Linear Models for Statistical Arbitrage in Event-Driven Trading

Project Group 10

Zhang Jianan Ha Nguyen Xu Long

***Abstract*—Statistical arbitrage is a popular quantitative investment and trading strategy in the financial industry. Statistical arbitrage models can hedge market and sector risk. However, existing statistical arbitrage models fail to counter company-specific risk, like the high-volatility periods around a company's earnings release dates. Therefore, benchmark statistical arbitrage may not perform well around the earnings period due to changes in earnings releases. Our research project refines from a baseline benchmark statistical arbitrage model with added features on 1) feature extraction, 2) multi-collinearity detection, 3) statistical arbitrage strategy optimization, and a new component – 4) weighted event-based filter. With the earning release data, our model beats the state-of-the-art baseline in almost all metrics in quantitative evaluation. In qualitative evaluation, our model reveals less variation in capital during earning release saturated months.**

I. Problem description

A. Context

Statistical arbitrage is a popular quantitative investment and trading strategy in the financial industry. Statistical arbitrage relies on a statistical and mathematical model to identify these temporary price divergences and perform trades between multiple assets. It doesn't rely on predicting the direction of the overall market. Instead, it captures the relative price movements among correlated cluster instruments. However, although factor-specific risks can be hedged among identified clusters, Statistical arbitrage models don't aim to hedge company-specific risks.

Company-specific risks lay among events related to each company. Press conferences or scandals can largely impact the company. Among these intangible events, earning releases are among those scheduled. Every quarter, every company reports its earnings to the public. This information includes the company's financial performance, like revenues, earnings, and future plans. Earnings releases are scheduled in advance at specific dates. Therefore, we can utilize this information to hedge the company-specific risks in our modified model. Also, by consolidating solid performance, our strategy can potentially be deployed on other scheduled events like new product release conferences for technology-related companies.

B. Problem statement

One of the important findings in the financial market is that corporate events, such as earnings releases, can cause high volatility in market expectations, leading stocks within the same industry to move synchronously.

However, while the influence of corporate events on stock movement is acknowledged, current statistical arbitrage strategies fail to adequately employ these data to hedge company-specific risk or capture company-specific opportunities. This oversight can lead to unanticipated losses, especially when earnings are released, the scheduled dates of which are known to the public.

C. Project Objectives and Success measures

We aim to refine baseline **linear factor models** to create a statistical arbitrage portfolio that captures the company's fundamental earning-release changes.

This aims to create a robust statistical arbitrage portfolio and improve baseline statistical arbitrage models' performance around company earning releases where the earning announcement causes lots of volatility. The entire portfolio should reduce the variance, less **Max Percentage Drawndow** and higher **Sharpe ratio** during the earning release period so that stock prices move significantly.

We prospect that our statistical arbitrage strategy with event-driven design can ideally have a $1.5 - 2$ **Sharpe ratio** with limited **Max Percentage Drawdown** to below 10% during the earning release period, especially during the earning release events. On the other hand, it should retain its ability as a statistical arbitrage as its baseline counterparts.

Max Percentage Drawdown MDD is the maximum percentage loss the portfolio suffered. $MDD = 1 - min_{t<s\in T}\frac{P_s}{P_t}$. Where P_s is the value of the portfolio value at peak value time, P_t is the value of the portfolio value at lowest value time. $P\% = \frac{P_N}{P_1}$

Sharpe Ratio measures how the risk is rewarded in terms of extra gain.

$$ShR = \frac{Mean(Pr) - i_r}{Var(Pr}$$

Where Pr is the portfolio return, i_r is the risk-free interest rate.

Therefore, A Sharpe ratio of $1.5-2$ with a limit MDD of 10% suggests that the portfolio generates significantly higher returns than a risk-free investment (like a government bond) after adjusting for the risk it carries. It is expected to deliver a return of 1.5 to 2 times the return of a risk-free environment.

D. Assumptions

1) We assume the underlying relationship among stocks is linear if we use linear models to construct the portfolio and extract linear risk factors.
2) For the sake of simplicity, this research assumes no transaction cost and no price slippage when entering a trade. In other words, we can enter a trade at the stated price without additional costs.
3) We only consider closing prices as data inputs at each trading date. We assume that the relationships of the highest price, lowest price, and opening price on different trading dates are similar to those of the closing price.

II. Literature Review

A. Time series Clustering

Time series clustering is a widely used tool in finance. It serves as a downstream task for Portfolio selections [10], [11], [18], Portfolio diversification [8], Stock market analysis [2], and many more. Statistical arbitrages [1], [5], [13] used time series clustering to identify similar stocks for pairs trading. The time series clustering methods can be divided into three categories [22].

Raw-data-based methods evaluate two-time series directly, which can be performed on both time and frequency domains. With normally sampled data from the same time interval, the raw-data-based method doesn't eliminate any redundant data.

Feature-vector-based methods diminish the impact of high-dimensional, highly noisy data. Using principal component analysis (PCA) or other filters, feature-vector-based methods involve a step of data preprocessing before clustering. [13] employed this method using principal component analysis (PCA) on both day and month intervals and removed the colinearity between them to form more representative risk factors. On the other hand, rather than statistical methods, [5] manually defined Lagged intra-day price returns (LR) and a set of technical indicators to represent the data more interpretably.

Fitted-coefficients-based methods consider each time series to be from some underlying probability distributions. Rather than clustering on the data, fitted-coefficients-based methods first fit a time-series model like Autoregressive integrated moving average (ARIMA) to the data. Fitted-coefficients-based methods cluster similar stocks by comparing the fitted coefficients and remaining residuals. [1] employed this approach by identifying Hurst Exponents that are irregular to Brownian motion.

B. Time series feature extraction

The clustering strategy in our baseline model [13] follows the **Feature-vector-based methods**. As a downstream task of time clustering, feature extraction methods aim to identify risk factors from raw data.

Statistical approaches have been identified as standard procedures in feature extraction related to financial time series. Among them, principal components analysis (PCA) [25], [29] and independent components analysis (ICA) [19], [30] emerge as two benchmark methods for extracting factors. PCA transforms raw data linearly to maximize the variance of each variable, and ICA aims to find the hidden statistically independent variables.

Machine learning approaches have emerged recently following the deep learning trends. Neural network principal components analysis (NNPCA) shows its effectiveness in retaining the information of the original data. Proposed by Scholz and Vigário [27] and adapted by Ladrón de Guevara Cortés et al. [20], NNPCA refits original data to itself, and loss was calculated by mean squared error. Variational autoencoder (VAE) has also been proposed for use in risk feature extraction methods [7], [23], [24] related to finance. Unlike NNPCA, in VAE, obtained variables are re-sampled from Gaussian distribution with the mean of itself and a trainable standard deviation after the encoder.

C. Statistical Arbitrage Strategy

Statistical arbitrage strategies serve as downstream tasks for a wide range of financial products, including equities, exchange-traded funds (ETFs), future contracts, options, fixed-income securities, forex, commodities, and many more. These strategies include various methods for studying and exploiting the relationship between underlying assets to construct a portfolio. Benchmark statistical arbitrage models include two-way steps: cluster formation and optimization.

Cluster formation based on factors: Our baseline model [13] constructs portfolios within similar stocks chosen from time series clustering to minimize risk factors. [3] constructs portfolios by extracting the residual momentum from Fama Factor models to construct momentum portfolios. The research does show that residual momentum performs better than total return with lower volatility, thanks to lower exposures to common factors. [17] uses the predicted distribution of a financial quantity called residual factors to construct portfolios. They propose a new system for stock price distributions based on spectral residual using a deep neural network. From the estimated means and variances of future spectral residuals, the system can construct a portfolio based on modern portfolio theory. [15] constructs arbitrage portfolios of similar assets from conditional latent

asset pricing factors. Then, they extract their time series signals with convolutional transformers to form an optimal trading policy that maximizes risk-adjusted returns under constraints. This arbitrage strategy does show consistently high out-of-sample returns.

However, though clustering can hedge cluster-specific or sector-specific risks, their strategies don't factor in idiosyncratic or company-specific risks.

Weights Optimization: Other research focuses on different optimization methods, namely, forming different representative trading baskets. Our baseline model [13] optimized the weights of all risk factors within a cluster to create a market-neutral portfolio. [31] uses a mean-reverting portfolio for pair trading strategies with convex optimization for portfolio weights. [21] involves trading among pairs of assets having co-integration. Then, form mean-reverting portfolios and solve a Hamilton-Jacobi-Bellman (HJB) partial equation to get an optimal portfolio.

However, none of these optimization methods considers timely anomalies, which can profoundly affect the stock market.

D. Regression methods

Regression analysis is a fundamental component of statistics and machine learning. It is used to predict a continuous outcome variable based on one or more predictor variables. In statistical arbitrage, it is often used for tasks like selecting a portfolio's weights.

Pooled Ordinary least square regression method is adapted in our baseline model [13] using the ordinary least-square (OLS) regression method several times in different time windows to select the weights of its portfolio.

Generalized linear regression models expand a wider regression family from linear regression. The Barra model [14] used the third power of market capitalization in the k-term spline function to conduct market forecasting.

Machine learning-based regression models are emerging powerful tools. Methods like support vector regressors, tree-based methods, and neural networks have all shown their application in the stock market [9], [14]. However, according to [26], single regression methods can have structural instability, which limits their performances on out-of-sample results, especially in the long run. This result coincides with the non-free lunch theory. [5] takes on a study among different regression methods and demonstrates **Ensemble regression models**, an average of different regression methods, can outperform individual methods.

E. Study on the effect of corporate events on stock markets

Event-triggered Return Anomalies have been an emerging topic. [4] examines how public information dissemination can affect return timings. They found that anomaly returns are surrounded in the month following the release of information and with a swift decay of such anomaly returns. Their observation, with statistical significance, concludes with the importance of analyzing financial data in the context of earnings release events and other time-sensitive information, like press conferences. On to data itself, [12] leveraged a sample of 97 anomalies and found these anomalies appear 50 percent higher on corporate news days and amplify sixfold on earnings release events. With their conclusion that anomaly returns usually appear driven by biased expectations, we find it important to hedge such risks for our statistical arbitrage models.

Volatility Changes indicate how the stock market can overreact to corporate events. Research shows that the earnings release event is positively related to the expected volatility. [28] [16] show that earnings expectations play a huge role for other sectors. For some statistical arbitrages, [6] explores whether earnings release events can cause overreaction of arbitrages and shows that short covering over positive news stocks generates overshooting in stock momentum and causes overreaction.

III. Project Requirement and Data Preprocessing

A. Data Requirement

We downloaded price data from Yahoo Finance API from 2016-01-01 to 2023-08-31. The price data includes *Open, Close, High, Low, Volume*. For each day, Open is the stock price when the market opens; Close is the stock price when the market closes; High is the Highest price during the day; Low is the Lowest price during the day; volume is the stock traded volume during the day.

Each year, a company will report earnings in each quarter, and the schedule for earning release will be scheduled in advance and announced to the public. We downloaded data from the Yahoo Finance API for each ticker in our universe for the earnings release calendar. The dataset includes four earnings dates for each stock from 2016 to 2023.

B. Data Pre-processing

We cleaned *36* columns (stocks) that were not continuous and had NaN values during the period we studied and split data into training and testing periods. We cleaned NaN columns because the models do not work with discontinuous data. Training data from 2016-01 to 2022-09. Testing period from 2022-09 to 2023-08. We test the models by 12-fold validation with each fold of one-month increment data.

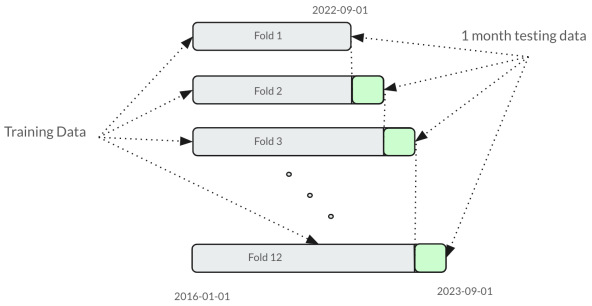


Fig. 1. 12 folds data

From the stock closing price, we calculate each stock's daily, monthly, and quarterly returns. We will use this stock return to feed into our model.

For our baseline model, we calculated six notable statistics: Percentage Profit, Profit Factor, Percentage Drawdown, Recovery Factor, Sharpe Ratio, and Sortino Ratio. With a 12-fold approach, we calculated their mean, standard deviation, and worst-case scenario (WCS) according to our baseline paper.

IV. Methodology (Process Overview)

Our baseline model is from one benchmark state-of-the-art paper [13]. This provides an overview of the process. The model includes four main parts:

A. Risk Extraction

From the closing price training data on each date, we first calculated returns based on different granularity, namely, daily, monthly, and granularity returns. Our feature extraction method will performed separately on two of these three granularities and identify seven principal components. In our baseline model, they perform PCA on stocks that reduce a 1977 (dates) by 464 (stocks) matrix to a 1977 (dates) by 7 (Principal components) matrix for daily return; the same is applied for monthly return and quarterly return. Note we modified feature extraction methods to neural network principal components analysis (NNPCA) and variational autoencoder (VAE.) Details are described in the next section.

However, the generated principal components of different granularities are both of the time domain. There exists strong colinearity between them. After extracting seven principal components for both granularities, we calculate the absolute value of the correlation coefficient of principal components from the larger time granularity with each principal component from daily granularity. We set the threshold to be 0.5, and if the correlation coefficient passes the threshold, we will eliminate the principal component from the larger granularity.

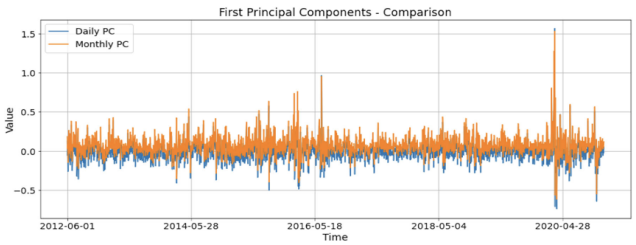
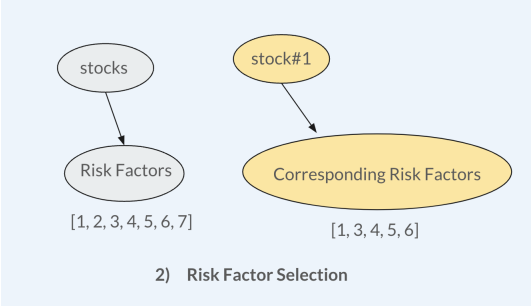


Fig. 2. Illustration of the multi-collinearity between monthly and daily granularity [13]

After the multi-colinearity elimination, the remaining PCs are identified as risk factors.

B. Risk Extraction



Using the penalized regression method Adaptive Lasso (A-Lasso), the baseline model determines risk factors associated with each stock. The process is straightforward. The weights related to negligible risk factors are set to zero. From the remaining weights, we can determine the corresponding risk factors for each stock.

C. Cluster formation

The baseline model classifies stocks with the same risk factors to different clusters. However, only clusters with 2 to 4 PCs are useful clusters, as we don't want to have too general clusters (below 2 PCs) or spend too much time on arbitrage strategy optimization (above 4 PCs.)

D. Statistical arbitrage strategy optimization

After clustering, there are three main parts including:

1) **Selecting the risk factor weight for each stock in one cluster.**

Once the clusters are obtained, we can construct a market-neutral portfolio. For each cluster C, let A_C be the relevant factor set for cluster C, T is time set, we can form a linear factor model as below:

$$r_t^j = \alpha^j + \sum_{i \in A_C} \beta_i^j F_{i,t} + \epsilon_t^j \qquad \forall j \in C, \forall t \in T \quad (1)$$

Where r_t^j is the return of stock j in time t. a^j is constant, β_i^j is the risk exposure of stock j to risk factor i, ϵ_t^j is the residual of stock j at time t.
We need to recalculate the coefficient α^j and β^j every time window TW. The time window can be daily or weekly, coherent with the investment holding period. We can apply Pooled Ordinary Lease Squares (OLS) Regression to obtain the coefficient for each time window. Finally, we can estimate the final parameters as below:

$$\alpha^j = \frac{1}{TW} \sum_{tw \in TW} \alpha_{tw}^j \qquad \beta^j = \frac{1}{TW} \sum_{tw \in TW} \beta_{tw}^j \quad (2)$$

2) **Selecting the weight for each stock in one cluster.** We want to create a linear combination weight vector from risk factor coefficients for each stock. $(w^1, w^2, ..., w^c)$ to remove the exposure for each risk factor.

$$Port_t = \sum_{j \in C} w^j r_t^j = \sum_{j \in C} w^j \alpha^j (\sum_{j \in C} w^j \beta_i^j) + \sum_{j \in C} w^j \epsilon_t^j \quad (3)$$

Since we want to delete the exposure of each risk factor, we have the below constraints.

$$\sum_{j \in C} w^j \beta_i^j = 0 \quad \forall i \in A_C \sum_{j \in C} |w^j| = 1 \quad (4)$$

Assuming there are more $|A_C|$ stocks in cluster C, then there is infinite number of linear combination weights $(w^1, w^2, ..., w^c)$. Hence, the equation (3) becomes:

$$Port_t = \sum_{j \in C} w^j \alpha^j + \sum_{j \in C} w^j \epsilon^j = \alpha^{Port} + \epsilon^{Port} \quad (5)$$

Where α^{Port} is the constant of the portfolio, ϵ_t^{Port} is a sum of Gaussian random variables. Since there are infinite portfolios that can satisfy constraints (4), we can define our objective to choose the portfolio that minimizes variance. We can choose *Sequential Least Squares Programming (SLSQP)* to solve nonlinear constraint objective function.

3) **Selecting the weight for each cluster in our final portfolio.** For each cluster, the model obtains the minimum variance portfolio $P_C = argmin_{P \in P_c} Var(P)$. Among clusters, we can pick the top 3 clusters with the minimum variance to reduce investment risk by diversification.

In our next section, Core Contribution, we propose ways to extend this baseline model to manage company-specific risk.

V. Methodology (Core Contribution)

The four-step contributions of our proposed statistical arbitrage are listed:

1) **Feature Extraction:** Compare among feature extraction methods for extracting risk factors and train our Neural Network-based principal components analysis
2) **Multi-collinearity detection:** Update from monthly granularity to quarterly granularity to suit our quarterly-based earning releases with more representative principal components
3) **Statistical arbitrage optimization:** Construct ordinary least square regression and ensemble regression for market neutral portfolio construction in each identified cluster and evaluate their performances
4) **Earning event-based weighted filter:** Design a weighted filter to capture changes and hedge risks from earning release events.

The first three contributions focus on refining an existing part of our baseline model, and the fourth contribution focuses on developing a new part that can hedge company-specific risks.

A. Risk Factor Extraction(NNPCA)

Neural Network Principal Components Analysis is designed to output itself from limited neurons. NNPCA is of an encoder-decoder architecture. In our model, we choose the inner layer to have 7 neurons as principal components that aim to reproduce the whole 464 columns with these 7 principal components. We used mean square error as loss and added regularize and early stopping mechanisms accordingly.

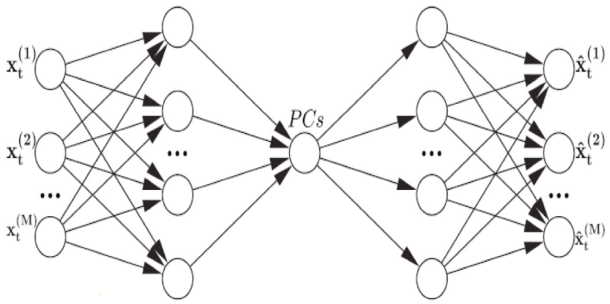


Fig. 3. NNPCA architecture [7]

Different versions of NNPCA are designed with different numbers of neurons in each layer and trained epochs. We use three layers as the encoder and the other two layers as the decoder. During inference, we only send our inputs to the encoder and obtain the inner 7 principal components. For different granularity, we also eliminate multi-colinearity, the

same as our baseline model.

Variational Autoencoder(VAE) is quite similar to the NNPCA. It is also designed to output itself from limited neurons and encoder-decoder architecture. The difference is that, for the inner layer of seven neurons, the encoder will attach to two different inner layers, one to output seven variables as mean and one to output seven variables as standard deviations. Then, before decoding, VAE will sample from a Gaussian distribution with corresponding mean and variance for each principal component and send these sampled values to the decoder.

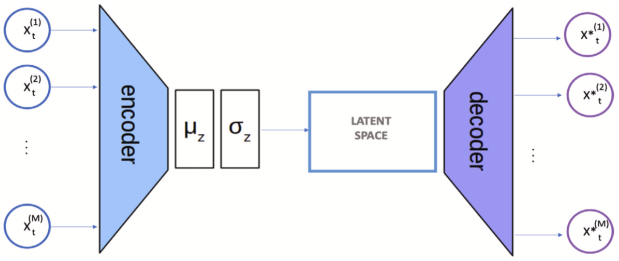


Fig. 4. VAE architecture [7]

Only one version of VAE is designed with 3 layers for the encoder and 2 layers for the decoder. We send our inputs to the encoder and only the "mean" prediction layer during inference. These 7 neurons are identified as the principal components of the VAE model. Also, multi-collinearity detection is conducted.

B. Multicollinearity detection

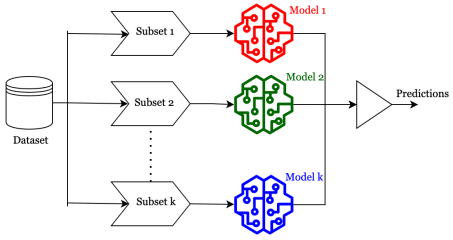
We implement daily and quarterly granularity and anticipate it to outperform daily and monthly. As earning release events are announced every quarter, we hope there could be some interactive effect between adapted earning event-based filters and granularity.

C. Statistical arbitrage optimization

We propose a different way to identify the **risk factors coefficients for each stock in cluster**
Once the clusters are obtained, we can construct a market-neutral portfolio. The first step is to identify the risk factor weights for each stock in the cluster. Here, we can apply the regression method to obtain the coefficient includes:
Pooled ordinary least squares(OLS) or **Pooled Ensemble regression** method to obtain the coefficient for each time window.

Here, the ensemble regression from Ordinary Least Square and Support Vector Regressor.

We compare the strategy performance of these two approaches to choose the best models.



D. Earning event based weighted filter

As discussed above in literature reviews, earning releases can impact stock prices before and after announcements. Statistical arbitrage models might not be able to account for these new company fundamental changes. Therefore, we can study these earnings' effects to improve our statistical arbitrage strategies.

1) **Earning Release window and hypothesis testing**
For each stock, we find an earning period window (p, q). Where p is the number of days before the earning date, q is the number of days after the earning date. We calculate the volatility within this period (p, q) and normal periods. We perform a grid search for p from 1 to 15 and q from 1 to 15 to find the period (p, q) with the maximum volatility difference from the normal period. Most of the (p, q) periods are $(4, 6), (5, 5), or (6, 4)$.

Null hypothesis H_0: $U_0 \geq U_1$ The means of volatility during earning periods is less or equal to the normal period. Alternative hypothesis H_1: $U_0 < U_1$. The volatility in the earnings period is significantly higher than the normal period.

We perform a T-statistic test for these two distributions using a significance level of 0.05. We reject the null hypothesis and conclude that the volatility in the earnings period is significantly higher than that in the normal period.

2) **Earning event based weighted filter:**
For each stock, we have found the earning period window (p, q). Our original strategy is to create a final portfolio $P = P_1 + P_2 + P_3$, where P_1, P_2, P_3 are the least variance portfolio among those clusters. We adjusted this strategy as below. For each cluster C in our final portfolio that contains stock in our strategy, if the cluster contains a stock within the earning period window, we expect this portfolio will increase volatility as earning releases for some stock in this portfolio come. We adjust the size of our final portfolio to α.
We perform a grid search for α from 0.1 to 1.9 with 0.1 increment to pick the optimal α to get the best strategy performance.

VI. Challenges

A. Running time issues

The full dataset contains 500 stocks and 464 stocks after cleaning. Since our experiments are conducted as 12-fold validation, each fold takes 2 hours to run. Totally, it takes 24 hours to perform one test. Therefore, we decided to use a reduced dataset, which contains 100 stocks, to perform the preliminary test, and then conduct experiment whole dataset in later experiments to save time.

Also, to mitigate computing resources. We first conducted two preliminary experiments on **Feature extraction method** (Mean squared error test) and **Multi-colinearity detection methods** (Mean absolute correlation coefficient test.) After the preliminary experiments, we selected candidates for performance experiments.

First, we isolate each component and evaluate its performance. After identifying the best-performing component in each group, we experiment with combining all these best-performing components and conducting ablation studies of it.

This can give us a grip on how each method performs on the reduced dataset, so we can test fewer variants on the full dataset.

VII. Experiment and result analysis

As discussed in the Challenges section, we will perform preliminary experiments on the reduced dataset to obtain the best performance model, and then we can infer the whole dataset. Then, we experiment with the whole dataset to obtain the final result.

A. Mean square error test on feature extraction methods

We compare the performance of PCA, multiple constructions of NNPCA (both are a 2-layer encoder and a symmetrical 2-layer autoencoder), and the VAE model. 5 is the MSE on the 12-fold test on the reduced dataset. PCA showed signs of underfitting, while NNPCA without early stopping tended to overfit. Interestingly, an increase in the number of neurons consistently led to a decrease in MSE, highlighting the importance of neuron count in model efficiency. However, variations in compile and activation functions appeared to have a negligible impact on the overall performance.
Based on the outcomes of our studies, we kept PCA, VAE, and the largest NNPCA model with a neuron configuration (1024,512) for further analysis.

B. Mean correlation coefficient test on multi-colinearity detection

In comparing different granularity like monthly and quarterly data, we computed the mean Correlation Coefficient between the principal components of daily data and those of monthly/quarterly data, yielding results of 0.233 and 0.229, respectively. Notably, we observed that quarterly returns exhibit better performance, extracting a greater number of principal components with lower correlation. This observation appears consistent with the pattern of earnings releases occurring each quarter, suggesting a potential alignment between data granularity and significant corporate events.

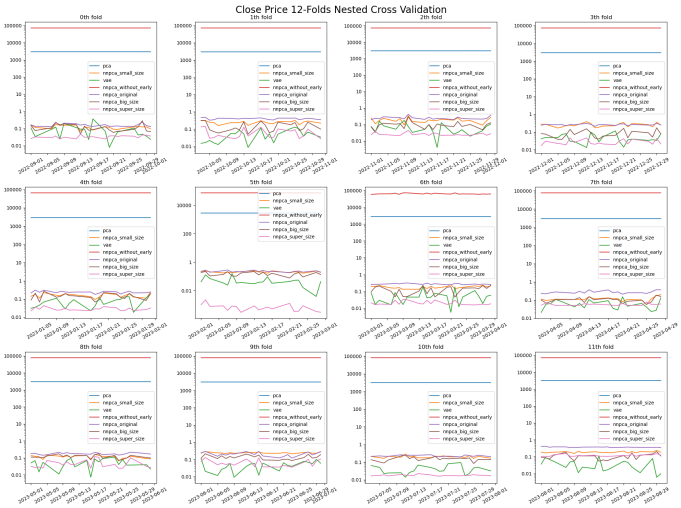


Fig. 5. MSE on 12 fold test

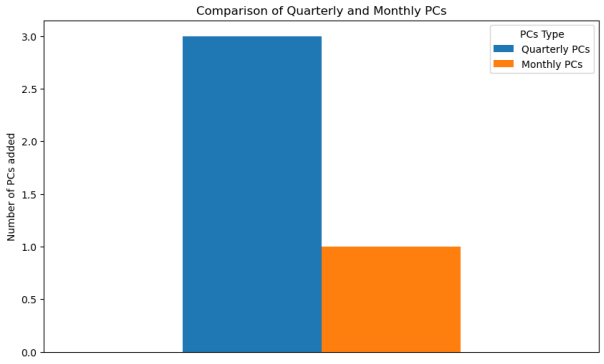


Fig. 6. number of PCs added

C. Isolated group test

In this part, we focus on each isolated component by comparing the mean, standard deviation (STD), and worst-case scenario(WSE) for the indicators identified in our study.

	pca			nnpca			vae		
	mean	std	wse	mean	std	wse	mean	std	wse
Percentage Profit	0.031206	1.021714	-1.269941	**0.231311**	**0.838983**	**-0.951320**	0.077576	0.876004	-0.956291
Profit Factor	1.037161	**0.630244**	0.319716	**1.647091**	1.394315	**0.482128**	1.210725	0.872462	0.364320
Percentage Drawdown	1.166719	**0.356855**	0.737387	**0.604415**	0.431405	**0.201115**	0.729624	0.450946	0.259331
Recovery Factor	0.149432	**1.013720**	**-0.905379**	**1.166188**	2.500207	-0.930939	0.710532	2.279397	-1.000000
Sharpe Ratio	-0.530560	**3.433618**	-6.570996	**1.595733**	4.514034	**-3.917847**	0.201859	3.977265	-5.599389
Sortino Ratio	-0.464355	**5.793339**	-11.190404	**3.586344**	7.862142	**-4.944337**	0.269082	5.950333	-8.760673

Fig. 7. table 1:feature extraction methods statistics.

1) Group 1 Feature extraction methods: In this group, we compared PCA, NNPCA, and VAE. The statistics table 7

presents our findings. Based on these results, NNPCA was identified as the most effective component for our purposes.

2) Group 2 Granularity: In this group, we compared daily + monthly and daily + quarterly, Also, the statistics table 8 presents our findings. Quarterly data was identified as the most effective component for our purposes. This is consistent with our findings regarding the mean correlation coefficient test earlier.

	monthly			quaterly		
	mean	std	wse	mean	std	wse
Percentage Profit	**0.031206**	1.021714	-1.269941	-0.080801	**0.617651**	**-1.169454**
Profit Factor	1.037161	0.630244	0.319716	**1.081326**	**0.570225**	**0.462507**
Percentage Drawdown	1.166719	**0.356855**	0.737387	**0.670911**	0.444895	**0.183757**
Recovery Factor	0.149432	**1.013720**	-0.905379	**0.469488**	1.562517	**-0.745792**
Sharpe Ratio	-0.530560	3.433618	-6.570996	**-0.129583**	**3.231765**	**-4.306970**
Sortino Ratio	-0.464355	**5.793339**	-11.190404	**0.018087**	5.922988	**-8.494166**

Fig. 8. table 2:granularity statistics.

	benchmark			ensemble			svr		
	mean	std	wse	mean	std	wse	mean	std	wse
Percentage Profit	0.031206	1.021714	-1.269941	**0.241888**	**0.522592**	**-0.672929**	0.061986	0.531240	-0.673040
Profit Factor	1.037161	**0.630244**	0.319716	**1.415428**	0.722510	**0.546018**	1.356137	0.992385	0.509087
Percentage Drawdown	1.166719	0.356855	0.737387	0.516062	0.303332	0.174274	**0.499683**	**0.290066**	**0.163617**
Recovery Factor	0.149432	**1.013720**	-0.905379	0.791344	1.421771	**-0.702557**	**0.963473**	2.080245	-0.741259
Sharpe Ratio	-0.530560	3.433618	-6.570996	**1.272110**	**2.875217**	**-3.856029**	0.720369	3.790671	-4.120684
Sortino Ratio	-0.464355	5.793339	-11.190404	2.138571	**5.229664**	-7.815810	**2.365175**	8.014170	**-6.522475**

Fig. 9. table 3:regression methods statistics.

3) Group 3 Regression methods: In this group, we compared OLS(benchmark), Ensemble, and SVR. The statistics table 9 presents our findings. Based on these results, Ensemble was identified as the most effective component for our purposes.

	earning_filter_0.1			earning_filter_0.5			earning_filter_1.0			earning_filter_1.8		
	mean	std	wse	mean	std	wse	mean	std	wse	mean	std	wse
Percentage Profit	**0.083142**	**0.782582**	**-0.739360**	0.057390	0.842630	-0.759403	0.024591	1.078912	-1.397275	-0.029340	1.627411	-2.411629
Profit Factor	**1.257759**	1.070317	**0.439114**	1.110825	0.796184	0.330119	1.025445	0.605222	0.318283	0.976871	**0.507454**	0.312210
Percentage Drawdown	**0.705463**	0.366793	**0.224529**	0.918047	**0.319002**	0.567068	1.231448	0.377044	0.771963	1.770790	0.618367	1.010356
Recovery Factor	**0.542977**	1.877728	**-0.900920**	0.224927	1.191320	-0.905090	0.139140	0.989533	-0.905400	0.112493	**0.913980**	-0.905335
Sharpe Ratio	**-0.236177**	3.775002	**-4.435630**	-0.436997	3.613710	-8.360273	-0.547855	3.397278	-6.569923	-0.617261	**3.210985**	-8.351681
Sortino Ratio	**0.558863**	8.424419	**-9.804078**	-0.019733	6.840843	-10.150259	-0.485237	5.592661	-10.596263	-0.705490	**5.470572**	-11.485148

Fig. 10. table 4:weights for event-based filter statistics.

4) Group 4 Weights for event-based filter: In this group, we compared event-based filters with weights from 0.1-1.9, However, the performance of the statistics was monotonous with weight. So we list a part of the statistics table as 10. Based on these results, weight 0.1 was identified as the most effective component for our purposes.

	best exclude earning_filter			best exclude granularity			best exclude regression			best exclude nnpca			best		
	mean	std	wse	mean	std	wse	mean	std	wse	mean	std	wse	mean	std	wse
Percentage Profit	-0.155897	**0.311072**	-0.586153	-0.020107	0.356328	-0.621782	-0.022430	0.653110	-1.042464	**0.265771**	0.697404	**-0.527067**	-0.159150	0.319597	-0.673778
Profit Factor	0.904286	**0.257052**	**0.573471**	1.059093	0.585562	0.504350	1.403952	1.053963	0.415931	**2.234915**	2.957391	0.380088	0.886150	0.406389	0.480147
Percentage Drawdown	0.681713	**0.229383**	0.355284	**0.393550**	0.254926	0.095141	0.513744	0.426200	0.115488	0.483833	0.257862	**0.064976**	0.463495	0.253155	0.090968
Recovery Factor	-0.158171	**0.504142**	**-0.702816**	0.294552	1.219365	-0.889586	0.863840	2.013957	-0.893417	**2.496626**	5.653775	-0.783785	-0.105333	0.891177	-1.000000
Sharpe Ratio	-0.779746	**1.613779**	**-3.233014**	-0.513358	2.685882	-3.986631	0.472102	3.900644	-5.078833	**0.996097**	4.625525	-4.730146	-1.237473	2.416606	-4.692667
Sortino Ratio	-1.191610	**2.582096**	**-4.857532**	0.010294	5.614751	-6.948734	1.409493	6.524573	-8.521902	**4.054772**	11.737137	-6.634304	-2.053576	4.631785	-11.001078

Fig. 11. table 5:best performed components and ablation studies statistics.

5) Best performed components and ablation studies: In this group, we compared the best-performed components alongside ablation studies and presented the results at 11. Based on these findings, we selected the best-performed component excluding NNPCA, for further investigation.

D. Experiments on full dataset

1) Metrics: In our analysis, we included a selection of alternative models alongside the best-performing model and baseline. Our focus was primarily on the average values of each indicator derived from the 12-fold test. The result is 12.

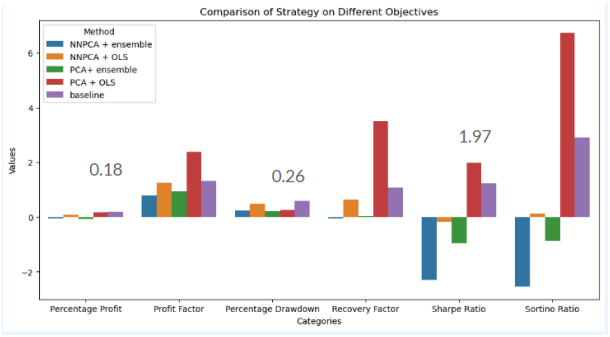


Fig. 12. model comparison on the full dataset: average indicators

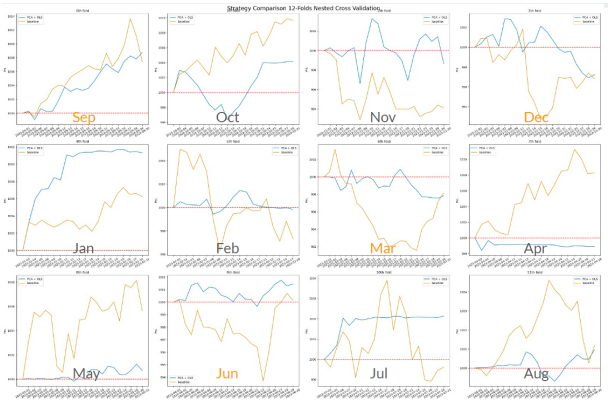


Fig. 13. model comparison on the full dataset: Profit and Loss on 12-fold CV

2) Analysis on 12-fold Cross validation: This graph (see Figure 13) illustrates the Profit and Loss (PnL) results obtained from a 12-fold cross-validation of both the best model and the baseline. Each fold corresponds to one month's performance. Notably, during months marked by more significant earnings events, such as September, December, March, and June, the enhanced model—incorporating PCA + OLS + Quarterly Return + Earning filters at 0.1—showcases enhanced returns and minimized drawdowns.

The visual representation indicates that our modified models outperform during these earnings months, exhibiting improved performance while mitigating risk and reducing variance.

There are notable disparities observed between experiments conducted on reduced and whole datasets. Specifically, models

utilizing NNPCA show underperformance in larger datasets due to convergence issues. Additionally, while the ensemble method surpasses OLS in the reduced dataset, its performance declines when applied to larger datasets. Addressing the convergence issues with NNPCA in larger datasets and understanding the varying effectiveness of models across different dataset sizes can be key focal points for further investigation and improvement.

VIII. Conclusion

Our study introduces several methods to enhance the baseline statistical arbitrage model, specifically targeting company-specific risks during earning seasons. These methods encompass feature extraction, identifying multi-collinearity among risk factors at various levels of detail, optimizing statistical arbitrage strategies, and implementing an earning event-based weighted filter.

In our feature extraction analysis, we discovered that PCA (Principal Component Analysis) outperforms NNPCA (Neural Network Principal Component Analysis) when extracting risk factors across the entire dataset. PCA's linear approach renders it more consistent compared to NNPCA. Interestingly, while NNPCA performs better in a reduced dataset, it encounters convergence issues when applied to the entire dataset, limiting its effectiveness in this context.

In the detection of multi-collinearity among risk factors of varying granularity, we've observed that quarterly returns outperform monthly returns in extracting these factors. Using quarterly returns tends to extract a greater number of principal components with lower correlation compared to monthly returns, making them more effective in this paper with quarterly earnings releases.

In our comparison of performance between Ensemble regression using Support Vector Regression (SVR) and Ordinary Least Squares (OLS) versus using OLS and SVR independently for statistical arbitrage optimization, we noticed distinct outcomes between the reduced and full datasets. Interestingly, the ensemble method demonstrates superior performance in the reduced dataset. We intend to investigate this further in future steps, as we suspect that the variation in the number of stocks within a cluster might be a contributing factor to the disparity observed between these two approaches.

From the ablation studies, we've found that the earnings-based weighted filter is the component that's truly operational. When supported by other elements, it significantly enhances performance across both large and small stock sets. Adopting smaller positions for stocks during earning periods appears to improve the strategy's performance by minimizing substantial risks.

Our statistical arbitrage model has shown substantial enhancements across nearly all metrics compared to our current baseline model. Beyond success in profit generation, it notably improves percentage drawdown and mitigates risks related to earnings events. The outcomes of our experiments have proven rewarding and aligned with our expectations, meeting the objectives of our project. Additionally, there are unexpected positive results that require further explanation, which, unfortunately, we cannot address within our current time constraints. Details will be in the subsequent section, Next Steps.

IX. Next Steps

A. Research on our hypothesis with NNPCA and PCA

The performance gap between NNPCA and PCA is notably significant when applied to small and large stock datasets. While NNPCA showcases superior performance in smaller datasets, it encounters convergence issues across some folds when tested on the entire dataset, hindering its reliability and effectiveness in this scenario. We suggest some hypotheses about this issue, including percentage account to the total capital market, cluster sizes, and incompatibility to linear regression methods.

B. Research on OLS and ensemble models performance on small and large dataset

The substantial difference in performance between OLS and the ensemble method becomes apparent when applied to both small and large stock datasets. As previously highlighted in our conclusion, this discrepancy might indeed be linked to the number of stocks present within individual clusters. Further investigation into this aspect could provide valuable insights into understanding and potentially addressing this observed effect.

C. Research on giving more weights to recent data

At present, we assign equal weight to all data in our models, yet we believe that recent data likely holds more pertinent information than older data. To address this, we can consider assigning greater weights to recent data. For instance, in determining the weight for each risk factor for individual stocks, we could prioritize a higher weight for the most recent time window.

D. Other corporate events

There are additional pre-scheduled corporate events, like dividend payouts, that we can identify in advance. However, our expectation is that these events might not significantly impact stock volatility, as this information typically doesn't influence company performance. To verify this, we can conduct hypothesis testing comparing the periods around dividend payouts to normal periods to ascertain their impact on stock behavior.

X. Team Member's contribution

Throughout the semester, we've diligently divided tasks and adhered to the project plans. However, it's important to note that additional responsibilities such as report writing, presentations, and similar tasks are not explicitly accounted for in our contributions outlined below. These tasks, while essential, might not be reflected in the specific contributions detailed below.

1) **Zhang Jianan**
   - Risk factors modification
   - Ensemble regression modification
   - Experiment with modified models
2) **Ha Nguyen**
   - Data Preparation and Preprocessing
   - Implement baseline model
   - Earning event-based weighted filter
   - Experiment with modified models
3) **Xu Long**
   - Risk factors modification
   - Evaluate model performance
   - Result analysis

REFERENCES

[1] BALLADARES, K., RAMOS-REQUENA, J. P., TRINIDAD-SEGOVIA, J. E., AND SÁNCHEZ-GRANERO, M. A. Statistical arbitrage in emerging markets: a global test of efficiency. *Mathematics 9*, 2 (2021), 179.

[2] BASTOS, J. A., AND CAIADO, J. Clustering financial time series with variance ratio statistics. *Quantitative Finance 14*, 12 (2014), 2121–2133.

[3] BLITZ, D., HUIJ, J., AND MARTENS, M. Residual momentum. *Journal of Empirical Finance 18*, 3 (2011), 506–521.

[4] BOWLES, B., REED, A. V., RINGGENBERG, M. C., AND THORNOCK, J. R. Anomaly time. *Available at SSRN 3069026* (2023).

[5] CARTA, S. M., CONSOLI, S., PODDA, A. S., RECUPERO, D. R., AND STANCIU, M. M. Ensembling and dynamic asset selection for risk-controlled statistical arbitrage. *IEEE Access 9* (2021), 29942–29959.

[6] CONTRERAS, H., AND MARCET, F. Arbitrageurs and overreaction to earnings surprises. *Finance Research Letters 43* (2021), 101994.

[7] CUOMO, S., GATTA, F., GIAMPAOLO, F., IORIO, C., AND PICCIALLI, F. An unsupervised learning framework for marketneutral portfolio. *Expert Systems with Applications 192* (2022), 116308.

[8] DE ANGELIS, L. Latent class models for financial data analysis: some statistical developments. *Statistical Methods & Applications 22* (2013), 227–242.

[9] DEMIR, S., STAPPERS, B., KOK, K., AND PATERAKIS, N. G. Statistical arbitrage trading on the intraday market using the asynchronous advantage actor–critic method. *Applied Energy 314* (2022), 118912.

[10] DIAS, J. G., VERMUNT, J. K., AND RAMOS, S. Clustering financial time series: New insights from an extended hidden markov model. *European Journal of Operational Research 243*, 3 (2015), 852–864.

[11] D'URSO, P., DE GIOVANNI, L., AND MASSARI, R. Garch-based robust clustering of time series. *Fuzzy Sets and Systems 305* (2016), 1–28.

[12] ENGELBERG, J., MCLEAN, R. D., AND PONTIFF, J. Anomalies and news. *The Journal of Finance 73*, 5 (2018), 1971–2001.

[13] GATTA, F., IORIO, C., CHIARO, D., GIAMPAOLO, F., AND CUOMO, S. Statistical arbitrage in the stock markets by the means of multiple time horizons clustering. *Neural Computing and Applications* (2023), 1–19.

[14] GU, S., KELLY, B., AND XIU, D. Empirical asset pricing via machine learning. *The Review of Financial Studies 33*, 5 (2020), 2223–2273.

[15] GUIJARRO-ORDONEZ, J., PELGER, M., AND ZANOTTI, G. Deep learning statistical arbitrage. *arXiv preprint arXiv:2106.04028* (2021).

[16] HVID, A. K., AND KRISTIANSEN, K. L. How news affects sectoral stock prices through earnings expectations and risk premia. Tech. rep., Danmarks Nationalbank Working Papers, 2021.

[17] IMAJO, K., MINAMI, K., ITO, K., AND NAKAGAWA, K. Deep portfolio optimization via distributional prediction of residual factors. In *Proceedings of the AAAI conference on artificial intelligence* (2021), vol. 35, pp. 213–222.

[18] IORIO, C., FRASSO, G., D'AMBROSIO, A., AND SICILIANO, R. A p-spline based clustering approach for portfolio selection. *Expert Systems with Applications 95* (2018), 88–103.

[19] LADRÓN DE GUEVARA CORTÉS, R., TORRA PORRAS, S., AND MONTE MORENO, E. Extraction of the underlying structure of systematic risk from non-gaussian multivariate financial time series using independent component analysis: evidence from the mexican stock exchange. *Computación y Sistemas 22*, 4 (2018), 1049–1064.

[20] LADRÓN DE GUEVARA CORTÉS, R., TORRA, S., AND MORENO, E. Neural networks principal component analysis for estimating the generative multifactor model of returns under a statistical approach to the arbitrage pricing theory: Evidence from the mexican stock exchange. *Computación y Sistemas 23* (06 2019).

[21] LI, T. N., AND PAPANICOLAOU, A. Statistical arbitrage for multiple co-integrated stocks. *Applied Mathematics & Optimization 86*, 1 (2022), 12.

[22] LIAO, T. W. Clustering of time series data—a survey. *Pattern recognition 38*, 11 (2005), 1857–1874.

[23] MANCISIDOR, R. A., KAMPFFMEYER, M., AAS, K., AND JENSSEN, R. Learning latent representations of bank customers with the variational autoencoder. *arXiv preprint arXiv:1903.06580* (2019).

[24] MONTESDEOCA, L., SQUIRES, S., AND NIRANJAN, M. Variational autoencoder for non-negative matrix factorization with exogenous inputs applied to financial data modelling. In *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)* (2019), pp. 312–317.

[25] OMRAN, M. F. Identifying risk factors within the arbitrage pricing theory in the egyptian stock market. *University of Sharjah Journal 2*, 2 (2005), 103–119.

[26] RAPACH, D. E., STRAUSS, J. K., AND ZHOU, G. Out-of-sample equity premium prediction: Combination forecasts and links to the real economy. *The Review of Financial Studies 23*, 2 (2010), 821–862.

[27] SCHOLZ, M., AND VIGÁRIO, R. Nonlinear pca: a new hierarchical approach. In *Esann* (2002), pp. 439–444.

[28] TSAFACK, G., BECKER, Y., AND HAN, K. Earnings announcement premium and return volatility: Is it consistent with risk-return trade-off? *Pacific-Basin Finance Journal 79* (2023), 102029.

[29] TZAGKARAKIS, G., CAICEDO-LLANO, J., AND DIONYSOPOULOS, T. Exploiting market integration for pure alpha investments via probabilistic principal factors analysis.

[30] YIP, F., AND XU, L. An application of independent component analysis in the arbitrage pricing theory. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium* (2000), vol. 5, pp. 279–284 vol.5.

[31] ZHAO, Z., ZHOU, R., AND PALOMAR, D. P. Optimal mean-reverting portfolio with leverage constraint for statistical arbitrage in finance. *IEEE Transactions on Signal Processing 67*, 7 (2019), 1681–1695.

APPENDIX

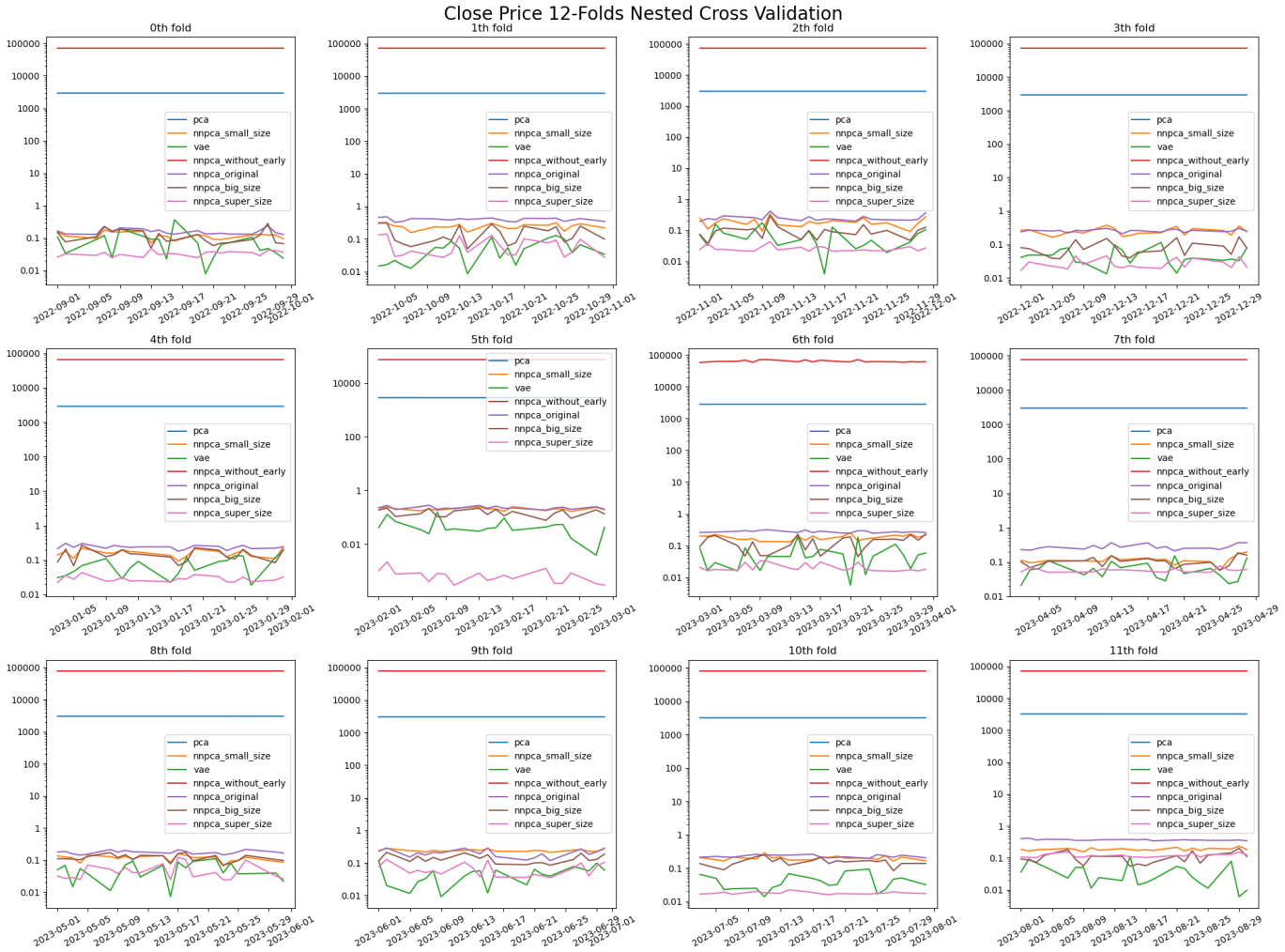


Fig. 14. MSE on 12 fold test

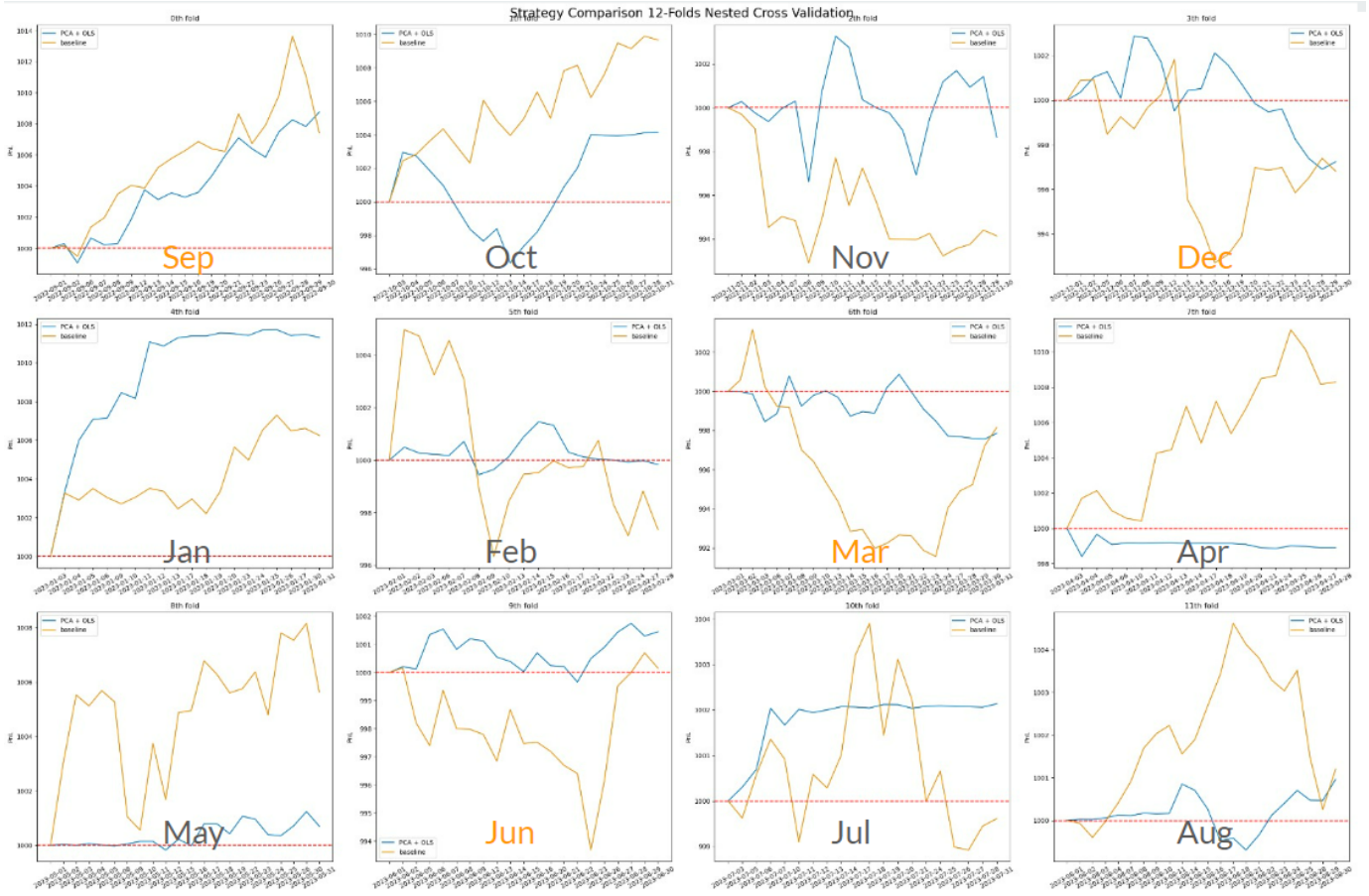


Fig. 15. MSE on 12 fold test