Conditional Autoencoder for Stock Trading

Analyzing the Impact of External Forces (Covid Pandemic) on Deep Learning Model Performance

Zelalem Abahana

Benjamin Arnosti

Deep Learning (Summer 2024)

Dr. Youakim Badr

PennState

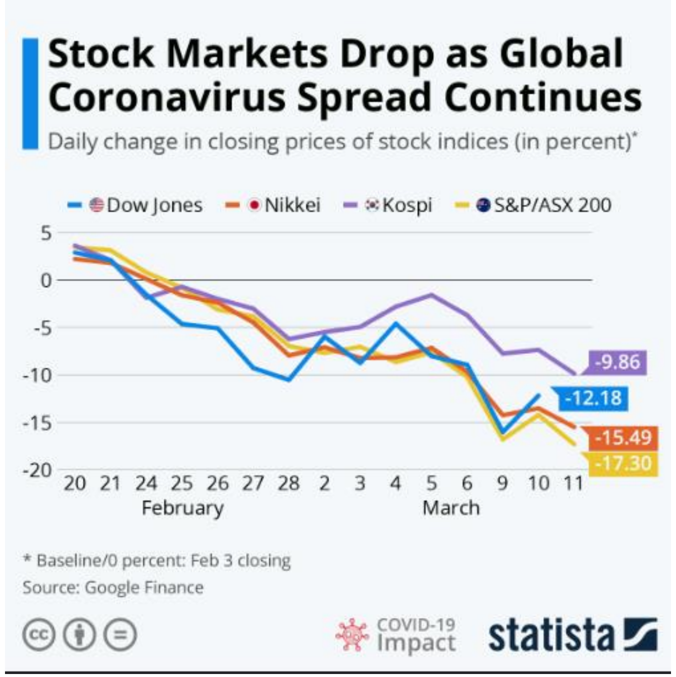
Deep Learning and Asset Pricing

- ❖ In finance, **asset pricing** is the study of how and why the prices of assets (e.g. stocks or commodities) change.

- ❖ We used a **Conditional Autoencoder** deep learning neural network to learn the underlying nonlinear relationships that influence stock prices.

- ❖ We then look at how external shocks to the market (**the Covid pandemic**) affect model performance.

- ❖ We tested the model performance using snapshots for **pre-covid, on-set of Covid, deep in-Covid period, recovery period,** and **post Covid period.**



Stock Markets Drop as Global Coronavirus Spread Continues

Daily change in closing prices of stock indices (in percent)*

* Baseline/0 percent: Feb 3 closing

Source: Google Finance

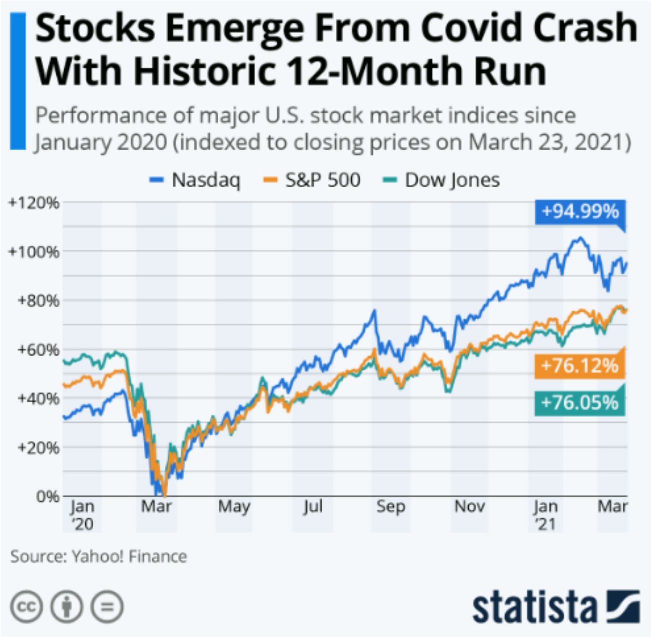
Asset Pricing Models

- Asset pricing models explain the returns from an asset using series of contributing elements called *factors*, each of which has a coefficient (*beta*) determining its strength.

$$\text{Returns (\$)} = \text{baseline} + (\text{beta}_1)\text{Factor}_1 + (\text{b}_2)\text{Factor}_2 + (\text{b}_3)\text{Factor}_3 + \ldots$$

- This approach is called a factor model, and there are many ways to design them.

- However, all share the same limitation: linearity!



Stocks Emerge From Covid Crash With Historic 12-Month Run

Performance of major U.S. stock market indices since January 2020 (indexed to closing prices on March 23, 2021)

Source: Yahoo! Finance

Problem and Challenges

- Factor models must rely on linear approximation for simplicity.

- BUT the true relationships are almost certainly nonlinear.

- Sadly, building a useful nonlinear factor model for stock market prediction is impractical.

- Deep learning comes to the rescue!

PennState

Problem and Challenges (cont.)

- ❖ Models struggle to recognize events not in historical data, like a stock selloff due to sudden global border closures.

- ❖ Market models struggle to account for external factors, such as public health crises.

- ❖ The 2020 pandemic upended markets and decreased model performance.

PennState

Related solutions / State of Art

Autoencoder Asset Pricing Models by Gu, S., Kelly, B., & Xiu, D. (2021)

- Introduced deep learning to stock pricing using conditional autoencoder models.

- Model performance was comparable to state-of-the-art linear models (IPCA).

- We use this work as a basis for the conditional autoencoder model framework, but we further assess how the model adjusts to new information as the Covid pandemic and learns without implicitly

Asset Pricing Variable and Related Features

Response Variable:

Returns is calculated using the adjusted close prices to account for corporate actions such as dividends, stock splits, and new stock issuance.

$$\text{Return} = \frac{\text{Price}_{\text{end}} - \text{Price}_{\text{start}}}{\text{Price}_{\text{start}}}$$

Features:

1. Volatility and Momentum

- **Close_Volatility**: Higher volatility typically indicates higher risk. High volatility can lead to higher potential returns but also higher potential losses.
- **Mom1m, Mom12m, Mom36m**: Momentum measures the tendency of securities to continue moving in their current direction. Positive momentum can attract investors, driving prices higher; negative momentum can lead to selling pressure and lower prices.
- **RetVol**: Similar to Close_Volatility, higher return volatility indicates higher risk and potential for both gains and losses.

Asset Pricing Variable and Features (Con.t)

2. Trading Activity and Liquidity

- **Turn**: Higher turnover indicates higher liquidity, making it easier to buy or sell the asset without significantly affecting its price. Liquid assets are generally less risky.
- **StdTurn**: Higher variability in turnover can indicate periods of increased trading activity, which may be associated with significant news or market events impacting the asset price.
- **DolVol**: Higher dollar volume indicates higher liquidity and investor interest, which can positively impact asset prices.

3. Risk Measures

- **MaxRet**: The maximum return over a specific period can indicate the asset's potential for high gains. Investors may be attracted to assets with high maximum returns, pushing prices higher.
- **Beta:** A higher beta indicates higher sensitivity to market movements. Assets with high beta may offer higher returns during market upswings but also higher losses during downturns.

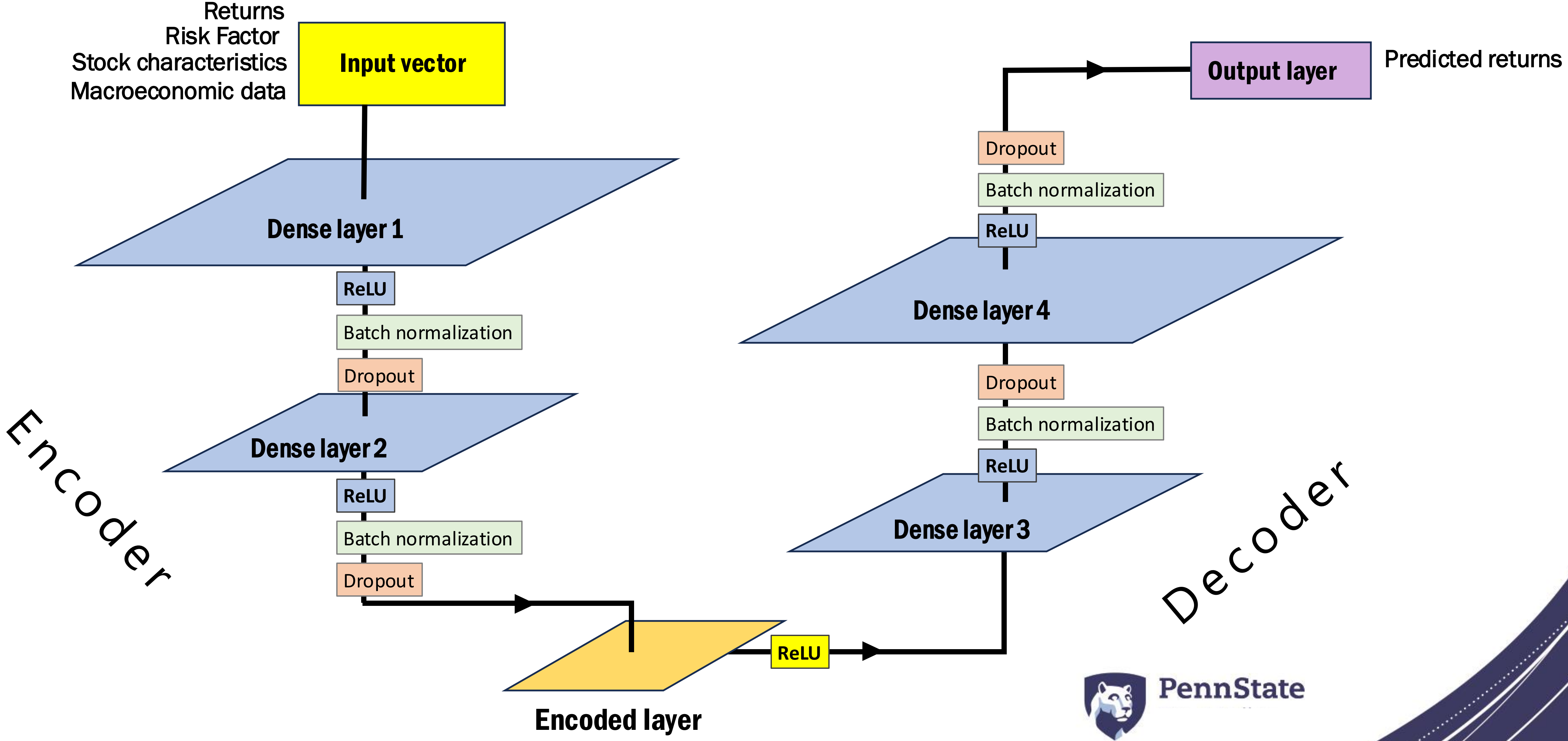
Asset Pricing Variable and Features (Con’d)

4. **Macroeconomic Factors**

- **GDP:** Higher GDP growth indicates a healthy economy, which can lead to higher corporate earnings and rising asset prices. Investors may prefer assets in economies with strong GDP growth.

- **Interest Rate:** Higher interest rates increase the cost of borrowing, reducing disposable income and business investment, which can negatively impact asset prices. Conversely, lower interest rates can stimulate spending and investment, positively impacting prices.

Conditional Autoencoder Architecture (actual)



Data Collection

❖Data sources:

- Yahoo Finance (yfinance API)
- Federal Reserve Economic Data (FRED)

❖Stock price data

- Queried closing prices and adjusted for each stock ticker
- Date range: Jan 2010 – Dec 2023
- Sampled 5 major companies from 5 industries (Tech, Finance, Retail, Industrials, Healthcare)

❖Macroeconomic data from FRED:

- Gross Domestic Product (GDP)
- Interest rates (DGS10)

Data Preprocessing

❖ Parsing/indexing

- ➢ Dates used as index (time series realization)
- ➢ Any end-of-month dates were set to beginning-of-month dates, in accordance with financial reporting practices.

❖ Cleaning up non-numerics/missing data

- ➢ Non-numeric values coerced to NaN
- ➢ Missing values handled with **forward filling.**

❖ Normalization

- ➢ Returns and stock characteristics normalized to ensure features are on the same scale

❖ Segmentation

- ➢ Data was split into **training**, **testing**, and **holdout (out-of-time)** sets by snapshot date for testing. After using the last two months for holdout data, the remainder was split 80/20.

PennState

Feature Engineering

- **Calculate Monthly Returns**:
  - Monthly returns are calculated using the percentage change in adjusted close prices for each ticker.
- **Calculate Rolling Volatility**:
  - Volatility is calculated as the standard deviation of monthly returns over a 12-month rolling window for each ticker.
- **Momentum Indicators**:
  - **1-Month Momentum (Mom1m)**: Calculated as the percentage change in adjusted close prices over 1 month.
  - **12-Month Momentum (Mom12m)**: Calculated as the percentage change in adjusted close prices over 12 months.
  - **36-Month Momentum (Mom36m)**: Calculated as the percentage change in adjusted close prices over 36 months.
  - **Maximum Return (MaxRet)**:
  - The maximum return is calculated as the maximum adjusted close price over a 12-month rolling window.

Feature Engineering (Con’t)

- **Turnover Metrics:**
  - **Turnover (Turn):** Calculated as the average adjusted close price over a 12-month rolling window.
  - **Standard Deviation of Turnover (StdTurn):** Calculated as the standard deviation of adjusted close prices over a 12-month rolling window.
  - **Dollar Volume (DolVol):**
  - Calculated as the product of adjusted close prices and trading volumes for each ticker.
  - **Return Volatility (RetVol):**
  - Calculated as the standard deviation of monthly returns over a 12-month rolling window.
- **Beta:**
  - Calculated as the covariance of the ticker's returns with the returns of a benchmark (e.g., AAPL) over a 12-month rolling window, divided by the variance of the benchmark's returns.
- **Interest Rate and GDP:**
  - Monthly average interest rates (10-Year Treasury Constant Maturity Rate from FRED).
  - Monthly GDP data, resampled to monthly frequency and forward-filled to handle missing values.

Methodology (Manual Tuning)

- **Model Building (manual tuning)**:
  - **Conditional Autoencoder**: Built a conditional autoencoder model using Keras with a combination of input layers for returns and characteristics, concatenation, and dense layers with ReLU activation.
  - **Regularization**: Applied dropout and batch normalization to prevent overfitting.
- **Model Training**:
  - Trained the model on the training set with early stopping and learning rate reduction on plateau callbacks.
  - Evaluated the model using Mean Squared Error (MSE) on the test set.
- **Model Evaluation**:
  - Predicted the returns for the holdout set and calculated performance metrics such as MSE, RMSE, and MAE.
  - Ranked tickers based on the forecasted average return for decisioning.
- **Data Normalization and Splitting:**
  - Normalized the returns and asset characteristics using StandardScaler.
  - Split the data into training, testing, and holdout sets using train_test_split.

Methodology (Hyperopt Tuning)

- ❖ Define Search Space:
  - ➢ Defined a search space for hyperparameters including the number of units in hidden layers, dropout rates, batch size, and learning rate.

  Objective Function:

  - ➢ Created an objective function that builds the model with given hyperparameters, trains it on the training set, and returns the validation loss.
- ❖ Hyperopt Optimization:
  - ➢ Used hyperopt library to run the optimization with the Tree-structured Parzen Estimator (TPE) algorithm.
  - ➢ Evaluated multiple combinations of hyperparameters to find the best set that minimizes validation loss.

Best Hyperparameters:

- ➢ Extracted the best hyperparameters found by Hyperopt.

Final Model Training:

- ➢ Built and trained the final conditional autoencoder model using the best hyperparameters obtained from Hyperopt.

Hyperparameter Optimization with *hyperopt*

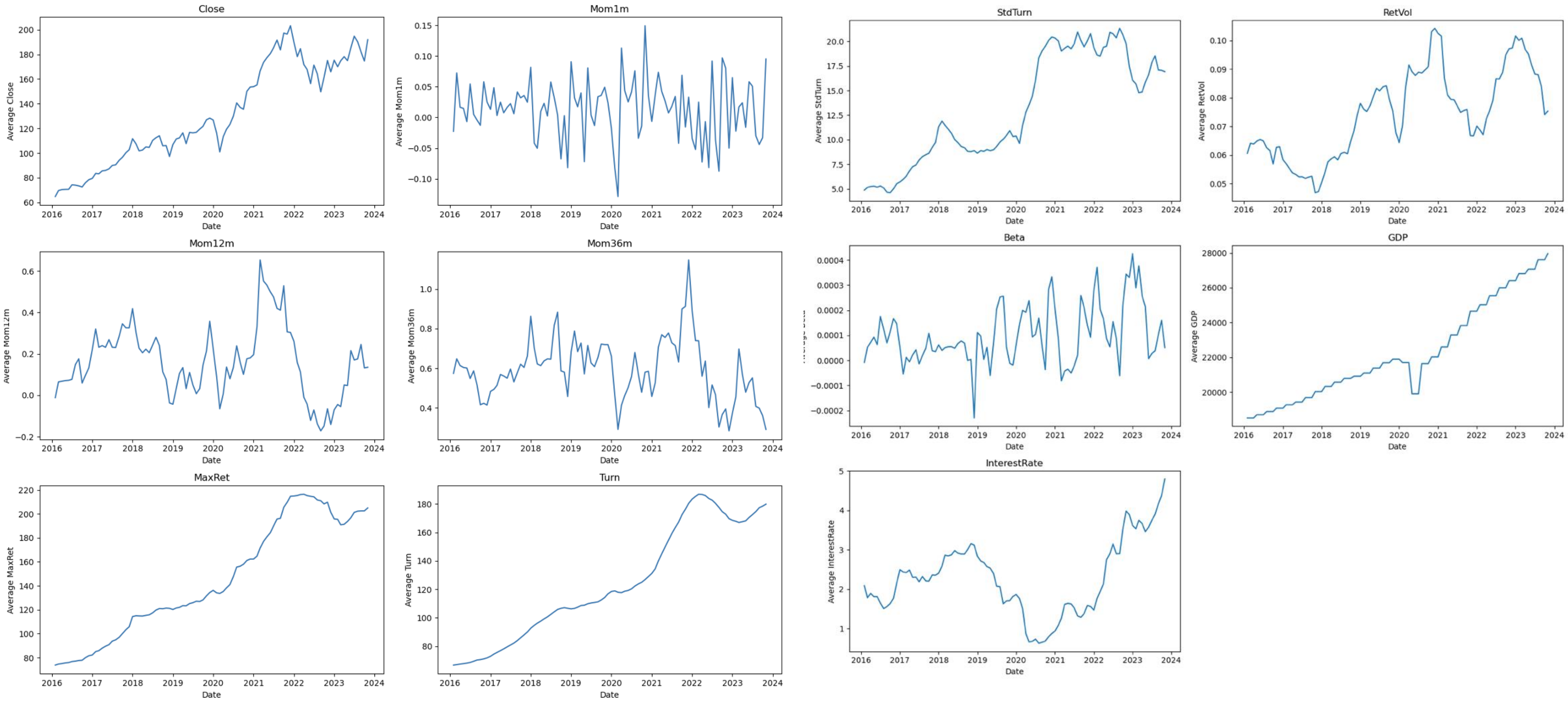
- Our search space

Neuron configurations:	[64,128,256], [32, 64, 128], [16, 32, 64]
Dropout rates:	0.1, 0.2, 0.3, 0.4, 0.5
Batch sizes:	16, 32, 64
Learn rates:	0.0001 – 0.01 (sampled, log-uniform range)

- Using hyperopt could result in <u>some</u> performance gains in some snapshots overall:

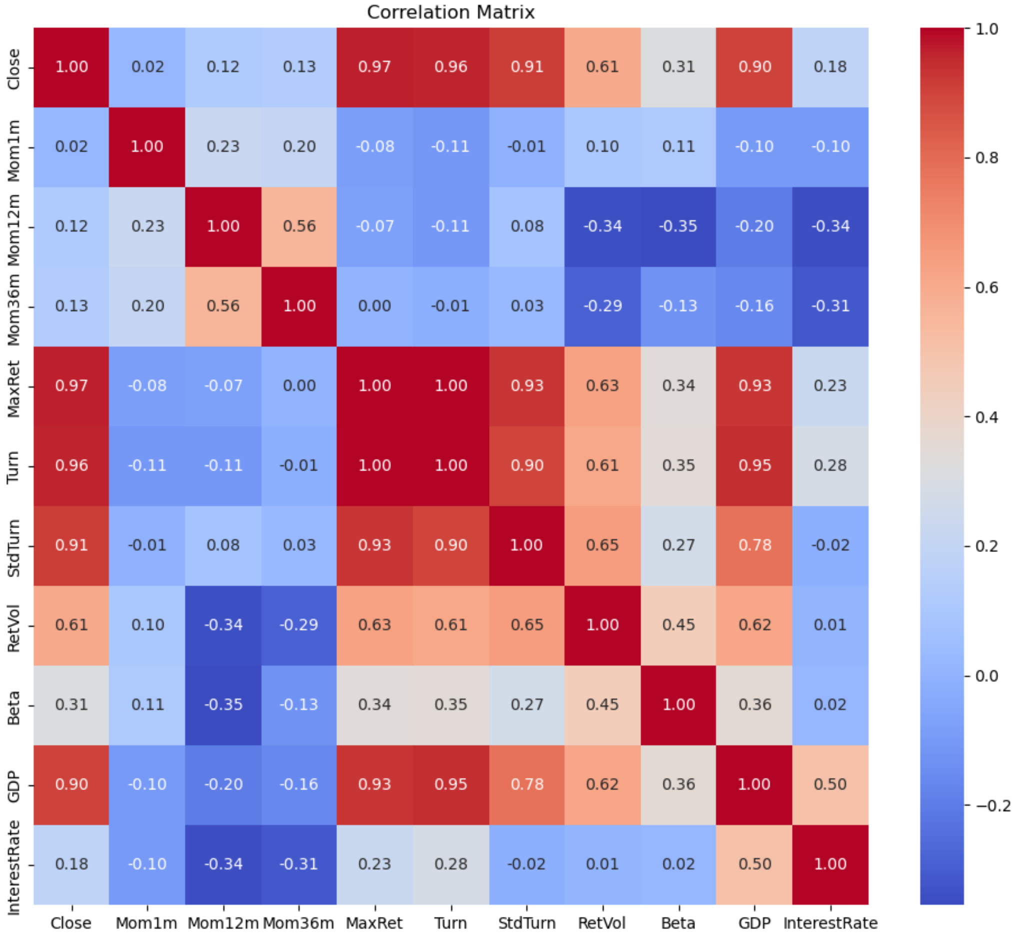
Final model training applied the optimized hyperparameters along with **early stopping** and **learn rate reduction** for performance plateaus.

Exploratory Data Analysis (Trends)



Time series trends of return, asset characteristics and macroeconomic factors in historical, pre and post Covid period.

Exploratory Data Analysis (Correlations)

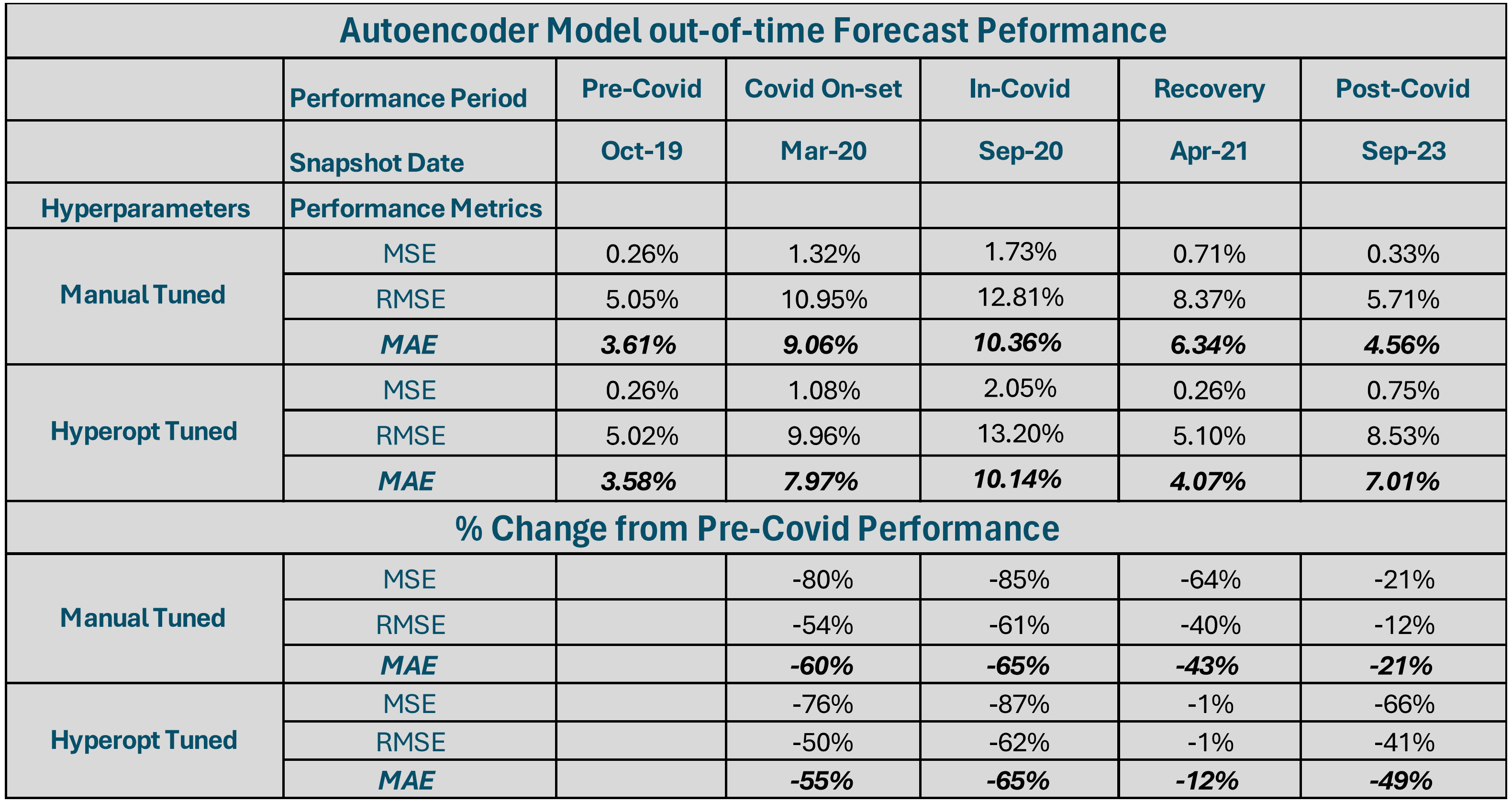

Correlation Matrix

	Close	Mom1m	Mom12m	Mom36m	MaxRet	Turn	StdTurn	RetVol	Beta	GDP	InterestRate
Close	1.00	0.02	0.12	0.13	0.97	0.96	0.91	0.61	0.31	0.90	0.18
Mom1m	0.02	1.00	0.23	0.20	-0.08	-0.11	-0.01	0.10	0.11	-0.10	-0.10
Mom12m	0.12	0.23	1.00	0.56	-0.07	-0.11	0.08	-0.34	-0.35	-0.20	-0.34
Mom36m	0.13	0.20	0.56	1.00	0.00	-0.01	0.03	-0.29	-0.13	-0.16	-0.31
MaxRet	0.97	-0.08	-0.07	0.00	1.00	1.00	0.93	0.63	0.34	0.93	0.23
Turn	0.96	-0.11	-0.11	-0.01	1.00	1.00	0.90	0.61	0.35	0.95	0.28
StdTurn	0.91	-0.01	0.08	0.03	0.93	0.90	1.00	0.65	0.27	0.78	-0.02
RetVol	0.61	0.10	-0.34	-0.29	0.63	0.61	0.65	1.00	0.45	0.62	0.01
Beta	0.31	0.11	-0.35	-0.13	0.34	0.35	0.27	0.45	1.00	0.36	0.02
GDP	0.90	-0.10	-0.20	-0.16	0.93	0.95	0.78	0.62	0.36	1.00	0.50
InterestRate	0.18	-0.10	-0.34	-0.31	0.23	0.28	-0.02	0.01	0.02	0.50	1.00

Correlations of return, asset characteristics and macroeconomic variables across time

Evaluations – Performance – Comparisons

- 2-month holdout (out-of-time) model performance using MSE, RMSE, and MAE are provided for phases of Covid time period including pre and post Covid time periods



Autoencoder Model out-of-time Forecast Peformance

	Performance Period	Pre-Covid	Covid On-set	In-Covid	Recovery	Post-Covid
	Snapshot Date	Oct-19	Mar-20	Sep-20	Apr-21	Sep-23
Hyperparameters	Performance Metrics					
Manual Tuned	MSE	0.26%	1.32%	1.73%	0.71%	0.33%
	RMSE	5.05%	10.95%	12.81%	8.37%	5.71%
	MAE	***3.61%***	***9.06%***	***10.36%***	***6.34%***	***4.56%***
Hyperopt Tuned	MSE	0.26%	1.08%	2.05%	0.26%	0.75%
	RMSE	5.02%	9.96%	13.20%	5.10%	8.53%
	MAE	***3.58%***	***7.97%***	***10.14%***	***4.07%***	***7.01%***

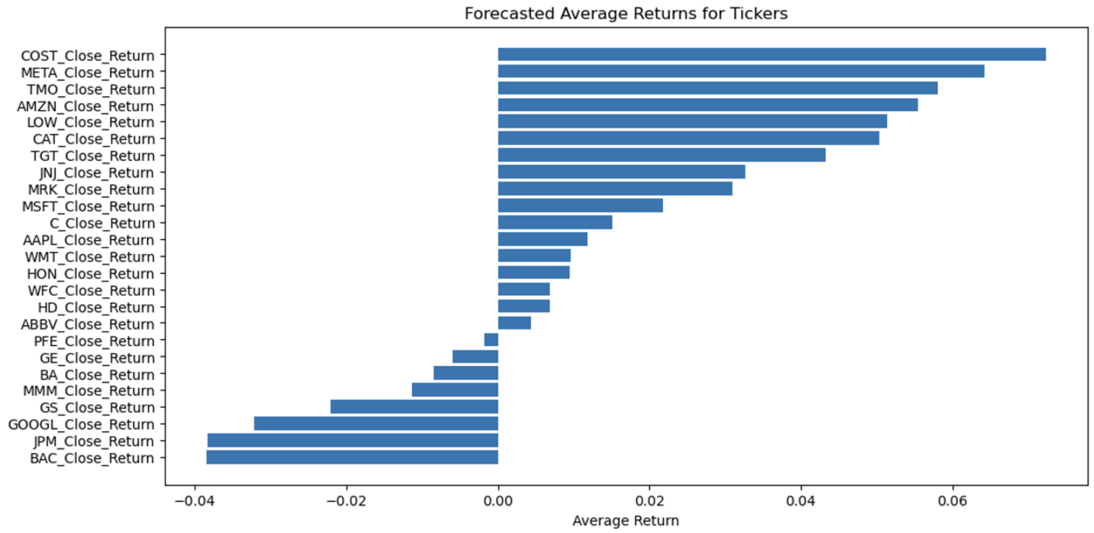
% Change from Pre-Covid Performance

Manual Tuned	MSE		-80%	-85%	-64%	-21%
	RMSE		-54%	-61%	-40%	-12%
	MAE		***-60%***	***-65%***	***-43%***	***-21%***
Hyperopt Tuned	MSE		-76%	-87%	-1%	-66%
	RMSE		-50%	-62%	-1%	-41%
	MAE		***-55%***	***-65%***	***-12%***	***-49%***

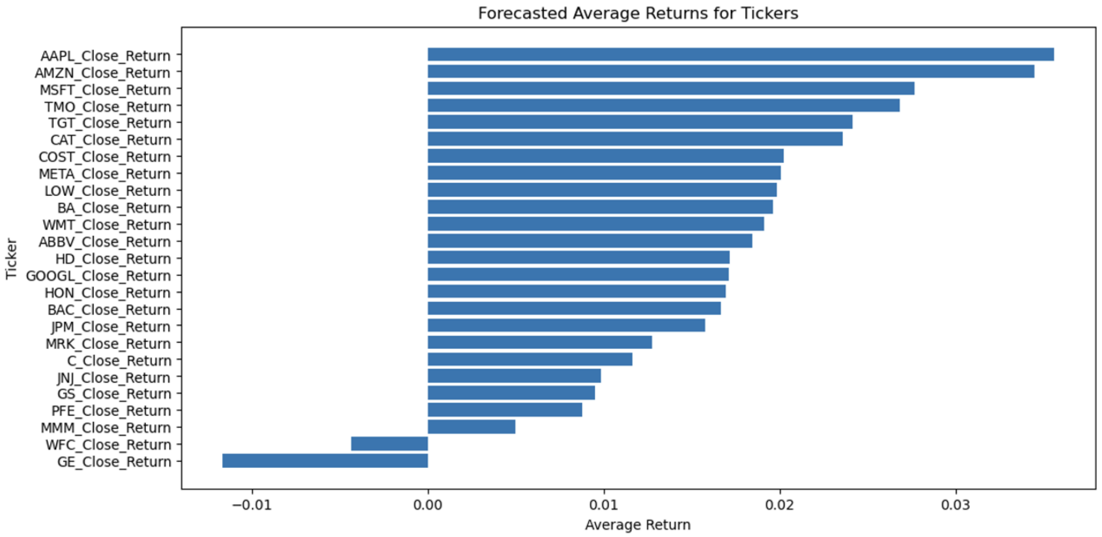
Model Use (Ranking Predicted Returns)

Ticker Forecasted Return Ranking (Post Covid Example)

Manually Tuned Model



Hyperopt Tuned Model



- The model can be used to forecast returns for many sticks to rank-order and select tickers that are more likely to bring in higher return. [model limitations factored in]

Outcomes/Findings

- Overall, the autoencoder model performed reasonably well in holdout time forecast performance evaluated over a 2-month out-of-time forecast.
- The model appears to have learned reasonably well in the encoder-decoder framework where it used a number of factors on asset characteristics, risk factors and macroeconomy; along with historical realizations
- The model appears to be off in performance on the Covid on-set and deep in Covid period, as the pandemic as external factor is a new information in the historical realization of stock returns and factors affecting them.
- In recovery period, the model shows improvement in performance indicating that it did not need a long period to adjust to the new reality
- Post Covid, the performance goes back close pre-Covid period showing that the model has absorbed the external shock without being implicitly told. (we did not use an indicator variable for Covid cycle)
- **Overall:**
  - The conditional autoencoder model demonstrated its ability to adapt to significant market disruptions like the COVID-19 pandemic.
  - The model's performance in differentiating between pre-COVID, during-COVID, and post-COVID periods indicates its robustness in handling varying market conditions and external shocks.

Limitations

- **Data Limitations:**
  - The model's performance is highly dependent on the quality and granularity of the input data. We used only 25 companies data using monthly average returns.
  - Any inaccuracies or biases in the historical data could impact the model's predictions.
- **Market Regime Shifts:**
  - The model might struggle with sudden market regime shifts that it hasn't encountered during the training phase.
- **Overfitting to Specific Events:**
  - While the model performed well during the COVID-19 period, it may overfit to this specific event, potentially reducing its generalizability to other types of market disruptions.
  - Ensuring a diverse training set that includes various market conditions can mitigate this risk.
- **Computational Complexity:**
  - The use of deep learning models, especially with extensive hyperparameter tuning, can be computationally intensive and require significant resources.
  - Balancing model complexity with computational efficiency is crucial for practical implementation.

Lessons learned and Perspectives

- ❖ Importance of Diverse Feature Engineering:
  - ➢ The inclusion of various asset characteristics (returns, volatility, momentum, etc.) and macroeconomic factors (interest rates, GDP) contributed significantly to the model's predictive power.
  - ➢ This highlights the importance of a comprehensive feature set that captures both asset-specific and broader economic indicators.
- ❖ Future Enhancements:
  - ➢ Incorporating additional macroeconomic indicators and sentiment analysis data could further enhance the model's accuracy.
  - ➢ Exploring alternative linear models and deep learning architectures like LSTM or GRU for capturing temporal dependencies in stock prices might provide better performance

Bibliographical References

- Gu, S., Kelly, B., & Xiu, D. (2020). Empirical Asset Pricing via Machine Learning. *The Review of Financial Studies, 33(5), 2223-2273. https://doi.org/10.1093/rfs/hhaa009*
- Mroua, M., & Lamine, A. (2023). Financial time series prediction under Covid-19 pandemic crisis with Long Short-Term Memory (LSTM) network. *Humanities and Social Sciences Communications, 10(1*), 1-15. https://doi.org/10.1057/s41599-023-02042-w
- Jansen, S. (2020). *Machine Learning for Algorithmic Trading*, Second Edition. Packt Publishing
- Gu, S., Kelly, B., & Xiu, D. (2019). Autoencoder Asset Pricing Models. Yale ICF Working Paper No. 2019-04, Chicago Booth Research Paper No. 19-24. Available at SSRN: https://ssrn.com/abstract=3335536 or http://dx.doi.org/10.2139/ssrn.3335536
- Kim, E., Cho, T., Koo, B., & Kang, G. (2023). Conditional autoencoder asset pricing models for the Korean stock market. *PLOS ONE*, *18*(7), e0281783. https://doi.org/10.1371/journal.pone.0281783
- Giglio, S., Kelly, B., & Xiu, D. (2021). *Factor Models, Machine Learning, and Asset Pricing*. Journal of Econometrics, 222(1), 429-450.

Demo!

PennState

Appendix

How Does Conditional Autoencoder Model Work?

Input Preparation:

•Inputs: Historical stock returns.

•Conditions: Asset characteristics like volatility, momentum, interest rates, and GDP.

Encoding Process:

- The encoder consists of several dense layers that combine the input and conditional information.
- The layers apply non-linear transformations to encode the combined information into a latent representation.
- Regularization techniques like dropout and batch normalization are applied to prevent overfitting.
- **Latent Representation**:
- The encoder outputs a latent space representation that captures the compressed and essential features of the input data and conditions.

Decoding Process:

- The latent representation is fed into the decoder, which also considers the conditional information.
- The decoder consists of several dense layers that transform the latent representation back into the original input space.
- The output is the reconstructed stock returns.
- **Training Objective**:
- The model is trained to minimize the reconstruction error, typically using mean squared error (MSE) as the loss function.
- The optimization process adjusts the weights of the encoder and decoder to achieve the best reconstruction accuracy.