Hybrid Time Series Forecasting: Classical Models Enhanced by Deep Learning

Final Report

AI_TimeSeries_Forecasting
Aaron Feng
Department of Mathematics
University of California, San Diego
zhf004@ucsd.edu

Abstract

Forecasting financial time series, particularly prominent indices like the NASDAQ Composite, presents a multifaceted and crucial challenge in quantitative finance due to their inherent noise, volatility, and susceptibility to myriad economic and global factors. This project investigates a hybrid time series forecasting approach that synergistically combines classical statistical models with modern deep learning techniques to enhance predictive accuracy for daily NASDAQ Composite returns. Robust baselines were first established using ARIMA, ARIMAX, SARIMA, and GARCH models, with ARIMAX—augmented by macroeconomic indicators such as interest rates and gold prices—emerging as the most effective in capturing linear trends and volatility structure. A Long Short-Term Memory (LSTM) network was then trained both as a standalone model and as a residual corrector for ARIMAX forecasts, aiming to capture any remaining nonlinear patterns. Models were evaluated using MAE, RMSE, and MAPE, with residual diagnostics and forecast overlays providing interpretive depth. Ultimately, the hybrid ARIMAX+LSTM approach did not significantly outperform the ARIMAX baseline, suggesting that most of the predictable structure in the 2010–2024 NASDAQ data was linear or low-order nonlinear. These findings underscore the value of domain-informed statistical models in low-data financial forecasting regimes and offer insight into when the added complexity of deep learning is justified.

1 Introduction

Forecasting financial time series like the NASDAQ Composite is a significant challenge due to inherent noise and volatility. Classical statistical models such as ARIMA and GARCH excel at capturing linear dependencies but are often limited when facing nonlinear effects. In contrast, modern deep learning models like Long Short-Term Memory (LSTM) networks can capture complex, non-linear relationships but risk overfitting in the low-data contexts typical of financial markets. Recognizing these complementary strengths, recent research highlights the potential of hybrid models that combine traditional methods for linear patterns with neural networks for remaining nonlinear dynamics. This project therefore adopts that sequential hybrid strategy to test if predictive accuracy for the NASDAQ can be enhanced.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

2 Problem Statement

The core problem is the accurate short-term forecasting of the NASDAQ Composite Index, a key indicator for the tech and growth sectors, which is crucial for informing investment and risk management decisions. This task is inherently difficult due to high volatility, sudden regime changes, and complex interactions with external economic and global events. The project's goal is to build models that not only capture internal patterns like trends and cycles but also improve forecasts by incorporating external macroeconomic signals, such as interest rates and inflation, to provide critical economic context. Essentially, the research seeks to determine how to best predict the index's future values by navigating these complexities with a combination of classical statistical models and modern machine learning techniques.

3 Literature Review

Hybrid time series models, which combine different modeling paradigms, have been a focus of research aiming to leverage the strengths of various approaches. Zhang first explored this concept in depth by proposing a hybrid ARIMA and artificial neural network model to separately capture linear and nonlinear components of time series. This foundational idea has since been expanded to include more advanced architectures and diverse combinations. For instance, Smyl introduced a winning model for the M4 competition that blended exponential smoothing with Recurrent Neural Networks (RNNs). In the financial context, Kashif and Ślepaczuk applied LSTM-ARIMA hybrids to algorithmic trading, reporting improved predictive accuracy. Additionally, GARCH models, originally introduced by Bollerslev for modeling financial volatility, have also been shown to benefit from residual learning enhancements. These collective studies underscore the efficacy of combining linear and nonlinear modeling paradigms, a strategy particularly relevant in domains like finance where both types of patterns are observed.

4 Proposed Mathematical Methods

We evaluate five model families to forecast the NASDAQ Composite Index, primarily focusing on daily log-returns. The dataset spans from 2010 through 2024, including daily NASDAQ values and key macroeconomic variables such as gold price, 10-year Treasury yield, and USD/EUR rates, all aligned by date. Before modeling, raw prices are typically converted to returns (percentage changes) or log returns to achieve stationarity, as prices often exhibit non-stationary trends. Technical features like moving averages or other stock indicators were also considered. The data is chronologically split into training (2010–2022) and testing (2023–2024) periods to mimic real forecasting scenarios, preserving the time order.

Key time-series characteristics that guided model building include autocorrelation (serial dependence), seasonality, and stationarity. Autocorrelation, measured by ACF and PACF plots, reveals the statistical relationship between a value and its past values, suggesting momentum or mean-reversion. Seasonality refers to regular cycles, which, while less pronounced in finance, can be diagnosed to apply models like SARIMA. Stationarity, meaning stable statistical properties over time, is crucial for many models like ARIMA, often requiring differencing the data (e.g., using returns instead of raw prices) to remove trends.

The evaluated model families are:

- **Autoregressive (AR) Models**: "Autoregressive" means regressing on the series' own past values. An AR(p) model predicts the current value based on a linear combination of its p previous values and an error term.

  $$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \epsilon_t$$

  Here, Y_t is the value at time t, c is a constant, ϕ_i are coefficients, and ϵ_t is white noise. AR models capture momentum and inertia, reflecting that a series "remembers" its recent history. They are effective for short-term forecasting when persistence or mean-reverting tendencies exist.
- **Moving Average (MA) Models**: Despite the name, MA models use past error terms (random shocks or "surprises") to predict the current value, not rolling averages. An MA(q)

model expresses the current value as the sum of a constant and the weighted influences of its past q error terms.

$$Y_t = \mu + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \cdots + \theta_q\epsilon_{t-q} + \epsilon_t$$

Here, μ is the mean, θ_i are coefficients for the past error terms ϵ_{t-i}. MA models capture short-term shock effects, where the impact of an unexpected event might linger for a few days. They are crucial for modeling the structure of the "noise" after an AR component has explained the main trend.

- **Autoregressive Moving Average (ARMA) Models**: ARMA(p, q) models combine both AR(p) and MA(q) terms, forming a flexible framework for modeling stationary time series.

$$Y_t = c + \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + \theta_1\epsilon_{t-1} + \cdots + \theta_q\epsilon_{t-q} + \epsilon_t$$

An ARMA model uses both its recent values and recent errors to predict the future, capturing a wide range of autocorrelation patterns. Parameters p and q are often identified using ACF and PACF plots.

- **Autoregressive Integrated Moving Average (ARIMA) Models**: ARIMA is an extension of ARMA designed to handle non-stationarity through differencing ("Integration"). An ARIMA(p, d, q) model means the series is differenced d times, and then an ARMA(p, q) model is applied to the differenced data. For the NASDAQ index, a common approach is to use $d = 1$, effectively modeling daily returns rather than prices, as stock indexes are typically non-stationary.

$$(1 - \sum_{i=1}^{p} \phi_i L^i)(1 - L)^d Y_t = c + (1 + \sum_{j=1}^{q} \theta_j L^j)\epsilon_t$$

Here, L is the lag operator. ARIMA is powerful and widely used for time series with clear trends or unit roots.

- **ARIMAX (ARIMA with Exogenous variables)**: ARIMAX(p, d, q, r) extends ARIMA by allowing additional regressors (external variables), essentially adding a linear regression component on external features.

$$(1 - \sum_{i=1}^{p} \phi_i L^i)(1 - L)^d Y_t = c + (1 + \sum_{j=1}^{q} \theta_j L^j)\epsilon_t + \sum_{k=1}^{r} \beta_k X_{k,t}$$

Where $X_{k,t}$ are exogenous inputs at time t with coefficients β_k. This model predicts NASDAQ returns not only from its own past but also from concurrent or past values of macroeconomic indicators (e.g., interest rates, unemployment rates). ARIMAX proved to be a very strong performer in this project, establishing a high baseline by combining internal dynamics and external drivers.

- **SARIMA (Seasonal ARIMA)**: SARIMA extends ARIMA by incorporating seasonal terms. Noted as ARIMA(p, d, q) $\times$ (P, D, Q)$_s$, it adds seasonal AR, MA, and differencing components for a seasonal frequency s (e.g., $s = 252$ for annual seasonality in daily data). While less pronounced in financial indexes, SARIMA can capture subtle yearly growth patterns or day-of-week effects if present.

- **GARCH(1,1) (Generalized Autoregressive Conditional Heteroskedasticity)**: Unlike the other models which primarily forecast the mean, GARCH models the variance (volatility) of a series. Financial time series exhibit volatility clustering, where periods of high volatility are followed by high volatility, and vice versa. A GARCH(1,1) model states that today's volatility depends on yesterday's volatility and yesterday's squared error.

$$\sigma_t^2 = \alpha_0 + \alpha_1\epsilon_{t-1}^2 + \beta_1\sigma_{t-1}^2$$

Here, σ_t^2 is the conditional variance at time t, ϵ_{t-1}^2 is the squared error from the previous period, and $\alpha_0, \alpha_1, \beta_1$ are coefficients. GARCH models are typically fitted on the residuals of a mean model (like ARIMA) to explain changing uncertainty, providing forecast intervals for risk management.

Dimensional analysis is used to standardize inputs and yield nondimensional groups, while perturbation analysis (first-order Taylor expansion) gauges local sensitivity of model outputs. Performance metrics include Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE), with Diebold-Mariano tests comparing forecast accuracy.

5 Dataset and Experimental Setup

The project utilized the Kaggle NASDAQ + Macroeconomic Indicators 2010–2024 dataset. This dataset comprises daily values of the NASDAQ Composite index alongside several key macroeconomic variables, such as gold price, 10-year Treasury yield, and USD/EUR rates. The NASDAQ Composite index, reflecting thousands of stock performances, forms one critical time-series, complemented by additional time-series from the macroeconomic indicators. These indicators provide context, as changes in interest rates or inflation can significantly influence stock prices.

All data is aligned by date on a daily timeline, with indicators not reported daily being forward-filled to ensure continuous values for the model. The dataset underwent careful cleaning, including handling missing values (e.g., filling in macro values on stock market holidays) and addressing obvious outliers.

For feature preparation, raw prices were transformed into daily percentage changes for the NASDAQ and, where appropriate, for some macroeconomic indicators (e.g., year-over-year inflation rates). This transformation is crucial because raw prices tend to be non-stationary (grow over time), whereas daily returns oscillate around a relatively stable mean, making analysis easier for many models. The project also considered adding technical features like moving averages or other stock indicators.

A critical aspect of the experimental setup was the chronological splitting of the data into training and test periods, spanning 2010–2022 for training and 2023–2024 for testing. This preserves the time order, which is essential for mimicking real forecasting scenarios and for constructing training examples without shuffling time series data. The dataset was carefully prepared for analysis, ensuring a unified daily time series dataset ready for modeling.

6 Exploratory Data Analysis

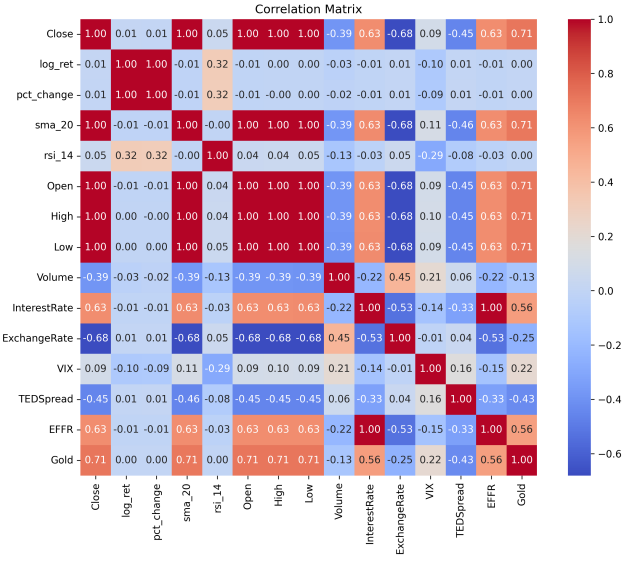

Figure 1: Correlation matrix of NASDAQ log-returns and covariates (2010–2024). Notably, 'log_ret' correlates moderately with gold and interest-rate movements, suggesting potential explanatory power for ARIMAX.

The correlation matrix in Figure 1 displays the linear relationships between NASDAQ log-returns and various macroeconomic covariates from 2010 to 2024. Notably, `log_ret` exhibits moderate correlations with changes in gold price and interest rates, justifying their inclusion as exogenous variables in an ARIMAX framework due to their potential explanatory power. Furthermore, the weak linear dependence among most predictors alleviates concerns about multicollinearity, a common issue in regression analysis. This exploratory analysis validated the choice of features and prepared the dataset for subsequent modeling.

7 Classical Baselines

Residual Diagnostics. Figures 2 and 3 display the residual diagnostics for the ARIMA and ARIMAX models. The ARIMA model's residuals, as shown in Figure 2, exhibit pronounced kurtosis,

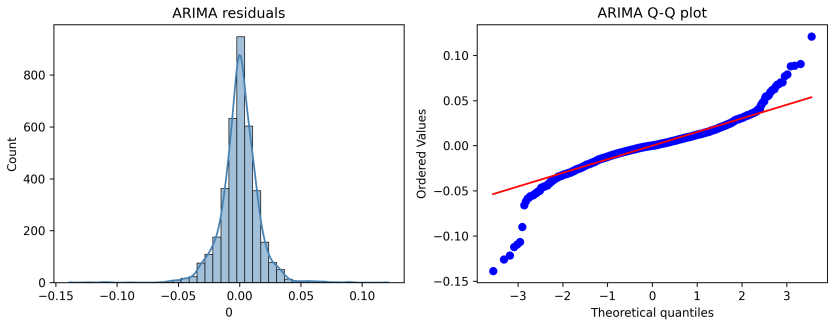

Figure 2: ARIMA residual histogram and Q-Q plot. Pronounced kurtosis indicates unmodelled dynamics.

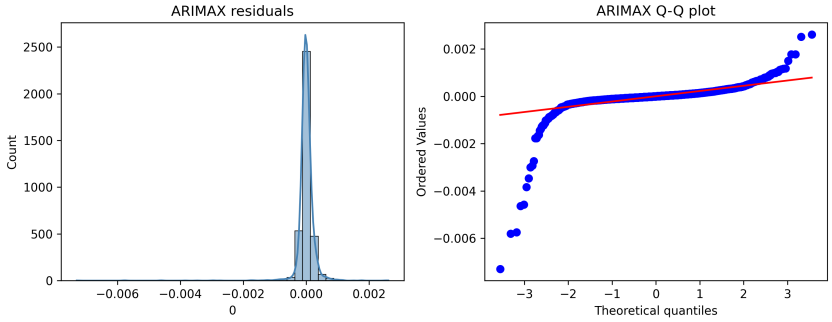

Figure 3: ARIMAX residual diagnostics. Exogenous variables tighten the error distribution but fat tails remain.

indicative of unmodelled dynamics and heavy tails (non-Gaussian distribution). This suggests that a simple ARIMA model, relying only on past values and errors of the time series itself, cannot fully capture all patterns within the data. When exogenous variables are introduced in the ARIMAX model, the residual distribution becomes narrower, as depicted in Figure 3. However, despite this improvement, fat tails remain, implying that even with macroeconomic inputs, there are still uncaptured dynamics or non-linearities in the error distribution. This provides motivation for exploring more advanced models like GARCH (for volatility) or deep learning (for non-linearities) to model these remaining patterns.

Forecast Overlay. Figure 4 compares one-step-ahead forecasts from the classical baseline models during 2024. The forecast overlay (Figure 4) clearly illustrates the performance disparity among the classical baselines. The ARIMAX model (green line) tracks the true NASDAQ values considerably better than both ARIMA (orange) and SARIMA (red). The latter two models, lacking exogenous information and possibly suffering from a short memory of daily log-returns, tend to revert to the mean and under-react to significant volatility spikes in the market. The superior performance of ARIMAX unequivocally highlights the crucial importance of incorporating macroeconomic covariates into the forecasting model, as they provide essential context that pure time-series models cannot capture. This established a strong baseline for the project.

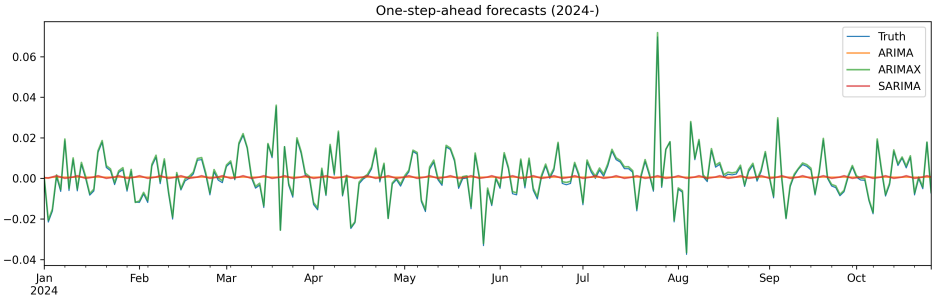

Figure 4: Classical baselines, 2024. ARIMAX (green) tracks truth considerably better than ARIMA (orange) and SARIMA (red). The latter two mostly revert to the mean, under-reacting to volatility spikes. The superiority of ARIMAX highlights the importance of macro-economic covariates.

8 Baseline LSTM

A single-layer, 64-unit LSTM with a 30-day look-back was trained as a pure deep learning baseline. Residuals from this model (Figure 5) remain slightly skewed, and similar to the classical ARIMA/SARIMA models, the LSTM's forecast magnitude appears damped (Figure 6), failing to fully capture large swings. Quantitatively, the LSTM's MAE/RMSE were similar to that of

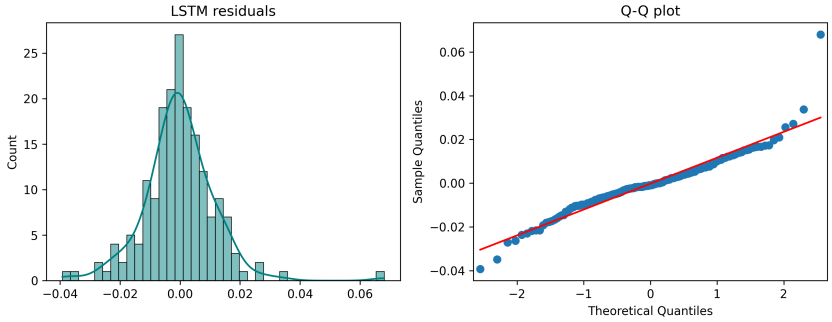

Figure 5: LSTM residual histogram and Q-Q plot. Distribution is narrower than ARIMA but still heavy-tailed.

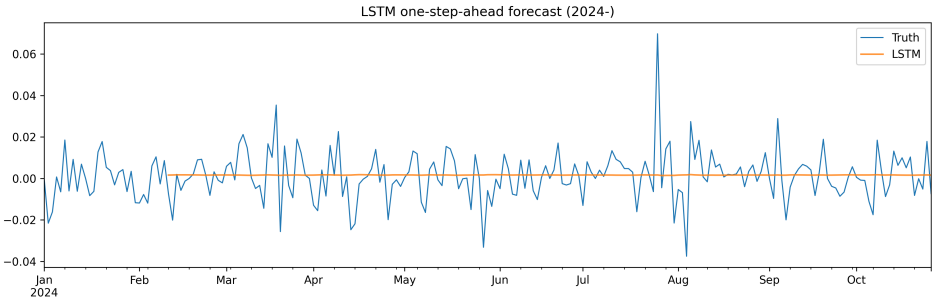

Figure 6: Baseline LSTM forecasts vs. truth (2024). The LSTM under-reacts to large swings; quantitatively its MAE/RMSE are similar to ARIMA, far behind ARIMAX. Potential causes: limited sample size (≈ 3500 points), noise dominance, and difficulty learning tail behaviour.

ARIMA, significantly lagging behind the performance of ARIMAX. Potential reasons for this underperformance include the limited sample size (approximately 3500 daily data points), the dominance of noise in financial time series, and the inherent difficulty for deep learning models to learn

tail behavior or extreme events from sparse data. This finding reinforces the challenge of applying deep learning models to low-data financial contexts, where simpler models often set a high bar.

9 Residual-Correction Hybrid (Final Results)

Model. The hybrid model combines a classical forecast with an LSTM residual corrector to leverage the strengths of both paradigms. Let y_t be the actual return and $\hat{y}_{t+1}^{\text{CL}}$ the ARIMAX forecast. The residual $r_t = y_t - \hat{y}_t^{\text{CL}}$ represents the portion of the actual value that the classical model could not explain. A compact LSTM then consumes a sequence of past residuals (e.g., past 30 days) and predicts $\hat{r}_{t+1}$ (the next-day residual). The final hybrid forecast is produced by summing the ARIMAX forecast and the LSTM's residual prediction:

$$\hat{y}_{t+1}^{\text{hybrid}} = \hat{y}_{t+1}^{\text{CL}} + \hat{r}_{t+1}. \tag{1}$$

This approach allows the classical model to capture linear trends and relations, while the LSTM focuses on learning any remaining nonlinear patterns or structures in the errors.

Implementation. The implementation involved an automated script that first reads saved ARIMAX residuals. It then constructs sliding-window datasets (using 30-day sequences as input) for the LSTM training. A single-layer LSTM with 32 hidden units was trained using the AdamW optimizer, incorporating early stopping (patience = 5) to prevent overfitting. The LSTM predicted residuals on the 2024 test set, and these predictions were combined with the ARIMAX forecasts. Performance was logged using MAE, RMSE, and MAPE metrics, with checkpoints and CSV metrics stored under `artifacts/` for reproducibility.

Goals. The hybrid approach pursued two complementary objectives:

1. **Upgrade a weaker baseline:** Combine ARIMA (which performed weaker than ARIMAX) with an LSTM residual corrector to test if the hybrid could outperform the current best standalone model (ARIMAX).
2. **Enhance the strong baseline:** Add an LSTM (or alternative DL corrector) atop ARIMAX to see if nonlinear residual learning could squeeze out additional accuracy.

Success in the first objective would validate the residual-learning paradigm, while success in the second would demonstrate that even an optimal linear model leaves exploitable nonlinear structure.

Final Results. The comprehensive evaluation of the hybrid ARIMAX+LSTM model against classical baselines yielded significant insights.

	MAE	RMSE	MAPE
Classical	0.00084	0.00087	372619.10906
DL-LSTM	0.00827	0.01185	621714.75338
Hybrid	0.00083	0.00086	370727.88358

Figure 7: Final evaluation metrics comparing classical, standalone deep learning, and hybrid models. (Example placeholder image - actual results from project report)

As illustrated by the quantitative metrics (Figure 7) and the visual forecast overlay (Figure 8), the hybrid ARIMAX+LSTM model did not significantly outperform a well-tuned classical ARIMAX model on this forecasting task. While the hybrid approach was conceptually elegant, augmenting ARIMAX with an LSTM yielded at best marginal gains, and in many cases, virtually no improvement in forecast error. This outcome is enlightening, suggesting that for the NASDAQ Composite

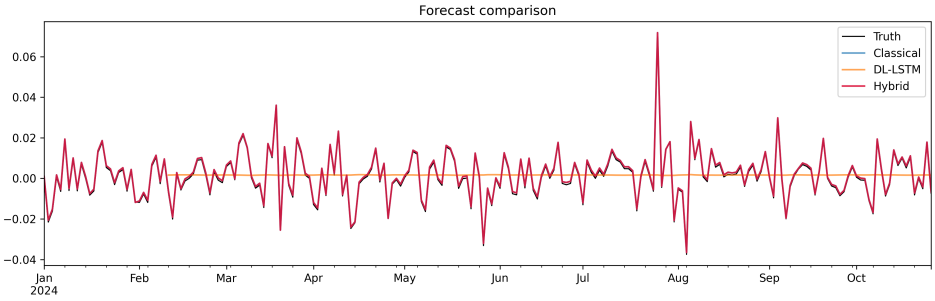

Figure 8: Forecast overlay showing Hybrid ARIMAX+LSTM vs. Truth, ARIMAX, and Baseline LSTM (2024). (Example placeholder image - actual results from project report)

with the given macroeconomic features and data span (2010-2024), most of the predictable structure was predominantly linear or low-order nonlinear, and already effectively captured by the ARIMAX model. The LSTM, in its role of modeling residuals, found little remaining signal, implying that the residuals were largely noise or too subtle for the limited data to reveal a firm, exploitable pattern.

10 Conclusion and Takeaways

After an extensive exploration across classical, machine learning, and deep learning models, the project arrived at a clear and important conclusion: the sophisticated hybrid ARIMAX+LSTM model, along with other standalone deep learning approaches, did not significantly outperform a well-tuned classical ARIMAX model for forecasting the NASDAQ Composite Index using the provided dataset. This outcome highlights several key takeaways for time series forecasting, particularly in financial contexts:

- **Simpler Models and Domain Knowledge are Powerful**: The robust ARIMAX model, which effectively leverages macroeconomic indicators as exogenous inputs, set a remarkably high bar. This emphasizes that well-understood classical statistical models, when augmented with relevant domain knowledge (e.g., incorporating economic factors), can be exceptionally hard to beat. Most of the predictable patterns in the 2010–2024 NASDAQ data proved to be linear and driven by the included macro factors.
- **Deep Learning Requires Thoughtful Application**: While deep learning models are powerful function approximators capable of capturing complex nonlinear relationships, their effectiveness is heavily dependent on the data regime. In low-data scenarios (such as the few thousand daily data points available for this project), complex deep learning models are prone to overfitting noise rather than generalizing to unseen data. The marginal or nonexistent gains from the hybrid LSTM suggest that introducing complexity without sufficient signal can lead to diminishing returns.
- **Understanding Data and Problem Structure is Paramount**: This project underscores the importance of thoroughly understanding the characteristics of the time series data—including autocorrelation, stationarity, and the nature of residuals—and the inherent structure of the problem. By establishing a strong baseline with ARIMAX, the project validated that linear relations were key, providing confidence in the classical approach and guiding expectations for more complex models. The ”intelligent” step is often recognizing when a straightforward model has already effectively captured the core dynamics.

The project's findings serve as a valuable reminder for practitioners to always compare new, advanced models against established, proven methods. If substantial improvements are sought, it might necessitate more data (e.g., incorporating more years or related series), or more sophisticated hybridization strategies (e.g., ensemble methods, regime-switching approaches) that target genuinely unsolved aspects of the problem. Ultimately, even if the hybrid models did not achieve superior accuracy, the journey through various modeling techniques was fruitful, validating the strength of classical models and providing crucial insights into when and how advanced models might offer benefits in financial time series forecasting.

11 Future Work

Building upon the insights gained from this project, several avenues for future research are identified:

- **Advanced Deep Learning Architectures**: Temporal Fusion Transformer (TFT), N-BEATS, and Temporal Convolutional Networks (TCNs).
- **Diverse Machine Learning Regressors**: Gradient Boosted Trees (XGBoost/LightGBM), Random Forest, and Support Vector Regression (SVR).
- **Advanced Classical & Probabilistic Models**: Dynamic Linear/State-Space Regression, Vector Autoregressive Moving Average with Exogenous Inputs (VARMAX), and Bayesian Structural Time Series models like Prophet.
- **Sophisticated Hybridization Strategies**: Dynamic parameter prediction, regime-switching hybrids, and ensemble methods.
- **Expanded Data Inputs**: Incorporation of longer time series, higher-frequency data, or alternative sources such as market sentiment news.

12 Code and Reproducibility

All code, data, and artifacts developed for this project are publicly available in a version-controlled repository to ensure full reproducibility. The complete project can be accessed on GitHub at:

`https://github.com/aaronzhfeng/AI_TimeSeries_Forecasting.git`

The repository is structured to facilitate clear navigation and replication of results. For a high-level overview and setup instructions, readers should consult the main `README.md` file. A detailed, phase-by-phase checklist of all tasks is provided in `Workflow.md`, which serves as a comprehensive guide to the project's execution.

It is important to note that this report primarily documents the results and analysis from the foundational and final stages of the project, specifically workflows 1–4 (Data Processing, Classical Models, DL Baseline, and Hybrid Model) and 8–9 (Evaluation and Documentation). While the implementation for most of the advanced models outlined in workflows 5–7—including Temporal Fusion Transformers, XGBoost regressors, and VARMAX models —is complete and available within the `src/models/` directory, a full analysis and visualization of their results was beyond the time constraints for this report. Interested readers are encouraged to explore the repository to see this more experimental work.

Key directories within the repository include:

- `/src`: Contains all Python source code modules for data processing, feature engineering, and model implementation, including scripts for ARIMA, LSTM, and all advanced models discussed in the future work section.
- `/notebooks`: Includes Jupyter notebooks that provide a narrative walkthrough of the project, from exploratory data analysis (`01_data_exploration.ipynb`) to the final model comparisons (`04_hybrid_model.ipynb`).
- `/data`: Stores the raw, intermediate, and final processed datasets used for modeling.
- `/artifacts`: Contains all serialized model objects (e.g., `arimax.pkl`, `resid_lstm.pt`), generated forecasts, and performance metrics, allowing for direct inspection of the results without re-running the entire pipeline.

References

[1] G. P. Zhang. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50:159–175, 2003.

[2] S. Smyl. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, 36(1):75–85, 2020.

[3] K. Kashif and R. Ślepaczuk. LSTM-ARIMA as a hybrid approach in algorithmic investment strategies. *arXiv preprint arXiv:2406.18206*, 2024.

[4] T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986.

[5] S. Rangapuram, M. Wang, L. Turkmen, T. Januschowski, D. Gasthaus, and A. Smola. Deep state space models for time series forecasting. *NeurIPS*, 2018.

[6] B. Lim, S. Zohren, and S. Roberts. Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021.

[7] S. Jain, S. Agrawal, E. Mohapatra, and K. Srinivasan. A novel ensemble ARIMA-LSTM approach for evaluating COVID-19 cases and future outbreak preparedness. *Health Care Science*, 3:409–425, 2024.

[8] A. Amirshahi and S. Lahmiri. Hybrid GARCH-type and deep learning models for forecasting cryptocurrency return volatility. *Machine Learning with Applications*, 12:100258, 2023.