Scenario Generation and Data Augmentation in Quantitative Wealth and Investment Management

(QWIM)

FE/FA-800 Project

Fall 2021

Team members: Abhimanyu Singh
Jeremiah Makaya
Sishir Yerra Doddi

Advisor : Dr. Cristian Homescu

CONTENTS

1.0 INTRODUCTION

In this project we applied neural networks and data augmentation techniques to generate scenarios. We compared the efficacy of the GAN Model, the Random Forest Model and the LTSM Model for the purposes of generating scenarios from financial time series data and empirically established the best method for scenario generation. We identified the LTSM model as the best model for generating scenarios. For data augmentation, we applied the Additive Gaussian data augmentation to financial time series data.

We were not satisfied just generating scenarios and predicting future dynamics of the financial markets, we went as far as to develop trading strategies based on data augmentation which can outperform a simple buy and hold strategy of the S&P 500. We able to develop a factor trading strategy using augmented historical prices of the S&P 500 sector ETFs from January 2011 to December 2020. The factor trading strategy did way better than the S&P 500 over the period January 2021 to June 2021.

We understand that some trading and investment strategies in the context of Quantitative Wealth and Investment Management only do better in unique market environments. To safeguard against perhaps the inappropriate application of our trading strategies in the future, we performed some portfolio and risk management analytics in order to gain a full understanding of the historical context in which the results are derived.

2.0 LITERATURE REVIEW

2.1.1 Data Augmentation

Data augmentation amounts to defining an underlying data-dependent distribution and generating new data points stochastically from this underlying distribution [Ziyin et al. 2021].

Considering a training loss function of the form $L = \frac{1}{N}\sum_i l(x_i, y_i)$ for N pairs of training data points $\{x_i, y_i\}_i^N$, a general way to define data augmentation is to start with a datum-level training loss and transform it to an expectation over an augmentation distribution $P\big(z|(x_i, y_i)\big)$ [Dao et al. 2019],

$$l(x_i, y_i) \rightarrow \mathbb{E}_{(z,g_i)\sim P(z,g|(x_i,y_i))}[l(z_i, g_i)],$$

and the total training loss function becomes

$$L_{aug} = \frac{1}{N}\sum_i \mathbb{E}_{(z,g_i)\sim P(z,g|(x_i,y_i))}[l(z_i, g_i)].$$

One common example of data augmentation is injecting isotropic Gaussian noise to the input [Shorten et al., 2020], which is equivalent to setting:

$$P\big(z, g|(x_i, y_i)\big)\sim\delta(g, y_i)exp[-(z - x_i)^T(z - x_i)/(2\sigma^2)]$$

for some specified strength σ^2.

2.1.2 Additive Data Augmentation

Under the Additive Gaussian Data Augmentation we inject some Gaussian noise into the historical stock prices data. The random noise we inject is:

$$\varepsilon_t \sim N(0,\ \rho^2)$$

The augmented data will then be

$$z_t = S_t + \varepsilon_t$$

When calculating returns the noise should not appear in the denominator else that will introduce a drift into the augmented data.

$$\tilde{r}_t = \frac{z_{t+1} - z_t}{S_t}$$

The optimal strength ρ is calibrated to maximise the true utility function for the additive Gaussian data augmentation. This true utility function is:

$$U_{Add} = \frac{r^2}{2\lambda\sigma^2 T}\mathbb{E}_{S_t}\left[\frac{(\sum_t r_t S_t)^2}{\sum_t (r_t S_t)^2}\Theta(\sum_t r_t S_t^2)\right]$$

Where Θ is the Heaviside step function. [Ziyin et al. 2021].

The result can be shown to be:

$$\rho^2 = \frac{\sigma^2}{2r}\frac{\sum_t (r_t S_t^2)^2}{\sum_t r_t S_t^2}$$

The proof is attached in the Appendix.

2.2.1 Factor Analysis

Factor analysis is particularly suitable to extract few factors from the large number of related variables to a more manageable number, prior to using them in other analysis such as multiple regression or multivariate analysis of variance. [*Noora* Shrestha et al. 2021].

Factor analysis achieves this by clustering similar variables into the same factor to identify underlying variables and it only uses the data correlation matrix.
In this study, Kaiser-Meyer-Olkin measure of sampling adequacy and Bartlett's test of Sphericity are used to assess the factorability of the financial time series data. To determine the number of factors to be extracted, Kaiser's Criterion and Scree test are examined. Varimax orthogonal factor rotation method is applied to minimize the number of variables that have high loadings on each factor.

2.2.1 Factor Model with 'm' Common Factors

Let $X = (X_1, X_2, \cdots, X_p)$ be a random vector with mean vector μ and covariance matrix Σ. The factor analysis model assumes that

$$X = \mu + \lambda F + \varepsilon$$

Where:

$\lambda = \{\lambda_{jk}\}_{p \times m}$ denotes the matrix of factor loadings

λ_{jk} is the loading of the j[th] variable on the k[th] common factor, $F = (F_1, F_2, \cdots, F_m)$ denotes the vector of latent factor scores

F_k is the score on the k[th] common factor and

$\varepsilon = (\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_p)$ denotes the vector of latent error terms;

ε^j is the j[th] specific factor.

2.2.3 Kaiser-Meyer-Olkin (KMO) Measure of Sampling Adequacy

KMO test is a measure that has been intended to measure the suitability of data for factor analysis. In other words, it tests the adequacy of the sample size. The test measures sampling adequacy for each variable in the model and for the complete model. The KMO measure of sampling adequacy is given by the formula:

$$KMO_j = \frac{\sum_{i \neq j} R_{ij}^2}{\sum_{i \neq j} R_{ij}^2 + \sum_{i \neq j} U_{ij}^2}$$

where:

R_{ij} is the correlation matrix and

U_{ij} is the partial covariance matrix.

The KMO values vary from 0 to 1. The KMO values between 0.8 to 1.0 indicate the sampling is adequate. KMO values between 0.7 to 0.79 are middling and values between 0.6 to 0.69 are mediocre. KMO values less than 0.6 indicate the sampling is not adequate and the remedial action should be taken. If the value is less than 0.5, the results of the factor analysis undoubtedly won't be very suitable for the analysis of the data. If the sample size is < 300 the average communality of the retained items has to be tested. [Tabachnick et al. 2013]

2.2.4 Bartlett's Test of Sphericity

Bartlett's Test of Sphericity tests the null hypothesis:

H_0: The variables are orthogonal i.e. The original correlation matrix is an identity matrix indicating that the variables are unrelated and therefore unsuitable for structure detection.

The alternative hypothesis:

H_1: The variables are not orthogonal i.e. they are correlated enough to where the correlation matrix diverges significantly from the identity matrix.

The significant value < 0.05 indicates that a factor analysis may not be worthwhile for the data set. In order to measure the overall relation between the variables the determinant of the correlation matrix $|R|$ is calculated. Under H_0, $|R| = 1$; if the variables are highly correlate, then $|R| \approx 0$. The Bartlett's test of Sphericity is given by:

$$\chi^2 = -\left(n - 1 - \frac{2p + 5}{6}\right) \times ln\,|R|$$

where,

p = number of variables,

n = total sample size and

R = correlation matrix

[Guttman et al. 1954].

2.2.5 Kaiser's (Eigenvalue) Criterion

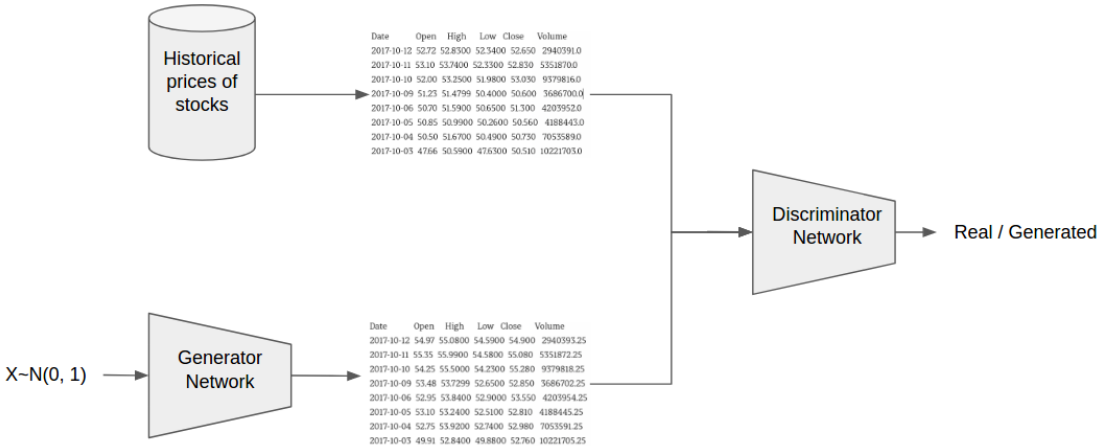
The eigenvalue of a factor represents the amount of the total variance explained by that factor. In factor analysis, the remarkable factors having eigenvalue greater than one are retained. The logic underlying this rule is reasonable. An eigenvalue greater than one is considered to be significant, and it indicates that more common variance than unique variance is explained by that factor[Pallant et al. 2010].

2.2.6 Scree Test

Cattell (1996) proposed a graphical test for determining the number of factors. A scree plot graphs eigenvalue magnitudes on the vertical access, with eigenvalue numbers constituting the horizontal axis. The eigenvalues are plotted as dots within the graph, and a line connects successive values. Factor extraction should be stopped at the point where there is an 'elbow' or levelling of the plot. This test is used to identify the optimum number of factors that can be extracted before the amount of unique variance begins to dominate the common variance structure [Hair et al. 1998].

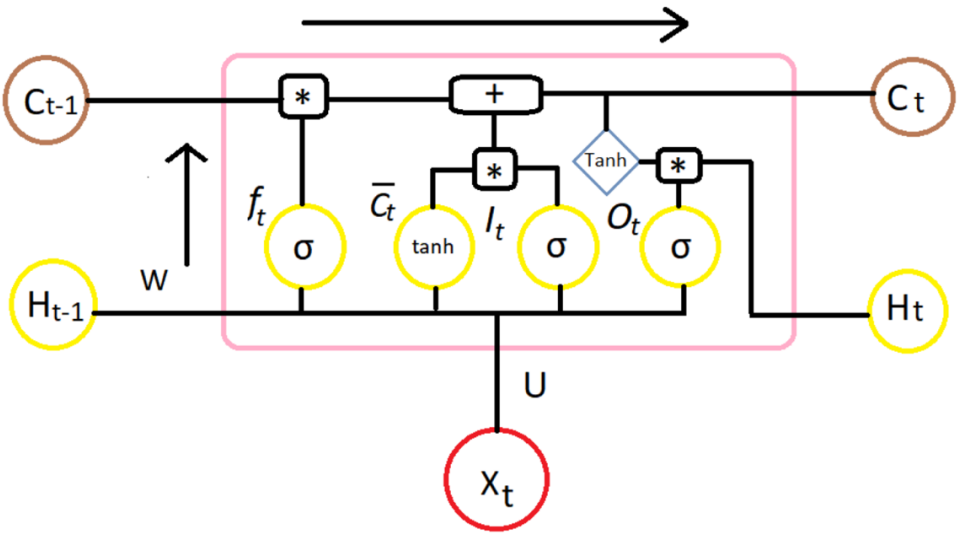
2.3.1 The Time GAN model

- Time GAN basically generates 2 neural networks.
    1. Creates new data with same statistics as training data also known as generator
    2. Differentiates between real and fake data also known as adversary/discriminator
- The Generator attempts to create data that fools adversary data.



2.3.2 The LSTM Model

- LSTM model is an example of gated RNN.
- It performs better than RNN when there is a time series data.
- LSTM has network cells that have an internal recurrence and outer recurrence of RNN that contains an additional state to remember the past.



$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
\tilde{c}_t &= \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
$$

- $\vec{x_t} \in \mathbb{R}^d$: input vector to the LSTM unit
- $f_t \in (0,1)^h$: forget gate's activation vector
- $i_t \in (0,1)^h$: input/update gate's activation vector
- $o_t \in (0,1)^h$: output gate's activation vector
- $h_t \in (-1,1)^h$: hidden state vector also known as output vector of the LSTM unit
- $\tilde{c}_t \in (-1,1)^h$: cell input activation vector
- $c_t \in \mathbb{R}^h$: cell state vector
- $W \in \mathbb{R}^{h \times d}$, $U \in \mathbb{R}^{h \times h}$ and $b \in \mathbb{R}^h$: weight matrices and bias vector parameters which need to be learned during training

2.4 ANALYSIS OF THE MARKET CONDITIONS

2.4.1 Capital asset pricing model (CAPM)

Every investment is associated with some level of risk. So investors would like to learn about the rate of return to compensate the risk factor. The capital Asset Pricing Model is a pricing method that helps evaluate the level of risk and return of investment based on the risk. CAPM model framework was developed in 1960 by Willian Sharpe and his team. [Mullins, 1982] CAPM model is defined as follows

$$R_a = R_{rf} + \beta_a * (R_m - R_{mf}), \text{ Where}$$

R_a- Expected Return on a security
R_{rf}- Risk-free rate
R_m- Expected Return of the market
β_a- The beta of the security
$(R_m - R_m)$ - Equity market premium

A stock's beta plays a crucial role in CAPM as it measures the stock's risk or relative volatility. It helps one understand the direction of stock jumps compared to the market. If stock price and market price move in the same line, beta can be considered close to 1. A stock will improve by 10% if the market improves by 10% and vice-versa if the beta is 1. Therefore, CAPM is a simple model which provides simple results to check which stock to invest in based on the risk of the stock [Mullins, 1982].

2.4.2 Monte Carlo Simulation

Monte Carlo simulation is a risk analysis technique that involves creating models of probable outcomes by substituting a set of values a probability distribution for any factor with inherent uncertainty. It then repeats the process, using different random values from the probability functions each time. A Monte Carlo simulation could take thousands or tens of thousands of recalculations to complete, depending on the amount of uncertainty and the ranges provided for them. Monte Carlo simulation generates probabilistic distributions of alternative outcomes [Raychaudhuri, 2008].

2.4.3 Kullback-Lieber Divergence

When working with a prediction model or a model that's been impacted by shocks, One would like to know the difference between actual and observed probability distribution. In such a scenario, Kullback-Lieber Divergence (KL) method can be utilized to determine the distance as it quantifies how much one's probability distribution differs from another probability distribution. For instance. The predicted stock data can be compared with its actual data to determine whether the machine learning model supports the original data.

KL divergence between two distributions, P and Q, is defined as follows:

If data is discrete,

$$D_{KL}(P \parallel Q) = \sum P(i) * Log(\frac{P(i)}{Q(i)})$$

If data is continuous,

$$D_{KL}(P \parallel Q) = \int P(x) * Log(\frac{P(x)}{Q(x)})$$

KL divergence value implies that when the probability for an event from P is large, but the probability for the same event in Q is small, there is a large divergence. When the likelihood from P is small, and the probability from Q is large, there is also a large divergence, but not as large as the first case. KL divergence score ranges from 0 to infinite. The distributions are considered to be identical when KL divergence value is 0.

3. METHODOLOGY

We built two factor trading strategies, one based on the actual historical data of the sector ETFs of the S&P 500 and another based on the augmented data.

Overview of the Factor Trading Strategy

Based on factor analysis, we choose 4 sectors to invest in from the 11 sector ETFs. These 4 sectors comprise of 2 sectors with the highest factor sensitivities, and another 2 with lowest factor sensitivities.

There are THREE major steps in our factor trading strategy:

- assessment of the suitability of the data for factor analysis
- factor extraction, and
- factor rotation, factor sorting and identification of sectors to invest in.

The first step is conducting statistical tests to assess the suitability of the historical data for factor analysis. We are using two test in our project to establish suitability of factor analysis for the time series data we are using in our project. The two tests are:

Bartlett's test of sphericity - checks whether or not the observed variables intercorrelate at all using the observed correlation matrix against the identity matrix. If the test turns out to be statistically insignificant, we do not employ a factor analysis.

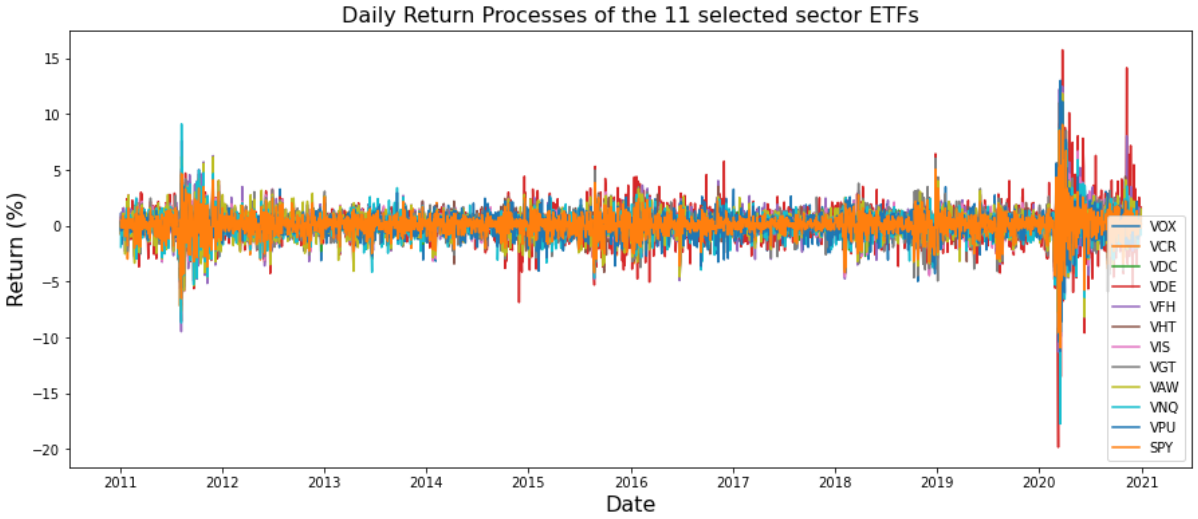
Kaiser-Meyer-Olkin (KMO) test - measures the suitability of data for factor analysis. It determines the adequacy for each observed variable and for the complete model. KMO estimates the proportion of variance among all the observed variables. KMO values range between 0 and 1. Value of KMO less than 0.6 is considered inadequate.

The next step is to determine the number of factors necessary to capture around 80% of the total variability of the data then fit the factor model with Varimax rotation and generate the factor loadings. We will then sort the sectors according to their loadings. In our investment strategy we will pick the top two and bottom two sectors for investment, the sectors in the middle are factor neutral, so we do not invest in them. We will repeat the entire process with augmented data and compare the results.

We will also generate scenarios using neural networks, however, we will mainly assess the efficacy of the different possible ways of generating scenarios.Some portfolio and risk management analytics will also be implemented to gain a full understanding of the historical context in which the results are derived.

4. RESULTS

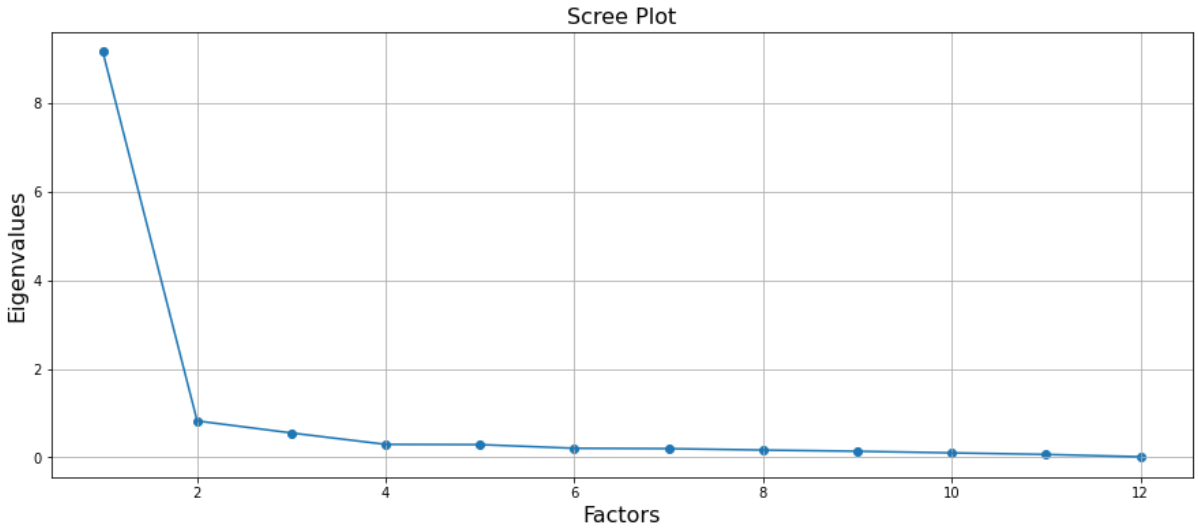
We downloaded historical prices of the S&P 500 sector ETFs from yahoo finance using a custom function get_tickers() we developed in python for the period 2011-01-01 to 2020-12-31. We computed the daily returns of the data. The graph of the daily returns of the S&P 500 11 sector ETFs for the period 2011-01-01 to 2020-12-31 is shown below:



The results of the suitability of factor analysis on the historical data using the KMO and Bartlett's tests of these returns are as follows:

	Bartletts Test	**KMO Test**
Test Value	43321.9	0.926299
Criteria	Test Value Significant	Test Value > 0.6

The results of the Scree Plot of the eigenvalues of the historical daily returns to determine the appropriate number of factors before Varimax Rotation are as follows:



The Scree Plot elbows after two factors showing that two factors capture the common variance of the historical daily returns. With two factors the next step is to fit a model with Varimax rotation. The results are shown below:

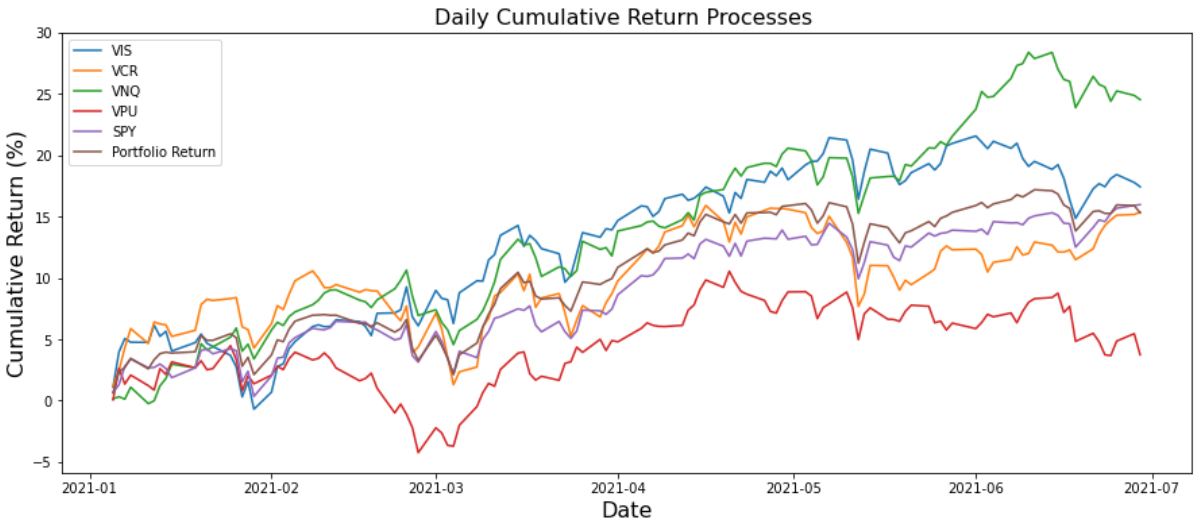
	Factor 1	Factor 2
VIS	0.879114	0.382987
VCR	0.86051	0.356644
VAW	0.853614	0.355088
VFH	0.814847	0.395478
VGT	0.813295	0.342165
VHT	0.729988	0.441545
VDE	0.724098	0.281526
VOX	0.716911	0.417309
VDC	0.566719	0.655348
VNQ	0.545062	0.652354
VPU	0.261503	0.913839

The factor trading strategy would be to invest in the highlighted sectors:

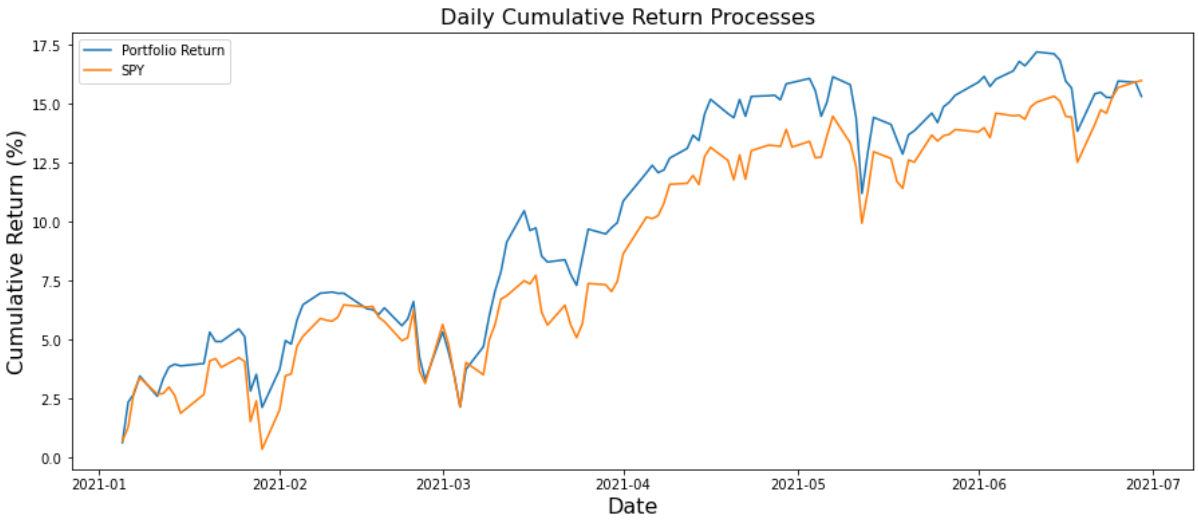
VIS VCR VNQ VPU

The sectors in the middle of the table above are relatively factor neutral. This is the outcome of the factor trading investment strategy to be applied going forward. The data used to build the strategy was drawn from 2011-2020. Implementing this strategy on the future (2021-01-01 to 2021-06-30) and comparing the results with S&P 500 benchmark yielded the following returns and financial metrics:

(1) The Daily returns of all the tickers, the portfolio formed by the four tickers and the S&P 500 benchmark are shown below:



(2) For a clearer comparison, just the S&P 500 cumulative returns and the portfolio are shown below:



	Portfolio	SPY
Sharpe Ratio	14.97%	14.68%
Total Returns	15.97%	15.30%

The factor trading strategy performed better than the S&P 500 index. The portfolio built using a factor trading strategy on the actual historical data yielded higher returns and a higher Sharpe Ratio too than the S&P 500.

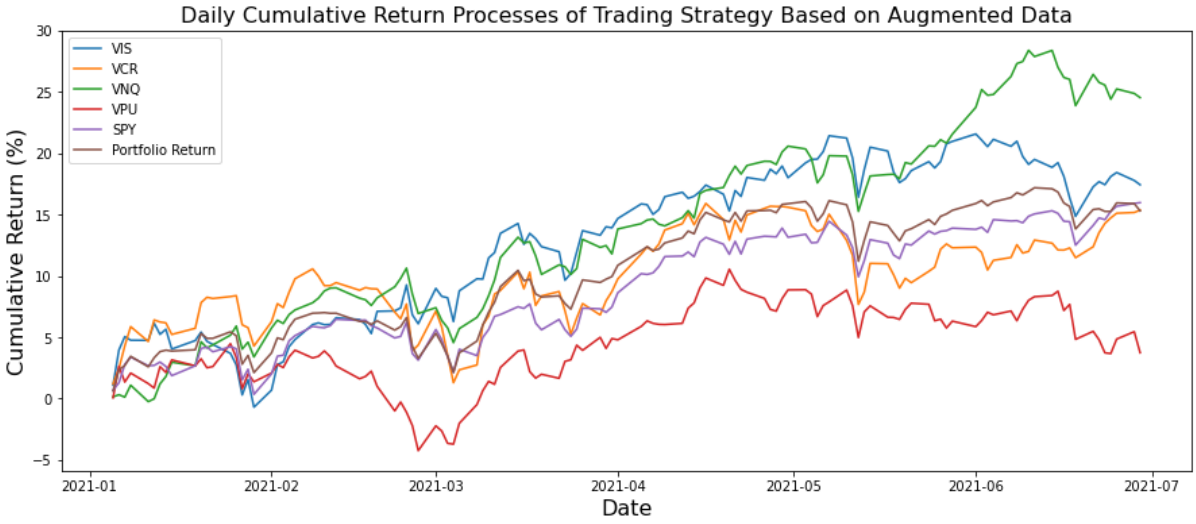
Part of our goal is to empirical assert whether using augmented data improves this investment outcome. It turns out augmenting data before applying factor analysis on it generates investment tickers which perform way better than if the actual data is used. The procedure is the same as with the implementation on the actual data. The only difference is that augmented historical returns data is used in place the actual returns data. The python codes for this implantation are attached. We tested the thousands of the possible scenarios and each time, the returns of the resulting portfolios were always at least as good as those of the actual data. Graphs depicting the general performance of the augmented strategy are shown below:

The outcome of the factor investment strategy on the augmented data is:

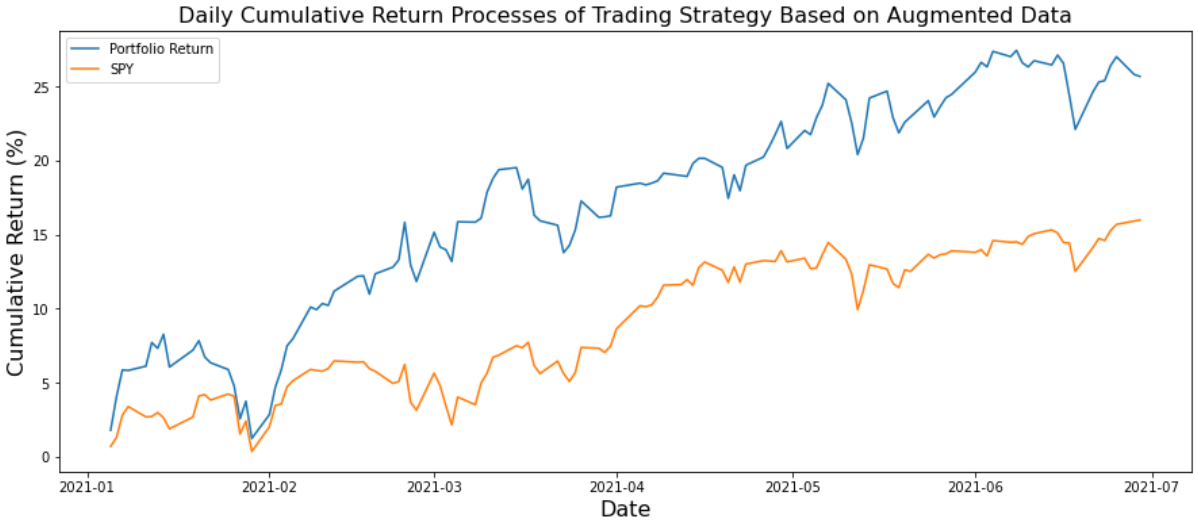
VDE VIS VGT VFH

There are still some common sectors in the outcomes, however, the performances are remarkably different.

(1) Daily cumulative returns of the augmented factor trading strategy:



(1) For a clearer comparison, just the S&P 500 cumulative returns and the portfolio are shown below:

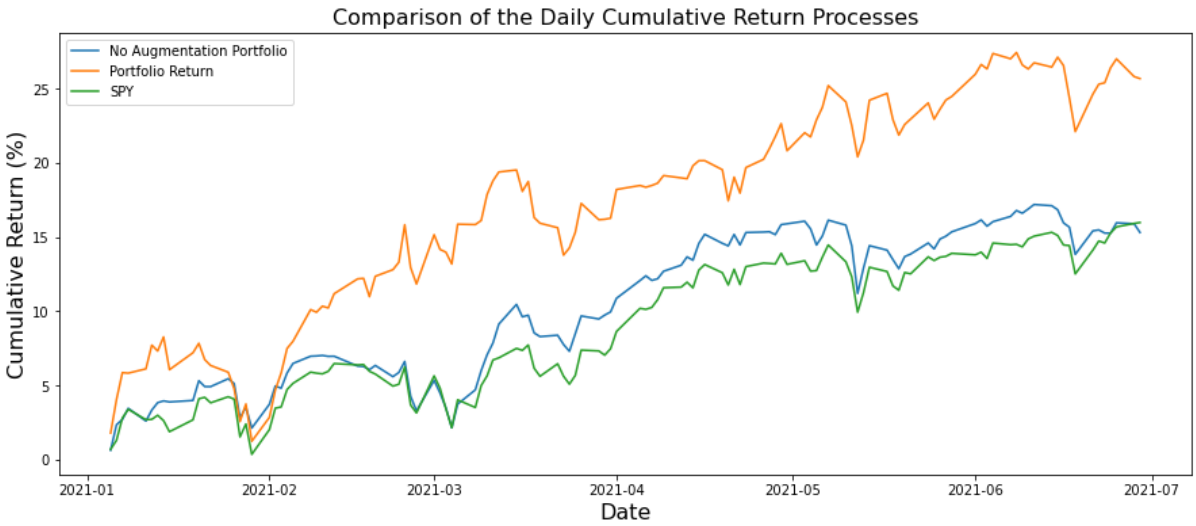


	Portfolio	SPY
Sharpe Ratio	17.67%	14.68%
Total Returns	25.68%	15.97%

To sum it all up:

	SPY (Benchmark)	Strategy 1 (Actual Data)	Strategy 2 (Augmented Data)
Returns	15.97%	15.30%	25.68%
Sharpe Ratios	14.68%	14.97%	17.67%

The graph below compares all the three strategies above:



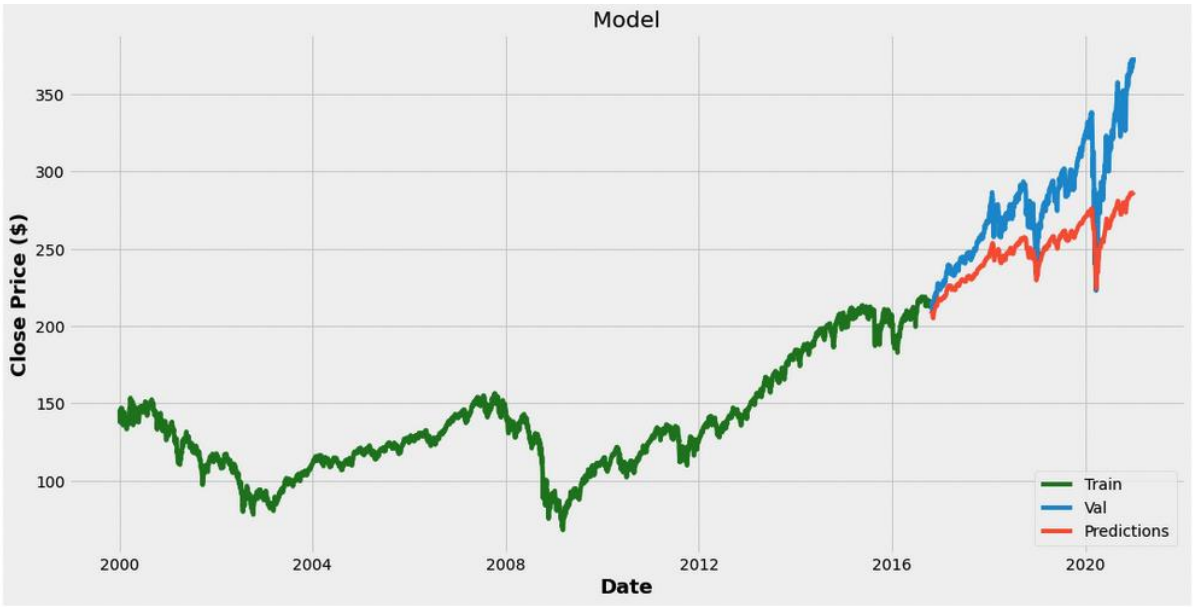
The returns and Sharpe ratios of the augmented data factor trading strategy are superior to those of the S&P 500 and the actual data factor trading strategy. In view of this, augmenting data not only efficiently produces new scenarios but is also produces superior trading strategies if used with factor trading strategies.

Results of Scenario generation using the time GAN model

- For scenario generation we took 2 machine learning models
- Firstly, we implemented GAN model on S&P 500
- Here is our results:

It can be seen from above that the time GAN model is not efficient for generating scenarios from financial time series. This could be due to over fitting.

Results of Scenario generation using the LTSM model



- LSTM model did better when compared to GAN model
- Root mean square value of GAN model was 35.73
- Root mean square value of LSTM model was 8.38

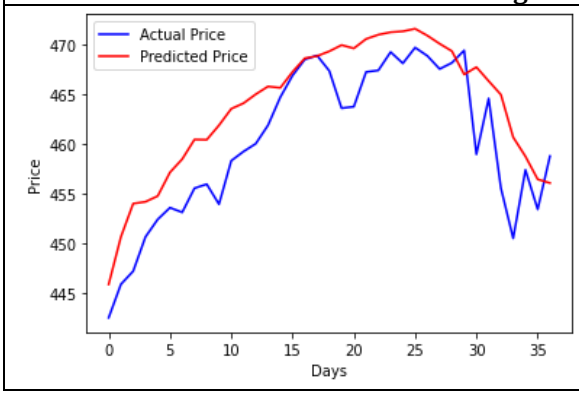
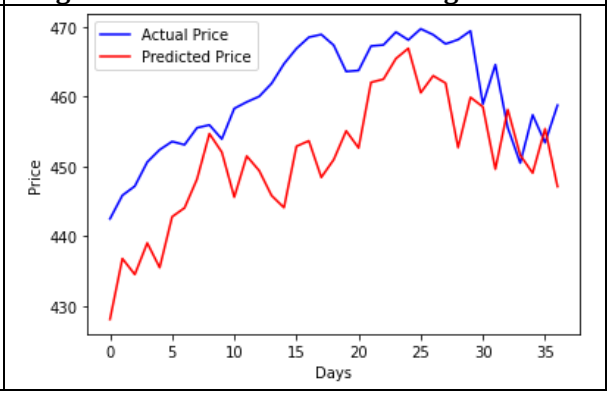
- Hence LSTM performed better than GAN
- LSTM performed better when compared to GAN because LSTM contains additional state to remember long and short term data, hence it knows which type of data to generate
- But GAN only follows the distribution of data it on which we train it.
- Hence, we will use LSTM to get our scenarios.

Results of using the LTSM model on augmented data

Data augmentation also enhances the prediction performance of the LTSM model. The results of the predictions from an LSTM model trained on:

(1) Augmented historical data
(2) Actual historical data

Are depicted below:

Actual Historical Data Used As Training Set	Augmented Data Used As Training Set
	

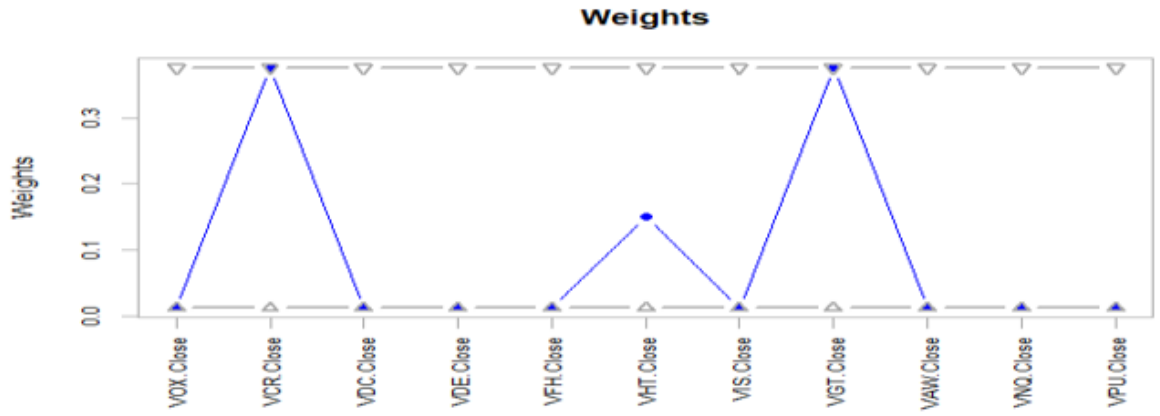
The data used in this case is the S&P 500 data from 2020-10-31 to 2021-11-30. 80% of the data is used for triaging and 20% for testing.

	Training Set	
	Actual Historical Data	**Augmented Data**
Mean Absolute Error	363.41446	359.57292
Future price after 1 days is	$454.79	$452.99
Accuracy Score	0.72222	0.30556

It is evident that augmenting the training set reduces over fitting when training the LTSM model of historical financial time series data. The absolute error in the data above is lower for predictions produced by the model trained on augmented historical data. The python code which produced the results above is attached.

Risk Environment Assessment

In order to gain a full understanding of the market dynamics upon which we build our trading strategies and our predictive models, we shocked the data of the 11 sector ETFs of the S&P 500 and established the following on mean-variance hypothetical portfolio:




```
Optimal Weights:
VOX.Close VCR.Close VDC.Close VDE.Close VFH.Close VHT.Close VIS.Close VGT.Close
VAW.Close VNQ.Close VPU.Close
   0.0125    0.3750    0.0125    0.0125    0.0125    0.1500    0.0125    0.3750
0.0125    0.0125    0.0125
```

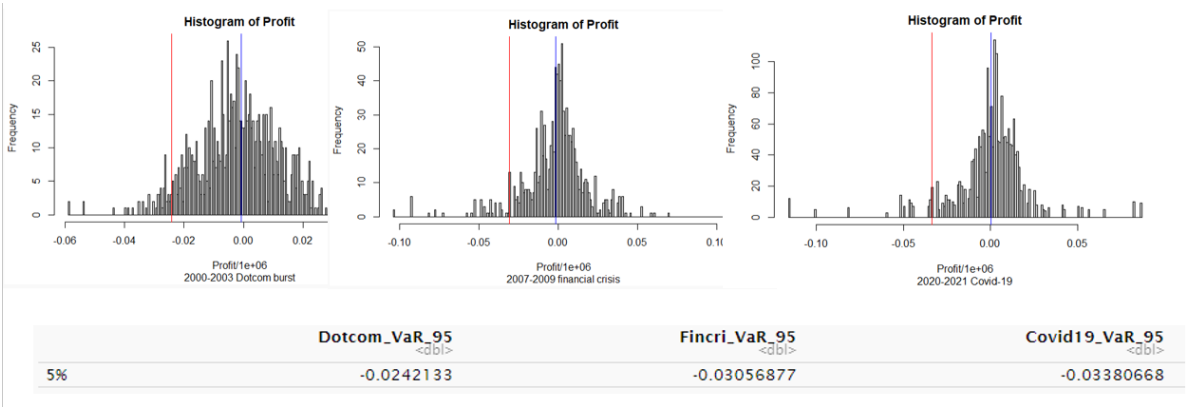
Upon applying the weights to the portfolio, we visualized that the optimally weighted portfolio performs better than an equally weighted portfolio after optimization. So we decided to work with an optimally weighted portfolio.

	BETA <chr>	COR <chr>	ALPHA <chr>	Pvalue_ALPHA <chr>	Pvalue_BETA <chr>	BM <chr>	RISK <chr>
RISK4	0.4523	0.695	0.007	0	0	VDE	Vanguard Energy Sector
RISK2	0.0152	0.0487	0.0069	0.01	0.01	FEDFUNDS	Fed Funds Rate
RISK5	0.65	0.7272	0.0039	0.04	0	VNQ	Vanguard Real Estate Sector Risk
RISK11	0.5979	0.5806	0.0036	0.11	0	VPU	Vanguard Utility Sector Risk
RISK10	0.7292	0.8544	0.0034	0.02	0	VAW	Vanguard material Sector Risk
RISK7	0.6752	0.8319	0.0031	0.04	0	VFH	Vanguard financial Sector Risk
RISK12	0.8428	0.8287	0.003	0.05	0	VOX	Vanguard Communication services Sector Risk
RISK8	0.8028	0.8901	0.002	0.12	0	VIS	Vanguard industrail Sector Risk
RISK14	0.9984	0.7692	0.0013	0.46	0	VDC	Vanguard Consumer Staple Sector Risk
RISK1	1.0242	0.9742	0.001	0.1	0	^GSPC	Market RISK
RISK6	0.9181	0.8753	-2e-04	0.89	0	VHT	Vanguard Health care Sector Risk
RISK9	0.8468	0.9642	-0.0013	0.08	0	VGT	Vanguard Informational technology Sector Risk
RISK3	0.8603	0.9569	-0.0012	0.15	0	XLK	Technology Sector
RISK13	0.9405	0.9627	-0.0011	0.15	0	VCR	Vanguard Consumer Discretionary Sector Risk

We assumed that the market improves by 15% and has a risk-free rate of 1.75%. We shocked our portfolio using the Capital asset pricing model with the help of the parameters from the linear model to determine the expected return from each of our indexes and other factors towards our portfolio. Assets with low beta and correlation showed a low growth rate between 2% - 7%, such as Fed Funds rate, energy sector, and utility sector. Whereas assets with solid beta and correlation showed that if the market improves by 15% that the portfolio returns can increase by 11.5% - 15.0%

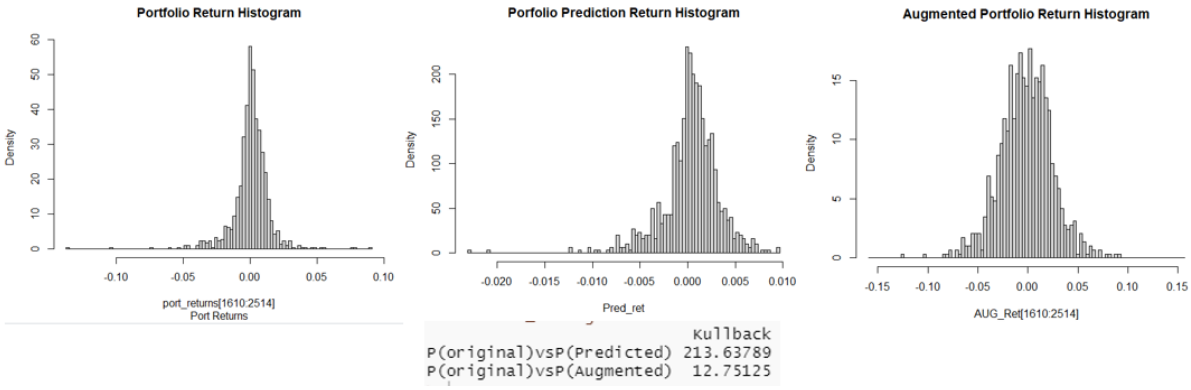
Similarly, after computing the expected return of 0.000337663 from the LSTM model and the expected return of -0.0032627 from augmented models. We shocked our portfolio again using CAPM to evaluate the impact and differences of both models. Expect for the last four assets, which showed robust evaluation, the remaining assets for both the models had similar growth rates with a minimal difference.

Comparing the last four assets for both the models, we determined a difference of .0030 or less between the models. It can be said that the LSTM model performed better; however, there wasn't a noticeable difference between both models.



	Dotcom_VaR_95 <dbl>	Fincri_VaR_95 <dbl>	Covid19_VaR_95 <dbl>
5%	-0.0242133	-0.03056877	-0.03380668

We determine the value at risk factor for different recession periods using Monte Carlo simulation in the next part. The more times we ran the simulation, the greater was the risk. We did this analysis to determine how much value our portfolio would lose if a recession were to occur. We will face the most loss during the Covid-19 crisis. This tells us that the composition of our portfolio was acceptable to the Covid-19 crisis. The discretion during this period can also be considered as a negative factor.




```
                          Kullback
P(original)vsP(Predicted) 213.63789
P(original)vsP(Augmented)  12.75125
```

Finally, we have three plots representing the probability distribution of our portfolio returns, predicted returns, and augmented data returns. We performed Kullback Lieber divergence to measure the distance of our models to the original data. As per the KL divergence method, the closer the value is to 0, the better the new data matches our actual data. KL-divergence is very high for both models, even though the distance between augmented and original data is comparatively lower.

5. CONCLUSIONS

- The factor trading strategies performed better the S&P 500 index.
- Factor trading strategy developed using augmented historical data is at least as good as the factor trading strategy developed using actual historical data.
- Augmenting financial time series data captures the volatility inherent in the process that a sample may not fully capture.
- Additive Gaussian Data Augmentation reduces over fitting significantly if used to train a model.
- The LTSM model is better suited than the time GAN model for generating financial time series data scenarios.
- Using the LTSM model on augmented data produces lower absolute mean errors owing to reduction in over fitting.
- From our risk analytics, the trading strategy and the two scenario generation methods, additive Gaussian data augmentation and the LTSM model are effective in all market conditions and types.

7. REFERENCES

1. Mammen, E. and Nandi, S. (2012). “Bootstrap and Resampling.”
2. Liu Ziyin[1], Kentaro Minami[2], Kentaro Imajo[2] . *Department of Physics, University of Tokyo. Preferred Networks, Inc.*
3. T. Dao, A. Gu, A. Ratner, V. Smith, C. DeSa, and C. Ré A kernel theory of modern data augmentation. In *International Conference on Machine Learning*, pages 1528-1537. PMLR, 2019.
4. C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. Journal of Big Data, 6(1):1{48, 2019
5. Liu Ziyin[1], Kentaro Minami[2], Kentaro Imajo[2] . *Department of Physics, University of Tokyo. Preferred Networks, Inc.*
6. Noora Shrestha, “Factor Analysis as a Tool for Survey Analysis.” *American Journal of Applied Mathematics and Statistics*, vol. 9, 1 (2021): doi: 10.12691/ajams-9-1-2.
7. Tabachnick, B.G. and Fidell, L.S., *Using multivariate statistics (6th ed.)*, Pearson, 2013.
8. Guttman, L., “Some necessary conditions for common-factor analysis,” *Psychometrika*, 19, 149-161. 1954.
9. Pallant, J., *SPSS survival manual: a step by step guide to data analysis using SPSS,* Open University Press/ Mc Graw-Hill, Maidenhead, 2010.
10. Hair, J.F., Anderson, R.E., Tatham, R.L., and Black, W.C., *Multivariate data analysis (5th ed.)*, N J: Prentice-Hall, Upper Saddle River, 1998.
11. Saxena, D. and Cao, J. (2020). “Generative Adversarial Networks (GANs): Challenges, Solutions, and Future Directions.”
12. Mammen, E. and Nandi, S. (2012). “Bootstrap and Resampling.”
13. Kritzman, M. and Turkington, D. (2021). “History, Shocks and Drifts: A New Approach to Portfolio Formation.”
14. MartínAbadi,PaulBarham,JianminChen,ZhifengChen,AndyDavis,Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In 12th {USENIX} symposium on operating systems design and implementation ({OSDI } 16). 265–283.
15. Felipe Barboza Oriani and Guilherme P Coelho. 2016. Evaluating the impact of technical indicators on stock forecasting. In 2016 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE
16. Xingyu Zhou, Zhisong Pan, Guyu Hu, Siqi Tang, and Cheng Zhao. 2018. Stock market prediction on high-frequency data using generative adversarial nets.

APPENDIX I

Theory Proofs

The true utility of the additive Gaussian data augmentation is

$$U_{Add} = \frac{r^2}{2\lambda\sigma^2 T}\mathbb{E}_{S_t}\left[\frac{(\sum_t r_t S_t)^2}{\sum_t (r_t S_t)^2}\Theta(\sum_t r_t S_t^2)\right]$$

Where Θ is the Heaviside step function.

Proof

For a time-dependent strategy π_t^*, the true utility is defined as

$$U(\pi^*) = \mathbb{E}_{S_0', S_1', \cdots, S_T', S_{T+1}'}\left[\frac{1}{T}\sum_{t=1}^{T+1}\pi_t^* r_t' - \left(\frac{\lambda}{2T}\sum_{t=1}^{T}(\pi_t^* r_t')^2 - \mathbb{E}_{S_0', S_1', \cdots, S_T', S_{T+1}'}[\pi_t^* r_t']^2\right)\right]$$

Where

$$S_1', \cdots, S_T', S_{T+1}'$$

is an independently sampled distribution for testing, and

$$r_1' := \frac{S_{t+1} - S_t}{S_t}$$

are their respective returns. Now, we note that we can write the price update equation (the GBM model) in terms of the returns:

$$S_{t+1} = (1 + r)S_t + \sigma_t S_t \eta_t \rightarrow r_t = r + \sigma\eta_t$$

Which means that

$$r_t \sim \mathcal{N}(r, \sigma^2)$$

obeys a Gaussian distribution. Therefore,

$$U(\pi^*) = \frac{r}{T}\sum_{t=1}^{T}\pi_t^* - \frac{\lambda\sigma^2}{2T}\sum_{t=1}^{T}(\pi_t^*)^2$$

We averaging over π_t^*, because we also want to average over the training set to make the true utility independent of the sampling of the training set. The strategy is defined as:

$$\pi_t^* = \begin{cases} 1, & if\ r_t \geq 0 \\ -1, & if\ r_t < 0 \end{cases} = \Theta(r_t \geq 1) - \Theta(r_t < 1)$$

For a training set $\{S_0, \cdots, S_T\}$, and Θ is the Heaviside step function. We thus have that

$$\begin{cases} (\pi_t^*)^2 = 1; \\ \mathbb{E}_{S_1,\cdots,S_{T+1}}[\pi_t^*] = \mathbb{E}_{S_1,\cdots,S_{T+1}}[\Theta(r_t \geq 0) - \Theta(r_t < 0)] = 1 - 2\Phi\left(-\frac{r}{\sigma}\right) \end{cases}$$

Where Φ is the Gauss c.d.f. We can use this to average the utility over the training set. Noticing that the training set and the test set are independent, we can obtain

$$U = \mathbb{E}_{S_1,\cdots,S_{T+1}}[\pi^*]$$

$$= \frac{1}{T}\sum_{t=1}^{T}\left[1 - 2\Phi\left(-\frac{r}{\sigma}\right)\right]r - \frac{\lambda}{2T}\sum_{t=1}^{T}\sigma^2$$

$$= \left[1 - 2\Phi\left(-\frac{r}{\sigma}\right)\right]r - \frac{\lambda}{2}\sigma^2$$

This finishes the proof.

Proof for Additive Gaussian Noise

Lemma 1. The maximum of the utility function with additive Gaussian noise is:

$$\pi_t^*(\rho) = \begin{cases} \frac{r_t S_t^2}{2\lambda\rho^2}, & if\ -1 < \frac{r_t S_t^2}{2\lambda\rho^2} < 1 \\ sgn(r_t), & otherwise \end{cases}$$

Proof. With additive Gaussian noise, we have

$$\begin{cases} \mathbb{E}_t[G_t(\pi)] = \pi_t \mathbb{E}_t[\tilde{r}_t] = \pi_t \mathbb{E}_t\left[\frac{S_{t+1} + \rho_t\epsilon_{t+1} - S_t - \rho_t\epsilon_t}{S_t}\right] = \pi_t \frac{S_{t+1} - S_t}{S_t} = \pi_t r_t; \\ Var_t[G_t(\pi)] = \pi_t^2 Var_t[\tilde{r}_t] = \pi_t^2 Var_t\left[\frac{\rho_t\epsilon_{t+1} - \rho_t\epsilon_t}{S_t}\right] = \frac{2\rho^2\pi_t^2}{S_t^2}, \end{cases}$$

Where the last line follows from the definition for the additive Gaussian noise that

$$\rho_1 = \cdots \rho_T = \rho$$

The training objective becomes

$$arg\max_{\pi_t}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_t[G_t(\pi)] - \frac{\lambda}{2}Var_t[G_t(\pi)]\right)$$

$$= arg \max_{\pi_t} \left(\frac{1}{T} \sum_{t=1}^{T} \pi_t r_t - \lambda \frac{\rho^2 \pi_t^2}{S_t^2} \right)$$

This can be maximised for every t by differentiation:

$$\frac{\partial}{\partial \pi_t} \left(\pi_t r_t - \lambda \frac{\rho^2 \pi_t^2}{S_t^2} \right) = 0$$

$$r_t - \frac{2\lambda \rho^2 \pi_t}{S_t^2} = 0$$

$$\therefore \pi_t^*(\rho) = \frac{r_t S_t^2}{2\lambda \rho^2}$$

By definition, we also have $|\pi_t| \leq 1$, and so

$$\pi_t^*(\rho) = \begin{cases} \frac{r_t S_t^2}{2\lambda \rho^2}, & if -1 < \frac{r_t S_t^2}{2\lambda \rho^2} < 1 \\ sgn(r_t), & otherwise \end{cases}$$

Which is the desired result.

Investors are risk averse so $|\pi_t| \leq 1$. If $\pi_t = 1$it means that the investor invests all his money in the financial market, since investors are risk averse, it is very unlikely. Mathematically, this means that it is often the case that $\lambda \geq \frac{|r_t| S_t^2}{2\lambda \rho^2}$

Proposition 5 For additive Gaussian noise strategy, the true utility is

$$U_{Add} = \frac{r^2}{2\lambda\sigma^2 T}\mathbb{E}_{S_t}\left[\frac{(\sum_t r_t S_t)^2}{\sum_t (r_t S_t)^2}\Theta(\sum_t r_t S_t^2)\right]$$

Proof

For a time-dependent strategy π_t^*, the true utility is defined as

$$U(\pi^*) = \mathbb{E}_{S_0', S_1', \cdots, S_T', S_{T+1}'}\left[\frac{1}{T}\sum_{t=1}^{T+1}\pi_t^* r_t' - \left(\frac{\lambda}{2T}\sum_{t=1}^{T}(\pi_t^* r_t')^2 - \mathbb{E}_{S_0', S_1', \cdots, S_T', S_{T+1}'}[\pi_t^* r_t']^2\right)\right]$$

Where

$$S_1', \cdots, S_T', S_{T+1}'$$

is an independently sampled distribution for testing, and

$$r_1' := \frac{S_{t+1} - S_t}{S_t}$$

are their respective returns. Now, we note that we can write the price update equation (the GBM model) in terms of the returns:

$$S_{t+1} = (1 + r)S_t + \sigma_t S_t \eta_t \rightarrow r_t = r + \sigma\eta_t$$

Which means that

$$r_t \sim \mathcal{N}(r, \sigma^2)$$

obeys a Gaussian distribution. Therefore,

$$U(\pi^*) = \frac{r}{T}\sum_{t=1}^{T}\pi_t^* - \frac{\lambda\sigma^2}{2T}\sum_{t=1}^{T}(\pi_t^*)^2$$

Plugging in the above Lemma, we have:

$$U(\pi^*) = \frac{r}{T}\sum_{t=1}^{T}\frac{r_t S_t^2}{2\lambda\rho^2} - \frac{\lambda\sigma^2}{2T}\sum_{t=1}^{T}\left(\frac{r_t S_t^2}{2\lambda\rho^2}\right)^2$$

This utility is a function of the data augmentation strength ρ. For a fixed training set, we would like to find the best ρ that maximizes the above utility.

$$(\rho^*)^2 = \begin{cases} \frac{\lambda\sigma^2}{2r}\frac{\sum_{t=1}^{T}(r_t S_t^2)^2}{\sum_{t=1}^{T} r_t S_t^2}, & if \sum_{t=1}^{T} r_t S_t^2 > 0 \\ \infty, & otherwise. \end{cases}$$

Plug in the lemma, we have

$$U_{Add} = \frac{r_t S_t^2}{2\lambda(\rho^*)^2 T} = \frac{r r_t S_t^2}{\sigma^2} \frac{\sum_t r_t S_t}{\sum_t (r_t S_t^2)^2} \Theta(\sum_t r_t S_t^2)$$

One thing to notice is that the optimal strength is independent of λ, which is an arbitrary value and dependent only on the investor's psychology. Plug into the utility function and take expectation with respect to the training set, we obtain

$$U_{Add} = \mathbb{E}_{S_1, \cdots, S_{T+1}} \left[U\left(\pi^*(\rho^*)\right) \right] \left[\frac{(\sum_t r_t S_t)^2}{\sum_t (r_t S_t)^2} \Theta(\sum_t r_t S_t^2) \right]$$

$$= U(\pi^*) = \frac{r}{T} \sum_{t=1}^{T} \pi_t^*(\rho^*) - \frac{\lambda \sigma^2}{2T} \sum_{t=1}^{T} [\pi_t^*(\rho^*)]^2$$

$$= \frac{r^2}{2\lambda \sigma^2 T} \mathbb{E}_{S_1, \cdots, S_{T+1}} \left[\frac{(\sum_t r_t S_t)^2}{\sum_t (r_t S_t)^2} \Theta(\sum_t r_t S_t^2) \right]$$

This finishes the proof.