[AICP 2023] Final Report

❒ Team Information

Research Topic	EN	Portfolio Strategy Development using Neural Networks			
	KR	신경망 모형을 활용한 투자 포트폴리오 전략 개발			
Team Name		**UNSIT Dong-Hak Ant 🐜**			
Professor				Dept.	
TA		Student No.	Name	Dept.	
Team Members		Student No.	Name	Intern-ship*	Dept.
	1	20236002	Namwoo Kwon		GSIM
	2	20211375	Daeun Oh		SBA
	3	20181369	Hyeonwook Oh		SBA
	4	20191055	Yejin Kim		IE
	5	20191327	Chanbeom Hur		IE
	6				

***[Undergrads Only] Research Internship**: 2023년 AICP 활동내역으로 연구인턴십 학점을 신청한 학부생들은 **본인의 신청학점(1~3학점)을 표기**해주시기 바랍니다. / If you have applied for research internship credits through the AICP activities, please **leave how many credits (1-3 credits) you have applied for.**

❐ Abstract

Keywords	Momentum Crash, Leverage, Anomaly Detection, Neural Network

The market error (a.k.a. Market Anomaly) that creates excess returns by buying financial products with good past returns (winners) and selling financial products with poor past returns (losers) is called the momentum effect. This effect has been seen consistently since the distant past, which is why many investors utilize momentum investing or momentum portfolios. However, sometimes the opposite happens, where a financial instrument with a poor past performance outperforms a financial instrument with a good past performance. This is called a Momentum Crash, and it can cause huge losses for Momentum investors.

We use a neural network (or Machine Learning) model, inheriting from previous research on predicting momentum crashes with a statistical methodology. We perform anomaly detection by considering Momentum Effect as a normal situation and defining Momentum Crash as an anomaly. If the model predicts an anomaly, it manages the Momentum Crash Risk by assigning low leverage to the Momentum Portfolio. And to compare with the existing Momentum Portfolio, we utilize the Sharpe Ratio metric, which shows the return to volatility ratio. And to compare with the existing Momentum Portfolio, we use the Sharpe Ratio, which represents the return on investment relative to the volatility.

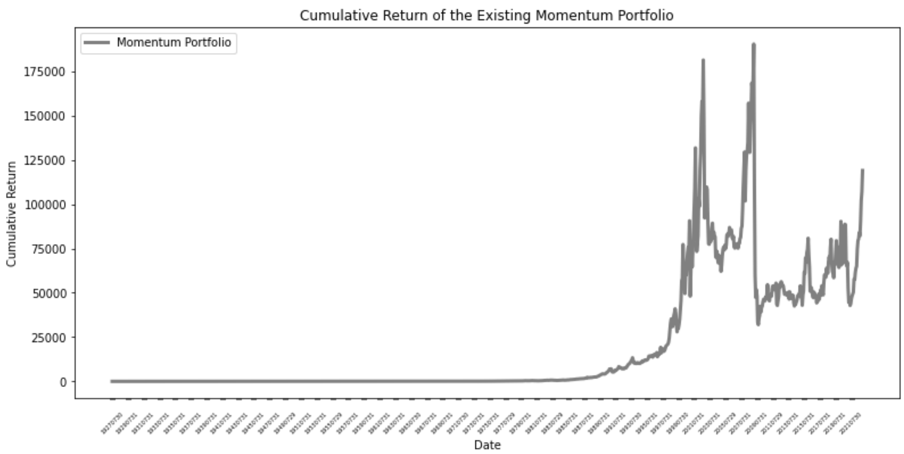
Our results show that our methodology outperforms the Sharpe Ratio of the original Momentum Portfolio due to our variable selection, model usage, and utilization of the Leverage Function. Our methodology effectively captures periods of high volatility in the returns of the Momentum Portfolio, driven by the Momentum Crash. We also found that applying the Isolation Forest model to Momentum Volatility and Winner related variables resulted in the highest Sharpe Ratio and higher cumulative return than the cumulative return of the existing Momentum Portfolio.

The methodology suggests that many improvements can be made in the selection of variables and models and the development of the Leverage Function. Our study only used momentum-related variables, so there may be important variables other than momentum, and models improved from overfitting problems and the use of new Leverage Functions may be effective in increasing the Sharpe Ratio.

❒ Research Contents

1. Introduction

Financial products with good past returns are defined as winners, and financial products with poor past returns are defined as losers. A portfolio that buys winners and sells losers during a specific period is called a Momentum Portfolio (or WML Portfolio). This is a strategy to gain excess profits by betting that winners will have higher future returns than losers. This strategy is based on the Momentum Effect, which consistently shows a positive value when the loser’s return is subtracted from the winner's return. However, sometimes, contrary to the Momentum Effect, a Momentum Crash occurs where a Loser reverses the Winner's returns. This can cause large losses for momentum investors who expect the Momentum Effect. The graph below shows that the momentum portfolio value of $1 from July 1927 to December 2022 gradually increased due to the momentum effect but decreased sharply after the momentum crash.



Previous studies have applied a stop-trade rule in which investors stop momentum trading during periods of extremely low losses over the past year (i.e., when momentum returns tend to be low and volatility tends to be high during this period). As a result, it was found to achieve a 40% higher rate of return (11.63%~16.18% per year) than the existing momentum and lower the maximum drawdown to -17.17%. We aim to increase the Sharpe Ratio over the existing momentum portfolio through a neural network model rather than a statistical methodology. Neural network models have the following advantages: 1) It has high prediction performance for high-dimensional and non-linear data. 2) It is effective in identifying time series patterns in the data. 3) It shows higher performance than statistical methodology for specific tasks.

We aim to develop a new momentum strategy that delivers higher returns and lower risk compared to existing momentum portfolios through artificial intelligence. The methodology we studied is an application of Dynamic Strategy, which trains various variables into a neural network model and assigns leverage to the momentum portfolio when making predictions based on the results of the validation data set. Through a new momentum portfolio analysis, this study is expected to derive economic channels for market errors that cause the Momentum Effect and develop possible tools that can be used to improve the profitability of various portfolio strategies other than momentum strategies.

2. Research Methods

2-1. Dataset

To build the dataset, we use monthly U.S. stock data from the Center for Research in Security Prices (CRSP). Our study sample utilizes all common stocks on the NYSE, AMEX, and NASDAQ (CRSP exchange codes 1, 2, and 3) from July 1927 to December 2022. And to build a momentum portfolio at time t, we track each company's stock returns from t-2 to t-12 (past 11 months). Based on the tracked returns, the top 10% (Winners) are purchased and the bottom 10% (Losers) are sold to obtain the Momentum Portfolio's monthly returns.

Based on variables that have been shown to be useful in predicting future momentum returns in previous research, 17 variables related to past momentum volatility and returns of winners and losers are used to learn the model. And we performed various feature selections to identify variables that are important in predicting Momentum Crash among the 17 variables. Momentum Crash is defined as when the momentum portfolio return at that time is negative.

Variables used in Model Learning & Inference (Features & Label)

winner_lag1	value weighted (VW) of winner’s return with lag 1
loser_lag1	value weighted (VW) of loser’s return with lag 1
wml_lag1	winner minus loser portfolio's return with lag 1
winvol_cum6	winner's cumulative 6 months momentum volatility

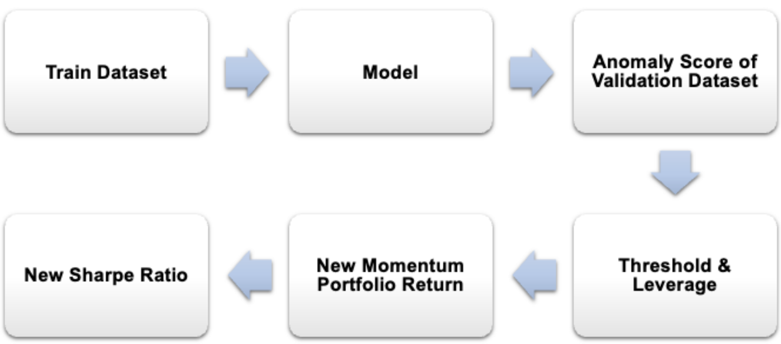
losvol_cum6	loser's cumulative 6 months momentum volatility
mvol_t_1	momentum volatility over past 1 month
mvol_t_2	momentum volatility over past 2 months
mvol_t_3	momentum volatility over past 3 months
mvol_t_4	momentum volatility over past 4 months
mvol_t_5	momentum volatility over past 5 months
mvol_t_6	momentum volatility over past 6 months
cum_winner_t_2_4	VW average of cumulative returns on winners over (t-2, t-4)
cum_winner_t_5_8	VW average of cumulative returns on winners over (t-5, t-8)
cum_winner_t_9_12	VW average of cumulative returns on winners over (t-9, t-12)
cum_loser_t_2_4	VW average of cumulative returns on losers over (t-2, t-4)
cum_loser_t_5_8	VW average of cumulative returns on losers over (t-5, t-8)
cum_loser_t_9_12	VW average of cumulative returns on losers over (t-9, t-12)
pos_wml (0 or 1)	Indicator which is 0 if return is positive, otherwise 0

Set the number of training datasets, validation datasets, and test datasets to 240 (20 years), 60 (5 years), and 1 (1 month), respectively. Then, through the Rolling Window algorithm, a window of a total size of 301 is used to move to the next month (one space) after model inference. As a result, the model presents a new Momentum Portfolio for each month from July 1952 to December 2022.

2-2. Dynamic Strategy

A dynamic strategy is an investment strategy that frequently rebalances its portfolio based on macro financial trends. We propose a strategy of underweighting the momentum portfolio if we believe a momentum crash will occur at that time. We use the following process to manage momentum crash risk with a dynamic strategy.

Dynamic Strategy Process



2-2-1. Feature & Model Selection

We selected 17 variables to use based on prior research. Due to the nature of financial data, a small number of datasets and many variables increases the likelihood of model overfitting due to the dimensionality curse. Therefore, we reduced the number of variables used to train the model by using backward elimination after using all the variables. This suggests which variables are important for the model to predict Momentum Crash, and we expect that it will help us to analyze Momentum Crash through the variables we used. We used many machine learning models because our research goal is to use neural network models, but they have the disadvantage of being easily overfitted. By properly matching the complexity of the data and the complexity of the model, we expect to show good performance in predicting Momentum Crash.

The main mechanism of the model we used is anomaly detection. Anomaly detection is the identification of data that is likely to be problematic for a particular task, and data with patterns that are not commonly encountered. Different anomaly detection models have their own strengths and weaknesses, meaning that Momentum Crashes defined as anomalies can have a variety of anomalous patterns, so the patterns that each model identifies well may differ. By applying different models, we hope to capture common and different patterns in Momentum Crashes across models.

2-2-1-1. PCA

Anomaly Detection using PCA is an algorithm that utilizes reconstruction error. By applying PCA to the original data, dimensionality is reduced, and the compressed data is restored to the same dimensions as the original data. And the reconstruction error is calculated through the difference between the original data and the reconstructed data. This assumes that anomalies will not be restored well and defines an anomaly as something that is not well restored and has a large reconstruction error.

PCA is a technique that finds a new axis that preserves the variance of the original data as much as possible and projects the data onto that axis. The process is as follows: 1) Find the axis with maximum variance in the original data. 2) Find the second axis that is orthogonal to the first axis found in this way and has the maximum variance. 3) Find a third axis that is orthogonal to the first and second axes and preserves variance as much as possible. 4) Find the original dataset with a certain number of dimensions using the same method as steps 1 to 3. 5) Inversely transform the dimensionally reduced data and restore it again.

2-2-1-2. LOF

LOF defines the anomaly score as how much each data deviates from specific clusters. In other words, the Anomaly Score indicates how isolated an entity is from its surrounding neighbors. To calculate this, the density of neighbors around the data site and the density ratio of the neighbors are calculated. If an data is anomaly, the density will be low because there are few neighbors around it, and the density of surrounding neighbors will also be low.

2-2-1-3. Isolation Forest

Isolation Forest is an application of Tree-based Machine Learning. This assumes that anomaly is easier than normal to isolate each observation using features. Anomaly Score defines the degree of separation of each observation using the number of splits in the tree (depth of the tree). Therefore, Anomaly tends to be located at a higher level in the tree structure, while Normal tends to be located at a lower level in the tree structure.

2-2-1-4. Auto Encoder

Auto Encoder, like PCA, is an algorithm that utilizes reconstruction error. The difference is that Auto Encoder is a neural network structure composed of an encoder that compresses input data into low dimensions and a decoder that restores compressed data. And Auto Encoder uses a non-linear function rather than a linear transformation like PCA, so it is advantageous for identifying non-linear patterns in data.

2-2-1-5. VAE

VAE has an encoder and decoder like Auto Encoder, but it additionally introduces Latent Variable. The encoder is responsible for receiving input data and generating parameters of a probability distribution. Probability distributions represent the representation of data in latent variable space. It mainly uses probability distributions such as normal distribution or multivariate normal distribution. Then, the latent variable (z) is sampled through the probability distribution generated by the encoder. The decoder network takes a latent variable (z) as input and uses the latent variable to restore the probability distribution of the original data. Therefore, VAE aims to minimize Kullback-Leibler divergence and reconstruction error, which represents the difference between the probability distribution and prior distribution of latent variables in model learning.

2-2-2. Model Training

It uses both Supervised Learning and Semi-Supervised Learning, which are the main mechanisms for training models. Supervised Leering informs the model whether there is a Momentum Crash when training the model, so it is expected that it will learn well what the Momentum Crash pattern is within the training data. However, when learning a Momentum Crash pattern that did not appear in the past, an error may occur that cannot be predicted as a Momentum Crash. Semi-Supervised Learning may have difficulty learning Momentum Crash patterns when learning a model, but it is expected that there will be flexibility in inferring Momentum Crash patterns that have not yet appeared in the future.

2-2-3. Inference Validation Dataset

The Threshold and Leverage Function to be applied to the test dataset are defined according to the process based on the Validation Dataset.

2-2-3-1. Define Threshold

Anomaly scores inferred for 60 validation datasets through the learned model were scaled using one of Standard Scaling, Min Max Scaling, and Robust Scaling. This is to compare the results by applying data of a similar scale to the leverage function to be applied later, as the Anomaly Score Scale is different for each model.

To define the threshold that identifies anomalies in the test dataset, the anomaly score of the scaled validation dataset is divided into a normal group and an anomaly group based on the label. There are three main processes for defining threshold. First, set the average of the maximum anomaly score of the normal group and the minimum value of the anomaly group to Threshold 1, and calculate the f1-score of the validation data set based on Threshold 1. Second, set the median of the anomaly score of the normal group and the average of the median of the anomaly group to Threshold 2, and calculate the f1-score of the validation data set based on Threshold 2. The reason for this process is to respond to situations in which the model made good inferences and situations in which it did not. Lastly, of the two thresholds, the one with the highest f1-score is used as the final threshold. If the f1-scores of two thresholds are the same, the lowest value of the two thresholds is used as the final threshold. This is because by using a low threshold, we are trying to conservatively infer whether the test dataset will experience a Momentum Crash.

2-2-3-1. Define Leverage Function

Due to the nature of financial data, even if the model shows good performance in the learning and validation datasets, it is difficult to expect it to show the same performance in the test dataset. Therefore, we propose the 'Certain' parameter, which indicates the expectation that the model will show the same performance on the test dataset as before, based on the number of validation datasets and their performance indicators. This transforms the leverage function to be applied to the test dataset differently each time based on the number of validation datasets and the performance of the validation dataset.

Certain Parameter

$$C_t = \frac{VF}{\sqrt{F}}$$

(C_t: Certain at time t, V: Validation dataset size, F_t: F1-score at time t)

Let’s assume two situations. First, the f1-score of the high validation dataset is given the same, and the number of validation datasets is different. If the number of validation datasets is large, it is expected that the test dataset will also show high prediction performance. Therefore, even if the anomaly score of the test dataset is close to the threshold, it will definitely give a leverage of 0 or 1. Conversely, if the number of validation datasets is small, even if the f1-score of the validation dataset is high, if the anomaly score of the test dataset is close to the threshold, rather than giving leverage of 0 or 1, it is slightly higher than 0 or slightly lower than 1. It is expected that providing leverage will be good in case of incorrect predictions.

Second, the number of sufficient validation datasets is the same, and the f1-scores of the validation datasets are different. If a sufficient validation dataset shows a high f1-score, it would be a good idea to clearly provide leverage of 0 or 1 even if the anomaly score of the test dataset is close to the threshold. Conversely, if a sufficient validation dataset shows a low f1-score, we would like to grant a leverage slightly higher than 0 or slightly lower than 1, rather than granting a leverage of 0 or 1 when the anomaly score of the test dataset is close to the threshold.

The leverage function is a modified version of the sigmoid function. This is because the sigmoid function converges to 0 or 1 at the extremes, so the leverage value can be set between 0 and 1, and since it has a non-linear structure, it is expected to be flexibly applied to each test dataset. Additionally, comparisons were made using the leverage function that gives a value of 0 or 1 based on the threshold. The leverage function, which is a modified version of the sigmoid function, has different function structures based on what is below and above the threshold.

The reason is that it assumes the following: If the model is unable to distinguish between subtle momentum effects and subtle momentum crashes, it is assumed that the anomaly score will be smaller than the set threshold. And if it is difficult to distinguish between a very large momentum effect and a very large momentum crash, it is assumed that the anomaly score will be greater than the set threshold. Therefore, in the section smaller than the threshold, a slight momentum crash is considered and leverage close to 1 is granted by acting less sensitive to the 'certain' of multiple momentum effects. Conversely, in sections larger than the threshold, leverage is provided fluidly by being very sensitive to 'certain' so that a certain amount of leverage can be given to the momentum effect while avoiding a very large momentum crash.

Leverage Function 1

Based on the set threshold, it clearly determines whether anomaly is present and outputs a leverage of 0 or 1.

$$lev_t = \begin{cases} 1 & x_t < T \\ 0 & x \geq T \end{cases}$$

(lev_t: Leverage at time t, x_t: anomaly score at time t, T_t: Threshold at time t)

Leverage Function 2

For everything below threshold - 1, a leverage of 1 is output. In addition, the section is divided into two based on the threshold and a different leverage function is applied. The difference between the two leverage functions is the degree to which they are sensitive to F1-Score. In particular, the function applied in the section above the threshold outputs relatively high leverage even if the anomaly score is high when the F1-Score is low.

$$lev_t = \begin{cases} 1 & x < T - 1 \\ \frac{1}{1+(x_t - T_t + 1)^{C_t}} & T - 1 \leq x < T \\ \frac{1}{1+(x_t - T_t + 1)^{F_t C_t}} & x \geq T \end{cases}$$

lev_t: Leverage at time t, x_t: Anomaly score at time t, T_t: Threshold at time t, F_t: F1-score at time t, C_t: Certain at time t

Leverage Function 3

Like leverage function 2, the function applied to the section above the threshold outputs relatively high leverage even if the outlier score is high when the F1-Score is low. However, the difference is that like Leverage Function 2, it does not give a leverage of 0.5 to all anomaly scores that match the threshold regardless of f1-score. Leverage Function 3 outputs relatively lower leverage to anomaly scores close to the threshold when the f1-score is low.

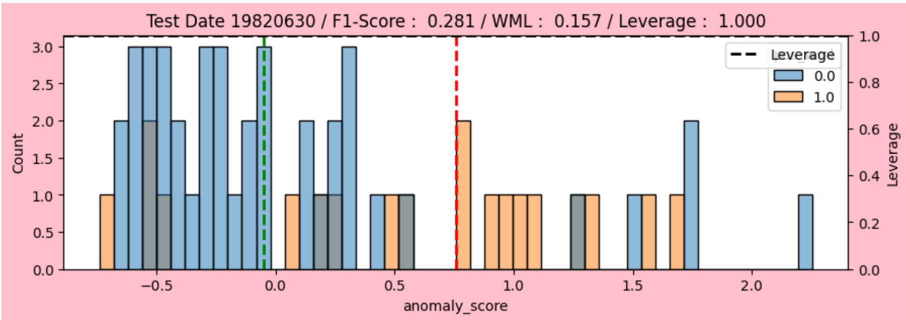
$$lev_t = \begin{cases} \frac{1}{1+(Fe^{C(x_t - t)})} & x < T \\ \frac{1}{(1+\sqrt{F})+(\sqrt{F}\,(x-t+1)^{\sqrt{F}\,\sqrt{C}})} & x \geq T \end{cases}$$

(lev_t: Leverage at time t, x_t: Anomaly score at time t, T_t: Threshold at time t, F_t: F1-score at time t, C_t: Certain at time t)

In conclusion, based on the threshold, two different leverage functions were structured differently using 'certain' and the f1-score of the validation dataset as parameters. The threshold and leverage functions defined according to the inference results of the validation dataset have final leverage determined by the anomaly score of the test dataset. And the portfolio return of the test dataset is multiplied by leverage to present a new momentum portfolio return.

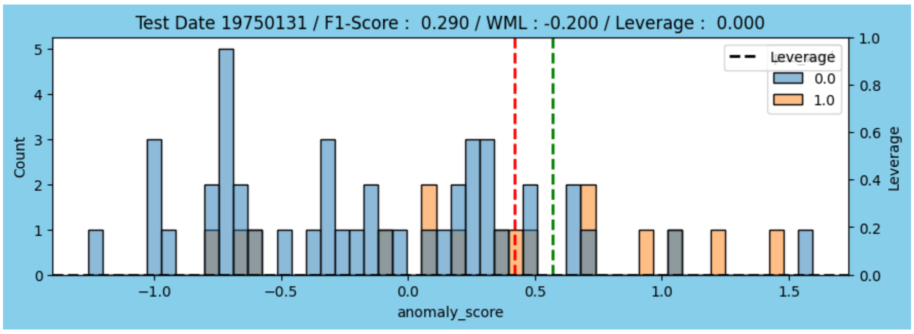
Example of leverage output using new methodology)

Situation 1) Very Large Momentum Effect



(Threshold: solid red line, Anomaly score of test dataset: solid green line, Leverage: solid black line)

Situation 2) Very Large Momentum Crash

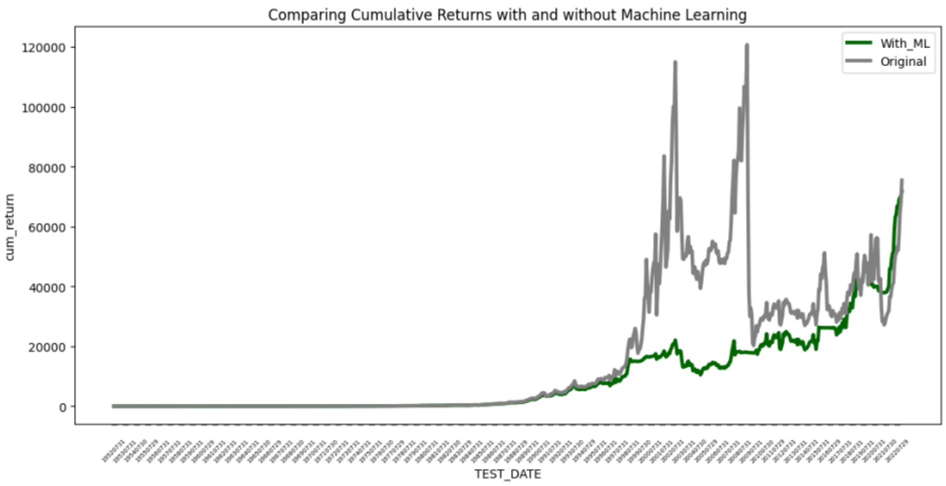


(Threshold: solid red line, Anomaly score of test dataset: solid green line, Leverage: solid black line)

2-2-5. Evaluation

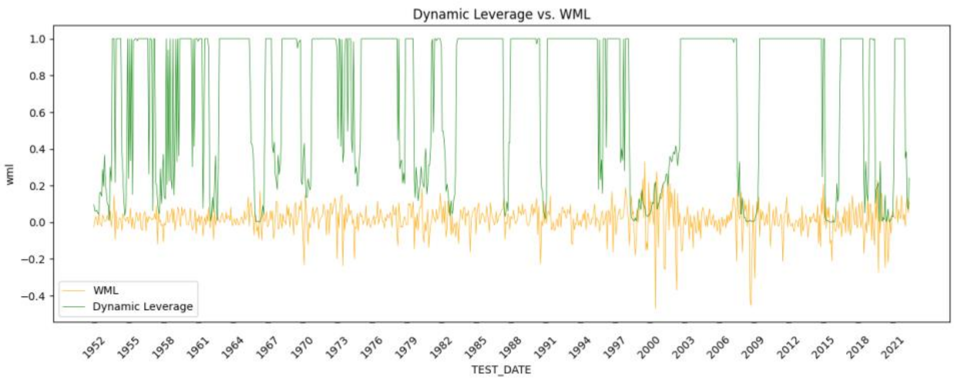
To check how effective our methodology is, we proceed with the following process. Based on the new momentum portfolio returns, we calculate the Sharpe Ratio and cumulative returns during the test dataset period and compare them with the existing momentum portfolio. Compare the leverage applied to each test dataset and the portfolio return at that point in time to check how liquid and low leverage was given in the event of a Momentum Crash. And check the f1-score of the validation dataset to check whether the variables and model used well predicted the momentum crash. Based on this, we improve Feature & Model Selection and perform hyperparameter tuning. Hyperparameter tuning includes improving the threshold setting process and leverage function architecture in addition to the parameters inherent in the model.

Example of Cumulative Return of Existing Momentum Portfolio and Cumulative Return of New Momentum Portfolio



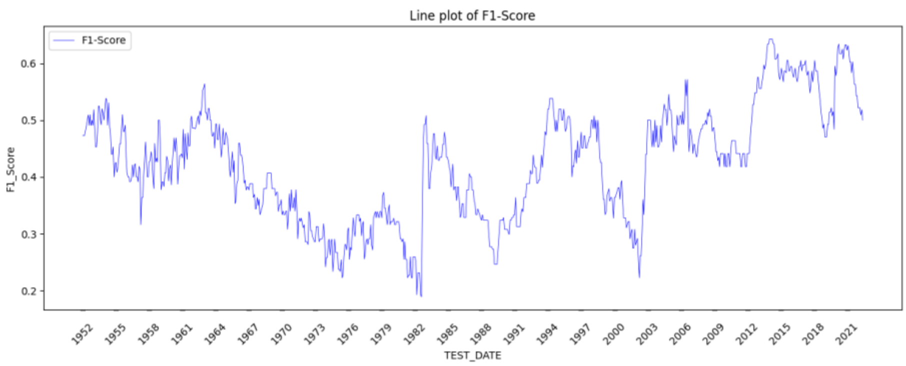
(Cumulative return of existing momentum portfolio: gray line, Cumulative return of new momentum portfolio: green line)

Example of Momentum Portfolio Return and Leverage



(existing momentum portfolio return: yellow line, leverage: green line)

Example of f1-score by validation dataset



(F1-Score: blue line)

3. Research Schedule

In common, we had time to meet with professors once every other week to report progress and results and receive feedback. In this process, various issues were raised, and ideas were created. We also purchased and studied related books to understand momentum, US financial markets, portfolios, machine learning, and deep learning.

3-1. April

We took the time to understand Dynamic Momentum Strategy by referring to linear research. Additionally, the data was visualized to confirm whether there was a unique pattern between Momentum Effect and Momentum Crash and whether a specific relationship existed between variables. Finally, the Rolling Windows algorithm for model learning and inference was implemented.

3-2. May

Based on the contents of the last meeting, the probability value, which is the final output of the Classification Model, was used as leverage. Leverage was derived through Tree-based Machine Learning and DNN series models, and the Sharpe Ratio was slightly increased. And through the Tree-based Machine Learning Model, variables that were important in making inferences for each test data were identified. Through this, it was confirmed that the most important variables in model inference changed after the sub-mortgage prime incident.

3-3. June

Since June, leverage has been derived using Anomaly Detection. When using the Isolation Forest model, greater improvement was achieved compared to using the existing probability value as leverage. This led to the use of Extended Isolation Forest, an improved version of the Isolation Forest model, but a better Sharpe Ratio was not obtained. I think this is because Extended Isolation Forest overfitting occurred more severely than before. And the same leverage function was used for all test datasets.

3-4. July

Based on last month's research direction, we used a neural network model. We also allowed different leverage functions to be applied for each test dataset. The machine learning model showed a higher Sharpe Ratio than the neural network model. We believe that this is due to the overfitting problem of the neural network model. And we summarized the contents of the last four months and gave an interim presentation.

3-5. August

There was an effort to find the optimal methodology by changing various threshold and leverage function settings. In this process, it was discovered that the anomaly score scale of the validation dataset was different for each model, and scaling was applied.

3-6. October & November

New variables necessary for model learning were added. And just like last month, we applied various new methods and compared them. And after trying various methods so far, the highest Sharpe Ratio was derived. And preparations were made for the final presentation.

4. Research Results

It showed a higher Sharpe Ratio than the existing momentum portfolio in all 75 cases that could be created by using 5 variable combinations, 3 leverage functions, and 5 models.

4-1. Supervised vs Semi-Supervised

There was no significant difference between supervised learning and semi-supervised learning in the five models. And for each model, the learning algorithm that showed superiority in Sharpe Ratio was different. It is inferred that the reason for this result is that the unique mechanisms of the models have different strengths and weaknesses. Based on this reasoning, we expect that we will be able to capture various patterns of momentum crashes by combining each model to compensate for its shortcomings.

4-2. Feature Selection

We present five groups by combining various variables. Through this, it was possible to infer which variables were important in predicting momentum crashes. And we used the group with the highest Sharpe Ratio to check what characteristics our methodology had.

Combination of 5 variables

Group 1	all 17 variables
Group 2	7 variables related to volatility
Group 3	6 variables related to Winner
Group 4	6 variables related to Loser
Group 5	9 variables with the highest Sharpe Ratio ‘wml_lag1’, ‘winvol_cum6’, ‘losvol_cum6’, ‘mvol_t_1’, ‘mvol_t_2’, ‘mvol_t_3’, ‘cum_winner_t_2_4’, ‘cum_winner_t_2_4’, ‘cum_winner_t_5_8’, ‘cum_winner_t_9_12’

Overall, when Group 2 was used, it showed a relatively high Sharpe Ratio. Sometimes the Sharpe Ratio showed that using Group 2 was better than using Group 1. This suggests that momentum volatility has many important patterns of momentum crashes. In fact, you can see that the volatility of the momentum portfolio increases after a very large momentum crash.

What is surprising is that in all cases, the Sharpe Ratio was higher when using Group 4 than when using Group 3. This suggests that momentum crashes are caused more by losers than winners. In other words, an increase in the loser's rate of return is a more important variable causing a Momentum Crash than a decrease in the winner's rate of return. This is consistent with the claims of previous research.

However, what is interesting about our study is that Group 5 has more variables related to winners than losers. Given these results, we infer that all variables related to losers do not have a significant effect in predicting momentum crashes. Based on this, we hope that we have selected meaningful variables among those related to loser.

4-3. Model Selection

Contrary to our expectations, the machine learning model showed a higher Sharpe Ratio than the neural network. This infers that an overfitting problem occurred in the neural network because the model was trained with a small train dataset of 240. And due to the feature of financial data, the future may show different patterns from the past due to various factors, so it is not easy to improve the generalization performance of the model. In general, machine learning models are thought to be less sensitive to the over fitting problem because the model complexity is lower than that of neural networks.

Among machine learning models, the isolation forest model showed a high Sharpe Ratio in all groups. It is inferred that the reason for this result is that each small tree used only some variables and then synthesized the results through the tree-based machine learning ensemble algorithm, resulting in excellent generalization performance.

Sharpe Ratio of Isolation Forest with Supervised Learning

Isolation Forest	Group 1	Group 2	Group 3	Group 4	Group 5
Leverage 1	0.949	1.030	0.882	0.988	1.165
Leverage 2	0.958	1.043	0.892	1.010	1.145
Leverage 3	0.974	1.039	0.917	1.038	1.126

Sharpe Ratio of Isolation Forest with Semi-supervised Learning

Isolation Forest	Group 1	Group 2	Group 3	Group 4	Group 5
Leverage 1	0.979	0.978	0.834	0.931	1.161
Leverage 2	0.991	0.978	0.864	0.955	1.141
Leverage 3	1.020	0.984	0.886	0.979	1.115

4-4. Leverage Function

In all groups except Group 5, Leverage 2 showed a higher Sharpe Ratio than Leverage 1, and Leverage 3 showed a higher Sharpe Ratio than Leverage 2. We argue that this is because by adjusting leverage through the f1-score of the validation dataset, the Momentum Effect output higher leverage than 0 in a situation with high volatility. If leverage 1 was used, leverage of 0 would have been granted to a very large momentum effect in a situation with high volatility after a very large momentum crash. However, if leverage 2 and 3 are used, a leverage slightly higher than 0 will be granted in the same situation.

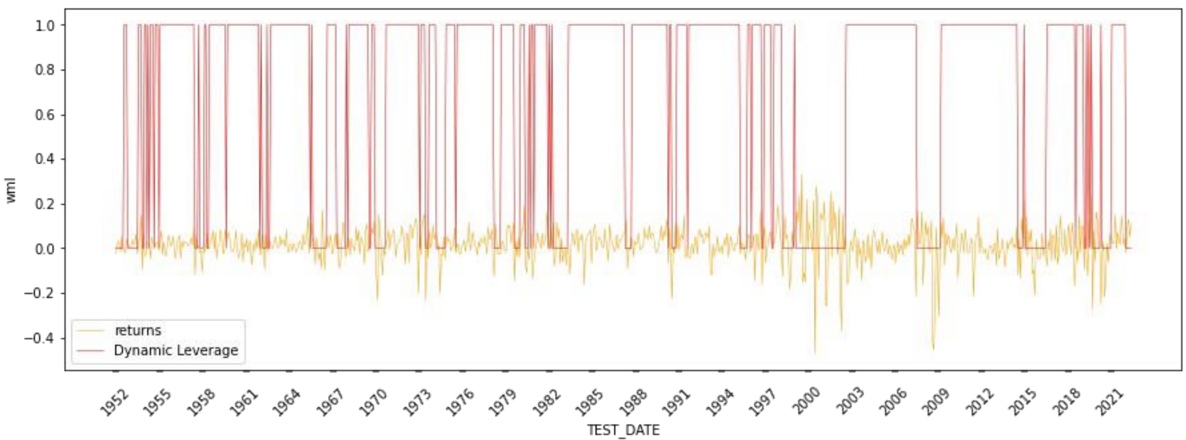
However, we believe that the reason this effect was not seen in Group 5 is as follows. Group 5 is the optimal combination of variables to predict momentum crash. Therefore, the threshold output based on this will be more optimal than the threshold of Group 1 2 3 4. This suggests that finding the optimal threshold is a more important issue in predicting a momentum crash than the structure of the leverage function.

4-5. Highest Sharpe Ratio Method

We tried various methods, and the method with the highest Sharpe Ratio is as follows. After training Group 5 in the Isolation Forest model with Supervised Learning, the Anomaly Score of the Validation Dataset is scaled with a Robust Scaler. Then, apply the Anomaly score of the Test dataset to Leverage Function 1. Through this, the Sharpe ratio increased by up to about 59.6% compared to the existing momentum portfolio, and the recent cumulative return increased by up to about 417.7%.

The reason why this improvement was able to be shown is because leverage of 0 was continuously output in a situation with high volatility after a very large momentum crash occurred, as shown in the graph below. What we can additionally see in the graph below is that we can see briefly that the methodology we proposed is effective in predicting momentum crashes.

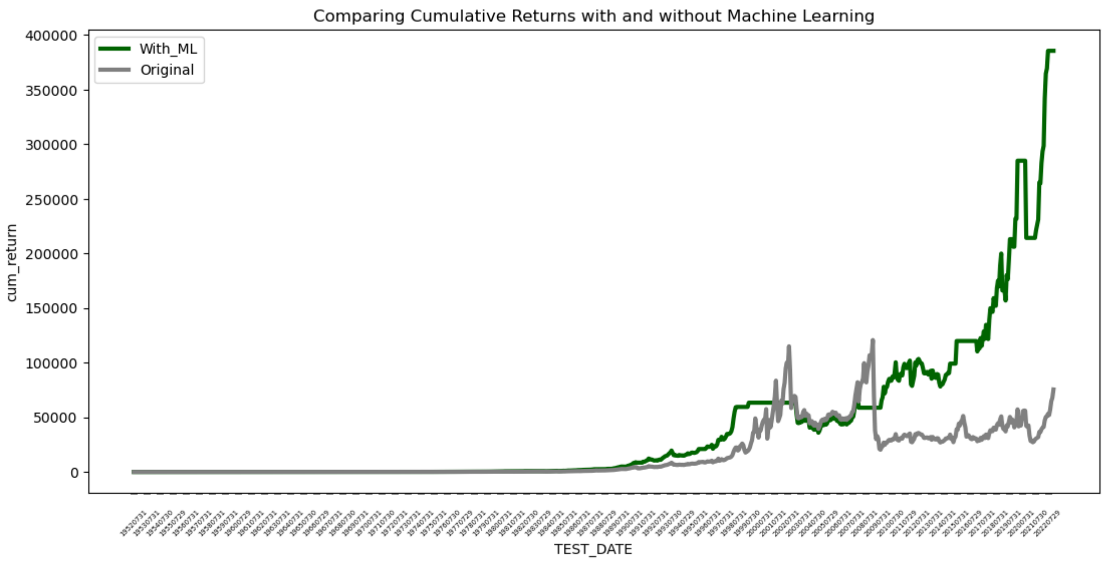
Existing Momentum Portfolio Returns & New Leverage



Looking at the graph below, there are some sections where the existing momentum portfolio has high cumulative returns, but the recent cumulative returns show a very large difference from the existing momentum portfolio. In the section where the existing momentum portfolio showed higher cumulative returns, there is no change because our methodology continuously output leverage 0 during that period. However, while the existing momentum portfolio suffers a momentum crash again and the cumulative return falls sharply, the new methodology shows no change or a relatively small decline.

Another section worth paying attention to is the section where momentum crash due to the relatively recent pandemic. During this period, our methodology appears to be unable to respond well to momentum crashes. This is likely because there is little data on financial crises caused by the virus, making it difficult for the model to predict.

Comparing of cumulative returns with existing momentum portfolios



5. Research Achievements

Through this study, it was confirmed that the Anomaly Detection algorithm is significantly effective in predicting Momentum Crash. In addition, important data features and patterns were identified in the process of predicting the Momentum Crash using the model.

Our results show that, across multiple trials, new methodologies build more attractive momentum portfolios. This suggests that a better factor model can be constructed through machine learning and neural network models.

6. Expected Contribution / Future Plans

Our methodology sets the momentum portfolio weight between 0 and 1 according to the situation predicted by the model. By extending this, leverage and shorting can be used to generate more profits and less losses. This allows for a variety of positions in the financial markets and gives investors more flexibility.

We found effective combinations in predicting momentum crashes through various combinations of variables and identified meaningful variables. However, in addition to the variables we used, there may be other variables that are effective in predicting momentum crashes. We expect to be able to achieve a higher Sharpe Ratio by additionally finding such variables. This process can be an opportunity to understand market anomaly.

Contrary to our expectations, the machine learning model showed better performance than the neural network. However, as neural network models are continuously being developed, new development possibilities can be shown if more effective neural network models are used.

7. Member's Role

	Name	Role	Details
1	Namwoo Kwon	Leader	Team Management, Presentation
2	Yejin Kim	Financial Expert	Data Analysis
3	Daeun Oh	Financial Expert	Data Analysis, Preparing for Presentation
4	Hyeonwook Oh	Developer	Modeling & Algorithm Development
5	Chanbeom Hur	Developer	Modeling and Algorithm Development

8. Participants' Comments

Professor

- **Hope Hyeun Han:**
- **Sang-Ook Shin:**

Team members

- **Namwoo Kwon:** Economic approach and interpretation are also important.
- Y**ejin Kim:** It was meaningful to apply machine learning models to understand momentum data and derive economic implications of the portfolio.
- **Daeun Oh**: It was a great experience to use multiple machine learning technique and with trial and error create a model that would yield optimal sharpe ratio.
- **Hyeonwook Oh:** Applying AI to the study of the momentum effect was a enjoyable experience.
- **Chanbeom Hur:** It was very interesting to see how momentum effect works, and it was a great experience to work with financial data.

We hereby submit the final report for the AICP in 2023 as stated above.

Date:	11 / 02 / 2023	
Professor:	Hope Hyeun Han	(Signature)
Professor	Sang-Ook Shin	(Signature)
Student leader:	Namwoo Kwon	(Signature)