Application of Neural network for Trade Signal generation in VIX Futures

Soham Mudalgikar
Department of Operations Research
smmudalg@ncsu.edu

Parimal Rohan Mehta
Department of Computer Science
pmehta3@ncsu.edu

Omkar Rajendra Madure
Department of Industrial Engineering
ormadure@ncsu.edu

Keywords—VIX Futures, Trading Signals, Contango, Deep Learning, Feedforward Neural Networks, Cross-Validation.

I. ABSTRACT

With the ever-increasing demand in predicting the market's volatility, there is a high need for models or techniques to generate trade signals. Hence machine learning models are in demand for generating such trading signals, especially in Volatility Index (VIX) Futures. In this report, we propose an approach for trading the VIX futures based on Neural Network based on the article "Trading signals in VIX Futures" [1]. VIX futures is a real-time index representing the market expectations and investors sentiments for the relative strength of the S&P500 index (SPX). This is derived from the SPX index options with near-term expiration; it generates a month forward projection of volatility.

The most common assumption in Volatility models is that they follow Markov property or they are Martigels. So, the trading term strategy that maximizes the expected utility function given the current shape and level of VIX future is known. Mathematically, we formulate the dependability between VIX futures curves, their position, and expected utility and approximate it using deep neural networks. We use the AutoRegressive (AR) model as the base model and use the Deep Feed Forward Neural network and the Recurrent neural Network to compare the performance and calculate the portfolio performance, maximum drawdown, and Sharpe ratio. This will generate the trade signal for the investors to either go long or short VIX futures depending on the market environment.

II. MOTIVATION

The Volatility Index (VIX) represents the market's expectation about the near-term price of the S & P 500 Index. The Chicago Board Of Exchange (CBOE) introduced VIX future in 2004, the shape of the VIX future curve is informative; however, it is likely to persist only for a short period. This could give a simple trade; for example, if the curve is hump-shaped, then there may be a long-short VIX futures position, which when the curve reverts to contango.[4] This fluctuation is very spontaneous and can generate a profit with near certainty. However, most trade involves a non-zero probability of losses. But, if trading strategies are constructed to optimize the expected utility function with multiple trading opportunities in the long term, we may experience fewer losses.

VIX is used as a fear gauge to suggest market participants' sentiments. VIX derivatives are one of the most liquid assets in the financial world. The shape of the VIX future curve is informative. Hence, if we can optimize the expected utility function with multiple trading opportunities, we may experience fewer losses.

III. INTRODUCTION

Over long time horizons, with multiple trading opportunities, the losses can be diminished if trading strategies are constructed to optimize the expected utility function. VIX futures are a good option for trading strategies as their curves can turn to contango, allowing for a turnaround before the next trading opportunity.

For a stationery VIX curve, the action that will maximize the expected utility function under a probability distribution is given by

$$a(x) = arg \max_{a \epsilon \mathcal{A}} \mathbb{E}[U(R_{t+1}(a))|X_t = x]$$

X_t = VIX futures curves at time t, vector-valued,
A = a set of possible trades/actions a, vector valued,
R_{t+1} (a) = change in position from time t to t + 1 if action a ∈ A is taken

$$R_{t+1}(a) = \sum_i a^i \left(\frac{\Delta I_t^i}{I_t^i} - r\Delta t \right)$$

U(.) = Utility function given by,

$$U(R) = \max(R, 0) + \gamma \min(R, 0)$$

Where t denoted time, E denoted expected value and $R_{t+1}(a)$ is the change in position from time $t \; to \; t+1$ if action a is taken given that we know the path of the function.

We assume $\mathcal{A}$ to be a finite set of trade, and X_t transition distribution is already known. The Expected utility function in the above optimization problem is estimated using a deep neural network, and parameters of X_t are estimated using historical VIX future data. The popular trade of the 2010 decade was to long (buy) one-month VIX futures and short (sell) five-month VIX future rolling positions.[1] So, we focus only on this trade

$$\mathcal{A} = \{(-1,2), (0,0)\}.$$

Where individual actions are,

$$(0,0) = \ no\ trade,$$

$$(-1,2) = \ short\ I^1\ and\ two * \ long\ I^5$$

IV. Dataset Description

The data is collected from an online site for 12 years daily to compare the correlations. We are using data on VIX Future prices with maturity in one month to nine months and the spot price for the period between 14th April 2008 to 31st December 2020, so the last date for nine-month maturity VIX future is 16th September 2021. The total dates or rows are 3204, and some parameters are created using these Futures prices, such as rolling value and CMFs values. From this raw data, the values of Constant-Maturity Futures (CMFs), roll, etc., will be computed, which will be our parameters of X_t. So there are a total of 22 columns in the dataset. The VIX future data is taken from http://vixcentral.com.

V. MODEL DESCRIPTION

While dividing the dataset, we use 50% data for training and 50% for testing the model. As we can simulate the VIX Future paths for training using the Autoregressive model or using Fractional Brownian motion using remaining 50% data will help us to validate the model and for backtesting, also considering the number of trade opportunities, we choosing a 50:50 ratio is the better choice

Evaluation of models will be done based on portfolio returns, Sharpe ratio, and maximum drawdown, computed by defining a time series portfolio. We compare all three models based on a rolling portfolio going short on 1 month and long on 5 months period on VIX Futures. The final objective would be to estimate the optimal expected utility function and create trade signals for VIX futures, which will be evaluated based on previously mentioned parameters.

The primary focus of this project is on approximating or estimating the optimal Expected utility function using a neural network. The use of Deep feed-forward Network (DFN)[2] with Vector Autoregressive model as baseline model and its results are given in paper "Trading Signal in VIX Futures." In general, financial time series data shows some auto-correlation to tackle this problem, Recurrent Neural Network (RNN) may be helpful in which feedback may serve as a memory that can help us approximate optimal utility function in a better way. We plan to implement RNN and compare the results between these two Neural Networks and some classification trees Models like Random Forest.

i) Baseline Model:

Historically volatility models have relied on Markovian property, so the current value depends on the immediately preceding value, which is an AR(1) model. We take the Auto-Regressive Model as a baseline model, so we say the future value of VIX Futures is based on the current/ past price of VIX futures. We will be using the AR model and then drawing samples from it to train the neural network.

$$\boldsymbol{\psi}_{t+1} = \boldsymbol{\mu} + \boldsymbol{A}\boldsymbol{\psi}_t + \boldsymbol{Z}_{t+1}$$

Where Z_t is an independent and identically distributed Gaussian random vector with mean 0 and covariance Σ. The least-squares estimator of A is

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

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

ii) Deep feed-forward Neural Network:

We plan to implement this baseline AR model as explained in the paper "Trading Signal in VIX Futures" and leverage the universal approximation theorem to ensure that DFN can effectively estimate the nonlinear mapping into a state action-value function. We will be performing some hyperparameters tuning to see if the results change. The network has eleven neurons in the input layer and one neuron in the output layer, suggesting whether or not to open a trade position; there are five hidden layers. The activation function used is Relu. The total number of epochs used are 500 and 1000.

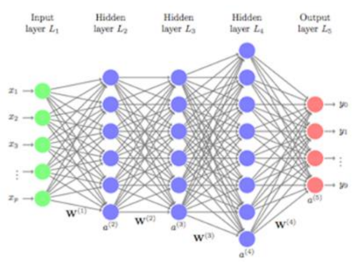

Figure 1: Schematic diagram of deep feed forward Neural Network.

iii) Recurrent Neural Network:

Generally, financial time series data shows some auto-correlation using a Recurrent Neural network with AR model as a baseline may provide a better approximation of state action-value function giving better performance in terms of trade signal. In the RNN model, we used 105 steps to the look-back window, which represents approximately five trading month time periods, and we set the algorithm to withhold 20% for validation. The input layer has four neurons, and there are five hidden layers with 250 neurons in each layer; the output layer has a sigmoid activation function with only one neuron to give output as to whether to open or not a trade position. Epochs for the training model are 500 and 1000.[2]

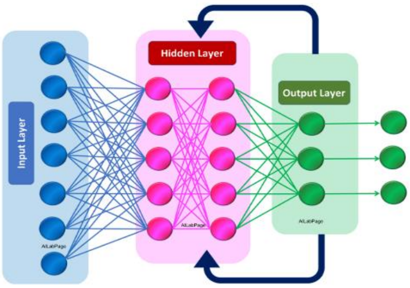

Figure 2:Schematic Diagram of Recurrent Neural Network.

VI. METHODOLOGY

We are using an autoregressive model, which can be represented by

$$\psi_{t+1} = \mu + A\psi_t + Z_{t+1}$$

where Z_t is an independent and identically distributed Gaussian random vector with mean zero and covariance Σ and A is the least-squares estimator. Using these AR models, we can find the value of the rolling portfolio and hence the expected utility function.

We can use a feed-forward network and neural network for better accuracy. We used hyperparameters configured as 500 epoch, 0.1 dropout rate, and sigmoid function as activation for the Feed Forward Neural network. We chose 5 layers of the dense feed-forward network as suggested in the "Trading Signals of VIX Futures" paper. A time-series or timestep dimension is added to create the required input shape for RNN, with a look-back window of 105 days. For RNN, we used sigmoid as activation function here; other parameters are 1000 epochs and 5 dense layers, four neurons at the input layer, and 250 neurons in each hidden layer.

- Data is divided into a training set consisting of 50% training data and 50% testing data. If needed, k-fold cross-validation will be performed.
- To evaluate models, a time series of trading signals portfolio will be defined to compute portfolio returns, Sharpe ratio, and maximum drawdown
- Deep Feed-Forward Neural Network (DFN) and Recurrent Neural Network (RNN) and classification tree models will be compared with a rolling portfolio of going long (buy) on 1-month and short (sell) on 5-month VIX Futures.
- The final objective would be to compare the plots of the Neural Networks and estimate the optimal expected utility function

VII. RESULTS

Training loss and accuracy started to improve from 0 to 100 epochs for DFN and RNN. The plot of DFN training accuracy suggests that training accuracy converges for selected hyperparameters after a certain point, and the same can be said about the training loss. On the other hand, we observed that validation loss is also reduced and converges after certain epochs, but validation accuracy behaves like noise or, in other words, it doesn't converge.

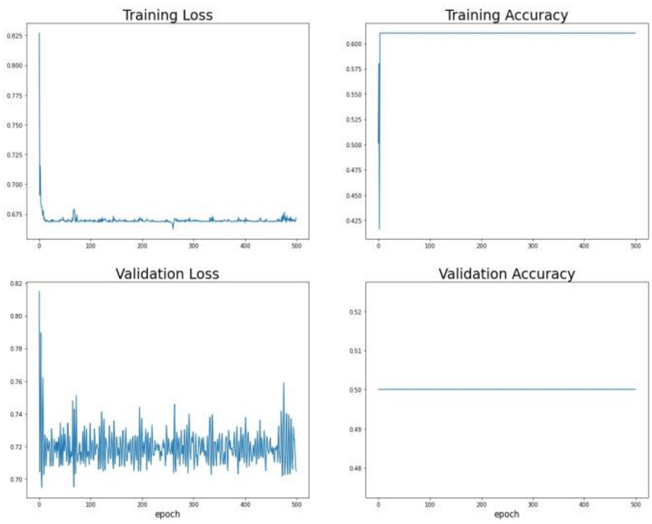

Figure 3:DFN Training and validation loss (left side) and Training and Validation Accuracy (right).

In the case of RNN, training accuracy and training loss improved instantly with respect to the number of epochs, but we can see that training accuracy is lower and training loss is higher for RNN compared with DFN. Also, validation loss is not converged to a fixed level; it also looks noisy.

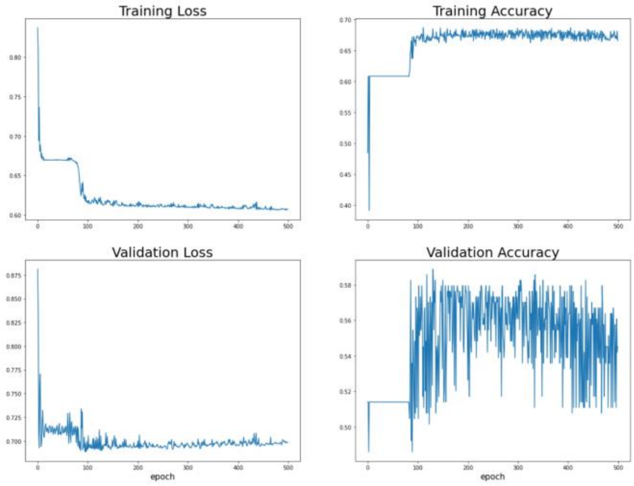

Figure 4:RNN Training and validation loss (left side) and Training and Validation Accuracy (right).

For both models, in-sample and out of sample ROC curves suggest that we are not choosing randomly, and our model is doing something better than just guessing. The ROC curve for both models looks almost similar, but the mean and optimal point in in-sample ROC curve for DFN is close, but the median is relatively at more distance. Whereas in the case of RNN, the mean, median, and optimal points are almost at the same position in the in-sample ROC curve. The out-of-sample ROC curve for both models is comparitivly at a lower level than in-sample ROC curve but still suggests that the model is doing more than a random guess. For both models, the optimal point on the out-of-sample ROC curve is at the middle, the mean is above the optimal point, and the median is below.

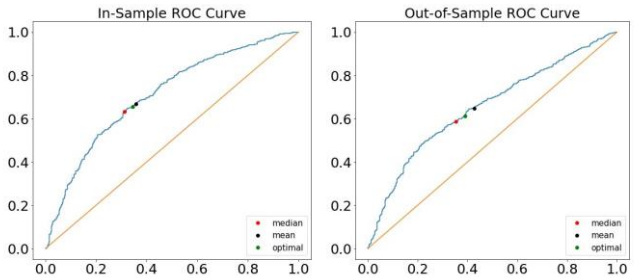

Figure 5: In-sample and out-of-sample ROC curve for DFN model.

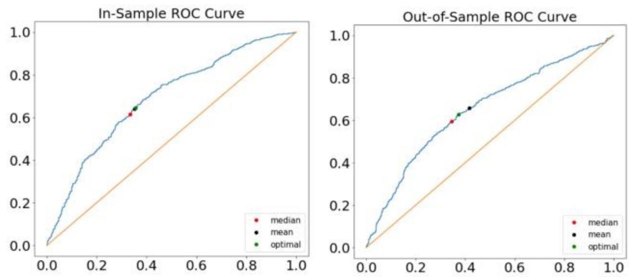

Figure 6:In-sample and out-of-sample ROC curve for RNN model.

VIII. EVALUATION

A rolling portfolio is created to evaluate these models, and the time series of values of the trading-signals portfolio is used. Performance metrics such as portfolio returns and maximum drawdown are computed using these rolling portfolios. As mentioned earlier, the historical data is divided into a training set consisting of 50% and a test set with the remaining 50%.

As observed in the figures below, the value of the portfolio of trade signal generated by the DFN model is consistent with that of the AR model portfolio till the training data date. However, after that, the portfolio's value with the DFN model increases more than the AR model portolio which suggests that the DFN model performs better than the AR model on out-of-sample data or validation data.

Figure 7: Rolling portfolio value for trade signal generated by DFN model (blue line) and AR model (Yellow line).

Similarly, the rolling portfolio constructed using the RNN model trade signal is consistent with the AR model portfolio for the training period, but the RNN model portofolio outperforms the AR model by a large margin for the out-of-sample period. This can be observed in the figure below,

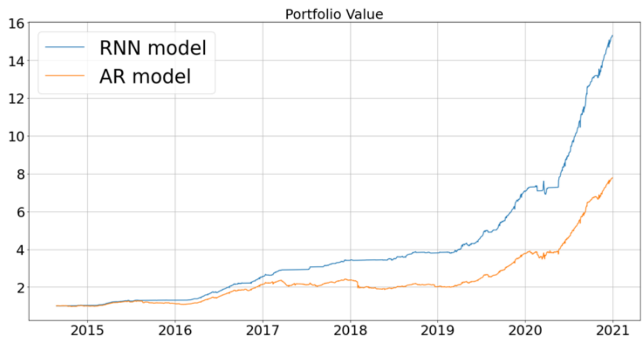

Figure 8: Rolling portfolio value for trade signal generated by the RNN model (blue line) and the AR model (Yellow line).

At last, DFN, RNN, and classification tree models will be compared with a rolling portfolio of going long (buy) on one-month VIX futures and short (sell) on five-month VIX futures. If needed, k-fold validation will be performed.[3]

The standard evaluation parameters for finance models are Portfolio Returns and maximum drawdown. In the case of trade signals, generally, returns and maximum drawdown are compared. The figure below yellow line is for the AR model, and the blue line is for the RNN model and DFN model; it can be observed that NN models have lower maximum drawdown.

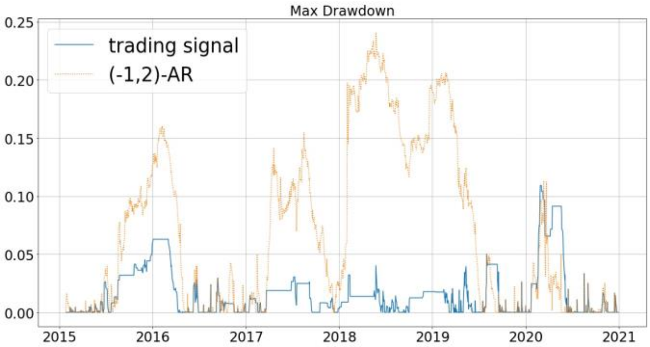

Figure 9:Max Drawdown for DFN model (blue line) and AR model (yellow line).

In the case of the Neural Network, we can observe that around mid-2015 (almost with the start of the out-of-sample time period), the time series of portfolio value started deviating from that of portfolio value of the AR model positively. The reasons for such outstanding performance are 1) DFN and RNN both have a better rate of returns as given in table 2) maximum drawdown is also less than 1% for NN models, whereas the AR model has a return of 34% and a maximum drawdown of 24%.

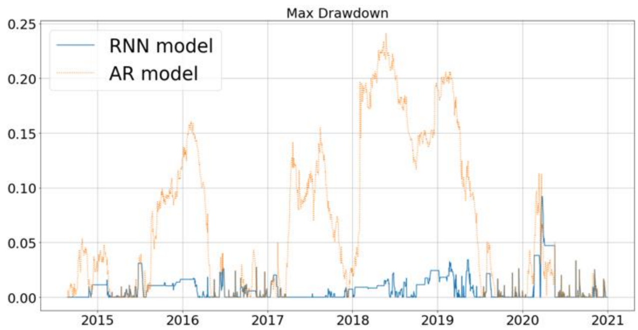

Figure 10: Max Drawdown for RNN model (blue line) and AR model (yellow line).

Performace Parameter	Return (%)	Max Drawdown(%)
AR Model	32.30%	24.07%
DFN model with 500 epochs	40.90%	0.63%
DFN model with 1000 epochs	38.16%	0.93%
RNN model with 500 epochs	42.15%	1.7%
RNN model with 500 epochs	43.43%	0.93%

The future direction for this work will be expanding the set of trading actions to consider some other sets of trades. Also, some constraints can be included along with transaction cost to see the profits. The utility function can be changed to something other than piecewise linear function, along with some complex models like emending random forest with Neural Network.

References

[1] Marco Avellaneda, T.N. Li. A. Papanicolaou, G. Wang. Trading Signals in VIX Futures. arXiv:2103.02016v2, 2021

[2] 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.

[3] Sylvain Arlot and Alain Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010. ISSN 1935-7516.

[4] 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.