A Work Project, presented as part of the requirements for the Award of a Master's degree in Finance from the Nova School of Business and Economics.

PREDICTING THE IMPLIED VOLATILITY SURFACE VIA DEEP LEARNING

NICOLAS YONG JOON LEE

Work project carried out under the supervision of:

Paulo M. M. Rodrigues

17-12-2024

Abstract: This thesis explores deep learning techniques for predicting the implied volatility surface (IVS), a critical component in options pricing and risk management. By applying ConvLSTM and Self-Attention mechanisms, the study evaluates their ability to capture spatial and temporal patterns across strikes and maturities. Results show that grid-based ConvLSTM excels in short-term forecasting, while Self-Attention enhances long-term accuracy by capturing global dependencies. The models were retrained and evaluated under volatile regimes, including the COVID-19 crash, testing their robustness in extreme market conditions. The findings contribute to improved IV surface predictions, benefiting strategies like volatility arbitrage and dynamic hedging.

Keywords: Implied Volatility Surface, ConvLSTM, Stochastic Volatility Inspired, Volatility Forecasting, Deep Learning, Self-Attention Mechanism, Neural Networks, Volatility Surface, IVS Calibration, Options Pricing

Acknowledgement: I would like to thank Professor Paulo M. M. Rodrigues for his guidance and teaching, which inspired my academic journey and encouraged me to pursue this research path.

This work used infrastructure and resources funded by Fundação para a Ciência e a Tecnologia (UID/ECO/00124/2013, UID/ECO/00124/2019 and Social Sciences DataLab, Project 22209), POR Lisboa (LISBOA-01-0145-FEDER-007722 and Social Sciences DataLab, Project 22209) and POR Norte (Social Sciences DataLab, Project 22209).

Introduction

The volatility of an asset measures the degree of variation of its returns. There are two measures of volatility: the historical volatility, which is the realized volatility at a certain point in time and observed from historical data, and the implied volatility which reflects market participants expectations regarding future volatility. The latter is derived inversely using the current option market price and the Black-Scholes equation. It is the volatility value that makes a certain option price with strike K and maturity T using the BS formula equal to the observed price in the market.

Modeling the volatility is important for several applications in finance such as option pricing, hedging, risk and portfolio management. In this context, one of the most classical models for option pricing is the Black-Scholes (BSM). However, one key limitation is its assumption that the volatility is constant, which is not observed in practice. In the market, the implied volatility varies considerably with the strike values and time to maturity. One way to clearly see that, is by visualizing the implied volatilities of options with different maturities and exercise prices in a three-dimensional plot called the implied volatility surface (IVS). One can see that the shape of the surface is not flat, meaning that the volatility for all strikes and maturities is not the same. Studies have found that the implied volatility varies systematically with the strike price and time until expiration. Moreover, the surface not only presents different shapes but also varies throughout time. The distortion between the implied volatility and strike values has been named the Volatility Smile, which shows that options with strike prices far from the current underlying asset's price, therefore out-of-the-money and in-the-money options have higher volatilities than at-the-money options. Another property that has been observed is that implied volatilities typically increase with time to maturity due to the larger degree of uncertainty regarding future

price movement, and this refers to the Volatility Term Structure. When these two properties are combined, it is possible to obtain the 3D figure of the volatility surface.

In the context of modeling the volatility surface, some challenges can be mentioned, such as its high-dimensional and non-linear nature, since the surface is defined across a range of strike prices (moneyness) and maturities that can behave in a non linear manner. In that sense, traditional models might fail to capture the complex and dynamic nature of the volatility surface. Recent advances in deep learning, however, have shown to be capable of capturing the complex and non-linear patterns present in high dimensional data such as volatility surfaces. The deep learning models that will be the focus of this study are Convolutional Long Short-Term Memory (ConvLSTM) networks and Self-Attention mechanisms. ConvLSTM is capable of modeling spatio-temporal dependencies, such as the relationships between different strikes at different maturities, as well as capturing how volatility changes over time. Incorporating Self-Attention further enhances this capability by allowing the model to focus on key spatial and temporal features, enabling it to learn long-term dependencies and interactions more effectively across the entire volatility surface. This study aims to evaluate the performance of these different models, and to provide insights into how the volatility surface can be more accurately modeled and predicted using deep learning for applications in options trading and risk management.

Literature Review

1. Option Pricing and Implied Volatility

The Black-Scholes model (BSM) significantly influenced finance by providing a theoretical framework for option pricing. The model assumes that the underlying asset price follows a Geometric Brownian Motion, represented by the stochastic differential equation:

$dS_t = \mu S_t dt + \sigma S_t dW_t$, where μ is the drift rate (expected return), σ is the volatility, and Wt is a Wiener process (brownian motion). Using Itô's Lemma, the price of a European call option can be expressed through a partial differential equation, commonly referred to as the Black-Scholes PDE. Solving this equation with the appropriate boundary conditions, corresponding to the payoff functions of European call and put options: max($St - K$, 0) and max($K - St$, 0), respectively, where K is the strike price - yields the closed-form solution for the European options prices. Detailed formulas are provided in Appendix A.

Implied volatility is the volatility obtained by using the market price of the option and inverting the Black-Scholes formula, i.e it is the volatility value (σ) that, when plugged into the Black-Scholes formula, makes the theoretical option price equal to the observed market price of the option, i.e., $\sigma(K, t) = BS^{-1}(C, S, K, t, r)$.

The Black-Scholes formula does not provide a direct solution for volatility, but its inverse function can be solved numerically. According to Orlando and Taglialatela (2017) the optimal method to compute the implied volatility is the Newton-Raphson method, which is an iterative numerical technique to find the roots of a function and that converges quickly depending on how close the initial guess is to the solution. The function is given by: $f(\sigma) = C_{Market} - C_{BS}(\sigma)$, where C_{Market} is the observed option price and $C_{BS}(\sigma)$ is the Black-Scholes theoretical option price for a given σ. Starting with an initial guess, the volatility is updated iteratively until the difference between observed and theoretical prices becomes small enough.

When the implied volatility is calculated for options with different exercise prices, one can obtain the volatility smile, which represents the implied volatility as a function of the option

strike price. If the different maturities are further considered, a three-dimensional volatility smile surface will be formed. The volatility surface is used by traders and analysts to gain deeper insights into option pricing, risk management, and potential trading strategies.

The calculated IV surface can appear uneven due to the discrete nature of strike prices and maturities combined with market microstructure factors such as differences in liquidity, bid-ask spreads, and noise. The discreteness causes the surface to have structural gaps, while the other factors might introduce local irregularities at specific strikes and maturities. Therefore, the surface must be smoothed in a way that ensures it does not introduce arbitrage opportunities. The non-arbitrage conditions include convexity with respect to strikes, monotonicity with respect to maturity, put-call parity, no butterfly spread arbitrage, and no calendar spread arbitrage. Detailed equations are provided in Appendix B.

Returning to the Black-Scholes model, one of its limitations is the assumption of constant volatility, which fails to capture the smile and skew present in the volatility and thus, might overestimate or underestimate volatilities of different options. Hull and White (1987) found that the BSM pricing model tends to underprice deep in-the-money (ITM) and deep out-of-the-money (OTM) options, while it tends to overprice at-the-money (ATM) options and that these effects increase with the time to maturity. To address this, Dupire (1994) proposed the local volatility model, which reflects how volatility varies across strikes and maturities.

2. Volatility Forecasting

The GARCH (Generalized Autoregressive Conditional Heteroskedasticity) model, introduced by Bollerslev (1986), is one of the most widely used time-series models for forecasting realized volatility due to its ability to capture volatility clustering effectively. Its formula is given by:

$$\sigma_t^2 = \omega + \sum_{t=1}^{q} \alpha_i \varepsilon_{t-1}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2$$

Where, σ_t^2 is the conditional variance, ε_{t-1}^2 the previous errors (shocks) squared, and σ_{t-j}^2 are the past conditional variances. Several studies have empirically tested the performance of different GARCH models and found that GARCH(1,1) often performs as well as or better than more complex models. Hanse and Lunde (2005) conducted an extensive comparison of 330 volatility models and found no evidence that other more sophisticated models consistently outperformed the GARCH(1,1) for out-of-sample volatility forecasting.

However, when it comes to forecasting the IV surface, the GARCH model faces several inherent limitations. GARCH is designed to forecast realized volatility over time for a single series (such as the underlying asset returns), making it effective for predicting temporal volatility patterns but unable to capture the cross-sectional variations across different strikes and maturities. This means GARCH can provide a single-path forecast of volatility for a specific strike (typically the ATM option), but it is not capable of modeling the entire volatility surface, which includes patterns like the volatility smile and skew.

3. Heston Model

The Heston model (1993), was developed to overcome some limitations of the Black-Scholes model by incorporating stochastic volatility. Unlike Black-Scholes, which assumes constant volatility, the Heston model allows the variance of the underlying asset to follow a mean-reverting stochastic process, making it more flexible and capable of capturing real-world market behaviors. It describes the dynamics of the underlying asset price S_t and its variance ν_t

through stochastic differential equations, provided in Appendix C. The Heston model is thus also widely used for pricing options since it can model complex option dynamics more realistically than the Black-Scholes model. The solution to the Heston model can be derived using a partial differential equation (PDE) or through semi-analytical approaches, such as the Fourier transform, to price options accurately.

On the one hand, the volatility smile is observed when plotting the implied volatilities of options with the same maturity but different strikes, forming a U-shaped curve. It indicates that deep ITM and OTM options have higher implied volatilities than ATM options. This pattern can be attributed to several hypotheses. One accepted explanation is the extreme price movement hypothesis, which suggests that traders price in the possibility of large price swings for deep ITM and OTM options, leading to higher implied volatilities at these strikes. Studies by Bakshi, Cao, and Chen (1997) showed that models incorporating stochastic volatility and jumps can better capture these extreme price movements, resulting in a U-shaped smile pattern. Additionally, the risk perception hypothesis argues that the smile reflects traders' concerns about large negative shocks in equity markets, leading to a higher demand for deep OTM puts as insurance against sudden crashes (Bates, 1991). The Heston model captures this smile effect due to the mean-reverting stochastic process for variance (ν_t), which allows the model to adjust the variance dynamically in response to price movements, and therefore the implied volatilities across different strike prices.

On the other hand, the volatility skew represents the asymmetry in the implied volatility smile (curve). For equity options, it is typically observed as a downward-sloping curve because out-of-the-money puts (lower strikes) often have higher implied volatilities than

out-of-the-money calls (higher strikes). This is known as the "put skew" and reflects the market's concern about large downward movements in asset prices, as traders seek protection through puts, causing their volatility to increase relative to calls.

The Heston model captures this volatility skew through the correlation term between the Brownian motions governing the underlying asset price (dW_t^S) and variance ($dW_t^\nu \nu$). When this correlation coefficient (ρ) is negative, a drop in the asset price typically causes an increase in volatility (often called the leverage effect). This relationship causes downward-sloping skews in implied volatilities because the model reflects that market participants anticipate higher future volatility when asset prices fall, which aligns with real-world observations. While the Heston model captures smile and skew effects for individual options, constructing a complete IVS across multiple strikes and maturities requires a model specifically designed for surface fitting.

4. Stochastic Volatility Inspired (SVI)

The Stochastic Volatility Inspired (SVI) model was originally developed at Merrill Lynch in 1999. Later, Gatheral (2004) formalized the model's implementation and calibration procedures, making it more widely adopted in both industry and academia. The model parametrizes the implied volatility smile using a small set of interpretable parameters. In the raw SVI parameterization of total implied variances:

$$\omega(k; \chi_R) = a + b\left\{\rho\,(k - m) + \sqrt{(k - m)^2 + \sigma^2}\right\}$$

where $\omega(k; \chi_R)$ is the implied variance for a given set of parameters $\chi_R = \{a, b, \rho, m, \sigma\}$ and k, which is the log-forward moneyness defined as $k = ln(\frac{K_t}{F(t,T)})$, where $F(t, T)$ is the forward

price at time t with maturity T. Additionally, the implied total variance is given by the formula $\omega(k) = (IV)^2 \times T$, where T is the time to maturity in years. The basic constraints are: $a \in \mathbb{R}$, $b \geq 0$, $|\rho| < 1$, $m \in \mathbb{R}$, $\sigma > 0$, and $a + b\sigma\sqrt{1-\rho^2} \geq 0$, which ensure that the implied variance is positive for all $k \in \mathbb{R}$. The description of the parameters is provided in Appendix D. Furthermore, it is important to ensure the appropriate behavior of the total implied variance at extreme strikes. Roger Lee's moment condition (2004) establishes a boundary for the growth rate of the implied variance at very high or low strikes. This condition translates into the total variance being bounded by a linear function at extreme strikes and having a maximum slope of two for the wings. The detailed asymptotes (wings) behavior of the total implied variance is provided in Appendix E.

The absence of arbitrage in the volatility surface requires meeting two conditions: (1) the surface must not exhibit calendar spread arbitrage, and (2) each time-to-maturity slice must be free of butterfly spread arbitrage. Calendar spread arbitrage is related to the monotonicity of option prices with respect to maturity. Specifically, for the same strike (i.e., at constant log-moneyness), option prices should be non-decreasing with time-to-maturity. That is, $\frac{\partial C_{BS}(k,\omega(k,t))}{\partial t} \geq 0$, where C_{BS} is the option price, $\omega(k,t)$ is the total implied variance for a given log-moneyness k and time to maturity t. Consequently, $\omega(k,t)$ must be strictly increasing with respect to t (i.e. $\partial_t \omega(k,t) \geq 0$). The absence of calendar arbitrage can be seen if there are no line crosses (intersections) of the total variance curves for different maturities. When the maturity increases, the SVI slice will shift upwards. And ensuring the absence of butterfly spread requires that the price of the butterfly spread is positive and the option price curve is convex. This requires that

$\frac{\partial^2 C_{BS}(k,\omega(k,t))}{\partial^2 k} \geq 0$, alternatively $\frac{\partial^2 \omega(k,t))}{\partial^2 k} \geq 0$. Ferhati (2022) proposed a robust multi-slices SVI calibration technique using a Sequential Least Squares Quadratic Programming (SLSQP) optimizer. The method simultaneously calibrates multiple time slices while preventing both arbitrages (calendar and butterfly). To ensure a more accurate fit in the most liquid regions, particularly around at-the-money (ATM) zone, Ferhati implemented a weighted calibration scheme. This approach prioritizes the ATM region by assigning higher weights while giving lower weights to deep out-of-the-money (OTM) and in-the-money (ITM) strikes.

Block and Book (2021) propose a method to perform multi-step forecasting of the SVI to predict the evolution of the IV surface using neural networks architecture. In particular, a ConvLSTM model is trained using historical SVI parameters from past IV surfaces and forward prices history for a fixed grid of maturities, which enable the model to learn the spatial and temporal relationships between strikes and time-to-maturities. After predicting the SVI parameters, the entire predicted IV surface is constructed for each maturity and strike, ensuring a smooth and arbitrage free surface. Their model was tested against a naive benchmark and demonstrated better performance on long term forecasts for short to medium range maturities.

5. Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM) neural network first introduced by (Hochreiter and Schmidhuber, 1997) is a type of recurrent neural network (RNN) which is used in learning problems involving sequential data. RNN (also known as “feedback neural networks”) contain feedback hidden layers (recurrent hidden states), whose output in each state depends on the values from all the previous states. In many prediction problems involving RNN, the time

interval of the relevant information can vary significantly. Some of the challenges faced by RNN that gave rise to LSTM were: when the time interval of the relevant information is short, the results of RNN can still be affected by historical information that is not very relevant. Whereas, when the important interval is long RNN might struggle to retain useful information across many time steps and therefore fail to capture key long term dependencies. This happens because during training, gradients (partial derivatives of the loss function) are multiplied at each time step through backpropagation, and if the values are smaller than 1, they can shrink exponentially. This results in the problem of vanishing gradients, which happens when they become so small that weights of the early layers are barely updated, making it difficult for the network to learn long-term patterns from early time steps. The architecture of LSTM was designed to address these challenges. LSTM introduced memory cells, which store information from previous time steps and a set of gates (input, forget, and output) that control the flow of information, allowing them to selectively retain (discard) relevant (irrelevant) information.

The forget gate (Gers et al, 2000) decides what fraction of the previous cell state should be kept. The input gate determines what new information should be added to the cell state. The output gate decides what should be the output of the hidden state. Specifically,

$$f_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f), \qquad G_t = tanh(X_t W_{xg} + H_{t-1} W_{xg} + b_g),$$

$$i_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i), \qquad C_t = f_t \odot C_{t-1} + i_t \odot G_t,$$

$$o_i = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_0), \qquad h_t = o_t \odot tanh(C_t)$$

where f_t, i_t, o_i are the forget gate, input gate and output gate at time t; W are weight matrices and b the bias terms, respectively; $\odot$ represents element-wise multiplication; σ and $tanh$ are sigmoid and hyperbolic tangent activation functions; C_t is the cell state; and h_t is the output of the LSTM unit at time t. The main limitation of using standard LSTMs for IV surface forecasting is that while they capture temporal dependencies, they lack the ability to model spatial relationships (in this case between strikes and maturities). This can result in a loss of crucial information that is necessary to accurately predict the IVS. Nevertheless, Chen and Zhang (2020) incorporated an attention mechanism into LSTM models for one-step-ahead IVS prediction.

6. Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) were initially introduced by LeCun et al. (1989) and became widely known after the success of AlexNet by Krizhevsky et al. (2012) in the ImageNet competition. The convolutional layer is the core component of a CNN, which is responsible for learning local patterns by applying a set of convolutional filters (kernels) to the input data. Each filter generates a feature map as follows:

$$Y[i,j] = \sum_m \sum_n X[i + m,\ j + n] \cdot K[m,n] + b$$

where, Y is the output feature map, X is the input matrix, K is the convolutional filter (kernel), and b is the bias term. After each convolution, a non-linear activation function is applied, such as the ReLU (Rectified Linear Unit): $f(x) = max(0, x)$.

To reduce the dimensions of feature maps, pooling layers are used. Pooling condenses the feature map while preserving the relevant information. By reducing the dimensionality, the process helps

to reduce overfitting and improve computational efficiency. The main types of pooling are max and average pooling:

$$Y[i,j] = max\{X[i + m, J + n]\} \quad \text{(Max Pooling);}$$

$$Y[i,j] = \frac{1}{mn}\sum_{m}\sum_{n} X[i + m, j + n] \quad \text{(Average Pooling).}$$

After feature extraction through multiple convolutional and pooling layers, in which early layers capture basic patterns and deeper layers detect complex structures, the resulting multi-dimensional feature maps are flattened into a one dimensional vector. This transformation is necessary to feed the data into the fully connected layers. These layers combine the learned features to make the final prediction. The primary strength of CNNs lies in their ability to capture spatial patterns and relationships between features. However, CNNs do not have a mechanism to capture temporal dependencies in sequential data.

7. ConvLSTM

ConvLSTM was first proposed by Shi et al. (2015) to overcome some of the challenges associated with traditional LSTMs in modeling spatiotemporal data. By extending the LSTM and integrating convolutional structures into both the input-to-state and state-to-state transitions, ConvLSTM was designed to capture spatial dependencies while maintaining the temporal dynamics across time steps. This design was initially applied to nowcasting precipitation and has demonstrated superior performance against standard LSTMs in various spatiotemporal datasets.

The convolutional filters (kernels) within ConvLSTM enable the model to learn local patterns, such as spatial relationships between neighboring points. When applied to IVS modeling, these

filters can detect subtle patterns in IV dynamics across different strikes and maturities. Furthermore, as Orosi (2012) points out, many practical methods for modeling and interpolating the Implied Volatility Surface (IVS) rely on non-linear techniques, such as fitting quadratic functions, cubic splines, or polynomials. Non-linearity is introduced through activation functions, such as ReLU, which are applied after the convolutional layer. This makes the ConvLSTM model a suitable choice for capturing the complex patterns of the IVS.

ConvLSTM can be particularly effective for IVS modeling due to its ability to capture both spatial and temporal dependencies in the implied volatility surface. Each strike-maturity point in the IVS is influenced by its neighboring strikes and maturities, forming local patterns such as volatility smiles and skews. The convolutional filters of ConvLSTM can learn these local spatial features, while the LSTM component can learn the temporal evolution of volatility. Bloch and Book (2021) explored this model to capture the evolution of SVI parameters for IVS prediction. Moreover, Medvedev and Wang (2021) showed the effectiveness of ConvLSTM for multistep IVS forecasting, outperforming traditional time series models.

8. Self-Attention

The Self-Attention mechanism was proposed by Vaswani et al. (2017) in the context of natural language processing with the Transformer model, however the mechanism has gained popularity in a wide range of deep learning applications, including time-series analysis and image processing. Self-Attention redefines how models capture dependencies between input elements by allowing each element in a sequence to directly attend to every other element. This overcomes some of the limitations faced by recurrent structures like LSTMs, which rely on sequential processing. At the core of Self-Attention is a set of transformations, the Query, Key, and Value

matrices that allow the model to assign different levels of importance to each input element by using weights derived from measuring the relationships and dependencies between them, i.e.,

$$Q = XW_Q, \; where \; W_Q \in \mathbb{R}^{d \times d_q} \quad \text{(Query Matrix);}$$

$$K = XW_K, \; where \; W_K \in \mathbb{R}^{d \times d_k} \quad \text{(Key Matrix);}$$

$$V = XW_V, \; where \; W_V \in \mathbb{R}^{d \times v} \quad \text{(Value Matrix);}$$

where X is the input sequence and each weight matrix W is randomly initialized and updated through backpropagation during training. After defining the Query, Key, Value matrices, the score between each pair of elements is determined using the dot product of the corresponding query and key vectors. This measures the degree of similarity or relevance of the elements to each other. For the input elements i and j, the attention score is defined as:

$$Score(Q_i, K_j) = Q_i \cdot K_j^T,$$

where Q_i is the query vector for element i, and K_j is the key vector for element j. To deal with the high dimensionality of the vector, the scores are typically scaled by $\frac{1}{\sqrt{d_k}}$,

$$Scaled \; Score(Q_i, K_j) = \frac{Q_i \cdot K_j^T}{\sqrt{d_k}}$$

where d_k is the key dimension. The scaled score goes through a softmax activation function to convert them into a probability distribution to determine the weights assigned to each input element, representing the relative importance of each input in the sequence.

$$\alpha_{i,j} = softmax\left(\frac{Q_i \cdot K_j^T}{\sqrt{d_k}}\right).$$

The weight $\alpha_{i,j}$ indicates how much focus is given to element j when updating the representation of element i. The output of the Self-Attention mechanism is a weighted sum of the value vectors, where each V_j is weighted by $\alpha_{i,j}$.

$$Self\ Attention(Q_i, K, V) = \sum_{j=1}^{n} \alpha_{i,j} V_j.$$

This output is a new contextualized representation of the input element i. However, in practice a single Self-Attention operation may not be enough to capture all the interactions present in the data. Multi-Head Self Attention can address this issue by employing multiple Self-Attention heads, where each head is assigned random initialized weights and computes its own set of Q, K, V matrices, resulting in different sets of attention outputs. The equations and final transformation details are provided in Appendix F**.**

The Volatility Surface

Data

This study focused on the modeling and prediction of the S&P 500 volatility surface since it is one of the most popular and actively traded derivatives offered by the Chicago Board Options Exchange (CBOE). While SPY(ETF) options are American-style, S&P 500 Index (SPX) options are European-style. The data comprises all option chains for the SPX for the period between 2010-01-04 and 2023-12-29. The main data sources used were OptionsDX, which provides end-of-the-day data for call and put option chains including values for the implied volatilities,

prices, strikes, days until expiration and also the greeks. Additionally, SPX historical data was obtained from Bloomberg. Furthermore, Treasury Constant Maturity rates were obtained from the Federal Reserve's website, providing daily risk-free interest rates across various maturities.

The market data for implied volatilities is not uniformly distributed across moneyness and maturity levels, varying across days. Therefore, to accurately model and predict the IV surface, a fixed grid of strikes and maturities was constructed. Using a grid allows for the interpolation of the IVS at consistent moneyness and maturity levels, which is crucial for the models that require structured input formats. It also facilitates comparison of IV surfaces between different dates. The grid used includes 40 fixed strikes expressed in log-forward moneyness in the range [-0.1, 0.1] and the following fixed 17 maturities [7, 14, 21, 30, 45, 60, 75, 90, 120, 150, 180, 210, 240, 270, 300, 330, 365] expressed in days. This choice for the grid tried to balance computational efficiency with applicability, focusing on regions of the surface where options are most liquid and heavily traded. To achieve this, maturities span from one week to one year, avoiding long-dated, less liquid contracts. Similarly, moneyness was limited to the range of -0.1 to 0.1, where the data is denser and less noisy. Outside this range, options generally also exhibit lower liquidity and IV values tend to be more sparse and noisy, which could potentially impact model performance. Additionally, log-forward moneyness calculated as $ln(\frac{K}{F})$, where K is the exercise price and F is the forward price of the underlying index, was chosen over simple moneyness ($\frac{K}{F}$) due to some advantages. For instance, log-forward moneyness naturally incorporates the time value of money by accounting for interest rates, ensuring a consistent scale across options with different maturities. Its use also aligns with the SVI model, where it is a standard input variable due to its ability to capture the symmetry of implied volatility around at-the-money strikes, which simplifies the curves fitting process. To calculate forward prices in each date for the

different maturities, the risk-free rates were estimated using the Nelson-Siegel yield curve fitting, due to its flexibility and ability to capture the term structure of interest rates. The model and its equation are provided in Appendix G. The interpolated risk-free rates were then used in the forward price formula: $F = Se^{rT}$. To generate the surface, made of 680 implied volatility values given as a 40x17 matrix per day, interpolation was applied. Smooth Bivariate Spline was employed, which is a two-dimensional extension of cubic spline interpolation. The method uses cubic spline fitting across the data grid in both the moneyness and maturity axis, ensuring a smooth representation of the surface. In case, the number of valid data points was insufficient (≤3), nearest-neighbor interpolation was used as a fallback. Examples of the interpolated surfaces are shown in Figure 1.

Models

1. SVI with ConvLSTM

The SVI with ConvLSTM model combines the Stochastic Volatility Inspired parameterization with a spatiotemporal deep learning architecture to predict implied volatility surfaces. For each trading date, the implied volatilities (IV) were converted to implied total variances $\omega(k)$. The SVI parameters for each maturity were then fitted through a two stage optimization. In the first stage, the 'trust-constr' method was used due to its robustness and ability to handle nonlinear constraints effectively. This stage enforced the following constraints: 1) Non-negativity of the implied total variance, $a + b\sigma\sqrt{1 - \rho^2} \geq 0$ and 2) Right-wing condition, $2 - b(1 + \rho) \geq 0$, which ensures $\omega(k)$ does not grow too fast in the positive log-forward moneyness region (call wing) that could result in unrealistic pricing. In the second stage, the values found in the first step were used as initial parameters to improve convergence. The method used was 'SLSQP' due

to its efficiency in handling more computationally demanding constraints. In this stage the additional no-arbitrage conditions were introduced: 1) Butterfly arbitrage, ensuring the total variance function is convex along the moneyness and that the risk neutral probability density function remains non negative is defined by: $g(k) \geq 0$, where $g(k) = \left(1 - \frac{k \cdot \omega'(k)}{2 \cdot \omega(k)}\right)^2 - \frac{(\omega'(k))^2}{4} \cdot \left(\frac{1}{\omega(k)} + \frac{1}{4}\right) + \frac{\omega''(k)}{2}$. 2) Calendar arbitrage, preventing shorter maturities from having higher total variance than longer ones, which would imply decreasing uncertainty over time. This condition given by $\omega(k, T_1) \leq \omega(k, T_2)$ for $T_1 < T_2$ was checked across maturities, allowing up to 4 crosses between SVI slices as suggested by Gatheral (2012). In both stages, the objective function was to minimize the squared error between the SVI total variances and the actual values obtained from the interpolated data. Additionally, weighting was applied to the loss function to emphasize the strikes near the at-the-money (ATM) region, as these are typically more liquid. The resulting SVI fits across maturities are shown in Figure 2. The reconstructed implied volatility surface, compared to the original surface, demonstrates the accuracy of the SVI fit and is illustrated in Figure 3.

After fitting the SVI parameters for each date and maturity using the optimization process described above, the resulting data contained the five SVI parameters (a, b, ρ, m, σ) for each of the 17 maturities across all available dates in the dataset and is shown in Figure 4. This parameter's history formed the basis for the training of the ConvLSTM. To ensure stable training and efficient convergence, a separate MinMaxScaler was applied to each parameter to scale them to the range between 0 and 1. The ConvLSTM model was designed to predict the evolution of the SVI parameters for multiple steps ahead, specifically t+1, t+5, t+10, t+20, t+30. These forecasts were generated iteratively, where the model's output for time t+n was fed as input to

predict the subsequent time step t+n+1. This autoregressive approach enabled the model to capture temporal dependencies over multiple forecast horizons. In this context, each prediction step represents a trading day, reflecting the structure of the dataset. The model was trained using the past 10 days of data (10 lags) as input to predict the SVI parameters for the next time step. The architecture consisted of two stacked ConvLSTM layers, each with 64 filters, to capture the temporal dynamics of the SVI parameters across multiple maturities. Each ConvLSTM layer employed a kernel size of 1×1, which simplified the convolutional operations to focus on temporal relationships rather than spatial dependencies. This choice aligned with the findings of Bloch and Book (2021), who showed that increasing the kernel size to capture spatiotemporal interactions did not yield performance improvements. To prevent overfitting, dropout and recurrent dropout (rates: 0.05) were applied within each ConvLSTM layer. Dropout randomly ignores a fraction of input connections during training, while recurrent dropout does the same for hidden-to-hidden connections, improving generalization in time-series data. Additionally, batch normalization was applied after each ConvLSTM layer to stabilize training and improve convergence. A final dropout layer (rate: 0.1) was applied to regularize the model's outputs further. The output tensor from the ConvLSTM layers passed through a 1×1 convolutional layer with a linear activation function. This layer ensured the predicted tensor retained the same shape as the input, producing the forecasted SVI parameters for all maturities. The model was trained using the Mean Squared Error (MSE) loss function, which is appropriate for regression tasks such as predicting the SVI parameters, with the Nadam optimizer and a learning rate of 0.001. Training was performed on batches of size 64 for up to 200 epochs, with early stopping and learning rate reduction to prevent overfitting and reduce training duration. A validation split of

20% was chosen, where the last 20% of the training data was used for validation, and the training process automatically stopped when no further improvements in validation loss were observed.

After predicting the SVI parameters for each maturity, the implied volatility surface was reconstructed by first calculating the implied total variance along the fixed strikes (log-moneyness) using the SVI formula. These total variances were then converted into implied volatilities by dividing each variance by its corresponding time to maturity (T) and taking the square root: $IV(k,\ T)\ = \sqrt{\frac{w(k)}{T}}$.

2. ConvLSTM (Grid-Based)

While the SVI with ConvLSTM approach focuses on predicting the evolution of SVI parameters, the ConvLSTM (Grid-Based) model takes a fundamentally different approach by directly predicting the entire surface. This method is more flexible, as it does not rely on a specific functional form for the IVS. Additionally, the grid-based model naturally aligns with the strengths of ConvLSTM, which excels at modeling sequences of grid-structured data. By directly modeling the IVS, the grid-based approach captures patterns across moneyness and maturities better, offering a richer level of detail and flexibility that parameterized methods might not be able to achieve. In this implementation, the IVS was represented as a two-dimensional grid with 17 maturities and 40 moneyness levels for each date, with the data normalized using a MinMaxScaler to ensure training stability and convergence. The architecture of the ConvLSTM model was designed to capture the temporal and spatial dependencies inherent in IVS data.The model consisted of two stacked ConvLSTM2D layers, both configured with 64 filters and a 3×3 kernel. The consistent use of 64 filters in both layers ensured sufficient capacity to model the complex temporal and spatial interactions within the IVS grid. The first ConvLSTM2D layer

processed the input sequence to capture temporal patterns while preserving spatial relationships, while the second ConvLSTM2D layer refined these patterns, aggregating the temporal information into a single time step to represent the next predicted IVS grid. Each ConvLSTM layer is followed by batch normalization, as well as dropout with a rate of 0.1 to prevent overfitting. To enhance the spatial resolution of the predicted IVS, the output of the ConvLSTM layers was passed through a Conv2D layer with a single filter, a 3×3 kernel and a ReLu activation function. This layer refined the spatial structure of the IVS, ensuring smooth transitions across maturities and moneyness levels. This was followed by a flattening layer, a fully connected dense layer, and a final reshape layer to restore the output to the original grid dimensions of 17 maturities by 40 moneyness levels. The model is trained to predict the IVS for the next trading day (t+1) using the past 10 days of data as input. Training was performed over 100 epochs with a batch size of 32, with the last 20% of the training data reserved as a validation set. The Mean Absolute Error (MAE) was used as the loss function, and the Nadam optimizer, with a learning rate of 0.0005, was employed to optimize the model weights. After training, the model was also extended to predict longer horizons (t+1, t+5, t+10, t+20, t+30) using an iterative prediction framework, where the predicted IVS for one step was used as input for the next step.

3. ConvLSTM with Self-Attention (Grid-Based)

Building on the strengths of the second model, the ConvLSTM-SA extended the architecture by incorporating Self-Attention mechanism to improve the model's ability to capture spatiotemporal dependencies within the IVS, particularly long-range relationships. Self-Attention introduces a global perspective by dynamically weighting the importance of different parts of the input sequence and grid. The interaction between a ConvLSTM layer and a Self-Attention layer is illustrated in Figure 5. This addition enables the model to capture long-range dependencies

across both spatial and temporal dimensions. The architecture begins with a ConvLSTM2D layer, configured identically to the ConvLSTM (Grid-Based) model, with 64 filters and a 3×3 kernel operating on a 4D input (time, maturity, strike, and channel). After this initial ConvLSTM block, batch normalization and dropout with a rate of 0.1 were applied to stabilize training and improve generalization. Following the first ConvLSTM block, a single-head Self-Attention layer, configured with 64 units and applying dropout (0.1) to its attention weights was introduced. This layer computes attention scores by scaling dot-product similarities between query and key vectors, emphasizing the most informative temporal and spatial regions of the input sequence. By doing that, the mechanism allows the model to focus on relevant temporal and spatial features. Although Multi-Head Self-Attention (MHSA) was initially considered to capture a richer set of dependencies, resource constraints limited the number of attention heads to two, which did not yield substantial improvement over a single-head configuration. Consequently, a single-head attention approach was retained for its balance of complexity, computational efficiency, and predictive accuracy. After the Self-Attention layer, the model processed the attention-enhanced representations through a second ConvLSTM2D layer with the same configuration (64 filters, 3×3 kernel), followed by another batch normalization and dropout step. The output of this second ConvLSTM block, now enriched by both local spatiotemporal patterns and global context from the attention mechanism, was reshaped into a 4D tensor for spatial refinement. A subsequent Conv2D layer with a single filter and a 3×3 kernel further refined the spatial structure of the IVS predictions. Finally, the output was flattened, passed through a dense layer to aggregate global features, and reshaped into the original grid dimensions of 17×40. The training procedure for this model mirrored that of the previous model, and included the use of 10 lags, a batch size of 32, and training for up to 100 epochs with early stopping and a validation set

comprising 20% of the training data. The consistency in the training setup ensured comparability between the models, allowing the contribution of the Self-Attention layer to be accurately assessed.

Results

A Vector Autoregression (VAR) model was employed as a baseline to evaluate the relative performance of the proposed models. The input in the VAR model consisted of the flattened vectors representing the volatility surface, using the same number of lags as the deep learning models. Moreover, all models, including the baseline and deep learning approaches, were trained and tested using the same dataset split. The train-test split was 80% for training and 20% for testing, ensuring consistency across evaluations. Specifically, the training data spanned from January 4, 2010 to March 18, 2021, while the testing data covered the period from March 19, 2021 to December 29, 2023. To evaluate the predictive performance of the proposed models, both Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE) were computed across the multiple forecast horizons. The results are summarized in Tables 1 and 2, and the daily smoothed MAPE values (using a 7-day rolling mean) are shown in Figure 6.

The SVI with ConvLSTM model exhibited the weakest performance among the other models. This was consistent across both metrics, with MAPE values ranging from 16.5161% at t+1 to 18.6258% at t+30, and RMSE values starting at 0.0392 and reaching 0.0442. The underperformance of this model can be attributed to the limitations of the parameterized SVI framework, which may impose restrictive assumptions on the volatility surface dynamics. Although the SVI model is effective for surface fitting, its reliance on a predefined functional form restricts its flexibility in capturing the non-linear and evolving spatiotemporal relationships present in the IVS. In contrast, the grid-based ConvLSTM model achieved the best short-term

performance in terms of MAPE, with an error of 8.9011% at t+1. This demonstrates the model's ability to effectively capture the surface dynamics for short-term horizons. Figure 7 illustrates the predicted vs. actual implied volatility surfaces (IVS) for the t+1 forecasts across the evaluated models. However, its predictive accuracy deteriorates slightly as the forecast horizon increases, with MAPE rising to 14.8957% at t+30. The ConvLSTM-SA model demonstrated better performance for longer forecast horizons, achieving the lowest MAPE values among all models for t+5, t+10, t+20, and t+30. This indicates that the inclusion of Self-Attention enables the model to better capture long-range dependencies, which becomes particularly valuable as the forecast horizon increases. By incorporating global temporal relationships, the Self-Attention mechanism helps mitigate error accumulation, ensuring more stable and accurate multi-step forecasting. Surprisingly, the VAR model outperformed all other models in terms of RMSE across all horizons. For instance, it achieved RMSE values of 0.0215 at t+1 and 0.0320 at t+30, which are consistently lower than the ConvLSTM-based models. This result suggests that the model provides more stable predictions with fewer extreme deviations. While RMSE is useful for measuring absolute deviations, MAPE is particularly relevant in the context of implied volatility surface prediction. By focusing on percentage-based errors, MAPE provides a measure for relative deviations in implied volatility levels across strikes and maturities. This is especially important given that implied volatilities are typically small in magnitude, some absolute errors can appear insignificant despite having meaningful relative impacts. Overall, the grid-based ConvLSTM models outperformed the other approaches in terms of MAPE, showcasing their ability to model the evolution of the IV surface effectively.

To further assess the robustness of the proposed models under unseen periods of extreme volatility, we retrained all models using data up until March 1, 2020, and evaluated them on the

period from March 2, 2020, to December 29, 2020. This period captures the beginning and aftermath of the COVID-19 market crash, a period marked by unprecedented volatility levels. Similar approaches have been utilized in previous studies, such as Kim et al. (2023), where their models were re-tested under volatile conditions during April 2020. The goal for this experiment was to analyze the models' ability to generalize under highly volatile regimes they were not explicitly trained on. High volatility periods often lead to rapid and nonlinear changes in the implied volatility surface, which can challenge the predictive accuracy of the models.

The results, summarized in Tables 3 and 4, and illustrated in Figure 8 highlight a general decline in performance for most models compared to the full sample testing period. The errors were the highest in March and April 2020, particularly at the beginning of the COVID-19 crash, before gradually stabilizing over the following months. This trend is evident across all forecast horizons, with smoothed MAPE values peaking in early March before declining as volatility started to normalize. The SVI with ConvLSTM model continued to perform the worst, with MAPE increasing significantly, particularly for longer horizons, reaching 22.1564% at t+30. Similarly, its RMSE values also deteriorated, reflecting the model's limited flexibility in handling extreme market dynamics. Among the remaining models, the grid-based ConvLSTM demonstrated notable resilience during this period. For short-term predictions (t+1), it achieved a MAPE of 8.5660%, a surprising improvement similar to the VAR model's MAPE of 8.4907%. However, unlike VAR, the grid-based ConvLSTM maintained superior accuracy over medium to long horizons. While the VAR model achieved lower RMSE values overall, consistent with its earlier performance, its predictive accuracy, as measured by MAPE, deteriorated significantly for longer horizons. Interestingly, the ConvLSTM-SA underperformed relative to the standard ConvLSTM during this period, for both short and long term horizons. This result may suggest

that the added complexity of the Self-Attention mechanism was less effective in capturing abrupt and localized changes in the volatility surface, which are typical during periods of extreme volatility. Self-Attention tends to focus on long-range dependencies, and in highly volatile markets, the dynamics of the IVS may exhibit more localized, short-term patterns that the grid-based ConvLSTM alone was better able to capture.

Conclusion

This study aimed to predict the implied volatility surface (IVS) using deep learning techniques, evaluating their performance across multiple forecast horizons and market regimes. The grid-based ConvLSTM model demonstrated superior performance for short-term horizons (e.g., t+1), achieving lower MAPE compared to other models. Its ability to capture the localized, short-term dynamics of the IV surface highlights its suitability for short-term trading option strategies. In contrast, the ConvLSTM-SA model delivered stronger results over longer forecast horizons, suggesting that the inclusion of Self-Attention improves the model's ability to capture long-range dependencies and broader trends in the volatility surface. The SVI model, despite being effective for surface fitting, when used with ConvLSTM to predict the evolution of its parameters and then reconstruct the predicted surface, showed limitations in predictive accuracy, particularly for non-linear, rapidly evolving volatility dynamics. Performances were also evaluated on unseen data from the COVID-19 period, characterized by high volatility and market stress using a different training sample. Results indicated that errors peaked during March and April 2020, coinciding with the initial market crash. This revealed the models' challenges in adapting to the very abrupt changes in volatility. Nevertheless, as uncertainty levels gradually stabilized over subsequent months, the grid-based ConvLSTM models displayed better performance. The findings of this study might also have potential applications in options trading

and hedging strategies, where the accurate prediction of implied volatilities is critical. Improved IV surface predictions can help traders to identify mispriced options by comparing forecasted volatilities with observed market quotes, informing strategies such as volatility arbitrage. Additionally, accurate volatility forecasts enhance options-based hedging, allowing traders to dynamically adjust their positions to manage volatility risk. For instance, during periods of high volatility, strategies such as straddles and strangles can be employed to capitalize on predicted increases in implied volatilities. Conversely, in low-volatility environments, strategies such as butterfly spreads and iron condors are preferred, as they are better suited to stable market conditions.

Future Steps

Building on the results of this study, future research could focus on fine-tuning model parameters, such as the number of lags, to further optimize predictive performance. Incorporating additional variables, such as the VIX index, macroeconomic indicators, or other market factors, could improve the models' ability to reflect diverse market conditions. A key limitation of this study lies in the interpolation process used to construct the implied volatility surface. On dates with sparse or noisy data, the interpolated surfaces were occasionally inaccurate or lacked smoothness, which may have affected the models' predictions. Future work could address this challenge by exploring more advanced interpolation methods. Finally, testing the models on other markets, such as European and Asian indices, or applying them to different assets, including individual stocks, would help evaluate their adaptability and effectiveness in various financial contexts. This presents valuable opportunities to refine the models and extend their applicability.

Appendix

Appendix A: Black-Scholes Formulas

The closed-form solutions for European call and put option prices under the Black-Scholes model are:

$$C(S,t) = SN(d_1) - Ke^{-r(T-t)}N(d_2);$$

$$P(S,t) = Ke^{-r(T-t)}N(-d_2) - SN(-d_1);$$

$$d_1 = \frac{ln(\frac{S}{K}) + (r + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}};$$

$$d_2 = d_1 - \sigma\sqrt{T-t}$$

Where C represents the price of the call, P the price of the put, S is the underlying asset price, K is the strike price, r is the risk-free interest rate, T−t is the time to maturity, σ is the volatility of the underlying asset, and N(x) is the standard normal cumulative PDF.

Appendix B: Non-Arbitrage Conditions

Non-arbitrage smoothing methods ensure that the volatility surface is smoothed and free of inconsistencies that would lead to arbitrage. The non-arbitrage conditions are:

$C(K, T)$ is a convex function of K;

$C(K, T)$ is a monotonically increasing function of T;

Put-Call Parity: $S0 + P = C + Ke^{-rT}$;

No Butterfly Spread Arbitrage: For $K_1 < K_2 < K_3$, $2C(K_2) \leq C(K_1) + C(K_3)$;

No Calendar Spread Arbitrage: $T_1 < T_2$, $C(S, T_1) < C(S, T_2)$.

Appendix C: Heston Model Equations

The Heston model describes the evolution of the price of an asset and its variance using a system of stochastic differential equations, which account for mean-reverting stochastic volatility:

$$dS_t = \mu S_t dt + \sqrt{\nu_t} S_t dW_t^S$$

$$d\nu_t = \kappa(\theta - \nu_t)dt + \sigma\sqrt{\nu_t} dW_t^\nu$$

Where S_t is the underlying asset price at time t, ν_t is the asset variance, μ is the drift of the asset, θ is the long-term mean of variance, κ is the rate of mean reversion, σ is the volatility of the variance, dW_t^S and dW_t^ν are two correlated Wiener processes of the price and variance, respectively.

Appendix D: SVI Parameters

The parameters of the SVI model correspond to the following:

a is the vertical shift of the smile. An increase in a raises the overall level of variance.

b is the slope of the wings. A higher b increases the steepness of both the left (put) and right (call) wings, resulting in a tighter smile.

ρ is the skewness parameter. When ρ increases, the left wing's slope decreases and the right wing's slope increases, therefore, producing a counter-clockwise rotation of the smile.

m is the horizontal translation of the smile. Increasing m moves the smile to the right.

σ is the curvature parameter. A larger σ reduces the curvature of the smile around the ATM point.

Appendix E: Asymptotes of the Total Implied Variance

At extreme strikes, the left and right asymptotes (wings) of the total implied variance are defined as:

$$W_L(k) = a + b(\rho - 1)(k - m) \qquad k \to -\infty$$

$$W_R(k) = a + b(\rho + 1)(k - m) \qquad k \to \infty$$

where, $b(\rho + 1) < 2$.

Appendix F: Multi-Head Self-Attention Equations

The Multi-Head Self-Attention mechanism is defined as follows:

$$head_i = Self\ Attention(QW_Q, KW_K, VW_V);$$

$$MultiHead(Q, K, V) = Concat(head_1, head_2, \ldots, head_h)W_O.$$

Each individual head output is concatenated and linearly transformed using the weight matrix W_O, where $W_O \in \mathbb{R}^{h \cdot d_v \times d}$. This final transformation ensures that the model learns a diverse set of relationships and patterns within the input sequence.

Appendix G: Nelson-Siegel Yield Curve Model

The Nelson-Siegel model is used to estimate the term structure of interest rates and is given by:

$$r(T) = \beta_0 + \beta_1 \cdot \frac{1-e^{-\lambda T}}{\lambda T} + \beta_2 \cdot \left(\frac{1-e^{-\lambda T}}{\lambda T} - e^{-\lambda T}\right)$$

Where T is the time to maturity in years and β_0, β_1, β_2, and λ are the parameters, which represent the level, slope, curvature and decay factor of $\beta_1$and β_2, respectively .

Appendix H: Figures

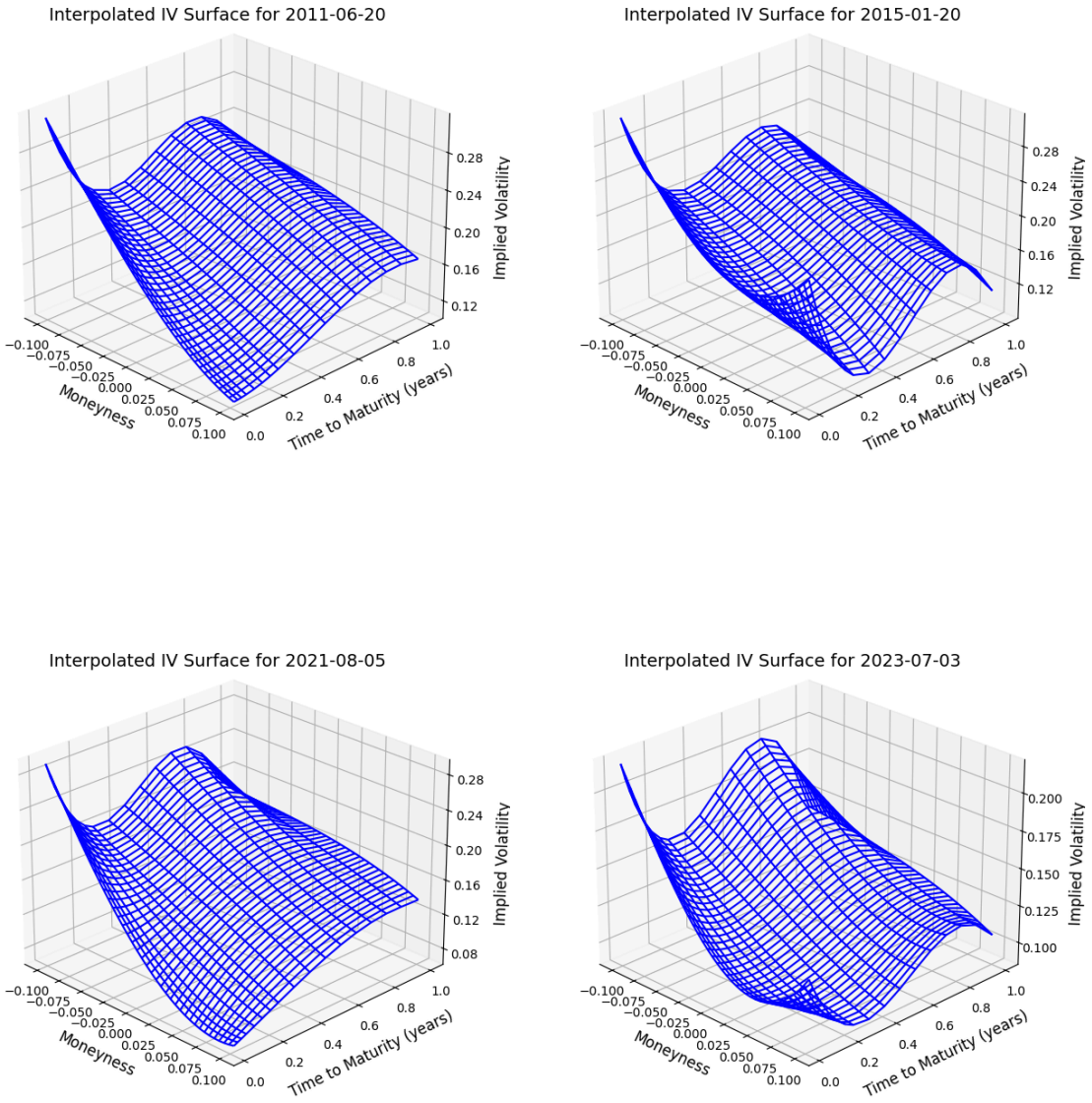


Figure 1: *Interpolated implied volatility surfaces (IVS) using Smooth Bivariate Spline interpolation.*

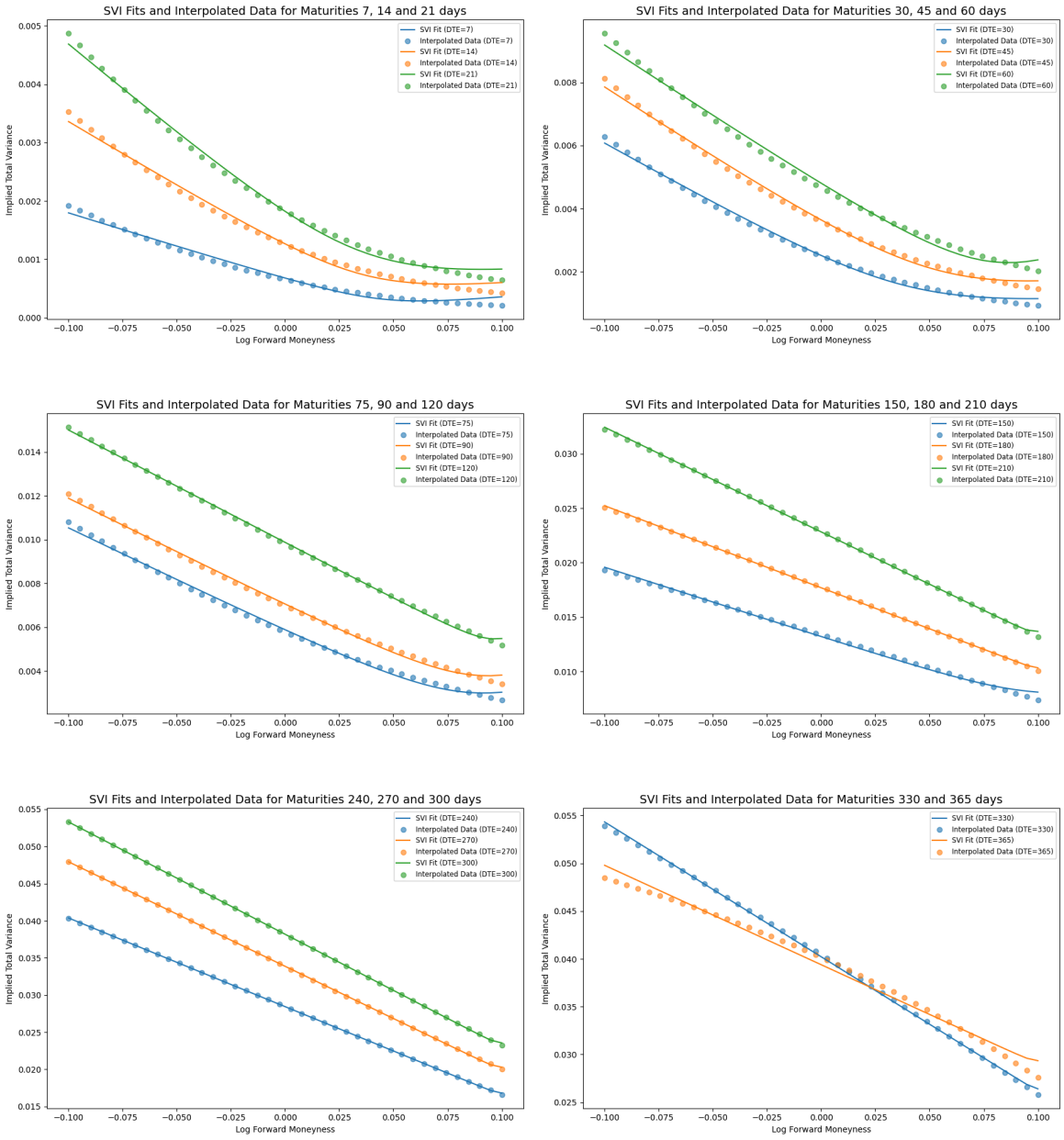


Figure 2: *SVI fits and interpolated total implied variances across different maturities.*

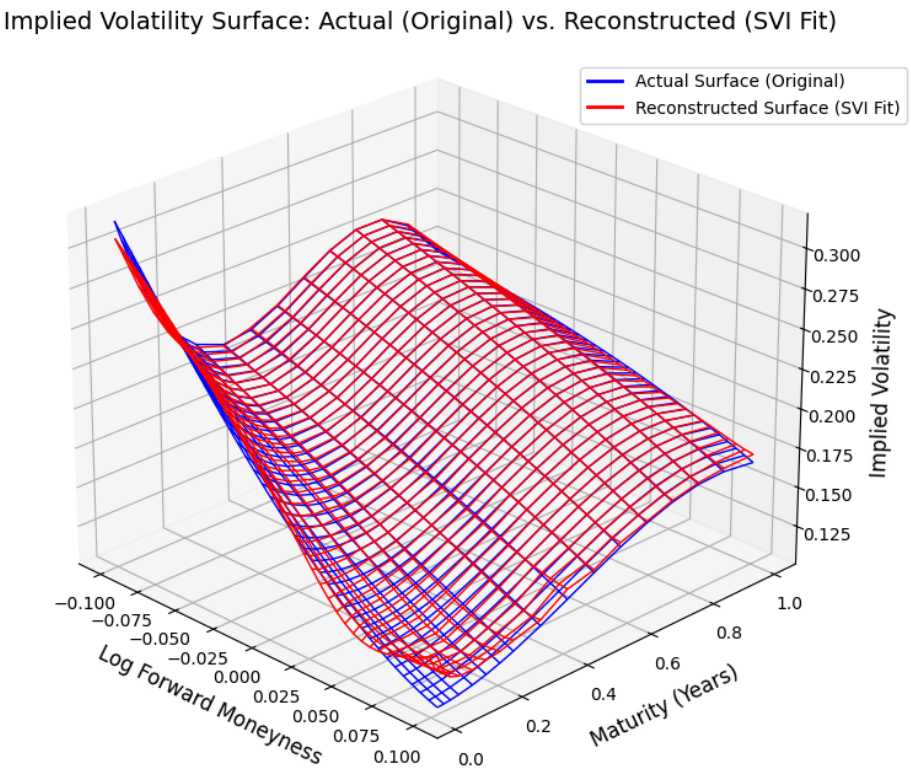


Figure 3: *Reconstructed implied volatility surface (SVI Fit) compared to the original surface.*

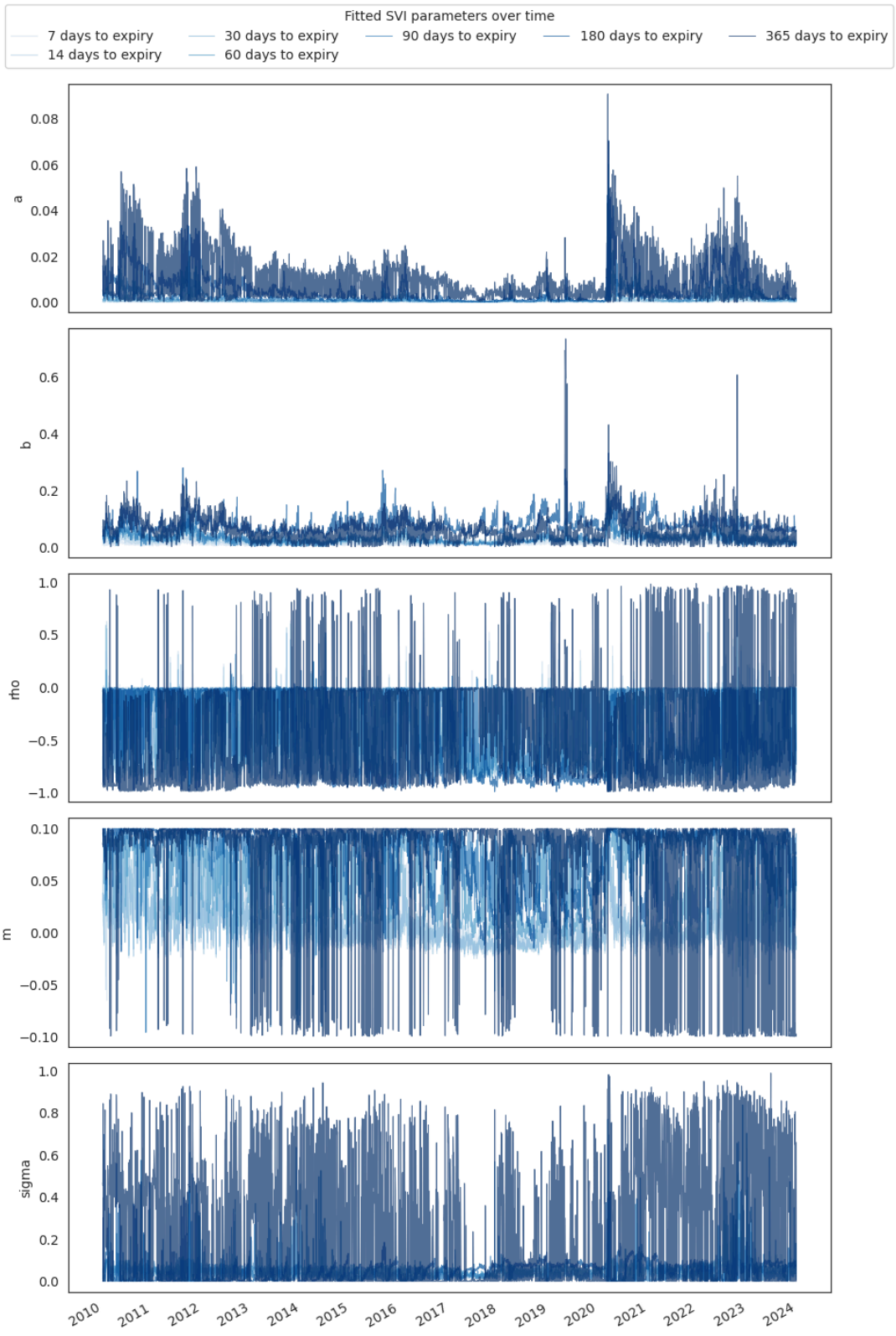


Figure 4: *Fitted SVI parameters (a, b, ρ, m, σ) over time for different maturities.*

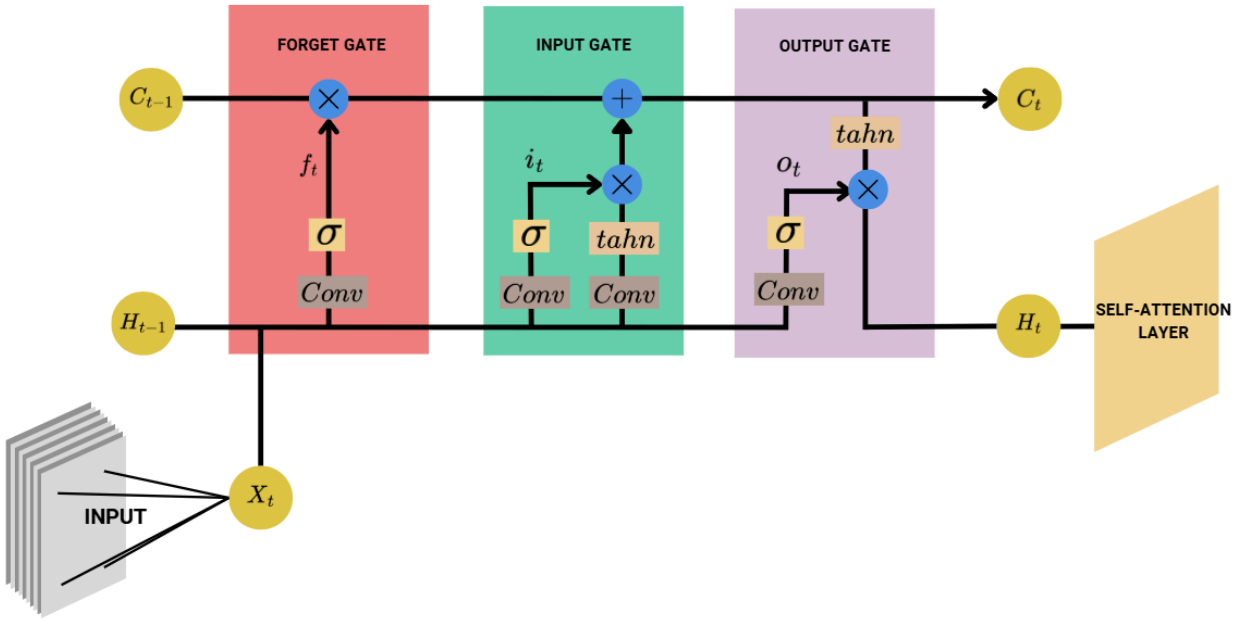
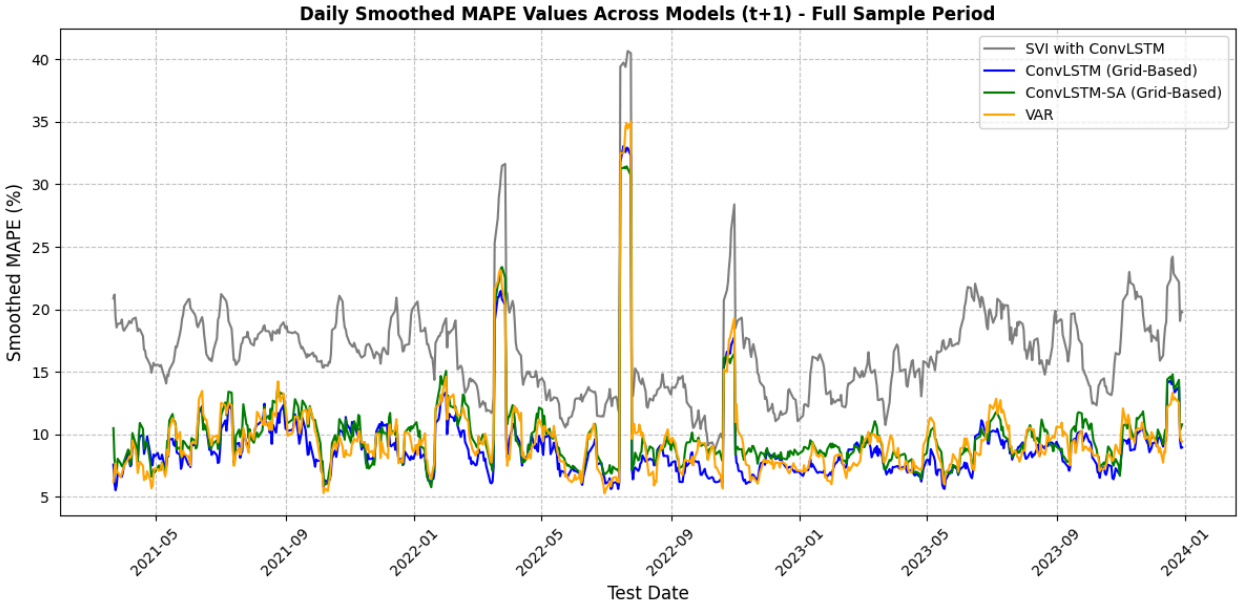
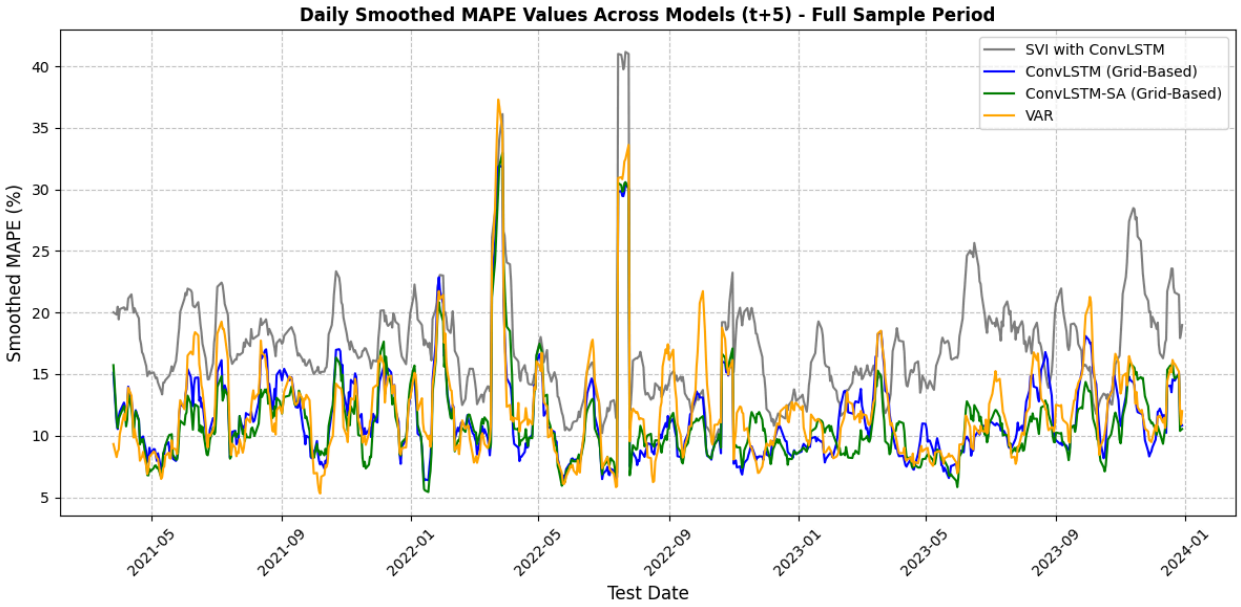
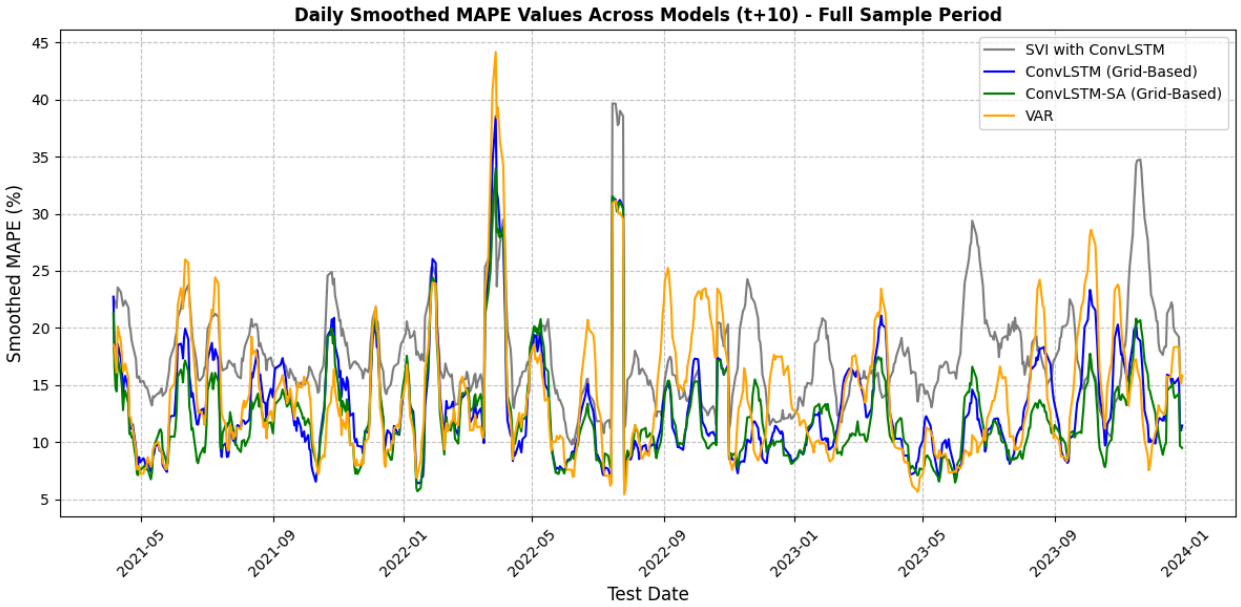


Figure 5: *ConvLSTM architecture with Self-Attention. Convolutional operations process spatiotemporal input through the Forget, Input, and Output gates, while the Self-Attention Layer captures global temporal dependencies.*




Daily Smoothed MAPE Values Across Models (t+5) - Full Sample Period
SVI with ConvLSTM
ConvLSTM (Grid-Based)
ConvLSTM-SA (Grid-Based)
VAR
Smoothed MAPE (%)
Test Date
2021-05
2021-09
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09
2024-01


Daily Smoothed MAPE Values Across Models (t+10) - Full Sample Period
SVI with ConvLSTM
ConvLSTM (Grid-Based)
ConvLSTM-SA (Grid-Based)
VAR
Smoothed MAPE (%)
Test Date
2021-05
2021-09
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09
2024-01

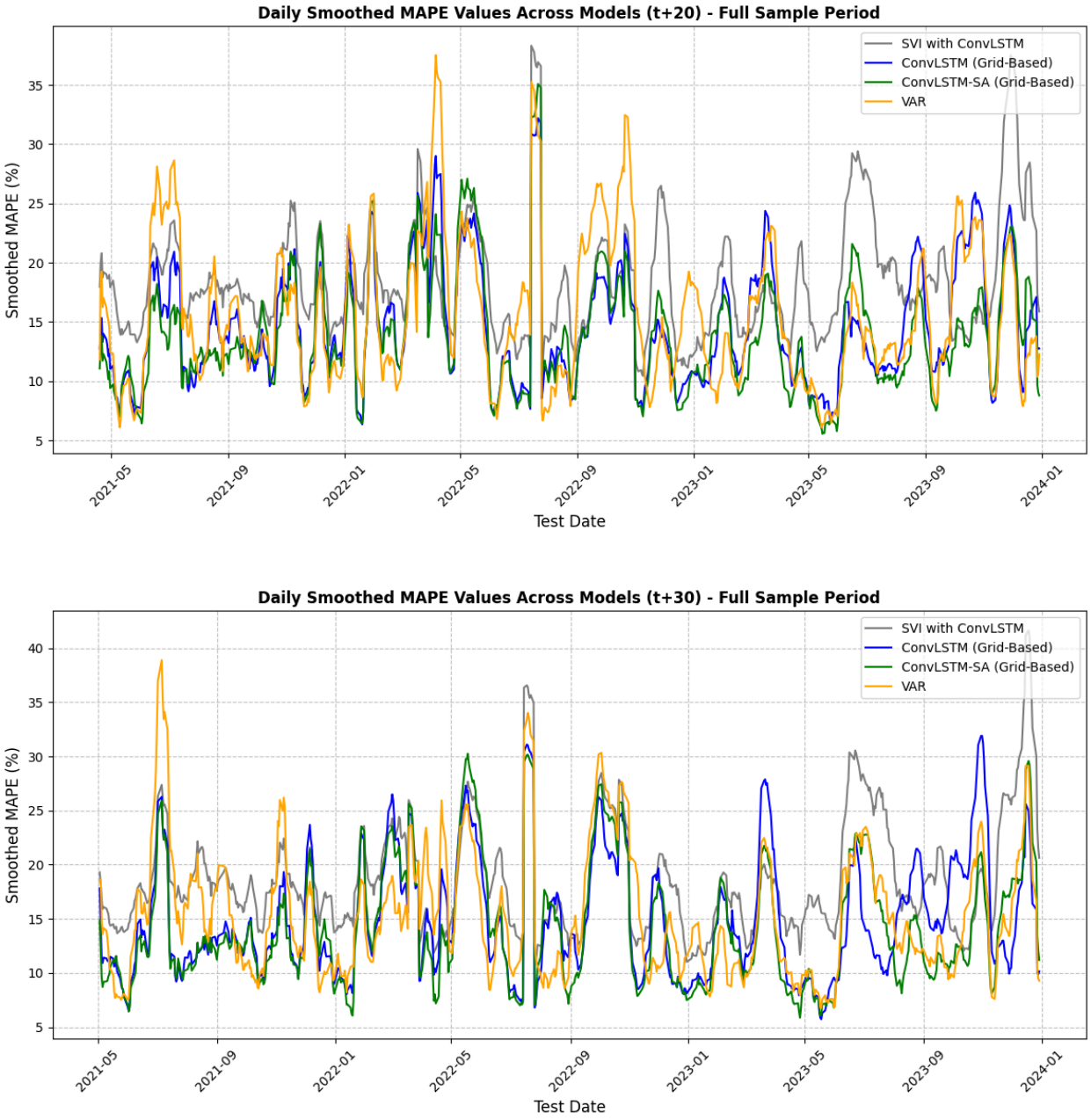
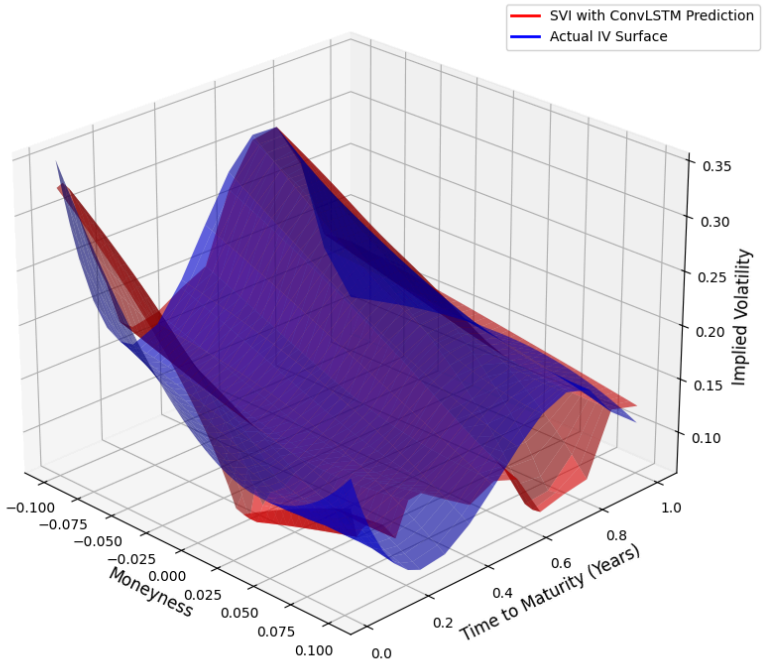
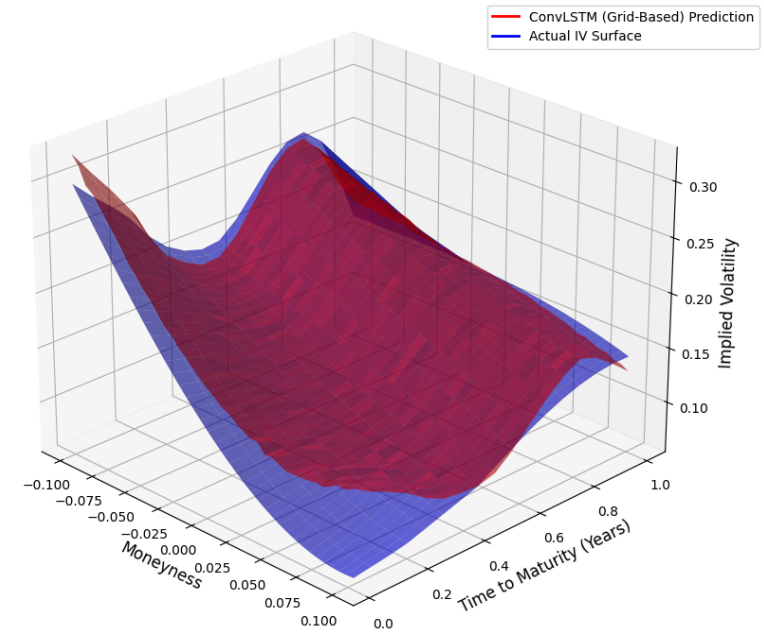


Figure 6: *Daily smoothed MAPE values across models (SVI with ConvLSTM, ConvLSTM, ConvLSTM-SA, and VAR) for different forecast horizons (t+1, t+5, t+10, t+20, and t+30) over the full sample period.*

SVI with ConvLSTM Predicted vs. Actual Implied Volatility Surface (t+1 Forecast)



ConvLSTM (Grid-Based) Predicted vs. Actual Implied Volatility Surface (t+1 Forecast)



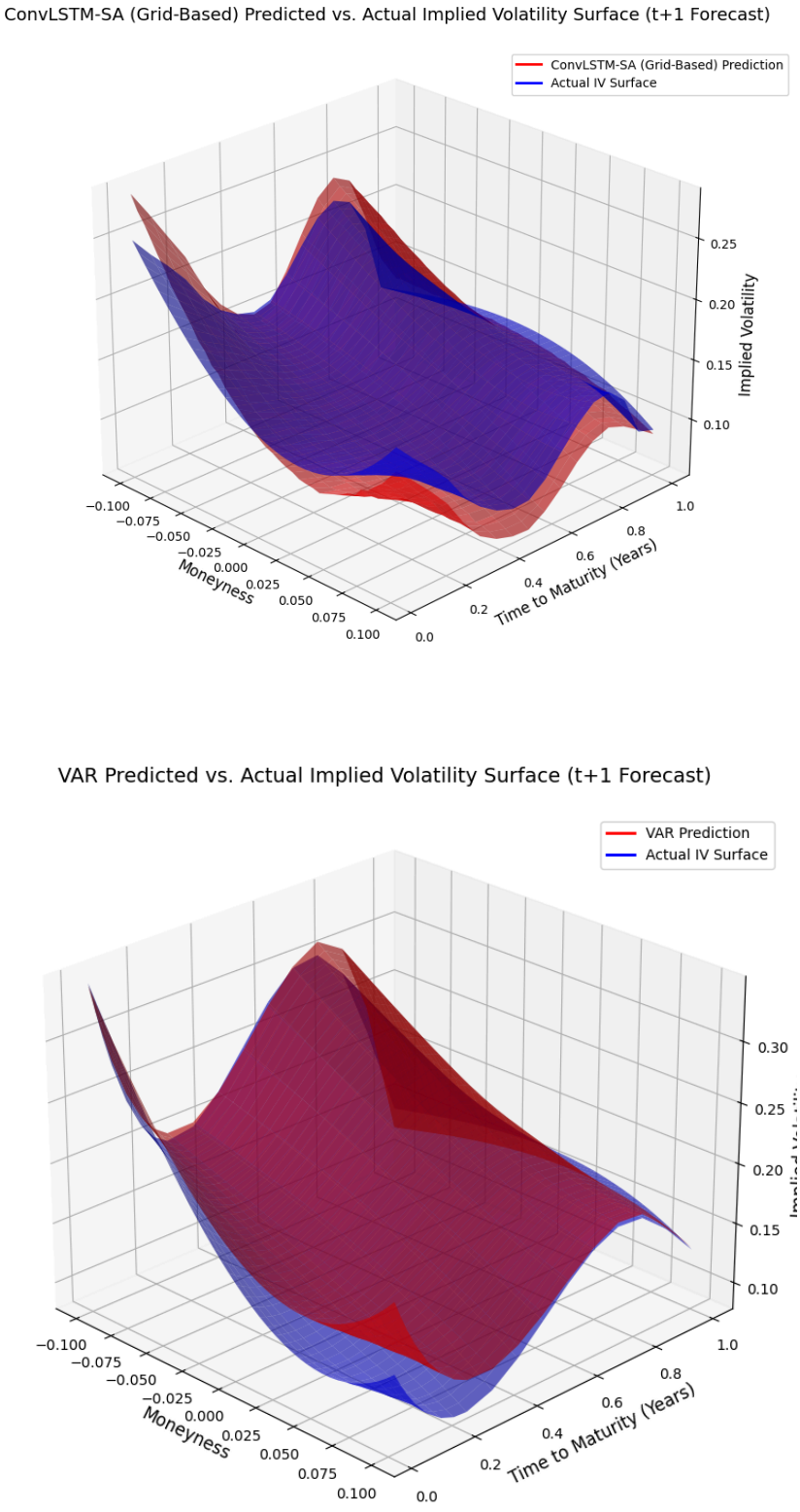
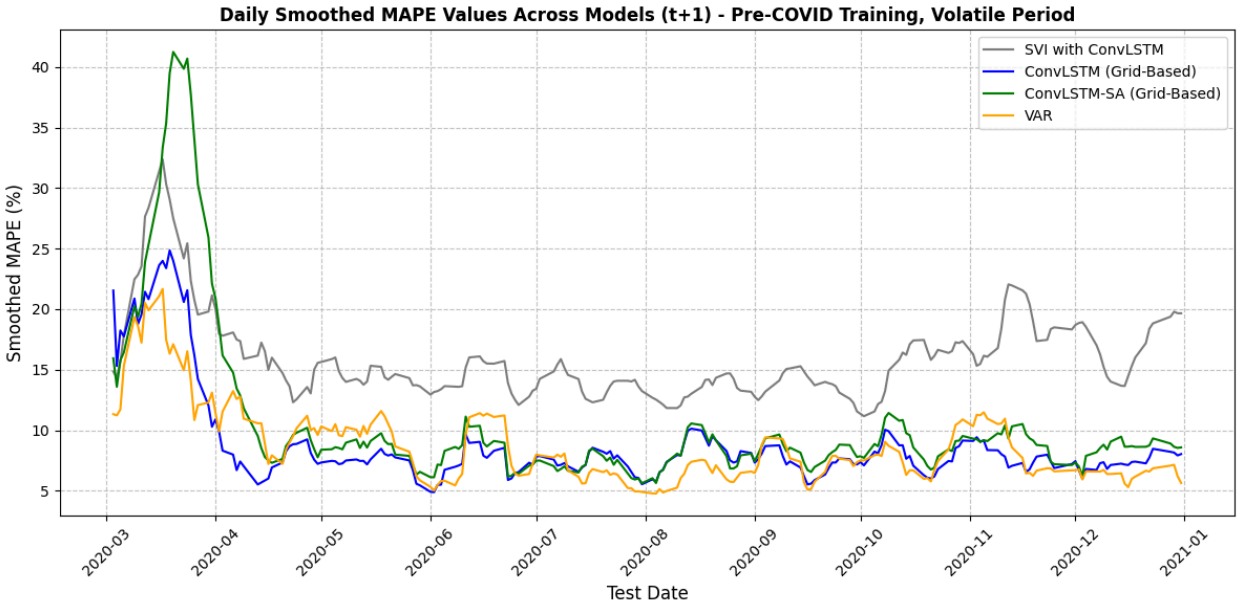
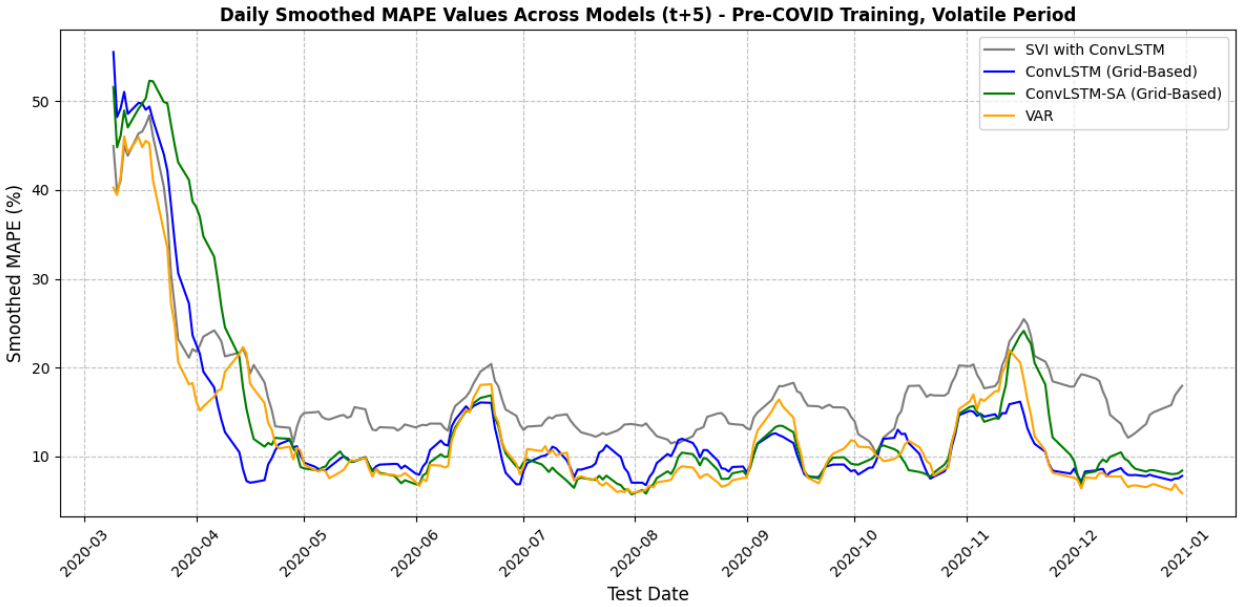
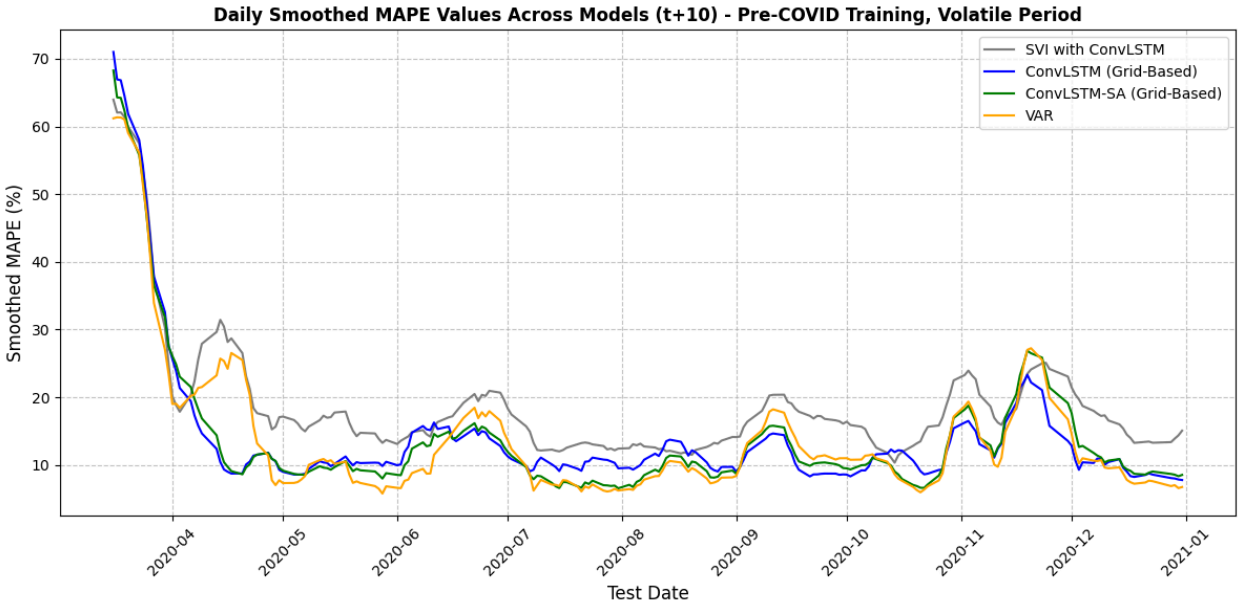
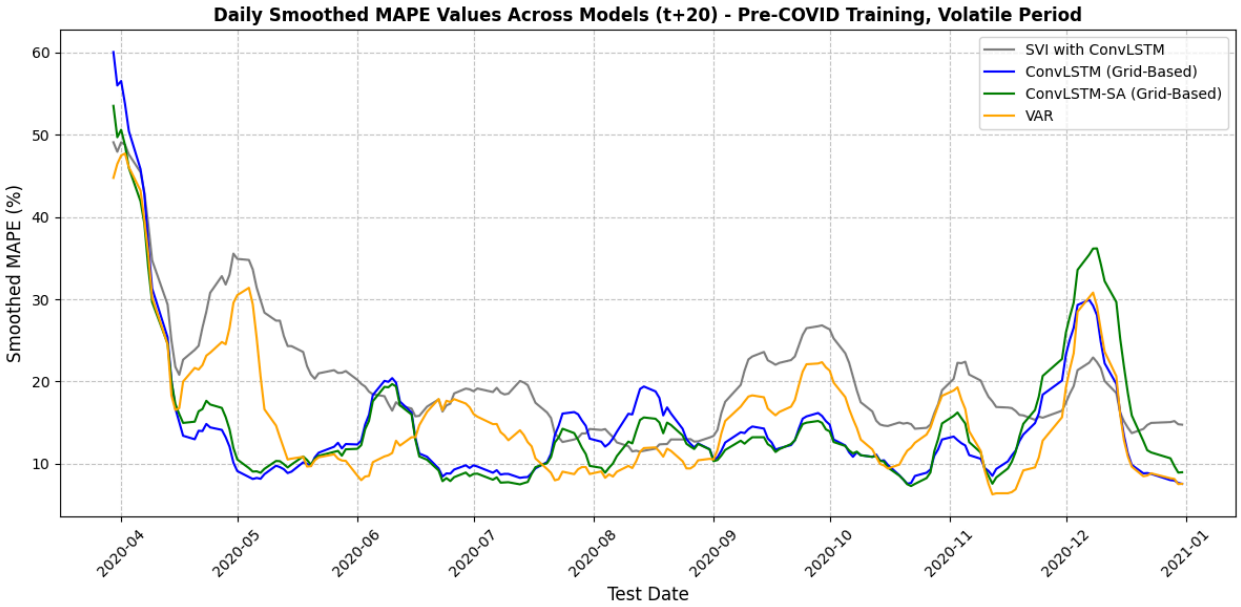


Figure 7: *Predicted vs. Actual Implied Volatility Surfaces (IVS) for t+1 forecasts across models.*


Daily Smoothed MAPE Values Across Models (t+1) - Pre-COVID Training, Volatile Period
SVI with ConvLSTM
ConvLSTM (Grid-Based)
ConvLSTM-SA (Grid-Based)
VAR
Smoothed MAPE (%)
Test Date
2020-03
2020-04
2020-05
2020-06
2020-07
2020-08
2020-09
2020-10
2020-11
2020-12
2021-01


Daily Smoothed MAPE Values Across Models (t+5) - Pre-COVID Training, Volatile Period
SVI with ConvLSTM
ConvLSTM (Grid-Based)
ConvLSTM-SA (Grid-Based)
VAR
Smoothed MAPE (%)
Test Date
2020-03
2020-04
2020-05
2020-06
2020-07
2020-08
2020-09
2020-10
2020-11
2020-12
2021-01


Daily Smoothed MAPE Values Across Models (t+10) - Pre-COVID Training, Volatile Period
SVI with ConvLSTM
ConvLSTM (Grid-Based)
ConvLSTM-SA (Grid-Based)
VAR
Smoothed MAPE (%)
Test Date
2020-04
2020-05
2020-06
2020-07
2020-08
2020-09
2020-10
2020-11
2020-12
2021-01


Daily Smoothed MAPE Values Across Models (t+20) - Pre-COVID Training, Volatile Period
SVI with ConvLSTM
ConvLSTM (Grid-Based)
ConvLSTM-SA (Grid-Based)
VAR
Smoothed MAPE (%)
Test Date
2020-04
2020-05
2020-06
2020-07
2020-08
2020-09
2020-10
2020-11
2020-12
2021-01

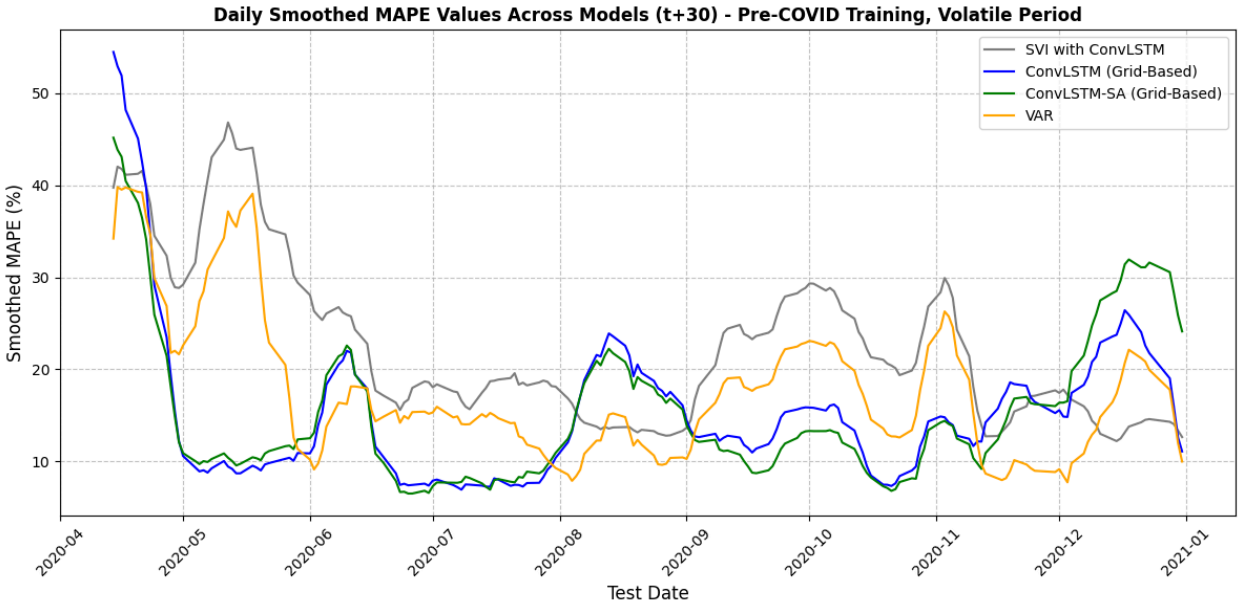


Figure 8: *Daily Smoothed MAPE values across models for t+1, t+5, t+10, t+20, and t+30 forecasts during the COVID stress testing period (March–December 2020).*

Appendix I: Tables

MAPE with Full Sample					
Models	**t+1**	**t+5**	**t+10**	**t+20**	**t+30**
SVI with ConvLSTM	16.5161%	17.0951%	17.5559%	18.3529%	18.6258%
ConvLSTM	8.9011%	11.3993%	12.9172%	14.5175%	14.8957%
ConvLSTM-SA	9.7925%	11.0297%	12.1932%	13.8056%	14.1674%
VAR	9.4684%	11.9769%	14.1660%	15.4784%	15.2305%

Table 1: Mean Absolute Percentage Error (MAPE) values for all models evaluated on the full sample period across forecast horizons t+1, t+5, t+10, t+20, and t+30.

RMSE with Full Sample					
Models	**t+1**	**t+5**	**t+10**	**t+20**	**t+30**
SVI with ConvLSTM	0.0392	0.0404	0.0415	0.0432	0.0442
ConvLSTM	0.0232	0.0279	0.0311	0.0349	0.0368
ConvLSTM-SA	0.0247	0.0274	0.0301	0.0341	0.0360
VAR	0.0215	0.0260	0.0300	0.0325	0.0320

Table 2: Root Mean Squared Error (RMSE) values for all models evaluated on the full sample period across forecast horizons t+1, t+5, t+10, t+20, and t+30.

MAPE with COVID Stress Testing					
Models	**t+1**	**t+5**	**t+10**	**t+20**	**t+30**
SVI with ConvLSTM	15.9356%	17.3145%	18.0315%	19.8243%	22.1564%
ConvLSTM	8.5660%	12.4457%	13.3015%	14.0246%	14.6167%
ConvLSTM-SA	10.3576%	13.6416%	13.0928%	14.2404%	14.8346%
VAR	8.4907%	12.1337%	13.1362%	15.1803%	17.1912%

Table 3: Mean Absolute Percentage Error (MAPE) values for all models during the COVID stress testing period (Mar.–Dec. 2020) across forecast horizons t+1, t+5, t+10, t+20, and t+30.

RMSE with COVID Stress Testing					
Models	**t+1**	**t+5**	**t+10**	**t+20**	**t+30**
SVI with ConvLSTM	0.0607	0.0752	0.0817	0.0699	0.0738
ConvLSTM	0.0439	0.0706	0.0743	0.0506	0.0457
ConvLSTM-SA	0.0620	0.0800	0.0725	0.0496	0.0440
VAR	0.0287	0.0415	0.0444	0.0438	0.0471

Table 4: Root Mean Squared Error (RMSE) values for all models during the COVID stress testing period (Mar.–Dec. 2020) across forecast horizons t+1, t+5, t+10, t+20, and t+30.

References

Bakshi, G., Cao, C., and Chen, Z. 1997. “Empirical Performance of Alternative Option Pricing Models.” *Journal of Finance* 52 (5): 2003–2049.

Bates, D. S. 1991. “The Crash of '87: Was It Expected? The Evidence from Options Markets.” *Journal of Finance* 46 (3): 1009–1044.

Black, F., and Scholes, M. 1972. “The Valuation of Option Contracts and a Test of Market Efficiency.” *Journal of Finance* 27 (2): 399–417. https://doi.org/10.2307/2978484.

Black, F., and Scholes, M. 1973. “The Pricing of Options and Corporate Liabilities.” *Journal of Political Economy* 81: 637–654. https://doi.org/10.1086/260062.

Bloch, D. A., and Böök, A. 2021. “Deep Learning Based Dynamic Implied Volatility Surface.” *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3952842.

Bollerslev, T. 1986. “Generalized Autoregressive Conditional Heteroskedasticity.” *Journal of Econometrics* 31 (3): 307–327.

Chen, S., and Zhang, Z. 2019. “Forecasting Implied Volatility Smile Surface via Deep Learning and Attention Mechanism.” *arXiv*. https://arxiv.org/abs/1912.11059.

Dupire, B. 1994. “Pricing with a Smile.” *Risk* 7 (1): 18–20.

Ferhati, T. 2020. “Robust Calibration for SVI Model Arbitrage-Free.” *HAL Working Papers*. https://hal.archives-ouvertes.fr/hal-02490029.

Gatheral, J., and Jacquier, A. 2013. “Arbitrage-Free SVI Volatility Surfaces.” *Quantitative Finance* 14 (1): 59–71. https://doi.org/10.2139/ssrn.2033323.

Gers, F. A., Schmidhuber, J., and Cummins, F. 2000. “Learning to Forget: Continual Prediction with LSTM.” *Neural Computation* 12 (10): 2451–2471. https://doi.org/10.1162/089976600300015015.

Hansen, P. R., and Lunde, A. 2005. “A Forecast Comparison of Volatility Models: Does Anything Beat a GARCH(1,1)?” *Journal of Applied Econometrics* 20: 873–889. https://doi.org/10.1002/jae.800.

Heston, S. L. 1993. “A Closed Solution for Options with Stochastic Volatility, with Application to Bond and Currency Options.” *Review of Financial Studies* 6: 327–343. https://doi.org/10.1093/rfs/6.2.327.

Hochreiter, S., and Schmidhuber, J. 1997. “Long Short-Term Memory.” *Neural Computation* 9 (8): 1735–1780.

Hull, J., and White, A. 1987. “The Pricing of Options on Assets with Stochastic Volatilities.” *Journal of Finance* 42 (2): 281–300.

Kim, S., Yun, S. B., Bae, H. O., Lee, M., and Hong, Y. 2022. “Physics-Informed Convolutional Transformer for Predicting Volatility Surface.” *arXiv*. https://arxiv.org/abs/2209.10771.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. 2012. “ImageNet Classification with Deep Convolutional Neural Networks.” *Advances in Neural Information Processing Systems* 25: 1097–1105.

LeCun, Y., and Bengio, Y. 1995. “Convolutional Networks for Images, Speech, and Time Series.” In *The Handbook of Brain Theory and Neural Networks*, 255–258. Cambridge, MA: MIT Press.

Lee, R. 2004. “The Moment Formula for Implied Volatility at Extreme Strikes.” *Mathematical Finance* 14 (3): 469–480.

Medvedev, N., and Wang, Z. 2022. “Multistep Forecast of the Implied Volatility Surface Using Deep Learning.” *Journal of Futures Markets* 42 (4): 645–667.

Orlando, G., and Taglialatela, G. 2017. “A Review on Implied Volatility Calculation.” *Journal of Computational and Applied Mathematics* 320: 202–220.

Orosi, G. 2011. “Empirical Performance of a Spline-Based Implied Volatility Surface.” *Journal of Derivatives & Hedge Funds* 18 (4): 361–376. https://doi.org/10.2139/ssrn.1911147.

Shi, X. J., Chen, Z. R., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-C. 2015. “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting.” *Advances in Neural Information Processing Systems* 28: 802–810.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. 2017. “Attention Is All You Need.” *Advances in Neural Information Processing Systems* 30: 5998–6008.