Deep Learning vs Traditional Quant Strategies: A Comparative Study Using LSTM for Portfolio Optimization

AMOD-5310H: Quantamental Investing and AI

Group 9

Quang Nguyen – ID: 843065

Junyi Chen – ID: 0861615

Yu Zhang – ID: 0862920

Instructor: Prof. Nicholas Faulkner

Table of Contents

Table of Figures

List of Tables

1. Introduction

1.1 Project Objectives

This project aims to synthesize traditional quantitative investment strategies with advanced deep learning techniques to construct and evaluate dynamic portfolio allocation models. Specifically, we implement and test a **CNN-LSTM-based prediction strategy** and compare its performance against established methods developed in Assignments 1 through 3. The core objective is to determine whether modern machine learning models can offer measurable improvements in return forecasting and portfolio optimization compared to traditional statistical approaches.

Our comparative analysis focuses on the following dimensions:

- **Return performance**, measured through cumulative and annualized returns;
- **Risk metrics**, including variance, Conditional Value-at-Risk (CVaR), and risk parity;
- **Model complexity and interpretability**, weighing the trade-offs between advanced predictive capabilities and implementation feasibility.

1.2 Motivation & Relevance

In recent years, the integration of **artificial intelligence (AI)** and **deep learning** into financial modeling has gained significant attention. Recurrent neural networks, particularly **Long Short-Term Memory (LSTM)** models, have shown strong potential in capturing complex, nonlinear, and temporal relationships in financial time series. These models are particularly well-suited to address the limitations of traditional approaches, which often rely on static assumptions and linear frameworks.

Evaluating whether LSTM-based strategies can outperform traditional models such as Modern Portfolio Theory or factor-based strategies is essential for informing both academic research and practical asset management. If machine learning models can deliver superior performance or risk control, they may justify the additional computational cost and complexity they entail.

1.3 Overview of Strategies

This project builds upon the foundational work in Assignments 1–3, which explored a range of classical portfolio construction techniques:

Strategy	Description
Modern Portfolio Theory (MPT)	A classical mean-variance optimization framework that seeks to maximize expected return for a given level of risk based on historical covariances.
Quantile Regression	A robust regression method used to forecast conditional quantiles of returns, enabling momentum-based portfolio construction across different deciles.
Fama-French Five-Factor Model	A factor-based approach that explains asset returns through exposures to market, size, value, profitability, and investment style factors.
Hybrid (MACD-HMM Regime-Switching Model)	A hybrid technique combining MACD indicators for trend detection with Hidden Markov Models to probabilistically detect market regimes.

CNN-LSTM-Based Prediction Strategy	A deep learning approach leveraging convolutional layers for local pattern detection and LSTM layers for temporal dependencies to forecast returns.

Table 1. Summary of methods used across all assignments

Table 1 shows the strategies that will be evaluated side-by-side under consistent market conditions and data periods, using common performance and risk metrics to ensure a fair and robust comparison.

2. Methodology

2.1 Summary of Traditional Strategies (Assignments 1–3)

2.1.1 MPT Portfolio

Modern Portfolio Theory (MPT) Portfolio was constructed based on the principle of mean-variance optimization, aiming to maximize portfolio utility by balancing expected return and risk. In our optimization, the objective function was:

$$\max_{\omega}(\mu^{\mathrm{T}}\omega - \lambda\omega^{\mathrm{T}}\Sigma\omega)$$

Where μ is the expected return vector estimated using exponentially weighted moving average, Σ is the covariance matrix of returns, and λ is the risk aversion coefficient. The portfolio was constrained to fully invest (weights sum to 1), long-only (each weight ≥ 0), and monthly turnover limit to 30% relative to the previous period. The optimization process was performed using a 12-month rolling window. Out-of-sample portfolio returns were calculated using next-month realized returns, while in-sample Sharpe ratios were recorded to assess risk-adjusted performance.

2.1.2 Quantile Regression

The Quantile Regression strategy forecasts was implemented to forecast return distributions rather than just the mean. For each stock, a 6-month rolling window regression were fitted for the 90$^{\text{th}}$ and 10$^{\text{th}}$ percentiles of return against time:

$$Quantile_q(R_{i,t}) = \alpha_q + \beta_q \cdot t$$

The beta spread ($\beta_{90} - \beta_{10}$) was calculated and then served as a momentum-like signal that reflects the asymmetry of return trends. To compare across stocks, beta spreads were standardized into z-scores. The top 5 ranked stocks were chosen to construct a z-score weighted portfolio, and the remaining stocks were initially weighted at zero. To mitigate transaction cost, we applied a soft turnover constraint of 20%. For example, if the overall monthly change in weights exceeded the 20% threshold, each stock's weight changes will only be limited to a max fraction ($\frac{20\%}{Raw\ turnover}$) over the targeted weight change. For comparison, an OLS-based momentum portfolio and an equal-weighted benchmark were also evaluated.

2.1.3 Fama-French Five-Factor Model

In Fama-French Five-Factor model, expected returns were derived from exposures to the five systematic risk factors: market excess return (MKT), size (SMB), value (HML), profitability (RMW), and investment (CMA). For each stock, we estimated factor loadings (betas) using historical excess returns (i.e. return minus risk-free rate):

$$R_{i,t} - R_{f,t} = \alpha_i + \beta_{MKT}MKT_t + \beta_{SMB}SMB_t + \beta_{HML}HML_t + \beta_{RMW}RMW_t + \beta_{CMA}CMA_t + \varepsilon_t$$

These estimated betas were then used to estimate each stock's expected return for the next period, and then covariance matrix was calculated accordingly. The portfolio was optimized monthly by maximizing the Sharpe ratio under the following constraints:

- Long-only weights
- Full investment ($\sum weights = 1$)
- Turnover limit less than 20%

Using updated factor loadings and expected returns, the optimizer tilted allocations toward stocks with strong multi-factor profile.

2.1.4 Hybrid Strategy (Signal Processing & Regime Detection)

The hybrid strategy combined technical indicators with regime switching theory to guide portfolio optimization. The signal processing generates a discrete weekly market signal based on a multi-layered technical signal – bullish (+1), bearish (-1), or neutural (0), which was calculated by the combination of three indicators - MACD histogram smoothing, Relative Strength Index (RSI), and volatility filtering. These indicators are shown in **Table 2**.

Indicator	Description
MACD histogram	Calculated from the portfolio's average return using the difference between the 12-week and 26-week exponential moving averages, further smoothed by subtracting a 9-week EMA signal line. A positive histogram implies upward momentum, while a negative value suggests weakening trends.
RSI	The average gains to average losses over a 14-week rolling window, transformed into $[0,100)$ range.
Volatility	5-week rolling standard deviation of portfolio returns.

Table 2. Technical indicators

A bullish signal (+1) was triggered when the smoothed MACD histogram was positive, the RSI exceeded 50, and portfolio volatility remained below the 70th percentile of its historical range. Conversely, a bearish signal (–1) was assigned when the MACD histogram was negative, the RSI fell below 50, and volatility remained low. If these conditions were not simultaneously met, the signal was considered neutral (0). To reduce short-term noise, the signals are smoothed over a three-week window that only use a signal when it has remained consistent for the past three weeks.

Meanwhile, a two-state Gaussian Hidden Markov Model (HMM) is trained using standardized MACD and MACD signal line. The HMM classifies each week into risk-on (regime 1) and risk-off (regime 0) states. Each week, expected return are estimated via regression against market and momentum factors, and portfolio optimization is performed according to different market regimes. In risk-on regimes, the objective is to maximize the Sharpe ratio, while in risk-off regimes, the objective shift to minimize CvaR (Conditional Value at Risk).

Both optimization objectives are subjected to four constraints:

- Long-only weights

- Full investment ($\sum weights = 1$)
- A soft turnover limit less than 20%
- Minimum momentum exposure level depending on the signal (Bullish: ≥ 0.5, Neutral: ≥ 0.25, Bearish: ≥ 0)

This hybrid strategy enables the portfolio to adapt dynamically across different market conditions, becoming more conservative when risk is elevated, and more opportunistic when conditions are stable.

2.2 LSTM-Based Strategy

2.2.1 Motivation for Using LSTM

Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) models, effectively model financial time series due to their capability to capture temporal dependencies and non-linear patterns (Lai et al., 2018). Traditional financial models often rely on fixed assumptions or static parameters, limiting their adaptability in dynamic market environments. LSTMs, however, are designed to retain relevant past information and selectively forget irrelevant data, enabling robust modeling of noisy, autocorrelated financial returns. Fischer and Krauss (2018) demonstrated that an LSTM-based portfolio achieved significant out-of-sample returns, outperforming traditional forecasting methods, highlighting the LSTM's suitability for predicting stock market behavior.

Convolutional Neural Networks (CNNs) have also shown considerable effectiveness in extracting short-term, local patterns within sequential financial data. Zeng et al. (2023) showed that combining CNN and transformer architectures improved the prediction accuracy by effectively capturing short-term market movements and reducing signal noise. Integrating CNN layers prior to LSTM layers can thus enhance predictive performance by extracting meaningful local features, making it easier for subsequent recurrent layers to model longer-term dependencies (Lai et al., 2018).

Self-attention mechanisms further enhance CNN-LSTM models by dynamically emphasizing critical time steps in input sequences. Vaswani et al. (2017) introduced attention mechanisms that improved the predictive accuracy of sequential models by effectively identifying and weighting important periods within the data. Similarly, Qin et al. (2017) found that incorporating attention mechanisms in recurrent networks significantly enhanced prediction performance on various time series tasks, emphasizing the attention mechanism's role in capturing global temporal dependencies and improving model interpretability.

Thus, integrating CNN, LSTM, and self-attention components creates a powerful framework that effectively combines local pattern extraction, long-term dependency modeling, and dynamic information weighting, leading to improved financial time series forecasting performance.

2.2.2 Model Architecture

To capture the complex temporal patterns in financial return data, we designed a deep sequence learning model that combines convolutional, recurrent, and attention-based components. The model begins with a one-dimensional convolutional layer (Conv1D), which helps **detect short-term patterns** and smooth local fluctuations in the input sequence. This is particularly useful for financial data, where market signals often emerge from localized return structures. Following the convolutional layer, we apply batch normalization to stabilize learning and a max

pooling layer to reduce dimensionality, which enhances computational efficiency and prevents overfitting by summarizing features over time intervals.

The core of the architecture is a Long Short-Term Memory (LSTM) layer, which is well-suited for sequential data and allows the model to **retain long-term dependencies** across time. This is essential for financial time series forecasting, where relationships may span multiple weeks. To further enhance the model's ability to focus on **informative time steps**, we incorporate a self-attention mechanism after the LSTM layer. This layer assigns weights to different parts of the sequence, allowing the model to emphasize the most relevant periods for return prediction.

The output from the attention layer is passed through two fully connected (Dense) layers with ReLU activation functions, interspersed with dropout layers to reduce the risk of overfitting. The final output is a single neuron that provides a regression prediction of the next-week return. The model is trained using the Huber loss function, which balances mean squared error and mean absolute error, making it robust to outliers in financial returns. Optimization is performed using the Adam optimizer, which combines momentum and adaptive learning rate techniques for fast and stable convergence.

2.2.3 Feature Engineering

To enhance predictive performance, we constructed a set of technical indicators derived directly from the weekly return series instead of raw price data. This approach allows the model to learn from more informative features that reflect underlying market behavior, such as trend momentum, volatility, and potential reversals. The first set of indicators includes the Moving Average Convergence Divergence (MACD), its signal line, and the MACD histogram, all computed using exponential moving averages of return series. These indicators are widely used in practice to detect momentum shifts.

We also compute the Relative Strength Index (RSI), which serves as a bounded momentum oscillator capturing overbought or oversold market conditions based on recent return strength. To measure risk, we include rolling volatility, defined as the standard deviation of returns over a 12-week window. Additionally, Bollinger Bands are calculated using a 20-week moving average and standard deviation, providing dynamic upper and lower bounds that signal breakout or mean-reversion opportunities.

Lagged returns from one, two, and three weeks prior are also included as features, allowing the model to directly observe recent return behavior. Finally, we add the lagged weekly return of the S&P/TSX Composite Index (SPTSX) to capture market-wide effects and systematic risk exposure. All features are normalized using MinMaxScaler to ensure that their magnitudes are consistent and suitable for training a neural network. This standardization also aids in convergence and helps prevent dominance of any particular input variable due to scale differences.

2.2.4 Training Process

To evaluate model performance in a way that mimics real-world trading conditions and avoids look-ahead bias, we implemented a **walk-forward training approach**. This method simulates how the model would operate in practice: it is trained on the most recent historical data, used to make predictions for a future period, and then retrained as new data becomes available.

In our setup, the model is retrained every few weeks using a **rolling training window of 104 weeks (approximately 2 years)**. After each retraining, the model forecasts returns for the **next 13**

weeks (a quarter), which serve as the out-of-sample test period. This process repeats throughout the dataset, creating a series of rolling forecasts and allowing the model to continuously adapt to changing market conditions.

Each training sample fed into the model consists of a **12-week sequence of historical features**, including technical indicators and lagged returns. This sequence length is designed to give the LSTM and attention layers enough context to learn temporal patterns in the data. After each training iteration, the model produces predictions for the immediate future, and these predicted returns are collected across all stocks for use in the portfolio optimization stage.

To evaluate the quality of the predictions, we compare the model's output to the actual observed returns using two common performance metrics: the **coefficient of determination (R^2)** and **root mean squared error (RMSE)**. These metrics provide insight into how well the model captures return dynamics and the magnitude of its forecast errors. This process is repeated for each stock individually across the full dataset, and the resulting predicted return matrix is then used in the optimization stage to construct a dynamically rebalanced portfolio strategy.

2.2.5 Prediction and Portfolio Construction

The predicted weekly stock returns from the CNN-LSTM model were used to construct four different portfolio strategies. Each one combined the model's predictive power with practical investment constraints and techniques to adapt to different market conditions.

The first strategy was the **CNN-LSTM Mean-Variance Portfolio**. In this approach, the model's predicted returns were treated as the expected return vector in a classical mean-variance optimization. The covariance matrix was calculated using a 12-week trailing window of historical returns. To ensure the strategy remained practical and avoided excessive trading, a turnover constraint limited the weekly change in portfolio weights to 20%. The portfolio was rebalanced weekly using updated predictions and risk estimates, allowing it to adjust dynamically as market conditions changed.

The second strategy was the **CNN-LSTM Signal-Based Portfolio**. This method added a technical signal component by analyzing the trend in predicted returns over the past four weeks. A linear regression model estimated the slope of the return trend, which was then divided by its volatility to compute a "trend score." This score indicated whether returns were trending upward or downward, and how strong the trend was. The optimizer used this score to set a target exposure to a macro-style factor (such as sensitivity to a moving average), encouraging the portfolio to tilt toward assets with favorable momentum.

Next, the **CNN-LSTM Regime-Aware Portfolio** used Hidden Markov Models (HMMs) to detect market regimes, such as aggressive (growth) or defensive (risk-off) environments. Each week, the HMM produced probabilities for each regime. If the probability of a defensive regime was higher than 60%, the portfolio focused on minimizing downside risk by optimizing for Conditional Value-at-Risk (CVaR) at the 95% level. If an aggressive regime was more likely, the strategy instead aimed to maximize the Sharpe ratio based on the model's predictions. This allowed the portfolio to switch between cautious and growth-focused behavior depending on market signals.

Finally, the **CNN-LSTM Hybrid Score Portfolio** combined both regime detection and signal-based exposure control. In defensive regimes, it focused on minimizing CVaR while keeping a

neutral stance on the signal. In aggressive regimes, it aimed to maximize the Sharpe ratio, but only if it also aligned with strong positive trend signals. The trend score was used to set a smooth exposure target through a controlled mapping function, helping to balance risk and reward based on both market regime and signal momentum.

Together, these strategies show different ways to turn CNN-LSTM predictions into investment decisions. Each one reflects a different philosophy, whether it's maximizing returns, managing risk, capturing trends, or adapting to changing markets. By testing and comparing these approaches, we can better understand how deep learning can support smart, data-driven portfolio management.

2.3 Risk Measures Applied

To comprehensively assess the portfolio strategies in this study, we applied three complementary risk measures: **variance**, **Conditional Value-at-Risk (CVaR)**, and **Risk Parity Deviation (RPD)**. Each captures a distinct dimension of portfolio risk, enabling a multidimensional evaluation of performance under varying market conditions.

Variance is the classical measure of risk in financial theory, quantifying the dispersion of returns around the mean. Higher variance indicates greater volatility and uncertainty in portfolio outcomes. In this project, variance served both as a benchmark for return stability and, in the case of mean-variance optimization, as the core objective function. Grounded in Modern Portfolio Theory (MPT), this approach assumes that investors prefer portfolios with lower volatility for a given level of expected return.

However, variance treats upside and downside deviations equally. To better capture **downside risk**, we employed **Conditional Value-at-Risk (CVaR)** at the 95% confidence level. CVaR measures the expected loss in the worst 5% of outcomes, providing insight into tail risk. This is critical in real-world markets where extreme losses can be disproportionately impactful. We calculated CVaR empirically from weekly return distributions, offering a more conservative complement to variance.

Lastly, we assessed portfolio diversification using **Risk Parity Deviation (RPD)**, which evaluates how evenly risk is distributed across assets. Rather than minimizing total risk, this metric considers the marginal risk contribution of each asset relative to an ideal risk-parity allocation. Although not used as an optimization constraint, RPD served as a diagnostic tool to gauge diversification quality, especially important for strategies, like those based on deep learning, that may unintentionally concentrate risk.

Together, these three measures (variance, CVaR, and RPD) form a robust framework for evaluating portfolios across dimensions of volatility, downside exposure, and risk balance. This multidimensional approach enables more informed comparisons between traditional, smart beta, and machine learning–driven strategies.

3. Data and Implementation

3.1 Data Collection

The dataset for this project consists of **weekly adjusted closing prices** of twelve large-cap stocks listed on the **Toronto Stock Exchange (TSX)**. The selected tickers include a diverse mix of sectors such as finance, energy, technology, and healthcare, aiming to provide a balanced representation of the Canadian equity market. The full list of tickers is: ["RY.TO", "MEG.TO", "ATH.TO", "CNQ.TO", "SIA.TO", "MFC.TO", "TECK-B.TO", "BHC.TO", "GIB-A.TO",

"CLS.TO", "NPI.TO", "BEP-UN.TO"]. These tickers were chosen based on liquidity, sector representation, and availability of long-term historical data.

To ensure a fair comparison with the portfolio optimization strategies developed in **Assignments 1 through 3**, which were evaluated using data beginning in **January 2020**, we aligned all strategy evaluations to start from that same point. However, because the **CNN-LSTM model** requires substantial training history to make reliable out-of-sample predictions, we collected a total of **eight years of historical data**, from **August 2017 to June 2025**. This extended dataset ensures that the model has sufficient lookback periods for training and validation while preserving the comparability of out-of-sample performance across strategies.

In addition to individual stock data, we retrieved the **weekly adjusted closing prices of the S&P/TSX Composite Index** to serve as both a **market benchmark** and an input feature for modeling. All price data was collected using the **Yahoo Finance API** via the yfinance library. Where necessary, we supplemented this with **Bloomberg Terminal** queries using **BQL** for cross-validation or enhanced data granularity.

3.2 Preprocessing

Before modeling, several preprocessing steps were performed to prepare the data for analysis and model input. First, the raw daily prices were resampled into **weekly closing prices**, aligned to **Fridays**, which is a common approach to reduce noise and reflect realistic portfolio rebalancing frequency. From these prices, weekly returns were calculated and used as the primary input for traditional models and for computing technical indicators. After return calculation, any residual NaNs were dropped to ensure consistency across time steps and assets. The result was a clean, aligned return matrix across all 12 stocks and the TSX index.

To enrich the input features for the deep learning models, we computed a variety of return-based technical indicators, such as MACD, RSI, rolling volatility, and Bollinger Bands, using the ta Python package. These indicators help the model detect patterns in momentum, trend strength, and risk, which are not always evident in raw returns. All features were scaled using MinMaxScaler before being passed to the LSTM network to ensure numerical stability and efficient convergence during training.

3.3 Implementation Overview

The project was implemented entirely in **Python**, leveraging a range of popular data science and machine learning libraries. For data manipulation and preprocessing, we used **Pandas** and **NumPy**. Technical indicator computation was done using the **ta package**, and in earlier stages, we experimented with TA-Lib as an alternative. Deep learning models were built using **TensorFlow** and Keras, while scikit-learn was used for scaling, regression metrics, and general utilities.

Each strategy was implemented in modular scripts to support large-scale simulation and retraining. The mean-variance optimization was solved using scipy.optimize.minimize, and CVaR was estimated from historical and simulated return distributions. For evaluating risk parity, we computed marginal contributions to portfolio volatility using matrix calculus applied to the covariance matrix.

Walk-forward training and prediction for the **CNN-LSTM model** were executed on a standard CPU environment with GPU T4x2 acceleration for faster training. Due to the manageable size of the dataset and the weekly frequency of training, GPU usage was not mandatory but helped reduce

model fitting time. Plots and diagnostic metrics such as R^2 and RMSE were visualized using Matplotlib to evaluate the prediction quality across time.

In summary, the implementation combined traditional financial modeling techniques with modern deep learning approaches, resulting in a robust framework for comparing multiple portfolio strategies under consistent data and execution settings.

4. Results and Analysis

4.1 Return and Risk Comparison

To evaluate the effectiveness of the portfolio strategies, we compare both traditional and CNN-LSTM-based models using a consistent set of risk-return metrics. **Table 3** summarizes key indicators, including Compound Annual Growth Rate (CAGR), volatility, Conditional Value-at-Risk (CVaR 95%), Sharpe ratio, maximum drawdown, total return, tracking error, and Risk Parity Deviation (RPD). Traditional strategies like **MPT Optimized** and **Quantile Regression Score** achieve strong returns, with MPT showing the highest **CAGR (0.8522)** and **total return (23.15)** but also the highest **volatility** and **CVaR**. Hybrid strategies such as **Hybrid (Regime + Signal)** and **Regime Only** deliver favorable **Sharpe ratios (~1.43–1.46)** with more balanced risk profiles. Notably, **CNN-LSTM-based portfolios** show competitive performance in terms of return and risk-adjusted measures, especially the **CNN-LSTM Hybrid Portfolio**, which yields a high **Sharpe ratio (1.3241)** and solid **CAGR (0.6896)**. However, most deep learning models exhibit slightly higher **RPD values**, indicating less effective risk diversification. Overall, hybrid and regime-aware strategies outperform equal weighting, while the integration of machine learning adds modest gains with increased complexity.

Strategy	CAGR	Volatility	CVaR 95%	Sharpe Ratio	Max Drawdown	Total Return	Tracking Error	RPD (↓ Better)
Equal Weight	0.2234	0.2527	-0.1409	0.8227	-0.3031	1.83	0.1476	0.60
MPT Optimized	**0.8522**	0.4989	-0.2102	**1.4243**	-0.3673	**23.15**	0.5387	1.73
Quantile Regression Score	0.2995	0.3051	-0.1524	0.9310	-0.3649	2.87	0.2160	1.08
Hybrid (Regime + Signal)	**0.7949**	0.4494	-0.1211	**1.4556**	-0.3770	**17.03**	0.3968	1.55
Signal Only	0.4020	0.2918	-0.0895	1.2025	-0.3331	4.32	0.2305	1.35
Regime Only	0.7568	0.4411	-0.1191	1.4259	-0.3728	15.22	0.3830	1.56
CNN-LSTM Regime Portfolio	0.6556	0.4389	-0.1200	1.2995	-0.3643	10.9800	0.3950	1.80
CNN-LSTM Mean-Var Portfolio	0.4216	0.4611	-0.1255	0.9261	-0.4979	4.6942	0.4046	1.80

CNN-LSTM Signal Portfolio	0.4584	0.4679	-0.1335	0.9951	-0.4793	5.4616	0.4496	1.73
CNN-LSTM Hybrid Portfolio	**0.6896**	0.4491	-0.1194	**1.3241**	-0.3657	**12.2415**	0.4044	1.79

Table 3. Performance Comparison of Portfolio Strategies

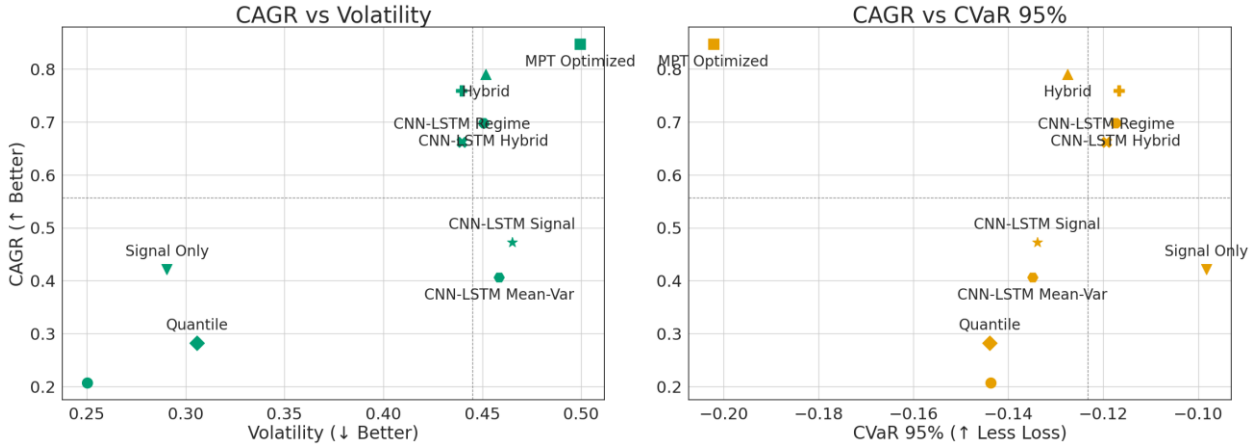
Figure 1 assesses portfolio strategies by comparing **Compound Annual Growth Rate (CAGR)** against three key risk metrics: **Volatility**, **CVaR 95%**, and **Risk Parity Deviation (RPD)**.

In the **CAGR vs Volatility** panel (top-left), the ideal strategies appear in the upper-left quadrant. **MPT Optimized** and **Hybrid (Regime + Signal)** dominate with the highest returns, while maintaining volatility at acceptable levels. **CNN-LSTM Regime** and **CNN-LSTM Hybrid** also perform well, though with slightly more volatility. Simpler models like **Equal Weight** and **Quantile** offer lower returns with lower risk.

In the **CAGR vs CVaR 95%** panel (top-right), **MPT Optimized** achieves the highest CAGR but suffers from the worst tail risk (CVaR = –0.2102). **Hybrid** stands out with strong return and more moderate downside risk. **Signal Only** is the most conservative in terms of CVaR, but yields modest growth.

In the **CAGR vs RPD** panel (bottom), **Equal Weight** exhibits the best diversification (lowest RPD) but the lowest return. **Hybrid** and **Regime-only** achieve strong CAGR with moderately high RPD, suggesting a more concentrated risk. CNN-LSTM models show moderate returns but relatively high RPD, indicating weaker risk parity.

Overall, the **Hybrid** strategy offers the best return-risk trade-off, the highest Sharpe ratio (1.46), strong CAGR (0.8), controlled CVaR, and lower RPD than all other strategies with CAGR > 0.5. While **MPT Optimized** delivers the highest returns, it consistently incurs the most risk. **CNN-LSTM Regime** and **Hybrid** approaches show promising results, balancing machine learning with regime awareness for improved portfolio performance. However, this comes with elevated **Risk Parity Deviation**, indicating less effective diversification.



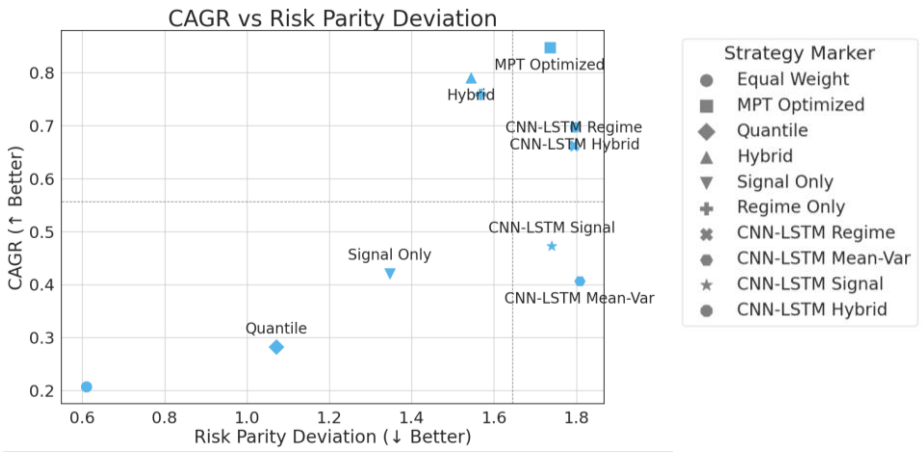


Figure 1. Quadrant-Based Multi-Metric Trade-Off Analysis of Portfolio Strategies

Figure 2 presents the cumulative returns of traditional smart beta portfolios and the LSTM-enhanced mean-variance portfolio. The **MPT Optimized** strategy achieved the highest cumulative return over the sample period, particularly after 2023, though it exhibited considerable volatility and drawdowns. In contrast, the **Equal-Weight** and **Quantile Regression Score** strategies delivered modest and steady growth, serving as benchmarks for passive allocation and alternative scoring methods.

The **CNN-LSTM Mean-Variance Portfolio**, which integrates deep learning predictions into a classic optimization framework, consistently outperformed the Equal-Weight and QR Score approaches. Although it did not surpass the MPT Optimized strategy in absolute return, it demonstrated enhanced risk-adjusted performance, validating the potential of deep learning in guiding portfolio construction.

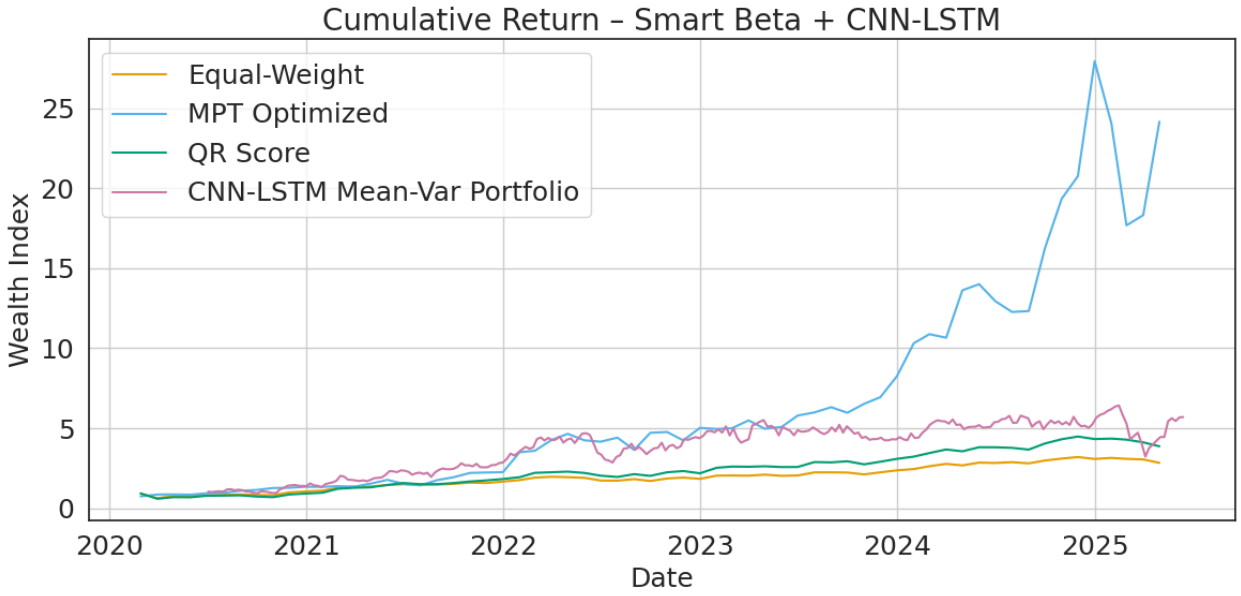


Figure 2. Cumulative Return Comparison between Smart Beta Portfolios

Figure 3 showcases the performance of smart alpha portfolios incorporating regime identification, signal processing, and deep learning. The **Hybrid (Regime + Signal)** strategy

outperformed all others, ending the period with the highest wealth index. This confirms the efficacy of combining macroeconomic regimes and momentum-based signals for adaptive allocation.

The **CNN-LSTM Hybrid Score Portfolio** closely tracked the Hybrid strategy's trajectory, suggesting that machine learning adds incremental value when overlaid on a strong structural foundation. Similarly, both the **CNN-LSTM + Regime Portfolio** and **CNN-LSTM Signal Portfolio** delivered robust performance, validating the standalone utility of each approach.

Meanwhile, the **Signal Only** and **Regime Only** portfolios, though simpler in construction, also achieved competitive results, outperforming naive strategies and demonstrating effective downside protection and responsiveness to market shifts.

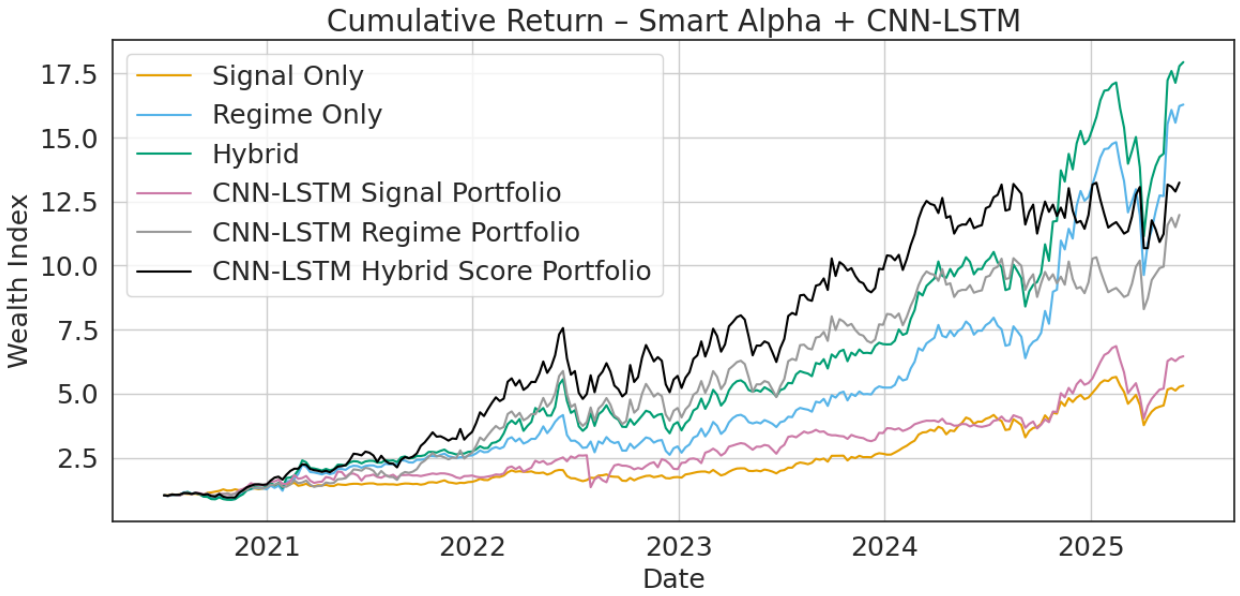


Figure 3. Cumulative Return Comparison between Smart Alpha Portfolios

The comparative analysis of portfolio strategies reveals that the **Hybrid (Regime + Signal)** portfolio outperformed across most metrics, achieving the highest Sharpe ratio (1.46) and strong CAGR (0.79), highlighting its superior balance of return and risk control. While the **MPT Optimized** strategy delivered the highest total return (23.15), it incurred higher volatility and the worst downside risk (CVaR: -0.21), underscoring the trade-off between return and stability. Among smart beta strategies, the **Signal Only** portfolio exhibited the best downside protection (CVaR: -0.0895), and **Regime Only** achieved strong risk-adjusted performance. Deep learning models further enhanced outcomes when combined with structural signals: the **CNN-LSTM Hybrid Score Portfolio** delivered a Sharpe ratio of 1.32 and robust cumulative returns, validating the effectiveness of hybrid learning-finance integration. In contrast, the standalone **CNN-LSTM Mean-Variance Portfolio** lagged due to higher drawdowns and limited diversification. Overall, the results emphasize that integrating domain-informed signals and regimes into deep learning models yields more adaptive, resilient, and high-performing portfolios.

4.2 Signal Stability and Turnover

Weight Turnover Comparison

To evaluate how frequently strategies adjust their holdings, we computed the average weekly turnover for each model. Traditional models like MPT, Quantile Regression, and Equal Weight exhibited more stable allocations due to longer rebalancing periods or static weights. In contrast, Hybrid (QR and Signal), Signal-only, Regime-only, and CNN-LSTM Signal Portfolio strategies exhibited higher turnover, reflecting their responsiveness to predicted returns.

Strategy	Rebalancing Frequency	Avg Turnover
Equal Weight	Static	0.00
MPT Optimized	Monthly	~0.09 (Monthly)
Quantile Regression Score	Monthly	~0.10 (Monthly)
Hybrid (QR + Signal)	Weekly	~0.13
Signal Only	Weekly	~0.12
Regime Only	Weekly	~0.13
CNN-LSTM Mean-Var Portfolio	Weekly	~0.07
CNN-LSTM Regime Portfolio	Weekly	~0.05
CNN-LSTM Signal Portfolio	Weekly	~0.15
CNN-LSTM Hybrid Score Portfolio	Weekly	~0.04

Table 4. Average Turnover across Strategies

Transaction Cost Implications:

Although dynamic strategies tend to respond faster to market signals, they incur **higher transaction costs** due to frequent weight adjustments. This trade-off must be considered when selecting a portfolio strategy in real-world applications. For instance, **CNN-LSTM Signal Portfolio**, while responsive, showed elevated turnover and higher drawdowns.

4.3 Computational Complexity

4.3.1 Training Times

Traditional portfolio strategies such as Modern Portfolio Theory (MPT), Quantile Regression, and the Fama-French model are highly efficient in terms of computation. These models typically complete execution within seconds to a few minutes on standard hardware, which makes them practical for rapid backtesting and deployment.

In contrast, CNN-LSTM models require significantly more computational resources. This is especially evident under the walk-forward retraining framework used in this study. Each retraining iteration, covering approximately 104 weeks of data, required between 60 and 120 seconds per stock when run on an NVIDIA T4 GPU. When aggregated across all stocks in the portfolio, the total training time reached approximately 10 hours. This highlights the need for access to high-performance computing infrastructure when working with deep learning models in finance.

4.3.2 Prediction Latency

Once trained, the CNN-LSTM model's prediction latency is negligible, less than 1 second per stock, making it suitable for weekly rebalancing cycles.

4.3.3 Model Explainability

Interpretability is a major differentiating factor between traditional and deep learning models.

- MPT uses explicit inputs such as expected returns and covariance matrices, which directly influence portfolio weights.
- The Fama-French model provides clear explanations for asset performance using predefined economic factors such as size and value.

On the other hand, CNN-LSTM models function as complex, nonlinear systems that are more difficult to interpret. Although attention mechanisms can highlight which time periods the model considers important, they do not fully explain the influence of each input feature on the final prediction. This limited transparency introduces a trade-off. While deep learning models may offer improved predictive accuracy, they often lack the interpretability that traditional financial models provide.

4.4 Trade-off Discussion

4.4.1 Complexity vs. Performance

While deep learning models such as CNN-LSTM introduce a high level of computational and implementation complexity, the performance gains they deliver can justify the added cost, particularly when combined with structural signals. For instance, the **CNN-LSTM Hybrid Score Portfolio** outperformed many traditional strategies in terms of Sharpe ratio and consistency of returns. However, the **CNN-LSTM Mean-Variance Portfolio**, which lacked structural guidance, underperformed, suggesting that deep learning models are not universally superior on their own. The marginal benefit of using LSTM over simpler models depends heavily on how well it is integrated with domain knowledge, such as regime classification or momentum signals. For investors with limited resources or shorter investment horizons, the additional complexity of deep learning may not be worthwhile unless the models are properly augmented and explainable.

4.4.2 Use Case Suitability

The choice of strategy should align with the investor's objectives and constraints.

- **Equal-Weight** and **Quantile Regression Score** portfolios are suitable for passive or rule-based investors seeking simplicity and broad diversification.
- **MPT Optimized** portfolios may appeal to return-seeking investors who are comfortable with higher volatility and tail risk, especially in bullish market conditions.
- **Signal Only** and **Regime Only** smart beta strategies offer a good middle ground, delivering improved risk-adjusted performance with interpretable logic and lower implementation burden.
- **Hybrid (Regime + Signal)** and **CNN-LSTM Hybrid Score** portfolios are most appropriate for advanced users aiming for robust, adaptive strategies capable of navigating varying market regimes. These are particularly well-suited for institutional or data-driven investors with the capacity to manage and maintain model infrastructure.

Ultimately, there is no one-size-fits-all solution, each strategy serves a different purpose depending on risk tolerance, data availability, and implementation capacity.

5. Conclusion

Summary of Key Findings:

This project conducted a comprehensive evaluation of portfolio construction methods, comparing traditional, smart beta, and deep learning–enhanced strategies. Among all approaches, the **Hybrid (Regime + Signal)** portfolio achieved the highest **Sharpe ratio** of 1.46 and a strong **Compound Annual Growth Rate (CAGR)** of 0.7949. It consistently performed well across all risk-return metrics, including a relatively moderate **Risk Parity Deviation (RPD)** of 1.55, outperforming both classical optimization and machine learning–only approaches.

The **MPT Optimized** portfolio achieved the highest **total return** (23.15) and **CAGR** (0.85) but suffered from elevated **volatility**, the most negative **CVaR** (–0.2102), and a high **RPD** (1.73), indicating concentrated and tail-risk exposure.

All **CNN-LSTM–based** strategies (including **CNN-LSTM Hybrid**, **Regime**, **Signal**, and **Mean-Variance**) exhibited **high RPD values** (≥1.73), signaling poor risk diversification. Nonetheless, the **CNN-LSTM Hybrid Score** strategy delivered a respectable Sharpe ratio of 1.32, demonstrating that blending deep learning with structural elements such as trend signals and regime filters improves performance compared to using predictions alone.

In contrast, the **CNN-LSTM Mean-Variance** portfolio, which relied solely on forecasted returns, underperformed across all key metrics. These findings suggest that integrating financial context into data-driven models yields more effective and balanced portfolio strategies.

Complexity and Justification of LSTM:

The CNN-LSTM models introduced substantial computational complexity and training overhead. However, their performance, particularly in hybrid configurations, supports their use for investors seeking adaptive and data-driven allocation strategies. When paired with structured alpha signals, LSTM-based models can outperform simpler models on a risk-adjusted basis. For investors or institutions with access to computational resources, the added complexity may be justifiable. For others, traditional approaches like MPT or Quantile Regression may remain preferable due to their transparency, ease of implementation, and lower operational cost.

Computational Constraints and Model Scalability:

A notable constraint in this study was limited access to GPU resources, which restricted the number of epochs per retraining iteration and limited the retraining frequency. The model was trained using rolling windows of approximately 104 weeks, with retraining every 13 weeks. Each iteration required approximately 35 to 60 minutes per stock, resulting in a total training time of nearly 10 hours.

With increased computational capacity, the model could be trained for more epochs or retrained more frequently. This improvement could lead to higher predictive accuracy and better portfolio responsiveness. The CNN-LSTM Mean-Variance strategy, which lacked auxiliary structure, may have been particularly affected by this limitation, as it relied entirely on predictive strength.

Strategic Recommendations

For retail or passive investors, simple strategies such as Equal-Weight or Quantile Regression are easy to implement and offer modest diversification benefits.

For investors willing to accept higher volatility in pursuit of higher returns, the MPT Optimized portfolio provides strong capital appreciation potential.

For those prioritizing stability and responsiveness, strategies such as Regime Only and Signal Only offer adaptive risk control without the need for deep learning.

For advanced practitioners or institutional investors, hybrid strategies that integrate LSTM-based forecasts with domain-specific signals and regime filters appear to deliver the most favorable balance of return and risk.

Future Research Directions

While the findings are promising, there remain opportunities for further exploration. Future research could include expanding the asset universe beyond twelve stocks, testing alternative model architectures such as transformers, and evaluating different forecasting horizons. Improvements in explainability techniques and integration of macroeconomic indicators could further enhance model utility. Lastly, scaling the infrastructure to support faster retraining and deeper model tuning could improve out-of-sample performance and portfolio adaptability.

5. References

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research, 270*(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054

Lai, G., Chang, W.-C., Yang, Y., & Liu, H. (2018). Modeling long- and short-term temporal patterns with deep neural networks. *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*, 95–104. https://doi.org/10.1145/3209978.3210006

Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., & Cottrell, G. W. (2017). A dual-stage attention-based recurrent neural network for time series prediction. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, 2627–2633. https://doi.org/10.24963/ijcai.2017/366

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*, 5998–6008. https://doi.org/10.48550/arXiv.1706.03762

Zeng, Z., Kaur, R., Siddagangappa, S., Rahimi, S., Balch, T., & Veloso, M. (2023). Financial time series forecasting using CNN and Transformer. *arXiv preprint arXiv:2304.04912*. https://doi.org/10.48550/arXiv.2304.04912

6. Appendices

A. Use of Generative AI

Category	Details
Tool Used	ChatGPT (OpenAI)
Prompts Given	- Requests for help debugging Python code - Clarifications on portfolio optimization methodologies
Actions Taken	- Generated Python code snippets for CNN-LSTM implementation - Provided guidance on debugging and optimizing financial models
Model Focus	CNN-LSTM for financial time series forecasting
Technical Areas Covered	- Time series feature engineering (MACD, RSI, MA, etc.) - CNN-LSTM architecture - Attention mechanism integration - Rolling training
Optimization Topics	- Mean-variance optimized objective function: Sharpe ratio - Risk parity – CVaR minimization,

Outcome	User received end-to-end support in building, debugging, and evaluating a hybrid deep learning-based portfolio strategy

Table 5. The Use of Generative AI

B. Supplementary Figures or Tables

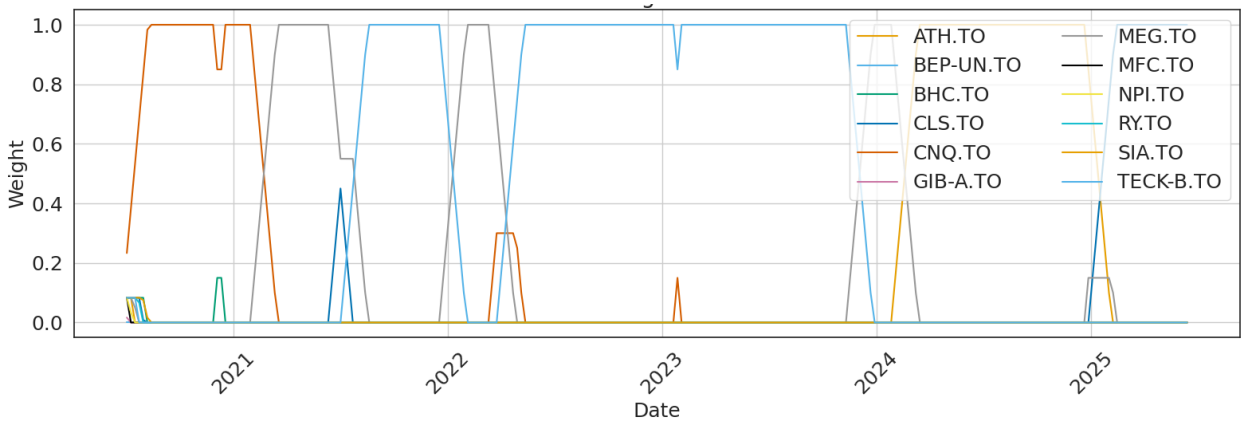


Figure 4. CNN-LSTM Mean-Variance Portfolio Weights Over Time

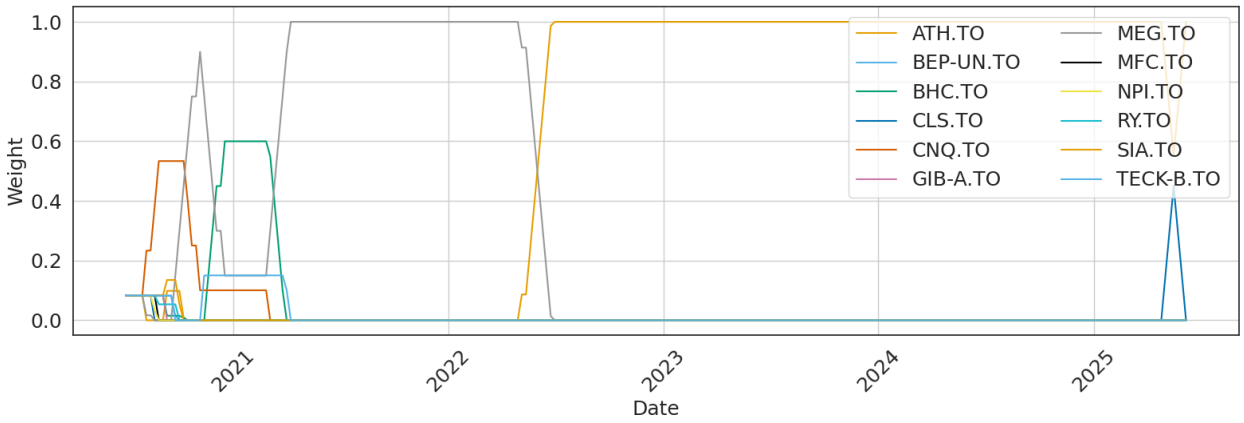


Figure 5. CNN-LSTM Hybrid Portfolio Weights Over Time

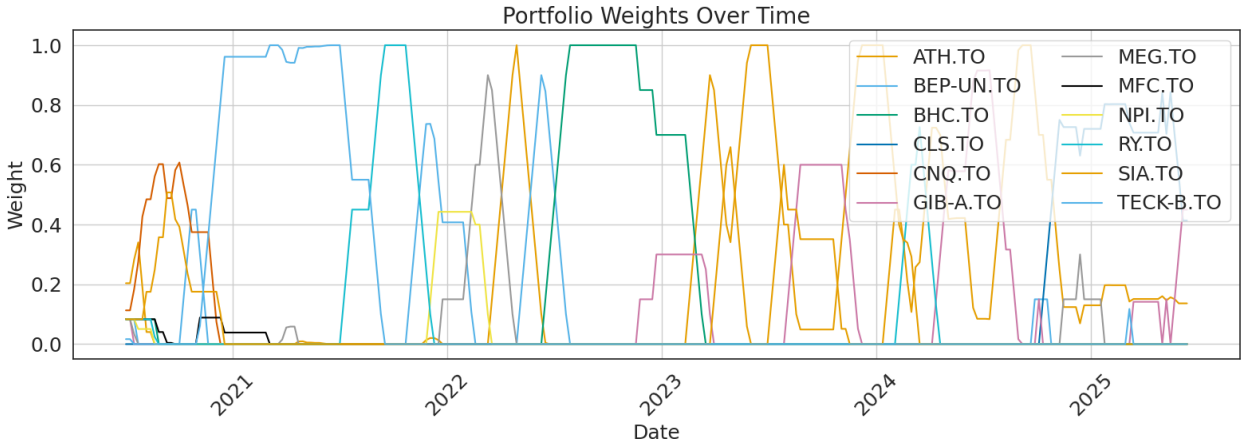


Figure 6. CNN-LSTM Signal Portfolio Weights Over Time

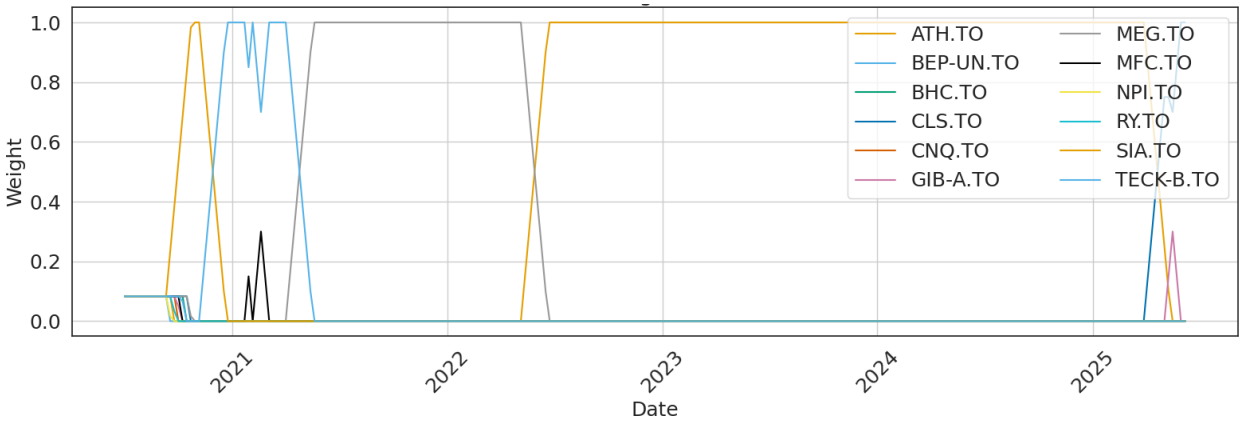


Figure 7. CNN-LSTM Regime Portfolio Weights Over Time

C. Full Code Repository

- GitHub code link: https://github.com/wangwang2111/canada-quantamental-investing-strategies.git