THE FLORIDA STATE UNIVERSITY
COLLEGE OF ARTS & SCIENCES

ARBITRAGE-FREE PREDICTION OF THE IMPLIED VOLATILITY SURFACE WITH TEMPORAL ATTENTION STRATEGY

By

ELI BUTTERS

A Thesis submitted to the Department of Statistics
in partial fulfillment of requirements for graduation with Honors in Statistics.

Degree Awarded:
Spring, 2025

Thesis Committee

Dr. Rongjie Liu	Department of Statistics,
Thesis Director	Florida State University
	University of Georgia
Dr. Alec Kercheval	Department of Mathematics,
Outside Committee Member	Florida State University
Dr. Joshua Loyal	Department of Statistics,
Committee Member	Florida State University

Signatures are on file with the Honors Program office

Abstract

We propose a model for predicting the future implied volatility surface (IVS) while ensuring that the forecast is free from static arbitrage. We select a discrete set of IVS points as well as 10 exogenous features to represent the broader market condition. Two feature extraction methods are considered: variational autoencoder and IVS sampling. A long short-term memory neural network with a multi-head temporal attention mechanism is used for prediction. This model will be trained using implied volatility surface data of options on the SPDR S&P 500 ETF (SPY), an exchange traded fund which tracks the S&P 500 index. Once an IVS is predicted, a deep neural network will be used to ensure that the extracted values are free from static arbitrage. We find that the IVS sampling approach provides superior predictive capability only when the multi-head temporal attention mechanism is used.

1 Introduction

Stock option contracts are priced using 5 numerical inputs which are strike price, underlying asset price, time to expiration, risk-free interest rate, and volatility. Unlike all other option pricing inputs, volatility is not a directly observable market variable. The widely used Black-Scholes-Merton Model, which was first proposed in 1973, assumes that volatility is constant throughout the life of an option contract [3, 20]. Under this assumption, every option contract on a single underlying asset would have the same volatility input regardless of its strike price, underlying asset price, maturity, or current market condition. In reality, we can observe that this assumption produces implied volatility surfaces (IVS) which are not free of static arbitrage [22].

The IVS can be empirically observed by inverting the Black-Scholes-Merton option pricing formula. The implied volatility of a specific contract $\sigma_t(K,T)$ with strike price K and maturity T depends on (K,T). The mapping function

$$\sigma_t : (K,T) \to \sigma_t(K,T)$$

shows this dependence and represents the IVS at time t.

There are two main features of the IVS which are insufficiently described by the Black-Scholes-Merton model and have drawn the attention of modellers and researchers. First, the existence of a non-flat profile, or term structure, shows the immediate flaw of the Black-Scholes-Merton model. This leads to surfaces referred to as a 'smile' or 'smirk' depending on the tail risk expected by market participants [9]. Second, the existence of a 'jump risk', or large sudden volatility movements from shifting supply and demand. This has been shown to cause a challenge in risk management models. These jumps largely come from important earnings announcements or other consequential macro-economic reports.

There have been many different attempts at modeling and understanding the movements of the IVS. Early models incorporated jump-diffusion to compensate for large volatility movements such as the Merton Jump-Diffusion Model in 1976 [21]. Others used the concept of stochastic volatility seen in the Heston Model proposed in 1993 [16]. More recently, we have seen the innovation of rough volatility, where the processes for volatility are seen as non-Markov [5]. This concept was first introduced by Gatheral et al [2018] [13].

2 Research Problem

Accurate predictions of the future IVS can be useful for tasks including volatility trading, derivatives pricing, and market analysis. While they will not be tested in this research, the uses of having accurate predictions for the future IVS can drastically improve market making operations and can provide alpha for volatility trading strategies. Zhao and Borochin [2023] showed that there is significant economic value for delta-hedged options portfolios in having accurate predictions of future implied volatility [26].

Strategies which speculate on short term price direction of front month VIX futures can see direct benefits from trading based on these accurate predictions. These types of strategies are prone to return distributions with fat tails, meaning that they experience sudden large spikes which produce a challenge for risk management. Therefore, being able to have an accurate indicator of potential large volatility moves can allow for much better hedging operations to be performed.

This study will focus on predicting the future expected IVS. Specifically, given the IVS at a time $\Sigma_t(K,T)$, we are interested in finding $\hat{\Sigma}_{t+1}(K,T)$. Past studies have investigated interpolation and feature extraction of the dynamic IVS [25]. However, given recent studies using new, previously untested, solutions for this problem, we believe that this requires further research. Furthermore, since capturing the bivariate dynamics of the IVS is quite challenging, we must make sure that these new, potentially advantageous, solutions allow for arbitrage-free conditions to be met.

In order to achieve this challenging prediction task, we will employ a variational autoencoder (VAE) to extract features from the IVS data. This approach will be tested with different latent vector dimensions, namely $d = 2, 5, 10, 20, 40$. It will also be compared with a simpler sampling approach which will use the entire standard IVS as a feature vector. Next, these different feature extraction approaches will be used to train a LSTM neural network with a multi-head attention mechanism. This LSTM with multi-head attention will be tested with different amounts of attention heads, specifically $n = 1, 8, 16, 32, 64$. In order to compare these different feature extraction and prediction challenges, we will compare them using 3 metrics for their out of sample tests. These 3 metrics are the root mean squared error (RMSE), mean absolute percentage error (MAPE), and the mean directional accuracy (MDA) which are defined as:

$$\text{RMSE} = \sqrt{\frac{1}{\sum_{i=T_1}^{T_n} |\Sigma_i|} \sum_{t=T_1}^{T_n} \sum_{(K,T)\in\Sigma_t} (\sigma_t(K,T) - \hat{\sigma}_t(K,T))^2},$$

$$\text{MAPE} = \frac{1}{\sum_{i=T_1}^{T_n} |\Sigma_i|} \sum_{t=T_1}^{T_n} \sum_{(K,T)\in\Sigma_t} \left| \frac{\sigma_t(K,T) - \hat{\sigma}_t(K,T)}{\sigma_t(K,T)} \right|,$$

$$\text{MDA} = \frac{1}{\sum_{i=T_1}^{T_n} |\Sigma_i|} \sum_{t=T_1}^{T_n} \sum_{(K,T)\in\Sigma_t} \mathbf{1}_{sign(\sigma_t(K,T)-\sigma_{t+1}(K,T))==sign(\hat{\sigma}_t(K,T)-\sigma_{t+1}(K,T))},$$

where T_1 is the first out of sample test day, T_n is the last out of sample test day, $\mathbf{1}$ is the indicator function, and $sign(x)$ is the sign of x.

3 Related Work

The IVS is a highly complex representation of the movements and risks that the market expects. This means that the IVS is not just driven by historic values, but is also influenced by other exogenous market factors. Zhang et al [2022] add that their model could be further improved through the use of these exogenous factors such as the VIX level and underlying index return [25]. Cao et al [2020] found that incorporating the VIX into the input vector of a neural network model designed to model movements in the implied volatility surface improved the mean squared error by 62.12% [6].

Chen and Zhang [2019] employed the use of an LSTM neural network with an attention mechanism and found its out of sample prediction error to be better than that of the LSTM and multi-layer perceptron [8]. Despite this, newer approaches including Zhang et al [2022] and Chen at al [2023] resorted to LSTM models without an attention mechanism [25, 7]. Attention is a mechanism proposed through the Google Brain and Google Research labs by Vaswani et al [2017] [24]. It allows the LSTM to have selective focus similar to the way in which humans brains interpret their environment. This helps the model deal with long feature vectors by focusing on the most relevant input sequences and provides a way for the LSTM to access important features of input sequences from the memory in the attention mechanism.

Chen at al [2023] make use of a conditional variational autoencoder (CVAE) in order to encode IVS data into a noise reduced latent vector to be used by an LSTM for future surface generation [7]. While this specific approach for feature extraction has not been compared to others, Zhang et al [2022] perform comparisons between a sampling and a VAE approach and found similar results with either [25]. This prompts the question of whether the VAE approach will prevail when dealing with more complicated non linear data from exogenous variables.

Bloch and Böök [2021] proposed a deep learning approach using multiple stacked convolutional long short-term memory (LSTM) layers to forecast future implied volatility smiles on S&P 500 option prices [4]. Their model integrates convolutional layers to extract spatial features and uses an LSTM to capture the temporal dependencies. Chen at al [2023] used a conditional variational autoencoder and an LSTM to generate future implied volatility surfaces conditional on a set of data [7]. This approach captures the probabilistic and temporal aspects of the volatility surface and leverages historical data to extract new volatility surfaces from past latent representations. Both of these approaches are capable of capturing the dynamics of real world options data on the S&P 500 and show that this data is predictable using modern deep learning techniques.

One problem with many of these approaches is that they do not necessarily produce an IVS which is free of static arbitrage. Static arbitrage is simply the existence of a risk-free profit opportunity that arises because of an inconsistency in the pricing of current options in the market. The conditions for a static arbitrage free IVS have been studied in depth by Roper [2010] [22]. Interestingly, Zheng et al [2021], Ackerer et al [2020], and Zhang et al [2022] built the static arbitrage free constraints into the loss functions of their models. In particular, Zhang et al [2022], reported that the empirical results from all feature generation methods using a deep neural network had no static arbitrage violations in their out of sample test. An alternative approach taken by Dellaportas and Mijatović [2014] predicts the implied volatility smile by fitting a stochastic model to the inputs of a SABR parameterisation [10]. While this approach did improve their baseline random walk approach when fitting to the volatility smile, this model does not fit the entire IVS well.

Bai and Cai [2023] studied the effects of different exogenous factors for predicting movements in the VIX [2]. Since the VIX is constructed from S&P 500 options that expire in an average of 30 days and are near the money, its prediction is related to the entire IVS. In particular, Bai and Cai found that the US weekly jobless report data, day of the week, and others heavily influence the direction that the VIX will trade in the coming days. This data is particularly interesting because there is no evidence of any study using a sophisticated deep learning technique to capture the relationship between these variables and the future IVS.

The model we propose seeks improvements to previous models in two primary problems. First, we will show that multi-head temporal attention outperforms all other LSTM approaches. Second, we will show that temporal attention reduces the need for dimension reduction techniques.

4 Real Data

The data that will be used in this research is from OptionsDX. This data contains the historical implied volatility surfaces of call options on SPY, which is an exchange traded fund that tracks the S&P 500 index. The dates range from February 2010 to September 2023 and the prices were quoted at the end of each trading day. The data was cleaned in the following two ways. First, options with less than 5 days until expiration were removed from the dataset. Second, options with no observed implied volatility or price were removed. This left 3,409 days of IVS observations with an average of 2,369.5 option prices per day. In total, we have 8,077,539 total implied volatility observations.

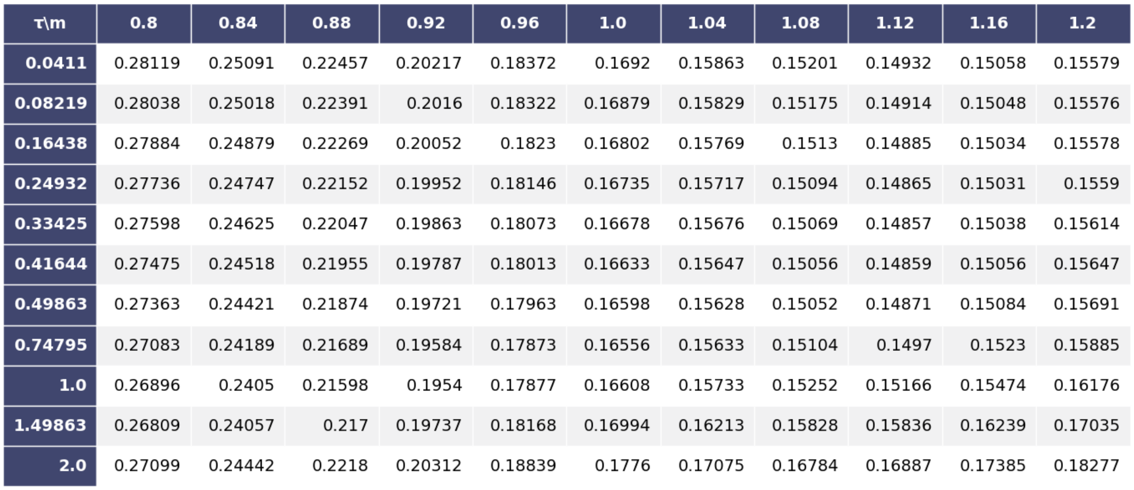
The IVS can be expressed using using the moneyness m and time to expiration τ pair (m, τ). The moneyness considered will range from 0.8 to 1.2 with an increment of 0.04 and τ = 15, 30, 60, 91, 122, 152, 182, 273, 365, 547, 730 days until maturity.

When turned into a grid, this creates 121 standard IVS points.

While options trading on a given day trade at nonstandard (m, τ) combinations, for each day, we will use an algorithm to interpolate the observed implied volatilities in order to fit our standard IVS points. We will consider a popular parametric model proposed by Dumas et al [1998] which we will refer to as DFW [11]. This model estimates the implied volatility given

$$\sigma(m, \tau) = max(0.01, a_0 + a_1 m + a_2 \tau + a_3 m^2 + a_4 \tau^2 + a_5 m \tau), \tag{1}$$

where there is a floor set at 0.01 in order to prevent implied volatility values from being negative. The terms $a_0, a_1, a_2, a_3, a_4, a_5$ are then estimated using regression.



τ\m	0.8	0.84	0.88	0.92	0.96	1.0	1.04	1.08	1.12	1.16	1.2
0.0411	0.28119	0.25091	0.22457	0.20217	0.18372	0.1692	0.15863	0.15201	0.14932	0.15058	0.15579
0.08219	0.28038	0.25018	0.22391	0.2016	0.18322	0.16879	0.15829	0.15175	0.14914	0.15048	0.15576
0.16438	0.27884	0.24879	0.22269	0.20052	0.1823	0.16802	0.15769	0.1513	0.14885	0.15034	0.15578
0.24932	0.27736	0.24747	0.22152	0.19952	0.18146	0.16735	0.15717	0.15094	0.14865	0.15031	0.1559
0.33425	0.27598	0.24625	0.22047	0.19863	0.18073	0.16678	0.15676	0.15069	0.14857	0.15038	0.15614
0.41644	0.27475	0.24518	0.21955	0.19787	0.18013	0.16633	0.15647	0.15056	0.14859	0.15056	0.15647
0.49863	0.27363	0.24421	0.21874	0.19721	0.17963	0.16598	0.15628	0.15052	0.14871	0.15084	0.15691
0.74795	0.27083	0.24189	0.21689	0.19584	0.17873	0.16556	0.15633	0.15104	0.1497	0.1523	0.15885
1.0	0.26896	0.2405	0.21598	0.1954	0.17877	0.16608	0.15733	0.15252	0.15166	0.15474	0.16176
1.49863	0.26809	0.24057	0.217	0.19737	0.18168	0.16994	0.16213	0.15828	0.15836	0.16239	0.17035
2.0	0.27099	0.24442	0.2218	0.20312	0.18839	0.1776	0.17075	0.16784	0.16887	0.17385	0.18277

Figure 1: Standard IVS values of S&P 500 options on 5/26/2011

In Figure 1, the columns represent the standard moneyness values and the indices represent the standard time to expiration values.

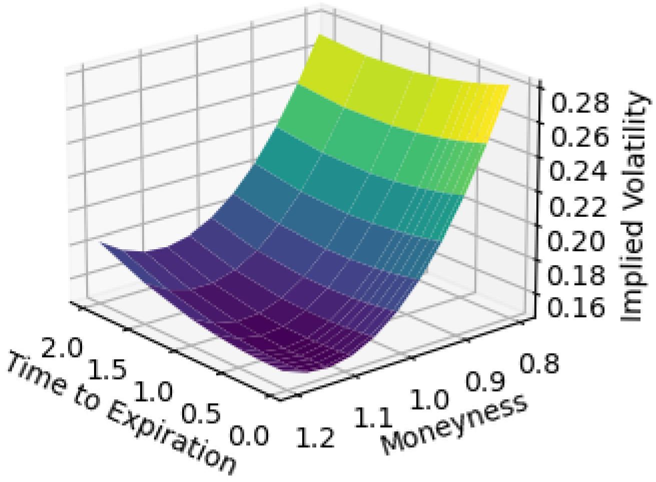


Figure 2: Implied volatility surface of S&P 500 options on 5/26/2011

In Figure 2, the standard moneyness and time to expiration values are on the x and y axes and the implied volatility is on the z axis.

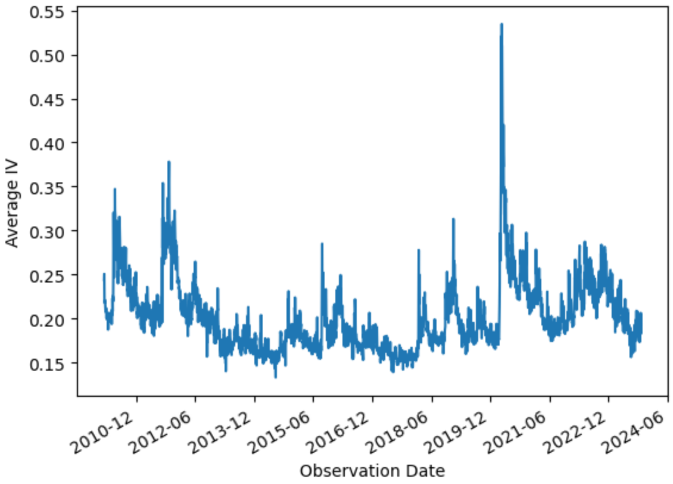


Figure 3: Average implied volatility throughout time

In Figure 3, the y-axis is the average IV and the x-axis is the observation date.

5 Proposed Model

First, we must define some notation. Let Z_t be the feature vector at time t. This feature vector can be defined as $Z_t = (H_t, E_t)$ where H_t is the latent vector representation of the IVS at time t and E_t is a vector of length 10 consisting of the following exogenous variables

$$\begin{aligned}
e_{t_1} &= log(\frac{p_t}{p_{t-1}}), \text{ where } p_t \text{ is the price of SPY at time } t,\\
e_{t_2} &= \frac{1}{22}\sum_{k=t-21}^{T} avg(\Sigma_k), \text{ i.e. the 22 day moving average IV,}\\
e_{t_3} &= \frac{1}{5}\sum_{k=t-4}^{T} avg(\Sigma_k), \text{ i.e. the 5 day moving average IV,}\\
e_{t_4} &= avg(\Sigma_t), \text{ i.e. the daily average IV,}\\
e_{t_5} &= \text{US Weekly Initial Jobless Claims,}\\
e_{t_6} &= \text{Day of the week,}\\
e_{t_7} &= \text{Day of the month,}\\
e_{t_8} &= \text{SPY daily volume,}\\
e_{t_9} &= \text{VIX index level,}\\
e_{t_10} &= \text{VIX index 60 day moving average.}
\end{aligned}$$

This input vector, Z_t, will be used to train the LSTM in order to provide context for macroeconomic conditions. The overall structure of the model architecture is outlined in Figure 4.

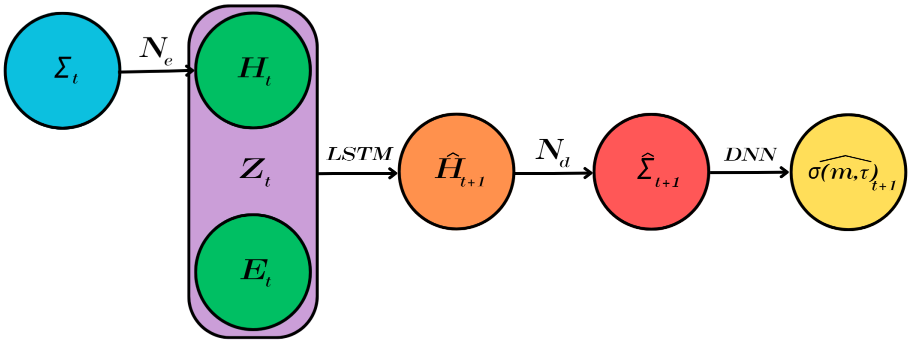


Figure 4: Path of data through prediction architecture

In Figure 4, Σ_t and $\hat{\Sigma}_{t+1}$ represent the IVS and predicted future IVS, N_e and N_d represent the encoder and decoder of the VAE, H_t and H_{t+1} represent the latent vector representations of the IVS and the predicted IVS, E_t represents the vector of exogenous variables, Z_t is the feature vector, and $\sigma(\hat{m}, \tau)_{t+1}$ represents the predicted IV for a given m and τ.

In order to better understand the reason for this architecture decision, it is important to highlight the mechanics of each of these models. We will now explore each model in depth to uncover their contribution to the overall prediction process.

5.1 VAE

Variational autoencoders (VAE) are generative models that map inputs to latent distributions for dimension reduction and then reconstruct these latent representations with minimal reconstruction error [19]. Our VAE consists of two parts. (1) An encoder N_e which takes Σ_t as its input and generates a latent vector H_t. This encoder, N_e, represents a deep neural network (DNN). The latent vector follows a multivariate normal distribution with a $\mu(\Sigma_t)$ vector and a $\sigma(\Sigma_t)$ vector. The latent vector can then be characterized by the equation $H_t = \mu(\Sigma_t) + \sigma(\Sigma_t) \ \odot \ \mathcal{N}(0, I_d)$, where I_d is the d x d identity matrix and $\odot$ is the Hadamard product. (2) A decoder, similarly modeled by a DNN and denoted N_d, takes H_t as an input and outputs ζ_t which is a representation of Σ_t with minimal reconstruction error.

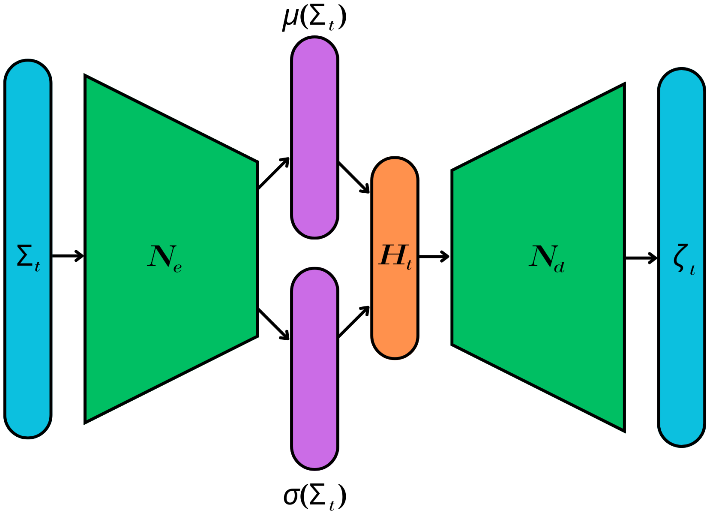


Figure 5: Path of data through VAE

In Figure 5, Σ_t and ζ_t represent the IVS and reconstructed IVS, N_e and N_d represent the encoder and decoder of the VAE, $\mu(\Sigma_t)$ and $\sigma(\Sigma_t)$ represent the multivariate normal distribution mean and standard deviation vectors of the IVS, and H_t represents the latent vector representation of the IVS.

The loss function for the VAE also consists of two parts. Namely, it contains a term for reconstruction error (RE) and a term for the Kullback-Leibler (KL) divergence. The RE term is simply the mean squared error between the original data and the interpolated data given by the equation:

$$RE = \frac{1}{B}\sum_{i=1}^{B}(\Sigma_i - \zeta_i)^2,$$

where B is the batch size of the training sample. The KL divergence provides a loss metric for ensuring that the encoded distribution is as close as possible to the normal distribution and it can be characterized as:

$$KL = \frac{1}{2}\sum_{i}(-1 - \log \sigma_i^2 + \sigma_i^2 + \mu_i^2),$$

with μ_i and σ_i being the mean and standard deviation of the i-th latent variable. Finally, the loss function is defined as:

$$\mathcal{L}(\Sigma_t; \theta_{VAE}) = RE + \beta KL.$$

5.2 LSTM

A long short-term memory neural network (LSTM) is a type of recurrent neural network (RNN) [17]. RNNs are extensions of feed-forward neural networks which include recurrent hidden states and whose outputs depend on the values of the previous state. This is ideal for our application because the future IVS is inherently adapted from the sequence of past IVSs. In order to deal with variable length intervals of useful information within the input space, the LSTM uses a special feedback structure in order to properly distinguish between long and short term dependencies. Originally proposed by Hochreiter et al [1997] and further improved upon by Gers et al [2001], LSTM uses an input gate, output gate, and forget gate to store information in a memory cell [17, 14]. These gates provide an interface to the long term memory cell state.

The input gate decides what information enters the cell state c_t given the output of the previous state (hidden state) h_{t-1}, the input x_t, and the previous cell state c_{t-1}. Similarly, the forget gate decides what information must be forgotten from the cell state c_t given the $h_{t-1}, x_t,$ and c_{t-1}. This gate structure allows LSTM to decide which information is most effective at informing future decisions. The structure of LSTM can be described as

$$
\begin{aligned}
&i_t = \sigma(W_i h_{t-1} + U_i x_t + Q_i c_{t-1} + b_i),\\
&f_t = \sigma(W_f h_{t-1} + U_f x_t + Q_f c_{t-1} + b_f),\\
&o_t = \sigma(W_o h_{t-1} + U_o x_t + Q_o c_{t-1} + b_o),\\
&g_t = \phi(W_g x_t + U_g h_{t-1} + b_g),\\
&c_t = f_t \odot c_{t-1} + i_t \odot g_t,\\
&h_t = o_t \odot \phi(c_t),
\end{aligned}
$$

where W, U, Q are weight matrices, b is an error vector, and σ, ϕ are the sigmoid and tanh function respectively. For any time t, $i_t, f_t,$ and o_t represent the input, forget, and output function at that time.

5.3 Attention

The attention mechanism used in this model, specifically temporal attention, will allow for more efficient and accurate prediction of the IVS using a technique that was adopted by our brains in order to screen high-value information from large information pools. This attention mechanism will be placed in between two adjacent LSTM hidden layers. This will allow the LSTM hidden states to assign more weight to the information deemed important by the attention layer. The key objective of the attention mechanism is to determine the current hidden state of the next LSTM layer in the sequence, s_t, which can be characterized by:

$$s_t = f(s_{t-1}, y_{t-1}, c_t),$$

where y_{t-1} is the target output of the previous step and c_t, the memory state, is:

$$c_t = \sum_{i=1}^{t} a_{ti} h_i.$$

The term a_{ti} is called the alignment score and it weights each previous hidden state by its relevance. These weights are determined by the formula:

$$a_{ti} = \frac{exp(e_{ti}(s_{t-1}, h_i))}{\sum_{j=1}^{t} exp(e_{tj}(s_{t-1}, h_i))},$$

where e_{ti} is an alignment model which takes in (s_{t-1}, h_i) as inputs. The model used is a single-layer perceptron which can be characterized by:

$$e_{ti}(s_{t-1}, h_i) = \lambda \cdot \phi(W_{eh} h_i + W_{es} s_{t-1} + b_e),$$

where W is a weight matrix, b is an error vector, ϕ is the tanh function, and λ is a bias.

Multi-headed temporal attention simply extends this notion of attention by employing multiple attention mechanisms, or heads, in parallel. Each head learns a unique relationship between each element in the input vector which provides more diverse context for the model. The method which combines these heads can be characterized by:

$$C_t = \lambda \cdot \phi(W_C \cdot [c_t^1, c_t^2, ..., c_t^k]),$$

where C_t is the combined memory state and $[c_t^1, c_t^2, ..., c_t^k]$ is a concatenation of the attention heads with k as the number of heads.

5.4 DNN for Arbitrage Free Surface Construction

Once the $\hat{\Sigma}_{t+1}$ has been predicted from our previously described model, we must now ensure that the standard IVS points of this surface are obtained with respect to static arbitrage constraints. For this task, we obtain the standard IVS using a deep neural network (DNN) which is a feedforward neural network. This DNN will take $\hat{\Sigma}_{t+1}$ as input and outputs the predicted IVS free of static arbitrage. This network will use the softplus function as the activation function for the output layer in order to ensure that each standard IVS value is non-negative and twice differentiable.

5.5 Static Arbitrage-Free Loss Function

In order to guarantee that the DNN output will be free of static arbitrage we need to satisfy 4 conditions. Our use of a softplus activation function for our output satisfies conditions 1 and 2 which are that the IVS must be positive and twice differentiable for any given point. In order to satisfy conditions 3 and 4, which are conditions prohibiting calendar arbitrage and butterfly arbitrage respectively, we must implement a loss function which will penalize the DNN if these conditions are violated. First, we must define conditions 3 and 4:

$$\ell_{cal}(m,\tau) = \sigma(m,\tau) + 2\tau\partial_\tau\sigma(m,\tau) \geq 0, \tag{2}$$

$$\ell_{but}(m,\tau) = \left(1 - \frac{mp_m}{\sigma(m,\tau)}\right)^2 - \frac{(\sigma(m,\tau)\tau p_m)^2}{4} + \tau\sigma(m,\tau)p_{mm} \geq 0, \tag{3}$$

where $p_m = \partial_m\sigma(m,\tau)$ and $p_{mm} = \partial_{mm}\sigma(m,\tau)$. Next, we must first consider a typical loss function, namely:

$$\mathcal{L}_S(\theta_\sigma) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{q_t}\sum_{i=1}^{q_t}(f(m_i,\tau_i,\hat{\Sigma}_t;\theta_\sigma) - \sigma_t(m_i,\tau_i))^2,$$

where q_t is the number of standard IVS points, 121 points in our case. In order to incorporate static arbitrage conditions into our loss function, we can take an approach inspired by Ackerer at al [2020] and Zhang at al [2022] [1, 25]. The loss functions for conditions 3 and 4 are then defined as:

$$\mathcal{L}_{C3}(\theta_\sigma) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|q_t|}\sum_{i=1}^{q_t}\max(0, -\ell_{cal}(m_i, \tau_i, \hat{\Sigma}_t; \theta_\sigma)),$$

$$\mathcal{L}_{C4}(\theta_\sigma) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|q_t|}\sum_{i=1}^{q_t}\max(0, -\ell_{but}(m_i, \tau_i, \hat{\Sigma}_t; \theta_\sigma)),$$

where $\ell_{cal}(m_i, \tau_i, \hat{\Sigma}_t; \theta_\sigma)$ and $\ell_{but}(m_i, \tau_i, \hat{\Sigma}_t; \theta_\sigma)$ are defined as 2 and 3 respectively with σ replaced with f.

Finally, our loss function for the DNN model is

$$\mathcal{L}(\theta_\sigma) = \mathcal{L}_S(\theta_\sigma) + \lambda(\mathcal{L}_{C3}(\theta_\sigma) + \mathcal{L}_{C4}(\theta_\sigma)),$$

where λ is a some non-negative hyperparameter. For our use, we will choose $\lambda = 1$ which is used in both Ackerer et al [2020] and Zhang et al [2022]. This loss function will ensure that every value used to construct the predicted standard IVS will be free from all 4 static arbitrage conditions.

6 Methods

Our data which consists of daily implied volatility observations for S&P 500 index options consists of 3409 trading days. The data was split into training, validation, and test data with a 0.7, 0.15, 0.15 ratio respectively. The training dataset spans from February 8, 2010 to August 27, 2019 with 2387 trading days. The validation dataset spans from August 28, 2019 to September 13, 2021 with 511 trading days. The test dataset spans from September 14, 2021 to September 29, 2023 with 511 trading days.

6.1 Feature Extraction Methods

We study the difference in two feature extraction methods which we described earlier:

- Sampling (SAM) uses the entire set of 121 standard IVS points from DFW interpolation (see 1), combined with the 10 chosen exogenous variables for a feature vector of length 131.
- Variational autoencoder (VAE) uses the latent representation of the standard IVS. We use a latent vector with dimension $d : 2, 5, 10, 20, 40$. This results in feature vectors of length 14, 20, 30, 50, and 90 respectively after being combined with the 10 chosen exogenous variables.

6.2 Model Training

In order to properly train the LSTM to predict the next state of the IVS, we do the following training procedures:

- We use Xavier Initialization to initialize our parameters which prevents the initial weights of the network from becoming excessively large or small [15]. This methods sets the weights of each layer, i, to follow a uniform normal distribution, namely:

  $$W^i = U\left[-\frac{1}{\sqrt{n^i}}, \frac{1}{\sqrt{n^i}}\right],$$

  where n^i is the number of neurons in the i-th layer.
- We use the Adam optimizer to perform gradient descent [18]. This will use batches, or subsets, of the data which is computationally cheaper than using all samples. The Adam optimizer uses the exponential weighted average of the gradients to guide convergence to the global minima of the loss function.

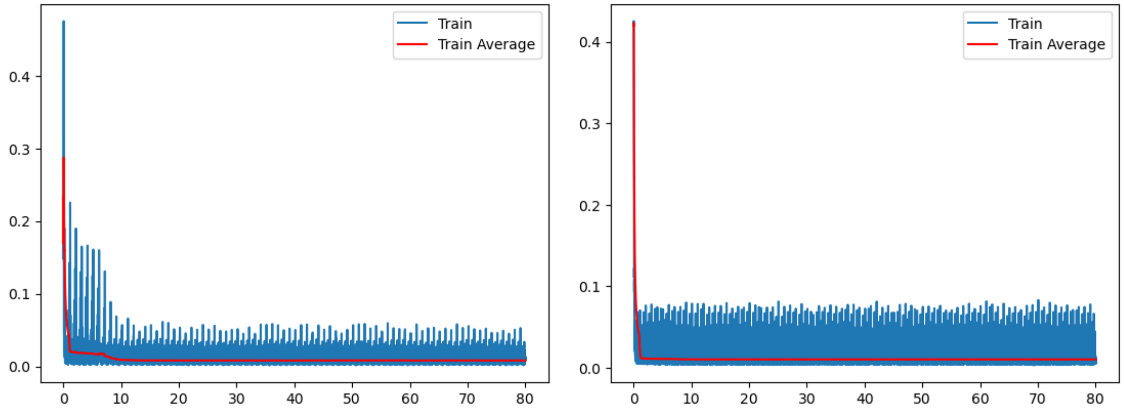
- We use a learning rate scheduler, specifically ReduceLROnPlateau, which is a technique that lowers the learning rate of the optimizer if the model stops learning. This technique was originally implemented as part of the Keras deep learning library.

- We use a dropout training mechanism to avoid overfitting our model [23, 12]. Dropout discards part of the data in the memory unit with a given probability, 0.25 was used in our case.

The training hyperparamaters for the VAE, LSTM, and DNN can be found in Table 1.

	Epochs	Batch Size	Hidden Size	Hidden Dim	Learning Rate
VAE	80	8	5	256	0.001
LSTM	80	8	8	256	Dynamic
DNN	120	1	4	128	0.001

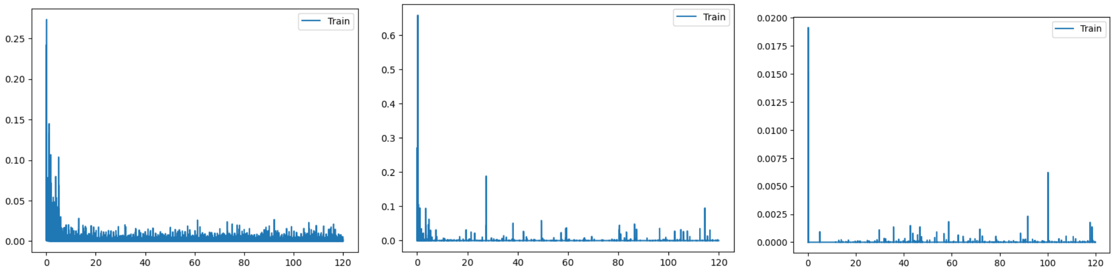
Table 1: Model Hyperparameters



(a) SAM

(b) VAE

Figure 6: The loss function for the SAM and VAE feature extraction approaches of the LSTM with 1 attention head. The x-axis is the epoch index.



(a) RMSE Loss

(b) Calendar Penalty

(c) Butterfly Penalty

Figure 7: The RMSE, calendar arbitrage, and butterfly arbitrage loss functions for the reconstruction error and static arbitrage penalties of the DNN. The x-axis is the epoch index.

Figures 6 and 7 show the progression of the loss functions of the DNN and LSTM during training. We can see a significant anomaly in the training data of the LSTM in Figure 6 (i.e., the spikes at every epoch). This is clearly caused by the flash crash of May 6th 2010 in which all major stock indices collapsed and rebounded rapidly, causing uncertainty in the market and high implied volatility levels. This data will help the model learn how to handle anomalies in the future as they are unavoidable in the real world. In Figure 7 we can see that the DNN is able to properly re-construct the IVS with very minimal reconstruction loss and only small static arbitrage violations.

7 Results

To evaluate the out of sample results, we use three common error metrics: root mean squared error (RMSE), mean absolute percentage error (MAPE), and mean directional accuracy (MDA). These three metrics, previously defined in Section 2, will be used to evaluate the error of the prediction $\sigma_t(m, \tau)$ for every t in our test data of 511 days.

Our first step is to compare the sampling (SAM) and variational autoencoder (VAE) feature extraction methods. We will assess their predictive capablities when using an LSTM and an LSTM with one head of temporal attention. This allows for a comprehensive comparison of the interaction between each feature extraction method and the attention mechanism. Results can found in Table 2.

Extraction Method	LSTM w/ Attention (1 Head)			LSTM		
	RMSE	MAPE	MDA	RMSE	MAPE	MDA
SAM	**0.0352**	**13.12%**	**64.27%**	0.0526	23.11%	59.98%
VAE (2D)	0.0548	22.14%	59.77%	0.0555	22.90%	58.51%
VAE (5D)	0.0623	24.41%	58.34%	0.0630	24.60%	57.85%
VAE (10D)	0.0528	20.82%	59.60%	0.0537	21.24%	58.87%
VAE (20D)	0.0635	21.59%	58.04%	0.0652	22.17%	58.61%
VAE (40D)	0.0628	26.23%	56.94%	0.0633	26.31%	56.84%

Table 2: Model results for all feature extraction approaches

While our results using LSTM align with studies such as Zhang et al [2021], the outcome changes significantly when attention is introduced [25]. The sampling extraction method produces similar and sometimes worse results than the VAE approach because it has trouble interpreting the high dimensionality (131) of the feature vector without attention. It is clear that the LSTM with a single head of attention improves the capabilities of either feature extraction method similar to the observations of Chen et al [2019] [8]. However, we can see that the attention mechanism disproportionately affects the sampling feature extraction approach. This result follows because the VAE reduces the dimensionality of the data and already extracts the most important features of the data. By giving the attention mechanism the entirety of the data, it can recognize more valuable relationships within the feature vector because it is using all of the data.

We now narrow our model comparison to the feature extraction methods of SAM and VAE (10D). Both models are tested using temporal attention with either 1, 8, 16, 32, or 64 heads. It is worth noting that while more attention heads can produce superior predictive power, there is a tradeoff in computational efficiency. The observed difference in training time between 1 and 64 heads of attention was on the order of 3x as long with our network size. This could be an important consideration for some low latency systems.

Attention Heads	SAM			VAE (10D)		
	RMSE	MAPE	MDA	RMSE	MAPE	MDA
1	0.0352	13.12%	64.27%	0.0528	20.82%	59.60%
8	0.0342	**12.47%**	64.78%	0.0536	21.07%	59.39%
16	**0.0339**	12.48%	65.06%	0.0540	21.11%	59.55%
32	0.0353	13.22%	**65.09%**	0.0534	21.04%	59.41%
64	0.0382	14.34%	63.71%	0.0534	21.00%	59.66%

Table 3: Performance of SAM and VAE (10D) approach when using multiple heads of temporal attention

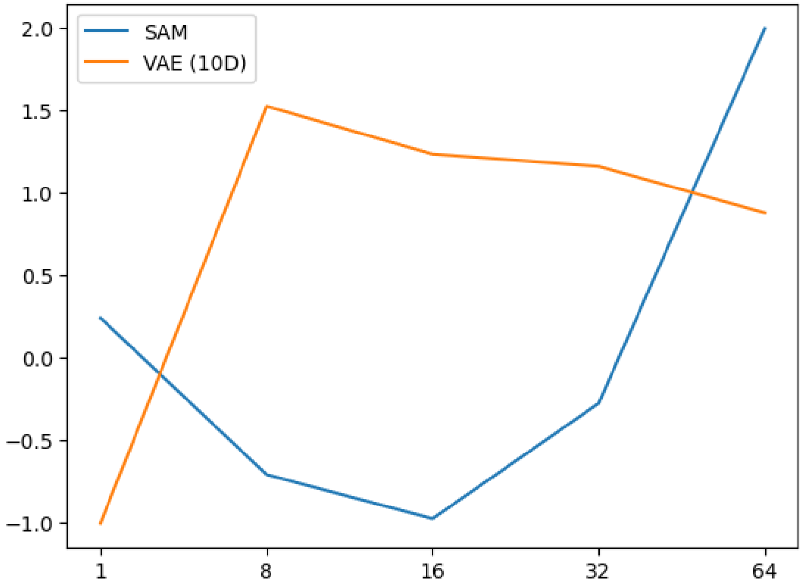


Figure 8: Composite metric for comparison of attention head count. The metric simply uses min-max scaling to normalize all metrics using the formula: $x_i^{'} = \frac{x_i - min(x)}{max(x) - min(x)}$ where x_i is the original value and $x_i^{'}$ is the normalized value. The composite metric can then be characterized as: $C = RMSE^{'} + MAPE^{'} - MDA^{'}$ where the inverse of $MDA^{'}$ is considered because higher directional accuracy is better. It is worth noting that the x-axis is the number of heads, while the y-axis is relative and completely arbitrary so the smaller the relative value, the better.

From Figure 8 we can see that the optimal choice for the superior SAM approach is 16 heads while the optimal choice for the VAE (10D) approach is just 1 head. These results follow quite clearly from the justification above, that the SAM approach provides more information rich data for the attention mechanism to draw value from.

Although, the increase in attention heads does provide diminishing marginal returns past a certain point. This can clearly be seen not only in computational efficiency, but also in predictive power when comparing the results of 16 heads of attention against 64 heads of attention. These diminishing returns are caused by overfitting which causes the model to fit to noise in the data and poorly generalize its approach to out of sample data.

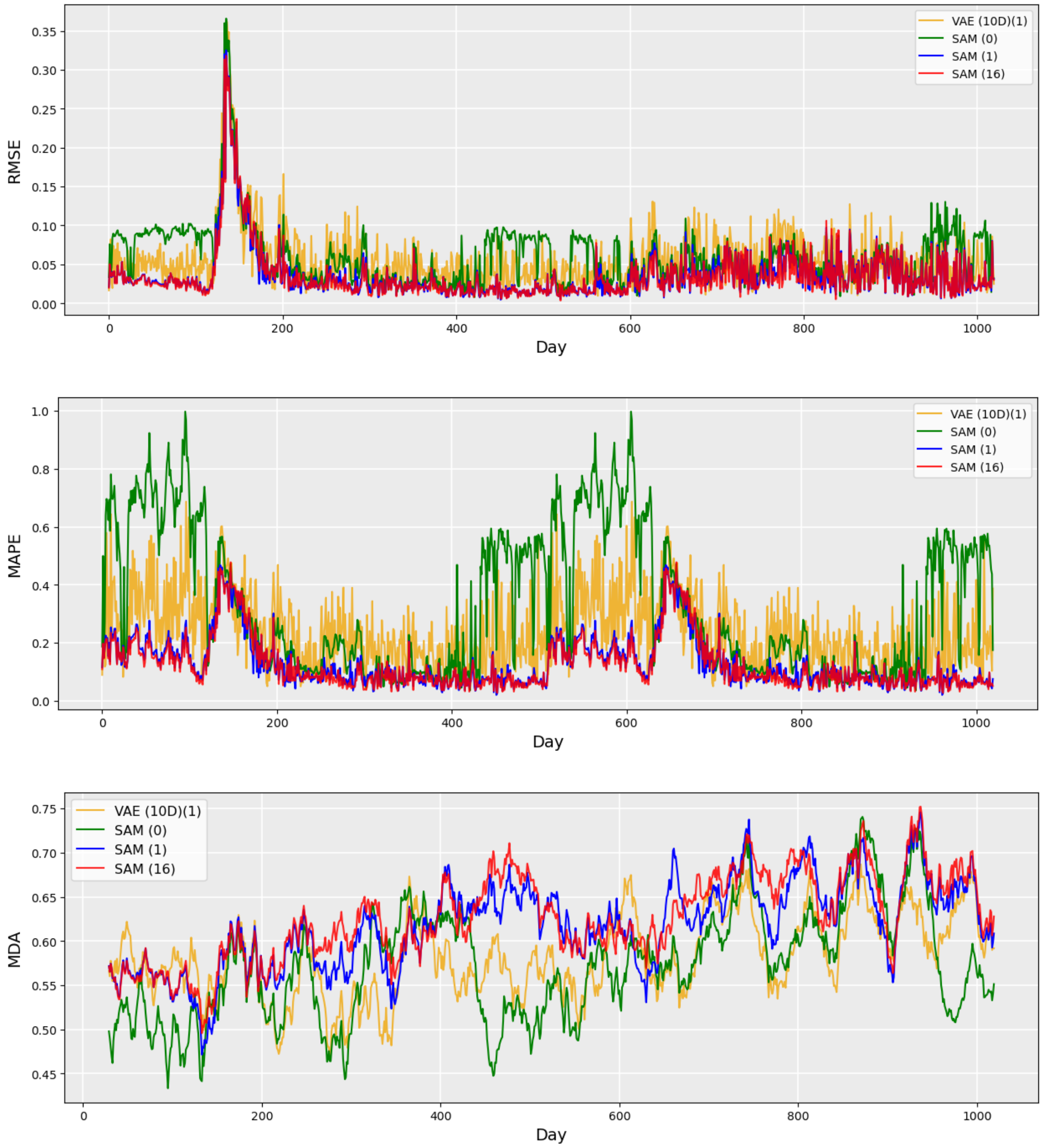

RMSE
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0
200
400
600
800
1000
Day
VAE (10D)(1)
SAM (0)
SAM (1)
SAM (16)
MAPE
0.0
0.2
0.4
0.6
0.8
1.0
0
200
400
600
800
1000
Day
VAE (10D)(1)
SAM (0)
SAM (1)
SAM (16)
MDA
0.45
0.50
0.55
0.60
0.65
0.70
0.75
0
200
400
600
800
1000
Day
VAE (10D)(1)
SAM (0)
SAM (1)
SAM (16)

Figure 9: The daily RMSE, MAPE, and 30 day moving average MDA during the validation and test period. The moving average MDA was used for interpretability as the raw data was far too noisy. Validation data is included to visually show the effects of the COVID-19 crash. (N) represents N heads of attention.

We now recall our static arbitrage constraints, and since a softplus activation function will satisfy our first 2 constraints, we must only consider the calendar and butterfly constraints defined as 2 and 3 respectively. Any value other than 0.0 reported in Table 4 is considered a static arbitrage violation and can be capitalized upon in a risk free manner. We observe that none of our feature extraction approaches violate any such constraints within our testing data and are therefore free from static arbitrage.

	SAM	VAE (2D)	VAE (5D)	VAE(10D)	VAE(20D)	VAE (40D)
$-\ell_{cal}$	0.0	0.0	0.0	0.0	0.0	0.0
$-\ell_{but}$	0.0	0.0	0.0	0.0	0.0	0.0

Table 4: Violation of static arbitrage constraints for calendar and butterfly arbitrage

8 Conclusion

The implied volatility surface is the basis of option pricing, hedging, and risk management. Its dynamic nature is inherently difficult to predict. We develop a flexible model architecture which is quite successful in predicting IVS levels the next day. This paper considers a discrete set of points to represent a standard set of IVS values and provides 10 exogenous variables to distinguish market regimes. Using the temporal attention mechanism with multiple heads, the loss function of the model converges. We find that IVS prediction is more effective with multi-head temporal attention than without and that IVS sampling outperforms a VAE feature extraction approach when using a temporal attention mechanism.

The results found in this study could be further improved in the following ways. We found that excess attention heads provide diminishing returns, however a model which prunes attention heads could successfully scale the temporal attention mechanism. Second, the use of a transformer based attention system could provide excess predictive capabilities because of its ability to capture longer range dependencies. Finally, further study could explore different exogenous variables which provide the model with more context for market sentiment.

References

[1] Damien Ackerer, Natasa Tagasovska, and Thibault Vatter. Deep Smoothing of the Implied Volatility Surface. *Advances in Neural Information Processing Systems* 33:11552-11563, 2020

[2] Yunfei Bai and Charlie X. Cai. Predicting VIX with Adaptive Machine Learning. 2023. https://dx.doi.org/10.2139/ssrn.4388132

[3] Fisher Black and Myron Scholes. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81(3):637-654, 1973

[4] Daniel Alexandre Bloch and Aruther Böök. Deep Learning Based Dynamic Implied Volatility Surface. 2021. https://dx.doi.org/10.2139/ssrn.3952842

[5] Jay Cao, Jacky Chen, John Hull, and Zissis Poulos. Deep Learning for Exotic Option Valuation. *The Journal of Financial Data Science Winter 2022*, 4(1):41-53, 2021

[6] Jay Cao, Jacky Chen, and John Hull. A Neural Network Approach to Understanding Implied Volatility Movements. *Quantitative Finance*, 20(9):1405-1413, 2020

[7] Jacky Chen, John Hull, Zissis Poulos, Haris Rasul, Andreas Veneris, and Yuntao Wu. A Variational Autoencoder Approach to Conditional Generation of Possible Future Volatility Surfaces. 2023. https://dx.doi.org/10.2139/ssrn.4628457

[8] Shengli Chen and Zili Zhang. Forecasting Implied Volatility Smile Surface via Deep Learning and Attention Mechanism. 2019. https://dx.doi.org/10.2139/ssrn.3508585

[9] Rama Cont, Josè Da Fonseca, and Valdo Durrleman. Stochastic Models of Implied Volatility Surfaces. *Economic Notes*, 31:361-377, 2002

[10] Petros Dellaportas and Aleksander Mijatović. Arbitrage-Free Prediction of the Implied Volatility Smile. 2014. https://arxiv.org/abs/1407.5528

[11] Bernard Dumas, Jeff Fleming, and Robert E. Whaley. Implied Volatility Functions: Empirical Tests. *The Journal of Finance* 53(6):2059-2106, 1998

[12] Yarin Gal and Zoubin Ghahramani. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. *Statistics*, 285-290, 2015.

[13] Jim Gatheral, Thibault Jaisson, and Mathieu Rosenbaum. Volatility is Rough. *Quantitative Finance*, 18(6):933-949, 2018

[14] Felix Gers, Douglas Eck, and Jürgen Schmidhuber. Applying LSTM to Time Series Predictable Through Time-Window Approaches. *International Conference on Artificial Neural Networks* 669-676. 2001

[15] Xavier Glorot and Yoshua Bengio. Understanding the Difficulty of Training Deep Feedforward Neural Networks. *Proceedings of Machine Learning research*, (9)249-265. 2010

[16] Steven Heston. A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options. *The Review of Financial Studies*, 6(2):327-343. 1993

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735-1780. 1997

[18] Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*, 2014

[19] Diederik Kingma and Max Welling. Auto-Encoding Variational Bayes. *International Conference on Learning Representations*, 2013

[20] Robert Merton. Theory of Rational Option Pricing. *The Bell Journal of Economics and Management Science*, 4(1):141-183, 1973

[21] Robert Merton. Option Pricing When Underlying Stock Returns are Discontinuous. *Journal of Financial Economics*, 3(1-2):125-144, 1976

[22] Michael Roper. Arbitrage Free Implied Volatility Surfaces. *The University of Sydney*, 2010

[23] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(1):1929-1958, 2014

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *Neural Information Processing Systems Conference*. 2017

[25] Wenyong Zhang, Lingfei Li, and Gongqiu Zhang. A Two-Step Framework for Arbitrage-Free Prediction of the Implied Volatility Surface. *Quantitative Finance*, 23(1):21-34, 2022

[26] Yanhui Zhao and Paul Borochin. The Economic Value of Equity Implied Volatility Forecasting with Machine Learning. 2023. https://dx.doi.org/10.2139/ssrn.4495087

[27] Yu Zheng, Yongxin Yang, and Bowei Chen. Incorporating Prior Financial Domain Knowledge into Neural Networks for Implied Volatility Surface Prediction. *ACM SIGKDD Conference on Knowledge Discovery*, 2021