A Neural Universal Differential Equation (UDE) Approach for Modeling and Forecasting NIFTY 50 Index (Indian Stock Market Index) Prices and Drifts

Ishaan Prasad Kulkarni
Department of Mathematics
University of Warwick
Coventry, United Kingdom
ishaan.p.kulkarni@warwick.ac.uk

Abstract

Modeling equity prices and equity drifts has been a major challenge, as it requires investors and traders to account for unexplained market dynamics. Despite the rapid progress of Scientific Machine Learning frameworks, the intersection of neural-augmented and physical-science-inspired models remains an unexplored avenue for systematic applications in the equity markets domain. In this study, we propose a two-state neural UDE (Universal Differential Equation) model approach to successfully capture market forces, predict equity price and equity drifts, and forecast these values over a fixed forecast period of 10 days. Through hyperparameter optimization and validation metrics, a 20 day training period gave the best overall results, producing low root mean square errors (RMSE) and achieving high directional hit rates (DHR) of 66.6% and 55.5% for prices and drifts. Our approach is adaptable, easily interpretable, and executable. This study opens a door to investigate applications of Scientific Machine Learning frameworks in forecasting tasks for the equity markets domain.

1 Introduction

Scientific Machine Learning is a growing field with a wide range of applications in various fields such as epidemiology, optics, circuit modeling, quantum circuits, fluid mechanics and financial markets[1,2,3,4,5,6,7,8,9,10]. In this paper, we further explore the application of scientific machine learning frameworks to the financial domain, which may be useful to investors while making investment decisions.

Financial market participants have been trying to model equity prices and drifts using traditional analytical and stochastic models. Classical models such as Geometric Brownian Motion, which assume constant volatility and the memoryless property of equity prices, may not be able to capture mean reverting behavior[11,12,13]. Stochastic calculus infused models may be difficult to calibrate in practice[14,15]. Data-hungry pure machine learning models such as recurrent neural networks (RNN) and long short-term memory (LSTM) models when directly applied to data in isolation, tend to capture noise, offer weak generalization and are not inherently intuitive in structure[16, 17, 18].

Therefore, we need a structured and adaptable tool that will improve our ability to model equity prices and drifts and at the same time will capture and preserve the unexplained "physics" or "dynamics" of equity markets, limiting noise. We propose to use a neural UDE[19] which retains the core scientific structure of differential equations, maintains interpretability by establishing clear relationships between involved states, limits noise and captures underlying patterns in data when trained properly.

Machine Learning and the Physical Sciences Workshop, NeurIPS 2025.

2 Methodology

We propose a two-state deterministic UDE with a lightweight, feed-forward neural network that captures those elusive, state-dependent drifts. Because the core structure is analytic, we retain interpretability while achieving the flexibility of modern machine learning.

Let $S(t)$ denote the equity price at time t, and let $\mu(t)$ represent its instantaneous drift. We model their joint evolution as:

$$\frac{dS(t)}{dt} = \mu(t), \tag{1}$$

$$\frac{d\mu(t)}{dt} = -\alpha\big(\mu(t) - \bar{\mu}\big) + N_{\boldsymbol{\theta}}\big(S(t), t\big), \tag{2}$$

with initial conditions $S(0) = S_0$ and $\mu(0) = \mu_0$. Here, $\alpha > 0$ governs the mean-reversion speed back to the long-run level $\bar{\mu}$. The neural network $N_{\boldsymbol{\theta}}$ supplies a data-driven excess drift term that can depend non-linearly on both the price level $S(t)$ and calendar time t.

An estimate of $\mu(t)$ can be obtained from observed prices $\{S_i\}$ at uniform time intervals Δt via the log return:

$$\mu(t_i) \approx \frac{\ln S_{i+1} - \ln S_i}{\Delta t}.$$

Equation (2) decomposes the drift's evolution into two parts: (i) a physics-inspired Ornstein–Uhlenbeck core term, $-\alpha(\mu - \bar{\mu})$, which enforces mean reversion, and (ii) a learned correction $N_{\boldsymbol{\theta}}(S, t)$, which captures nonlinear effects.

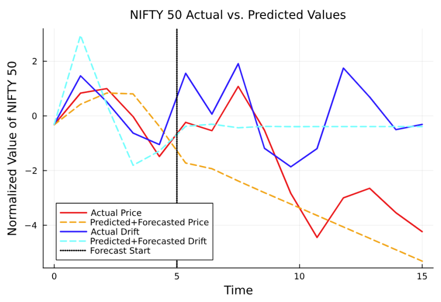
We fix a 10 day forecasting period from 21st July 2025 to 1st August 2025. The training (prediction) periods that we will use are 5, 10, 20, 30, 40 and 50 days before 21st July 2025. These training data segments of logarithmic transformed NIFTY 50 index prices and their drifts are extracted and normalized. The dynamics is modeled through our neural UDE. The model parameters are trained by minimizing a weighted squared error loss with regularization ℓ_2, optimized in two stages: stochastic gradient descent with ADAM for coarse convergence, followed by BFGS refinement. Using the trained parameters, the UDE is simulated forward to generate forecasts of prices and drifts for next 10 days, whose accuracy is assessed via root mean square error (RMSE) and directional hit rate (DHR).

For validation, we also analyze the special case $N_\theta \equiv 0$, comparing the closed-form solution against numerical integration.

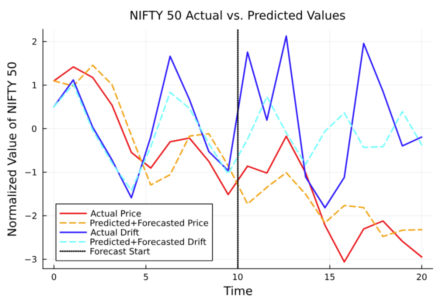
Finally, predicted and actual trajectories are visualized over the training and forecast periods to enable qualitative evaluation.

3 Results and discussion

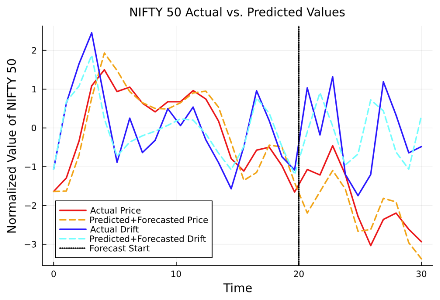
Six cases were considered for the training process of the deep learning-based model.



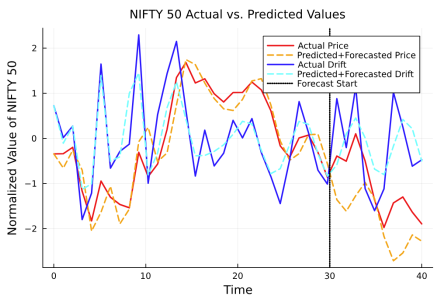
Case (a) Training period: 5 days, forecast period: 10 days



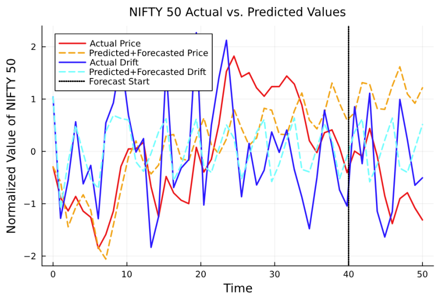
Case (b) Training period: 10 days, forecast period: 10 days



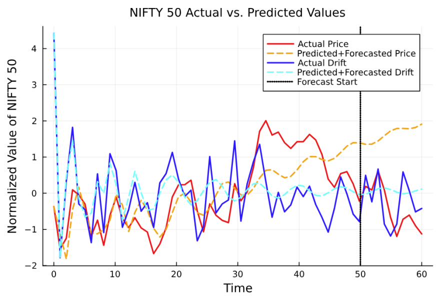
Case (c) Training period: 20 days, forecast period: 10 days



Case (d) Training period: 30 days, forecast period: 10 days



Case (e) Training period: 40 days, forecast period: 10 days



Case (f) Training period: 50 days, forecast period: 10 days

Figure 1: Comparison of actual and predicted NIFTY 50 index log-prices and drifts for different training window sizes. All forecasts are fixed at 10 days, while the training window is varied from 5 to 50 days.

Table 1: Validation metrics across cases (a)–(f).

Validation Metrics	Case (a)	Case (b)	Case (c)	Case (d)	Case (e)	Case (f)
Forecast RMSE (Price)	1.7	0.6	0.5	0.8	1.8	2.2
Forecast RMSE (Drift)	1.3	1.5	1.1	0.8	1.2	0.7
Directional Hit Rate (Price %)	66.6	33.3	66.6	66.6	44.4	22.2
Directional Hit Rate (Drift %)	55.5	22.2	55.5	55.5	33.3	33.3
Closed Form vs. Numerical RMSE on Price	1.7×10^{-5}	2.9×10^{-6}	4.1×10^{-13}	5.6×10^{-9}	2.1×10^{-12}	3.6×10^{-5}
Closed Form vs. Numerical RMSE on Drift	4.4×10^{-6}	4.2×10^{-7}	8.1×10^{-17}	1.6×10^{-11}	8.3×10^{-17}	3.3×10^{-6}

In Case (a) in Figure 1, for a training period of 5 days, we observe that the neural UDE performs poorly in both the training (prediction) and forecasting periods. Even though our model captures the overall decreasing behavior of the price and the sideways behavior of the drifts, it misses short-term directional movements and, in fact, predicts values in the opposite direction.

In Case (b), we notice an improvement in the performance of our model as we increase the training period to 10 days. The model effectively learns the behavior of the prices and drifts and maintains a consistent, reasonable level of accuracy in matching their trajectories in both the training (prediction) and forecasting periods. However, just like Case (a), the model does not closely capture directional movements in price and drift values.

In Case (c), an almost perfect matching can be observed between the approximated and actual values of prices and drifts. We observe that the approximated prices and drifts slightly deviate from actual values in the forecasting period. Overall, results are outstanding. Interestingly, in this case, we used a training period of 20 days, which is exactly twice the length of our forecasting period of 10 days.

Case (d), with a training period of 30 days, delivers results that are nearly comparable to those of Case (c). However, it is evident that there is some abrupt overfitting of the model due to the noisy data in both the training (prediction) and forecasting periods.

In Cases (e) and (f), as we further increase the training dataset/period, there is a noticeable breakdown in the performance of the neural UDE model. The model abruptly overfits in the training period and fails in forecasting future price and drift values. This suggests that when the training period becomes considerably larger relative to the forecast period, the model begins to capture noise and short-term fluctuations rather than the underlying patterns of the data.

Thus, these results contradict the common notion that increasing the training dataset size for neural UDE model necessarily improves forecasting accuracy. Instead of the sheer size of the training data, what matters is optimum training data period and recency(how fresh and recent the data is).

Our observations and insights are further reinforced by Table 1. As a sanity check, from Table 1, we note that when $N_\theta \equiv 0$, the negligible RMSE between its closed-form and numerically integrated solutions confirms the correctness of our solver. Since the stock price is the primary target forecasting variable for market participants (drift is supportive), we crucially discover that Case (c) produces the most optimum results for our neural UDE model, showcasing brilliant forecasting abilities. Case (c) also yields excellent joint directional accuracy, with a DHR of 66.6% for prices and 55.5% for drifts. This highlights that, in practice, the model is not only approximating values closely but also capturing the correct market direction, which is often more important in equity markets forecasting.

4 Architecture summary and computation

Thorough experimentation led to the 2->14->1 neural network framework with tanh activation functions and ADAM and BFGS optimizers. The computing time consumed to arrive at this architecture was 2 hours. The programming language used was Julia and the average computation time for training the model was 4 hours (total computing time for the project was 6 hours) using a m16 R2 Alienware personal laptop with GPU Intel(R)Arc(TM) Graphics. Code for Case (f), data, alpha and mu values used for each case can be found at our **Github Repository.** Similar code is used for other cases.

5 Limitations

For training our model, the input price data was taken from daily closing NIFTY 50 index prices from 1st June 2025 to 21st July 2025. As work on this paper started around this period, this period was selected randomly as a sample model training period. For further validation, we need to run this model for different periods in the past and investigate the results.

6 Broader impact

As per latest Security and Exchange Board of India (SEBI) reports, 91% of Indian retail traders are losing money in trading [20,21]. We propose making our model freely accessible to retail investors, promoting democratization of advanced deep learning tools and leveling the field against large institutions. If efforts are not made to highlight and popularize such high end tools among retail investors, only hedge funds will continue to use them, leading to further unequal distribution of wealth.

7 Further modifications and applications

Figures 4, 5, 6 and 7 for this section can be found on our **Github Repository.**

7.1 Changing forecasting period to 5 days and 10 days

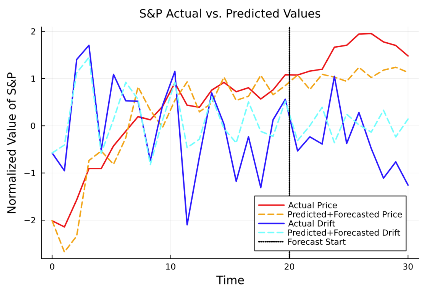
From Figure 4, we find that reducing the forecasting period to 5 days yielded a satisfactory accuracy level overall for both prediction (training) and forecasting of NIFTY 50 prices and drifts. Graphs for the 40 and 50 day training periods were omitted due to computational constraints. Moreover, we believe that the results for this case are comparable to those discussed in the earlier setting of 10 forecasting days and that a training period of 20 days produces the best results overall.

Increasing the forecasting period to 20 days intuitively suggests a deterioration in the performance our neural UDE model. As shown in Figure 5, only one graph could be generated due to computational limitations, and the model performs poorly for this longer-term forecasting task. This suggests that our model is not useful for long term investors and traders.

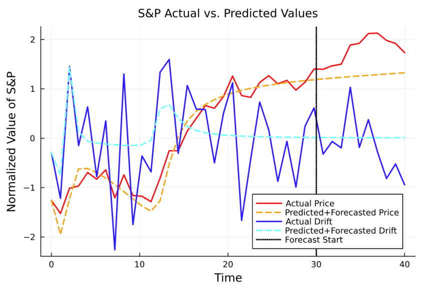
Moreover, since we did not gain any significant improvement in accuracy by altering the forecasting period, we stick to the 10 day forecasting period for further applications.

7.2 Application to S&P 500 index

As observed in Figure 6, our neural UDE model is not very effective in predicting and forecasting S&P 500 index prices and drifts. For reference, we provide a sample of two graphs, as given below.



Case (a) Training period: 20 days, forecast period: 10 days

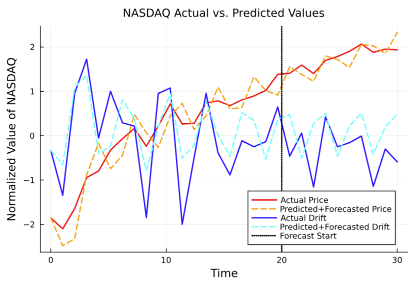


Case (b) Training period: 30 days, forecast period: 10 days

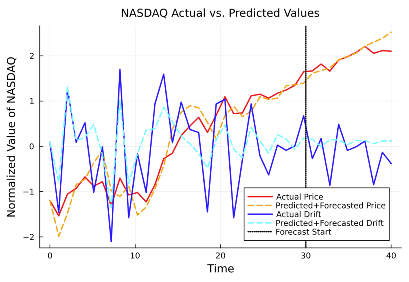
Figure 2: S&P 500 index forecasts for two training periods of 20 and 30 days respectively and a fixed forecasting period of 10 days.

7.3 Application to NASDAQ index

As seen in Figure 7, our model performed extremely well on the NASDAQ index even for longer training periods and we again note that a 20 day training period gave the best result overall. For reference, we provide the graph for the 20 and 30 day training periods, as given below.



Case (a) Training period: 20 days, forecast period: 10 days



Case (b) Training period: 30 days, forecast period: 10 days

Figure 3: NASDAQ index forecasts for two training periods of 20 and 30 days respectively and a fixed forecasting period of 10 days.

8 Conclusion

In this paper, we successfully trained varying segments (5, 10, 20, 30, 40, 50) of normalized NIFTY 50 index prices and drift values through our neural UDE model and synergistically used ADAM and BFGS optimizers to minimize training loss. Then we effectively simulated forecasted price and drift values for the 10 day forecast period and through appropriate validation metrics, concluded that a training period of 20 days produced the best results, successfully capturing underlying patterns in the data and limiting noise. Varying the forecasting period did not lead to any significant improvements. We further applied our model to the S&P 500 and NASDAQ indices, observing poor performance on S&P 500 but strong results on NASDAQ, particularly with a 20-day training period.

References

[1] Nathan Baker, Frank Alexander, Timo Bremer, Aric Hagberg, Yannis Kevrekidis, Habib Najm, Manish Parashar, Abani Patra, James Sethian, Stefan Wild, et al. (2019). Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence. Technical report, USDOE Office of Science (SC), Washington, DC (United States). https://www.osti.gov/servlets/purl/1478744

[2] Raj Dandekar, Chris Rackauckas, and George Barbastathis. (2020). A machine learning-aided global diagnostic and comparative tool to assess effect of quarantine control in COVID-19 spread. Patterns, 1(9). https://www.sciencedirect.com/science/article/pii/S2666389920301938

[3] Raj Dandekar, Shane G. Henderson, Marijn Jansen, Sarat Moka, Yoni Nazarathy, Christopher Rackauckas, Peter G. Taylor, and Aapeli Vuorinen. (2020). Safe blues: A method for estimation and control in the fight against COVID-19. medRxiv, pages 2020–05. https://www.medrxiv.org/content/10.1101/2020.05.04.20090258v1

[4] Raj Abhijit Dandekar. (2022). A new way to do epidemic modeling. PhD thesis, Massachusetts Institute of Technology. https://dspace.mit.edu/handle/1721.1/147364

[5] Weiqi Ji, Franz Richter, Michael J. Gollner, and Sili Deng. (2022). Autonomous kinetic modeling of biomass pyrolysis using chemical reaction neural networks. Combustion and Flame, 240:111992. sciencedirect.com/science/article/am/pii/S0010218022000116

[6] Alexander Bills, Shashank Sripad, William L. Fredericks, Matthew Guttenberg, Devin Charles, Evan Frank, and Venkatasubramanian Viswanathan. (2020). Universal battery performance and degradation model for electric aircraft. arXiv preprint arXiv:2008.01527. https://arxiv.org/abs/2008.01527

[7] Zhilu Lai, Charilaos Mylonas, Satish Nagarajaiah, and Eleni Chatzi. (2021). Structural identification with physics-informed neural ordinary differential equations. Journal of Sound and Vibration, 508:116196. https://www.sciencedirect.com/science/article/abs/pii/S0022460X21002686

[8] Emily Nieves, Raj Dandekar, and Chris Rackauckas. (2024). Uncertainty quantified discovery of chemical reaction systems via Bayesian scientific machine learning. Frontiers in Systems Biology, 4:1338518. https://www.frontiersin.org/articles/10.3389/fsysb.2024.1338518

[9] Allen M. Wang, Darren T. Garnier, and Cristina Rea. (2023). Hybridizing physics and neural ODEs for predicting plasma inductance dynamics in tokamak fusion reactors. arXiv preprint arXiv:2310.20079. https://arxiv.org/abs/2310.20079

[10] Ali Ramadhan. (2024). Data-driven ocean modeling using neural differential equations. PhD thesis, Massachusetts Institute of Technology. https://dspace.mit.edu/handle/1721.1/153671

[11] MathCalculate. (n.d.). Geometric Brownian Motion. https://mathcalculate.com/learning-center/geometric-brownian-motion

[12] QuantMatter. (2025). What is Geometric Brownian Motion? https://quantmatter.com/what-is-geometric-brownian-motion/

[13] Wikipedia. (n.d.). Geometric Brownian Motion. https://en.wikipedia.org/wiki/Geometric_Brownian_motion

[14] Yiran Cui, Sebastian del Baño Rollin and Guido Germano. Full and fast calibration of the Heston stochastic volatility model. https://ar5iv.labs.arxiv.org/html/1511.08718

[15] Yu Feng, Ralph Rudd, Christopher Baker, Qaphela Mashalaba, Melusi Mavuso and Erik Schlögl (2021). Quantifying the Model Risk Inherent in the Calibration and Recalibration of Option Pricing Models. https://www.mdpi.com/2227-9091/9/1/13

[16] Cheng Zhang, Nilam Nur Amir Sjarif, and Roslina Ibrahim (2023). Deep learning models for price forecasting of financial time series: A review of recent advancements: 2020–2022. https://wires.onlinelibrary.wiley.com/doi/full/10.1002/widm.1519

[17] Qi Tang, Tongmei Fan, Ruchen Shi, Jingyan Huang, Yidan Ma, and Wei Yin. Prediction of financial time series using LSTM and data denoising methods. https://arxiv.org/pdf/2103.03505

[18] Thirupathi Kandadi, and G. Shankarlingam (2025). DRAWBACKS OF LSTM ALGORITHM: A CASE STUDY. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5080605

[19] Rackauckas, C., and Nie, Q. (2020). Universal differential equations for scientific machine learning. Proceedings of the National Academy of Sciences, 117(13), 770–777. https://arxiv.org/abs/2001.04385

[20] Vivekam. (2025). 91% of retail traders lost money in F&O. https://www.vivekam.co.in/91-of-retail-traders-lost-money-in-fo/

[21] The Economic Times. (2025). 91% of individual F&O traders lost money in FY25 despite SEBI curbs: Study. https://economictimes.indiatimes.com/markets/stocks/news/91-of-individual-fo-traders-lost-money-in-fy25-despite-sebi-curbs-study/articleshow/122302956.cms?from=mdr