Analysis of Volatility Surfaces with Computer Vision

Alec Janowski
Stanford University
Computer Science
alecj@stanford.edu

Matt Johnson
Stanford University
ICME
mattj949@stanford.edu

Annette Jing
Stanford University
Statistics
ajing@stanford.edu

Abstract

This paper examines several architectures for predicting future VIX index values using historical equity options surface data. We provide final accuracy and fit metrics, as well as qualitative analysis of the superior or inferior performance of each model given the horizon of desired VIX forecasts. Specifically, this study experiments with CNN, CONV-LSTM, and Transformer based approaches towards volatility prediction, none of which are currently industry standard techniques. We conduct accuracy and residual analysis in the experimental results, and find that CONV-LSTM and Transformer approaches exhibit predictive power above other techniques.

1. Introduction

An options volatility surface is a 3-dimensional plot depicting option implied volatility (expected standard deviation of the daily percentage price changes) as a function of time until expiration (tenor) and strike price. Traders study these surfaces to gain insight into the financial markets. Since an options volatility surface depicts expected future volatility, it theoretically should depict the market's aggregated/consensus view on the future, given current information. Since traders visually interpret these plots in hopes of predicting future market behavior, we wish to see if a computer vision algorithm can do the same, or better.

Our approach is to first apply baseline models like linear regression, as well as more tailored volatility models like Generalized AutoRegressive Conditional Heteroskedasticity (GARCH). We then move onto various neural network architectures such as a 2-dimensional CNN, a 3-dimensional CNN, a CONV-LSTM model, and a Transformer.

The VIX index is the market's expectation for volatility (ie: standard deviation of prices) over the next 30 days. Accurately modeling and predicting the VIX index – or volatility more generally – is exceedingly important for financial institutions needing to approximate and hedge risks over time. Failure to account for potential spikes in volatility can result in loss of capital, regulatory penalties, and even bankruptcy in extreme scenarios.

2. Related Work

Fernandes et al. [4] show that the VIX has a long-term dependence, and use basic parametric and autoregressive time series models to forecast future VIX values. Hosker et al. [7] find that vanilla RNNs and LSTMs are better than more basic statistical models at forecasting future VIX values, using input data consisting VIX futures and options data, as well as various technical indicators. Majmudar et al. [9] use the GARCH model for prediction. Martens et al. [10] describe how option implied volatilities (such as those contained in a volatility surface) are superior predictors of future volatility than time series models just based on historical returns of the underlying. Padhi and Shaikh [12] also further the notion that options implied volatilities contain useful, predictive information of future volatility.

Hirsa et al. [6] use neural networks to approximate the VIX index using a different basket of underlying futures contracts; while successful, they don't explicitly seek to predict the VIX, and further they don't explore the predictive ability of volatility surfaces. Carr et al. [2] use ridge regression and fully connected neural networks to forecast future realized volatility using S&P 500 index and options data. However, forecasting future volatility is still different from forecasting the VIX index itself. Yu [18] and Vrontos [17] use statistical methods to try and predict the direction of the VIX's movement in the future, however these works are classification problems rather than a regression problem concerning the actual value of the VIX. Rosillo et al. Work by Osterrieder et al. [11] is probably the most similar to our own, as they formulate a regression problem of predicting the VIX on an intraday scale. However, we are interested in longer time scales.

In sum, and quite crucially, the existing literature does not consider the informational content of volatility surfaces to predict future VIX values. Further, the deep learning approaches used to forecast the VIX are limited to simple fully connected networks and LSTMs.

3. Methods

3.1. Baseline methods

For the ***AR model***, we fit a simple linear regression using lagged target values to predict the current target value. Denoting the VIX index at time t by v_t and the lag by ℓ, we fit the model

$$v_t = \alpha + \beta v_{t-\ell} + \varepsilon_t$$

by ordinary least squares.

For the ***GARCH model***, we fit the prior year of S&P 500 returns to GARCH(1,1), and then we use this to model the current VIX index. Intuitively, GARCH models a moving average of variance over time with adjustable model parameters that impact the degree to which prior variances impact the current forecast. The model specification is given below for reference,

$$\begin{aligned} r_t &= \sigma_t \epsilon_t, \quad \epsilon_t \overset{iid}{\sim} \sqrt{\frac{\nu-2}{\nu}}\, t(\nu), \\ \sigma_t^2 &= \omega + \alpha \epsilon_{t-\ell}^2 + \beta \sigma_{t-\ell}^2, \end{aligned}$$

where r_t is the return at time t and $t(\nu)$ referes to the student-t distribution with degrees of freedom ν. The tunable parameters are α, β, ω, and ν under this setting. The GARCH conditional volatility estimates $\hat{\sigma}_t^2$ are used as a close predictive proxy to the VIX values v_t we wish to estimate, and we calculate all metrics using this output as the prediction of the VIX level. Following the standard approach, each new prediction is generated from refitting with a rolling-window approach.

3.2. Network Architectures

For our ***2D CNN*** architecture we stack 20 trading days of options surfaces to form an N channel input. This can be thought of as a "video" of the surfaces movement over time, though still using 2d kernels. Our CNN architecture is as follows:

- CONV(32), MAXPOOL,
- CONV(64), MAXPOOL,
- CONV(128), MAXPOOL, FLATTEN,
- DENSE(256), DENSE(128), DENSE(64).

We use 3×3 filters on all CONV layers, 2×2 pooling, as well as L2 regularization of .0001 on all layers. We use ReLU activations on all layers and linear output on the final layer. We also use Huber loss as most financial data contains many outliers and the signal-to-noise ratio of the data is low.

For our ***3D CNN*** architecture we follow the same structure, except all CONV layers use $3\times3\times3$ filters and pooling layers $2 \times 2 \times 2$ filters.

For our ***CONV-LSTM*** architecture, we use 20 trading days in training and test to predict volatility at the desired lag. Images are all single channel, and the architecture is as follows:

- CONV LSTM(64),
- CONV LSTM(128),
- CONV LSTM(256),
- DENSE(64).

Similar to before, L2 regularization of .0001 is included for all CONV-LSTM layers, we use 3x3, 2 dimensional kernels, and we use ReLU activations. For equivalent reasons to above, we used Huber loss as performance was substantially more consistent. We chose this architecture because it is a common approach to video frame predictions, and our volatility surfaces can be interpreted as videos over a number of days. CONV-LSTM layers are nearly identical to vanilla LSTM layers in architecture, but internally convolutions are used for input-to-state and state-to-state transitions [15]. In particular, the inputs X_t, cell outputs $\mathcal{C}_t$, hidden states H_t, and gates i_t, f_t, o_t are all 3D tensors. The equations for the CONV-LSTM layers are similar to vanilla LSTMs and are included below for reference:

$$\begin{aligned} i_t &= \sigma(W_{xi} \star X_t + W_{hi} \star H_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i), \\ f_t &= \sigma(W_{xf} \star X_t + W_{hf} \star H_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f), \\ \mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} \\ &\quad + i_t \circ \tanh(W_{xc} \star X_t + W_{hc} \star H_{t-1} + b_c), \\ o_t &= \sigma(W_{xo} \star X_t + W_{ho} \star H_{t-1} + W_{co} \circ \mathcal{C}_t + b_o), \\ H_t &= o_t \circ \tanh(\mathcal{C}_t). \end{aligned}$$

Here, "$\star$" denotes the convolution operator and "$\circ$" the Hadamard product.

For our ***Transformer*** architecture we only slightly modify the standard Transformer model [16] by removing the last softmax layer, and setting the last linear layer to have and output dimension of 1. An illustration of the original transformer network architecture can be found in figure 1. We took inspiration from a paper by Dosovitskiy et.al [3] on transformers for visual recognition by first flattening each volatility surface and applying positional encoding. This process serves as our input embedding. The input sequence is thus 20 of these "embedded" vectors. The model input is of size $(batches, 20, tensors * deltas)$ output is of size $(batches, 20, 1)$, and in calculating the loss we use only the very last temporal output of the transformer; all other outputs (19 of them) are ignored. We do not use a mask. We use the Adam optimizer [8] with a step scheduler which reduces the learning rate by 1% after each epoch.

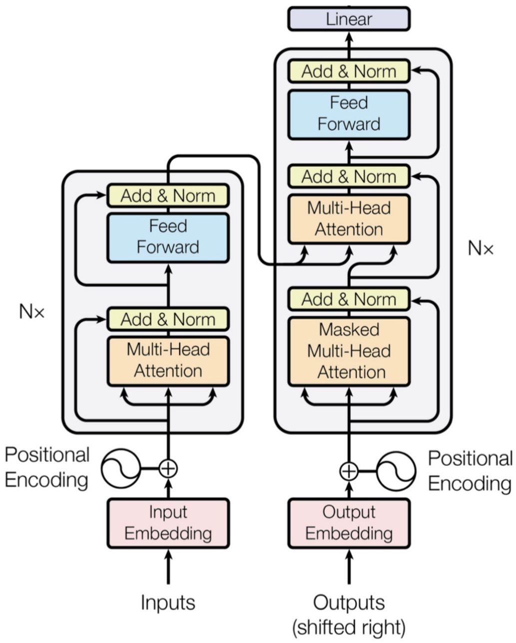

Figure 1. Transformer architecture, minus softmax layer. Credit: Modified from [16].

3.3. Additional methods for the analysis of results

For both the CONV-LSTM and Transformer models, we include histograms of the model residuals on our test dataset. We wish to highlight the dramatic negative skew, which is a byproduct of the VIX index's behavior: it is much more likely to spike upwards than downwards, and hence our models can sometimes severely under-predict, but rarely over-predict the VIX index by a significant margin. This highlights how metrics such as R^2, MSE, and MAE can be misleading.

We also provide saliency maps for the 3D CNN model to show which parts of the volatility surface the trained model pays more attention to in its prediction. Since our task is a regression problem rather than a classification problem, we directly look at the absolute value of the gradient of the input without taking maximum over the channels. We emphasize that the input data is a stack of 20 volatility surfaces, which results in a 4D tensor of implied volatility, delta (amount the option is expected to move based on a $1 change in the underlying), tenor (time until expiration), and lag (date of the surface relative to the prediction target). To analyze this 4D input, we fix one dimension at a time (except for implied volatility) and plot the saliency map for that cross-section.

4. Dataset and Features

In this project we use historical volatility surface data to predict future market VIX values. The historical volatility surface data is taken from OptionMetrics on the S&P 500 Index; an example volatility surface is shown in figure 2. For response data, we use time series price data on the VIX Index.

We first process our volatility surface data into grayscale images, like that shown in figure 3, and then feed these images into our models. The target values are the value of the VIX index at time periods one, five, and 10 days in the future. Our error is the Huber error between the predicted VIX value, and the actual VIX value. We also look at R-Squared, average absolute error, as well as the standard deviation of the error/residual. We also perform comparative analysis of the residuals of our various models. Predicting outlier price moves in the VIX is just as important as predicting the absolute level of the VIX. Therefore, an analysis of the residuals of our models will help shine light on this aspect of model performance.

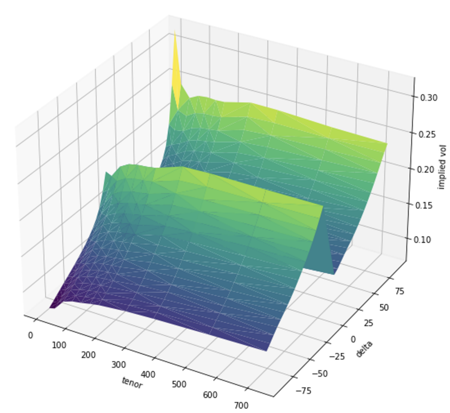

Figure 2. Example of a volatility surface, derived from both calls and put options

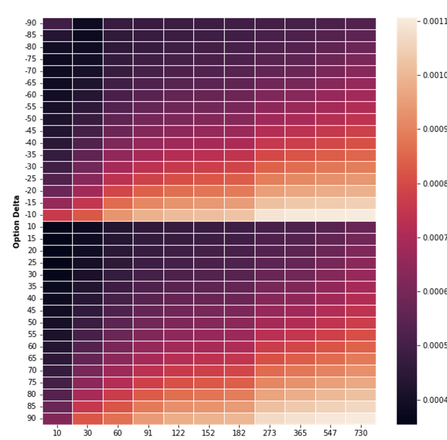

Figure 3. Heatmap representation of a volatility surface

5. Results

We conducted hyperparameter tuning and found that we were able to use consistent hyperparameters across all models (except for the Transformer) with reasonable results and little overfitting. All following results used the Adam optimizer, L2 regularization on all layers (where applicable) of .0001, ReLU nonlinearities, and Huber loss. This regularization was chosen because it achieved the highest validation set accuracy, and Huber loss was chosen to avoid issues with frequent outliers within financial data.

For evaluation, we train the models using only data prior to January 1, 2019; and test their performances on the VIX index after that. We did not conduct cross validation as the usual procedure does not work for time series data due to temporal dependency between observations.

5.1. Lagged linear regression (AR model)

One of the simplest models used in finance to predict future VIX levels is a lagged linear regression. In our testing, the model achieves an R-squared of 0.919 and a has a highly significant coefficient. However, while this model is performant in predicting most VIX values, it completely misses out on large movements since the model is essentially using the latest observation as its prediction.

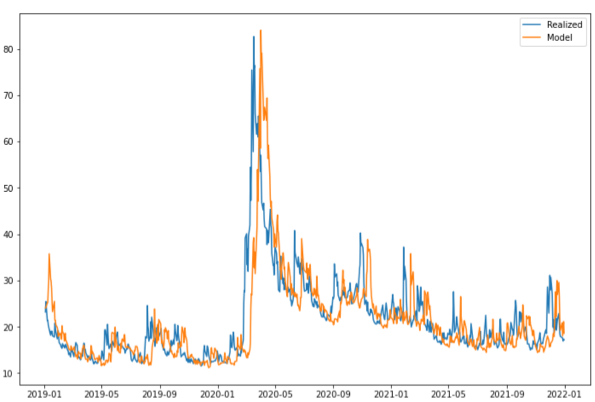

Figure 4. Out-of-sample AR(1) predictions vs realized VIX using 10-day lagged data.

5.2. GARCH model

While GARCH is an industry standard model for predicting volatility of autoregressive time series returns, we found that attempting to predict VIX levels rather than S&P500 volatility directly (which is what GARCH attempts to fit) destroyed much of the predictive power of the model. We were surprised by the performance and the spike in predicted volatility (see figure below) leading to poor error metrics. This model also does not appear to exhibit as much predictive power as we would like to observe.

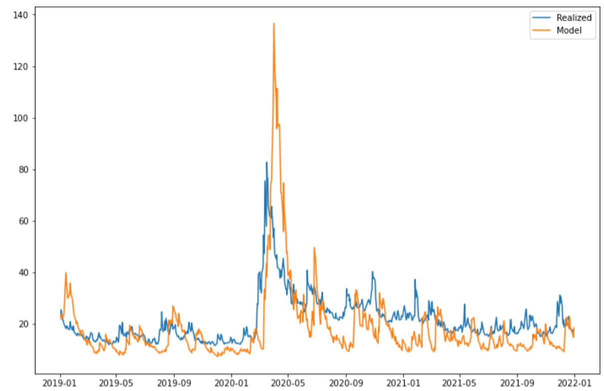

Figure 5. Out-of-sample GARCH(1, 1) predictions vs realized VIX using 10-day lagged data.

5.3. 2D CNN

Our CNN resulted in an out-of-sample fit with a lower R^2 than regression as expected (regression can simply predict the mean with high R^2 but little predictive power), however we observed fairly reasonable fits with high target, prediction correlations. The initial results are reasonable but require additional analysis as we have a low amount of data (compared to traditional applications) due to only having access to daily data from 1996. With access to additional data in the future, we would like to observe whether this pattern persists with different equity surfaces as input. Despite this, however, the CNN holds marginally more predictive power than a simple regression, at least according to these results.

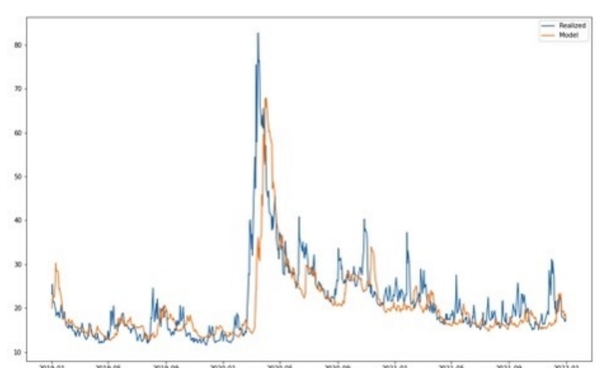

Figure 6. Out-of-sample CNN predictions vs realized VIX using 10-day lagged data.

5.4. 3D CNN

The 3D CNN model performs slightly worse than the 2D CNN model likely due to overfitting. As shown in the analyses about saliency maps below, only the most recent volatility surfaces have much impact on our prediction results. This implies that the additional dimension of a 3D

CNN is unnecessary to achieve a performance on par with that of its 2D counterpart.

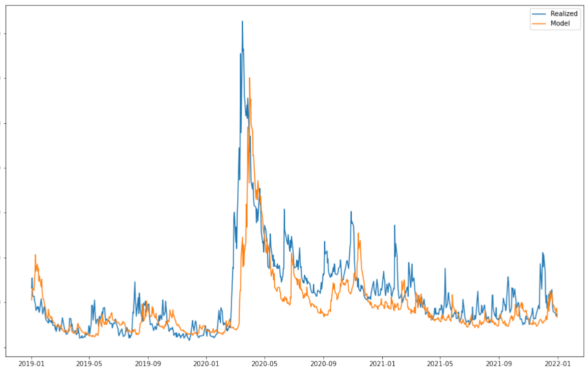

Figure 7. Out-of-sample 3D-CNN predictions vs realized VIX using 10-day lagged data.

The results of our saliency map analysis can be broken down into three parts: the *delta-tenor surface*, the *delta-lag surface*, and the *lag-tenor surface*. For each part, we fix a cross-section in the omitted dimension and analyze how the remaining dimensions affect prediction results.

To better visualize what the saliency maps represent, we also include a 3D plot and its contour of the corresponding cross-section for each part. For the 3D plots and their contours, red is associated with higher implied volatility and blue lower. As for the saliency maps, red indicates higher contribution to the loss relative to the true target and blue lower.

In our example of the *delta-tenor* surface (Figure 8), we observe that with lag fixed, the model pays most attention to near-term (small tenor) close-to-the money (small delta) put options. This seems reasonable as put options protect investors against losses, and can be thought of as insurance against stock losses. Therefore, any large movements in the price of this insurance-like option might be indicative of investor's fearing the future, and hence expecting large future volatility.

The *delta-lag* saliency map of our example (Figure 9) shows that if we fix a given tenor (days until option expiration), the model pays a lot of attention to the large magnitude delta call and put option implied volatility derived from the most recent volatility surfaces: the lag 1 and lag 2 volatility surfaces. This makes sense because the lag 1 and lag 2 volatility surfaces contain the most up-to-date information on investor's expectations, and further, large magnitude delta call and put options are inherently the most sensitive to movements in stock prices. Hence, they will be the first options to be affected by future stock market volatility (movements in the VIX index).

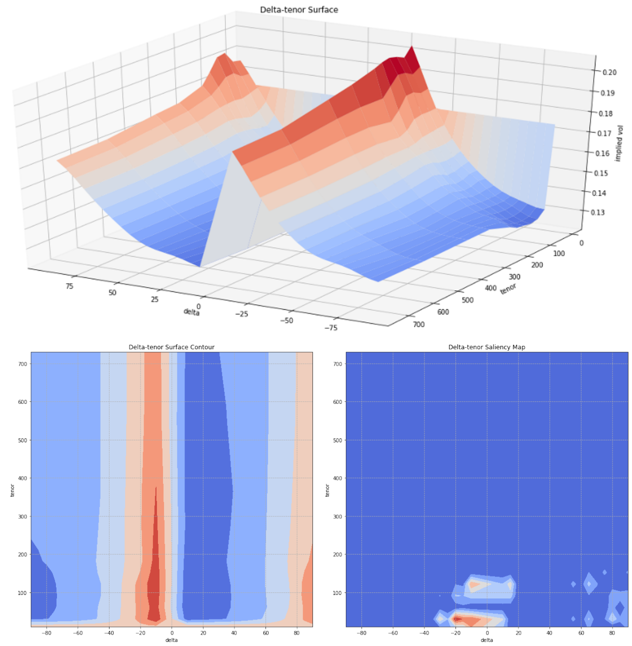

Figure 8. A delta-tenor surface (top), its contour (bottom left), and saliency map (bottom right).

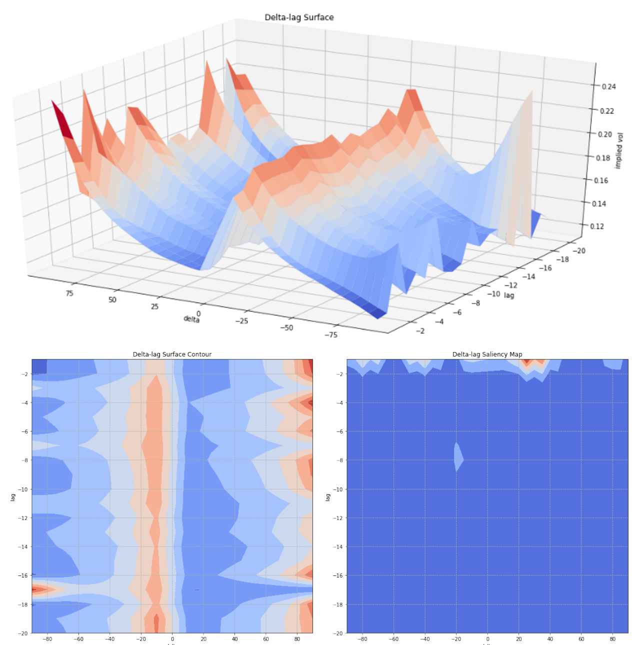

Figure 9. A delta-lag surface (top), its contour (bottom left), and saliency map (bottom right).

Finally, the *lag-tenor* saliency map of our example (Figure 10) shows that for a fixed option delta, the model pays the most attention to the smallest lags and small tenor options (fewest days until option expiration). The former observation is consistent with what we've observed for the delta-lag surface. The later can be attributed to the fact that options close to expiration are more likely to be affected by current events, and are thus more relevant for forecasting the contemporaneous volatility index.

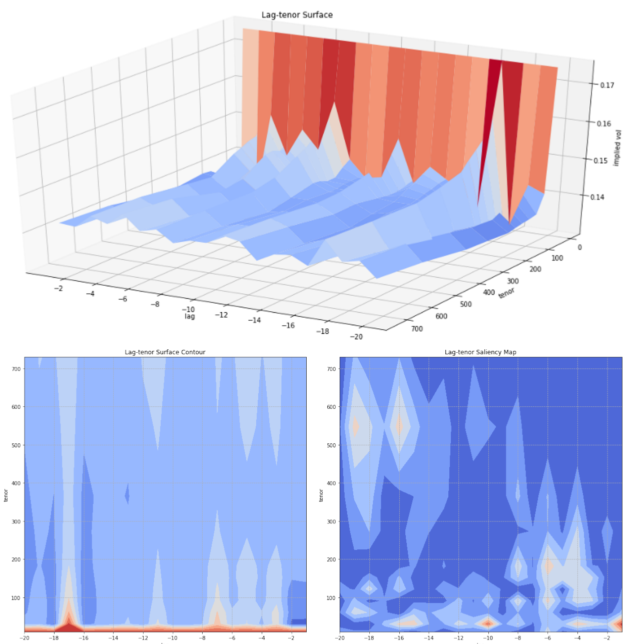

Figure 10. A lag-tenor surface (top), its contour (bottom left), and saliency map (bottom right).

5.5. CONV-LSTM

The CONV-LSTM based approach was more difficult to achieve reliable training convergence than other methods, likely due to the recurrent nature of the LSTM architecture coupled with high noise within the data. However, with additional epochs and a lower learning rate, we were able to achieve consistent train set convergence. We achieved a reasonable train, test accuracy gap and do not believe the data was overfit due to tuning of L2 normalization parameters.

The results are strong for short lags of 1 trading day, the highest of any technique excluding regression, and the prediction test fit visually appears to hold more predictive power than regression which is essentially lagging predictions behind the realized VIX level. Furthermore, the mean average error is remarkably consistent across prediction lags from 1 to 10 days and even declines slightly. This indicates that an LSTM-based approach may be more robust to changes in the desired prediction horizon, which makes sense given the recurrent nature of the model.

Additionally, this model has the desirable property of directly outputting predicted volatility surfaces (rather than only VIX values) if the last layer is changed. Further, it can parse desired inputs of a varying number of days depending on data availability. This is important in finance as data is often missing or incomplete to the point that it cannot be used, so in some cases a lower number of days of surface data must be used.

We also conducted analysis of the residuals as the LSTM approach visually appeared to hold more predictive power than other models. The plot is included below. While visually this may be the case, there is still excess kurtosis in the lower tail of the residual distribution, meaning that the model has difficulty predicting upward spikes in market volatility (spikes of the VIX in a downward direction generally do not occur frequently). Such spikes are generally caused by external factors or market events and are thus exceedingly difficult to predict, but even with this in mind, residual analysis shows that the LSTM fairs only marginally better than other models in that respect.

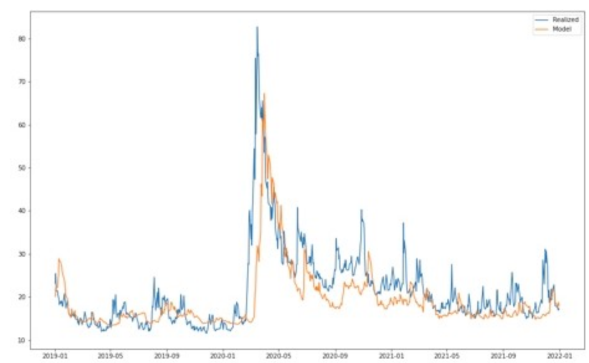

Figure 11. Out-of-sample CONV-LSTM predictions vs realized VIX using 10-day lagged data.

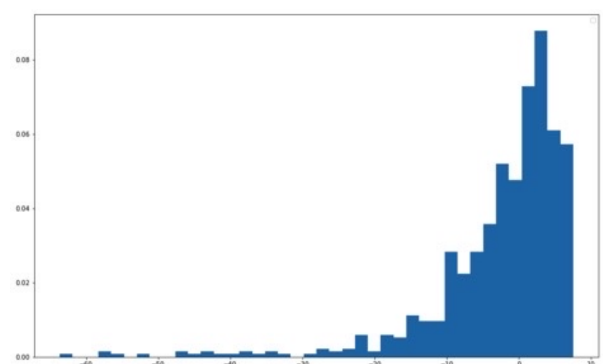

Figure 12. Out-of-sample CONV-LSTM residual distribution using 10-day lagged data.

5.6. Transformer

While the transformer performed well, it was not necessarily the best predictor. It appears to understate the VIX less than the other models, but it is not nearly as sensitive to the higher-frequency fluctuations in the VIX. In other words, its output is lower variance.

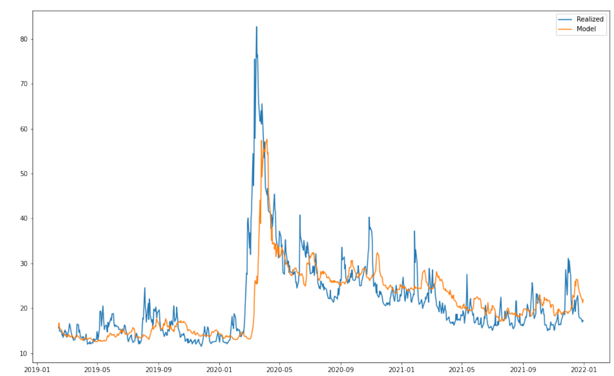

Figure 13. Out-of-sample Transformer predictions vs realized VIX using 10-day lagged data.

The 1 and 5 day forecasts were trained for 500 epochs, while the 10 day was trained for 600 epochs. All models exhibited a curious loss curve, similar to that shown below for the 5 day forecast, see figure 14. The loss curves would plateau for a significant period of time, after which they would start decreasing rapidly. We believe this to be emblematic of a saddle point in the model's optimization problem. It is possibly that better initialization of our model parameters could help with this issue. Sadly, we were limited by time and computing resources in our exploration of the hyperparameter space for the Transformer; given this limitation, we suspect that we might achieve superior performance with a more carefully-chosen set of hyperparameters. In our experience, changes in learning rate, learning rate decay, and number of epochs did not seem to have significant affects beyond those already mentioned. Expanding the transformer depth beyond 1 layer, and better hyperparameter tuning could be areas of future study.

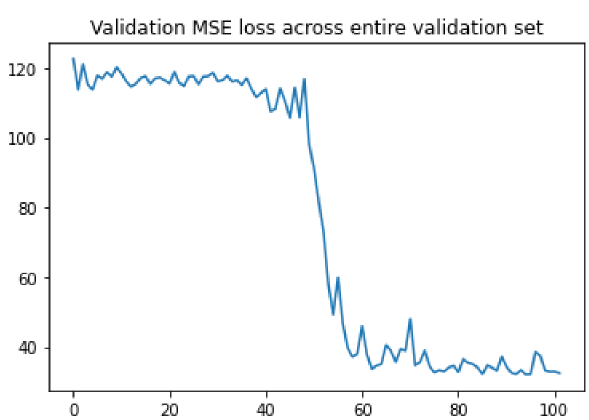

Figure 14. Validation loss for Transformer; 5 day forecast

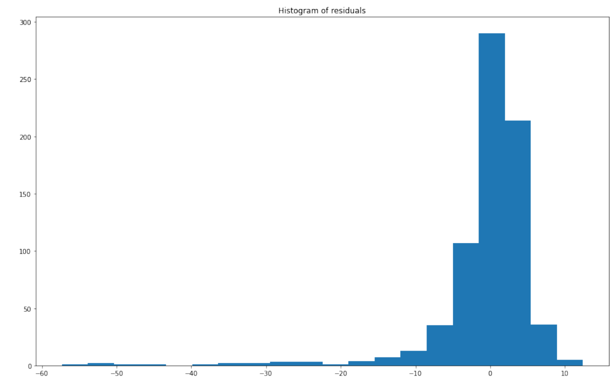

Figure 15. Histogram of Transformer prediction residuals for using 10-day lagged VIX data.

5.7. Comparisons

We summarize the performance of all models across different metrics in Table 1.

Despite lagged linear regression having better fit in many cases, it can be seen in our previous graphs that it is always "off by the number of lags." When the lag is sufficiently large (such as 10, as depicted in the prediction line graphs), it becomes clear that linear regression is delayed in responding to both sudden upward and downward movements. On the other hand, the CNN models, CONV-LSTM, and Transformer model appear to do relatively well in predicting large downward movements of the VIX index following a sudden upward spike.

This phenomenon is especially clear when we look at the large peak in the first quarter of 2020, which is caused by Covid-induced market uncertainty. None of the models were able to predict this sudden event, but the network models were able to adjust to it better than the baseline methods.

6. Conclusion

Unsurprisingly, regression based approaches to volatility modeling are superior for next-day estimates, however, it does seem that other convolutional, recurrent, and attention based approaches have predictive power beyond that of basic regression given longer horizons. While raw error scores are generally lower for regression, residual analysis appears to show that LSTM and Transformer based approaches hold more predictive power in forecasting volatility spikes, which is incredibly important in financial management, often more so than accuracy. We specifically believe that a Transformer based approach for moderate to long time horizons holds promise for volatility forecasting, and our results support this with 10 day forecast results of our Transformer being almost superior across the board in all quantitative metrics. Residual analysis also shows a distribution closer to normal than the other techniques.

Metric \ Model	Regression (AR)	GARCH	2D CNN	3D CNN	CONV-LSTM	Transformer
R^2	0.90	0.08	0.82	0.77	0.89	0.84
	0.69	-0.21	0.62	0.67	0.63	0.67
	0.43	-0.58	0.50	0.43	0.44	0.50
MSE	9.07	80.64	9.81	21.38	10.10	14.67
	27.44	105.83	27.48	29.75	33.80	30.64
	50.04	138.57	45.20	51.51	50.70	46.78
MAE	1.83	6.25	1.86	2.75	8.51	2.70
	3.08	6.81	2.87	3.04	8.31	3.56
	4.05	7.53	3.60	4.10	7.82	3.67
AE STD	2.39	6.45	2.52	3.72	10.06	2.71
	4.24	7.71	4.38	4.53	9.94	4.40
	5.80	9.05	5.67	5.89	9.37	5.77

Table 1. Model performance using lagged data (the rows correspond to 1, 5, and 10 day lags).

Given our results, we would like to conduct future work with more complex/additional Transformer based models for varying financial forecasting problems, including additional volatility problems in non-equities markets. We would also like to gain access to more data and potentially combine surfaces from multiple equities into one larger surface and see if that improves performance. Finally, we would like to explore additional accuracy metrics that encourage predictive, rather than lagged, out-of-sample results, even if numerical error may be higher.

7. Contributions & Acknowledgements

Equal contribution across all team members.
GitHub: github.com/mattj949/CS231n

Alec: Problem formulation, 2D CNN, CONV-LSTM.
Annette: AR/linear regression model, GARCH, 3D CNN, saliency maps, LaTeX beautification.
Matt: Data sourcing, problem formulation, AR/linear regression model, Transformer.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] P. Carr, L. Wu, and Z. Zhang. Using machine learning to predict realized variance. *arXiv:1909.10035*, 2019. 1

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv:2010.11929*, 2020. 2

[4] M. Fernandes, M. Medeiros, and M. Scharth. Modeling and predicting the cboe market volatility index. *Journal of Banking & Finance*, 40:1–10, 2014. 1

[5] C.R. Harris, K. J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T.E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020.

[6] A. Hirsa, M.B. Hadji, J. Osterrieder, W. Cao, Y. Fu, H. Sun, and K. Wong. The vix index under scrutiny of machine learning techniques and neural networks. Available at SSRN: https://ssrn.com/abstract=3796351, 2021. 1

[7] J. Hosker, S. Djurdevic, H. Nguyen, and R. Slater. Improving vix futures forecasts using machine learning methods. *SMU Data Science Review*, 1(4), 2018. 1

[8] D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 3

[9] U. Majmudar and A. Banerjee. Vix forecasting. Available at SSRN: https://ssrn.com/abstract=533583, 2004. 1

[10] M. Martens and J. Zein. Predicting financial volatility: High-frequency time series forecasts vis-à-vis implied volatility. *Journal of Futures Markets*, 25:1005–1028, 2004. 1

[11] J. Osterrieder, D. Kucharczyk, S. Rudolf, and D. Wittwer. Neural networks and arbitrage in the vix. *Digital Finance*, 2(1):97–115, 2020. 1

[12] P. Padhi and I. Shaikh. On the relationship of implied, realized and historical volatility: evidence from nse equity index options. *Journal of Business Economics and Management*, 15(5):915–934, 2014. 1

[13] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A.

Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8024–8035. Curran Associates, Inc., 2019.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[15] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Proceedings of the 28th Conference on Neural Information Processing Systems*, page 802–810, 2015. 2

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 3

[17] S.D. Vrontos, J. Galakis, and I.D. Vrontos. Implied volatility directional forecasting: a machine learning approach. *Quantitative Finance*, 21(10):1687–1706, 2021. 1

[18] M. Yu. *Predicting the volatility index returns using machine learning*. PhD thesis, University of Toronto (Canada), 2017. 1