.329	3.025	-0.159	0.344	-0.175	-0.050	-1.403	3.439
	Maximum Drawdown	-0.225	-0.262	-0.163	-0.262	-0.329	-0.311	-0.306	-0.601	-0.542	-0.240
$\varepsilon = 30\text{bps}$	Profit (%)	138.855	-28.295	247.105	87.970	-11.656	-1.441	-31.083	-28.788	-55.441	123.111
	Sharpe Ratio	2.709	-0.488	4.460	1.919	-0.324	0.119	-0.561	-0.087	-1.492	2.315
	Maximum Drawdown	-0.244	-0.473	-0.187	-0.307	-0.346	-0.333	-0.367	-0.601	-0.569	-0.279
$\varepsilon = 40\text{bps}$	Profit (%)	85.275	-84.254	103.636	-7.369	-43.848	-42.591	-76.575	-57.851	-80.190	11.942
	Sharpe Ratio	1.787	-1.677	2.096	0.046	-1.458	-0.898	-1.658	-0.509	-2.170	0.456
	Maximum Drawdown	-0.269	-0.857	-0.377	-0.510	-0.515	-0.507	-0.765	-0.681	-0.802	-0.528

Table 3.4: Portfolio metrics with transaction cost deduction for out-of-sample tests in  $k$ -fold cross validation with trading signal constructed with piece-wise linear utility function (3.5), and the state-action function  $Q(\mathbf{X}_t, \mathbf{a})$  obtained by training DFN given by equation (3.3). The transaction costs per contract are given by equation (3.8) and the portfolio returns are given by equation (3.9).

Statistics \ Fold		0	1	2	3	4	5	6	7	8	9
$\varepsilon = 0$	Profit (%)	143.296	64.787	23.421	139.082	-19.477	-44.923	-33.712	-91.306	-49.599	118.240
	Sharpe Ratio	2.738	1.477	5.505	2.649	-0.470	-0.871	-0.565	-0.967	-1.063	2.098
	Maximum Drawdown	-0.274	-0.184	-0.224	-0.263	-0.381	-0.563	-0.443	-0.920	-0.574	-0.328
$\varepsilon = 20\text{bps}$	Profit (%)	131.973	52.441	305.771	133.121	-19.477	-44.931	-33.716	-91.371	-49.600	103.910
	Sharpe Ratio	2.553	1.239	5.245	2.551	-0.470	-0.871	-0.565	-0.968	-1.063	1.892
	Maximum Drawdown	-0.274	-0.185	-0.224	-0.267	-0.381	-0.563	-0.443	-0.921	-0.574	-0.328
$\varepsilon = 30\text{bps}$	Profit (%)	99.883	-10.819	194.296	56.765	-24.365	-53.164	-50.059	-94.346	-53.466	55.666
	Sharpe Ratio	2.015	-0.019	3.545	1.259	-0.618	-1.075	-0.943	-0.989	-1.161	1.173
	Maximum Drawdown	-0.297	-0.348	-0.254	-0.311	-0.402	-0.596	-0.515	-0.948	-0.605	-0.333
$\varepsilon = 40\text{bps}$	Profit (%)	64.643	-78.125	61.369	-58.328	-55.578	-91.129	-95.012	-131.543	-87.536	-29.006
	Sharpe Ratio	1.405	-1.354	1.336	-0.902	-1.566	-1.809	-1.624	0.794	-1.887	-0.192
	Maximum Drawdown	-0.330	-0.808	-0.416	-0.772	-0.596	-0.912	-0.950	-1.289	-0.890	-0.534

Table 3.5: Portfolio metrics with transaction cost deduction for out-of-sample tests in  $k$ -fold cross validation with trading signal constructed with exponential utility function (3.6), and the state-action function  $Q(\mathbf{X}_t, \mathbf{a})$  obtained by training DFN given by equation (3.3). The transaction costs per contract are given by equation (3.8) and the portfolio returns are given by equation (3.9).

Date	$P$	$\omega$	$a^1$	$a^5$	$n^1$	$n^2$	$n^5$	$n^6$	$\sum_i n^i$
2020-12-28	100.00	0.65714	-1	2	-3	-1	5	3	4
2020-12-29	101.25	0.62857	-1	1	-3	-1	2	1	-1
2020-12-30	103.20	0.60000	0	0	0	0	0	0	0
2020-12-31	103.03	0.57143	0	0	0	0	0	0	0
2021-01-04	103.03	0.45714	-1	2	-2	-2	4	4	4
2021-01-05	102.23	0.42857	-1	2	-2	-2	3	4	3
2021-01-06	101.15	0.40000	-1	2	-2	-2	3	5	4
2021-01-07	104.18	0.37143	0	0	0	0	0	0	0
2021-01-08	103.88	0.34286	-1	1	-1	-3	1	3	0
2021-01-11	101.23	0.25714	-1	1	-1	-3	1	3	0
2021-01-12	103.22	0.22857	-1	1	-1	-3	1	3	0
2021-01-13	105.05	0.20000	-1	1	-1	-3	1	3	0
2021-01-14	105.45	0.17143	-1	1	-1	-4	1	3	-1
2021-01-15	103.63	0.14286	-1	1	-1	-3	1	3	0
2021-01-19	105.99	0.02857	-1	2	0	-4	0	8	4
2021-01-20	103.19	0.00000	0	0	0	0	0	0	0
2021-01-21	102.89	0.96429	-1	2	-4	0	7	0	3
2021-01-22	103.37	0.92857	0	0	0	0	0	0	0
2021-01-25	103.10	0.82143	-1	1	-3	-1	3	1	0
2021-01-26	105.44	0.78571	-1	1	-3	-1	3	1	0
2021-01-27	92.60	0.75000	-1	2	-2	-1	5	2	4
2021-01-28	90.38	0.71429	-1	2	-2	-1	4	2	3
2021-01-29	89.03	0.67857	-1	2	-2	-1	4	2	3
2021-02-01	92.25	0.57143	-1	2	-2	-1	4	3	4
2021-02-02	93.27	0.53571	-1	2	-2	-2	3	3	2
2021-02-03	96.28	0.50000	0	0	0	0	0	0	0
2021-02-04	96.03	0.46429	-1	1	-2	-2	2	2	0
2021-02-05	96.69	0.42857	-1	1	-2	-2	1	2	-1
2021-02-08	98.28	0.32143	-1	1	-1	-3	1	2	-1
2021-02-09	98.66	0.28571	-1	1	-1	-3	1	2	-1
2021-02-10	98.85	0.25000	-1	1	-1	-3	1	3	0
2021-02-11	100.93	0.21429	-1	1	-1	-3	1	3	0
2021-02-12	104.66	0.17857	-1	1	-1	-3	1	3	0
2021-02-16	105.57	0.03571	1	-1	0	4	0	-3	1
2021-02-17	115.42	0.00000	-1	1	0	-4	0	4	0
2021-02-18	115.25	0.96429	-1	1	-4	0	4	0	0
2021-02-19	118.17	0.92857	0	0	0	0	0	0	0

Table A.1: Real-time backtest from 2020-12-28 with weekly re-training of network, for piecewise linear utility function (3.5), the number of contracts  $n^i$  in  $F_t^i$ , the portfolio value  $P$  as given in equation (2.13), the CMF roll weight  $\omega$  appearing in equation (2.2), with the same for all  $i$  as explained in Section 3. The net position in VIX futures is  $\sum_i n^i$ . The position is long if the net is positive, short if the net is negative, and neutral if the net is zero. Of note is the loss seen from Jan. 26th to 27th during the GameStop trading freeze, and then recovery of losses in the following days.

## Appendix B Metrics for the SPY and VIX ETFs

Results that are reported in Section 3.1 should be compared with some standard benchmarks. Table B.1 presents, for SPDR S&P 500 Trust ETF, the same metrics that were used to evaluate the

VIX futures trading-signal portfolios for the same ten time periods of the  $k$ -fold cross validation. Table B.2, Table B.3, Table B.4, Table B.5, and Table B.5 display the same metrics for different VIX futures ETFs and ETNs, such as ProShares VIX Short-Term Futures ETF, iPath Series B S&P 500 VIX Short-Term Futures ETN, ProShares VIX Mid-Term Futures ETF, iPath Series B S&P 500 VIX Mid-Term Futures ETN, and iPath S&P 500 Dynamic VIX ETN.<sup>6</sup> The data that we used are downloadable from the Yahoo Finance website. In these tables, the metrics for some folds are missing, it is because the corresponding data are not available.

By comparing with the metrics for DFN-based trading signals that are displayed in Table 3.2 and Table 3.3 in Section 3.1, we can observe that the results that are produced by the neural network algorithm that we propose have reasonably good returns, profits, and Sharpe ratios, but also have high volatility and high drawdowns.

Statistics Fold \ \backslash	$\mathbb{E}[\mathbf{R}_t(\mathbf{a}(\mathbf{X}_t))]$	$\text{std}[\mathbf{R}_t(\mathbf{a}(\mathbf{X}_t))]$	Profit (%)	Sharpe Ratio	Maximum Drawdown
0	-0.172	0.415	-29.111	-0.439	-0.514
1	0.216	0.180	25.147	1.144	-0.157
2	0.138	0.211	14.347	0.607	-0.186
3	0.192	0.125	23.383	1.452	-0.096
4	0.222	0.105	27.567	2.014	-0.056
5	0.100	0.144	11.191	0.624	-0.119
6	0.097	0.130	11.171	0.676	-0.128
7	0.155	0.114	18.722	1.271	-0.101
8	0.095	0.145	10.514	0.583	-0.193
9	0.218	0.304	20.637	0.685	-0.337

Table B.1: Metrics for SPDR S&P 500 Trust ETF (ticker symbol: SPY) for the same ten folds listed in Table 3.1, which are used in the  $k$ -fold cross validation of Section 3.

Statistics Fold \ \backslash	$\mathbb{E}[\mathbf{R}_t(\mathbf{a}(\mathbf{X}_t))]$	$\text{std}[\mathbf{R}_t(\mathbf{a}(\mathbf{X}_t))]$	Profit (%)	Sharpe Ratio	Maximum Drawdown
2	-0.020	0.726	-25.155	-0.041	-0.516
3	-0.693	0.656	-82.766	-1.070	-0.833
4	-0.448	0.473	-58.787	-0.968	-0.706
5	-0.163	0.692	-40.503	-0.250	-0.614
6	-0.580	0.667	-74.397	-0.884	-0.839
7	-0.270	0.677	-48.582	-0.413	-0.665
8	-0.149	0.617	35.283	-0.259	-0.564
9	0.331	0.955	-15.597	0.336	-0.681

Table B.2: Metrics for ProShares VIX Short-Term Futures ETF (ticker symbol: VIXY) for the same ten folds listed in Table 3.1, which are used in the  $k$ -fold cross validation of Section 3.

---

<sup>6</sup>The Yahoo Finance website: <https://finance.yahoo.com/>

Statistics Fold \ \backslash	$\mathbb{E}[\mathbf{R}_t(\mathbf{a}(\mathbf{X}_t))]$	$\text{std}[\mathbf{R}_t(\mathbf{a}(\mathbf{X}_t))]$	Profit (%)	Sharpe Ratio	Maximum Drawdown
7	5.572	1.123	43.709	4.951	-0.306
8	-0.143	0.617	-34.642	0.247	-0.564
9	0.321	0.942	-15.396	0.330	-0.683

Table B.3: Metrics for iPath Series B S&P 500 VIX Short-Term Futures ETN (ticker symbol: VXX) for the same ten folds listed in Table 3.1, which are used in the  $k$ -fold cross validation of Section 3.

Statistics Fold \ \backslash	$\mathbb{E}[\mathbf{R}_t(\mathbf{a}(\mathbf{X}_t))]$	$\text{std}[\mathbf{R}_t(\mathbf{a}(\mathbf{X}_t))]$	Profit (%)	Sharpe Ratio	Maximum Drawdown
2	-0.092	0.370	-15.913	-0.276	-0.276
3	-0.528	0.301	-63.331	-1.783	-0.646
4	-0.287	0.230	-36.741	-1.286	-0.493
5	-0.051	0.349	-13.297	-0.175	-0.307
6	-0.228	0.302	-31.823	-0.789	-0.452
7	-0.258	0.275	-34.368	-0.973	-0.456
8	-0.041	0.267	-92.530	-0.190	0.244
9	0.791	0.511	77.000	1.526	-0.266

Table B.4: Metrics for ProShares VIX Mid-Term Futures ETF (ticker symbol: VIXM) for the same ten folds listed in Table 3.1, which are used in the  $k$ -fold cross validation of Section 3.

Statistics Fold \ \backslash	$\mathbb{E}[\mathbf{R}_t(\mathbf{a}(\mathbf{X}_t))]$	$\text{std}[\mathbf{R}_t(\mathbf{a}(\mathbf{X}_t))]$	Profit (%)	Sharpe Ratio	Maximum Drawdown
7	1.562	0.553	24.962	2.805	-0.082
8	-0.080	0.250	-13.413	-0.360	-0.241
9	0.789	0.512	76.718	1.521	-0.263

Table B.5: Metrics for iPath Series B S&P 500 VIX Mid-Term Futures ETN (ticker symbol: VXZ) for the same ten folds listed in Table 3.1, which are used in the  $k$ -fold cross validation of Section 3.

Statistics Fold \ \backslash	$\mathbb{E}[\mathbf{R}_t(\mathbf{a}(\mathbf{X}_t))]$	$\text{std}[\mathbf{R}_t(\mathbf{a}(\mathbf{X}_t))]$	Profit (%)	Sharpe Ratio	Maximum Drawdown
8	-0.054	0.144	-2.340	0.444	-0.055
9	1.131	0.505	121.335	2.218	-0.295

Table B.6: Metrics for iPath S&P 500 Dynamic VIX ETN (ticker symbol: XVZ) for the same ten folds listed in Table 3.1, which are used in the  $k$ -fold cross validation of Section 3.

## Bibliography

- Irene Aldridge and Marco Avellaneda. *Big data science in finance*. Wiley, New York City, New York, USA, 1st edition, 2020. ISBN 9781119602989.
- Carol Alexander, Julia Kapraun, and Dimitris Korovilas. Trading and investing in volatility products. *Financial Markets, Institutions & Instruments*, 24(4):313–347, 2015. ISSN 1468-0416.
- Sylvain Arlot and Alain Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010. ISSN 1935-7516.
- Robert D. Arnott, Campbell R. Harvey, and Harry Markowitz. A backtesting protocol in the era of machine learning. *The Journal of Financial Data Science*, 1(1):64–74, 2019. ISSN 2640-3943.
- Marco Avellaneda and Andrew Papanicolaou. Statistics of VIX futures and applications to trading exchange-traded products. *International Journal of Theoretical and Applied Finance*, 22(1), February 2019. ISSN 0219-0249.
- Christian Bayer, Peter Friz, and Jim Gatheral. Pricing under rough volatility. *Quantitative Finance*, 16(6):887–904, 2016. ISSN 1469-7688.
- Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. In *ICML Thirty-Fifth International Conference on Machine Learning*, 2018.
- Christoph Bergmeir and José M. Benítez. On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213, 2012. ISSN 0020-0255.
- Christoph Bergmeir, Rob J. Hyndman, and Bonsoo Koo. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, 120: 70–83, 2018. ISSN 0167-9473.
- Prabir Burman and Deborah Nolan. Data-dependent estimation of prediction functions. *Journal of Time Series Analysis*, 13(3):189–207, 1992. ISSN 0143-9782.
- Philippe Casgrain, Brian Ning, and Sebastian Jaimungal. Deep Q-learning for Nash equilibria: Nash-DQN. *arXiv preprint arXiv:1904.10554*, 2019.
- Vitor Cerqueira, Luis Torgo, and Igor Mozetič. Evaluating time series forecasting models: An empirical study on performance estimation methods. *Machine Learning*, 109(11):1997–2028, October 2020. ISSN 0885-6125.
- George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989. ISSN 0932-4194.
- Robert Fry Engle and Clive William John Granger. Co-integration and error correction representation estimation and testing. *Econometrica*, 55(02):251–276, 1987. ISSN 0012-9682.
- Omar El Euch and Mathieu Rosenbaum. Perfect hedging in rough heston models. *The Annals of Applied Probability*, 28(6):3813–3856, 2018. ISSN 1050-5164.
- Jianqing Fan, Zhaoran Wang, Yuchen Xie, and Zhuoran Yang. A theoretical analysis of deep q-learning. In Alexandre M. Bayen, Ali Jadbabaie, George Pappas, Pablo A. Parrilo, Benjamin Recht, Claire Tomlin, and Melanie Zeilinger, editors, *Proceedings of Machine Learning Research (PMLR): Learning for Dynamics and Control*, volume 120, pages 465–475. ML Research Press, June 2020. ISSN 2640-3498.
- Jean-Pierre Fouque, George Papanicolaou, and K. Ronnie Sircar. Mean-reverting stochastic volatility. *International Journal of Theoretical and Applied Finance*, 3(1):101–142, 2000. ISSN 0219-0249.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. Adaptive computation and machine learning series. The MIT Press, Cambridge, Massachusetts, USA, November 2016. ISBN 9780262035613.

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*, 2019.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518 (7540):529–533, 2015. ISSN 0028-0836.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. Adaptive computation and machine learning series. The MIT Press, Cambridge, Massachusetts, USA, 2nd edition, December 2018. ISBN 9780262039406.

Allan Pinkus. Approximation theory of the MLP model in neural networks. *ACTA Numerica*, 8: 143–195, 1999. ISSN 0962-4929.

Justin Sirignano and Konstantinos Spiliopoulos. Stochastic gradient descent in continuous time. *SIAM Journal on Financial Mathematics*, 8(1):933–961, 2017. ISSN 1945-497X.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction*. Adaptive computation and machine learning series. The MIT press, Cambridge, Massachusetts, USA, 2nd edition, 2018. ISBN 9780262039246.

Robert E. Whaley. The investor fear gauge. *The Journal of Portfolio Management*, 26(3):12–17, 2000. ISSN 0095-4918.

Robert E. Whaley. Understanding the VIX. *The Journal of Portfolio Management*, 35(3):98–105, 2009. ISSN 0095-4918.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *Fifth International Conference on Learning Representations*, 2017.