ENSTA IP PARIS



AI X Finance
Models to predict stock return

KASETSART UNIVERSITY

FISE27

Autonomous robotics

October 1, 2026

Clément Sartre
clement.sartre@ensta.fr

Acknowledgement

First of all, I would like to thank Juliette Lourdault and Florian Milin—students at the ENSTA Brest campus, who, like me, completed their internships at Kasetsart University under the supervision of Piya Limcharoen. We shared the same office, and although our internship topics differed, all three projects focused on Machine Learning (ML). This proximity allowed us to support one another throughout the internship. I also wish to thank Piya Limcharoen for his dedication to our internship and for the time he devoted to us, particularly during our weekly meetings. I am also grateful for his many pieces of advice and suggestions regarding both our work and our integration in Thailand. He was always available and open to our questions, which greatly assisted us in both our professional tasks and our discovery of Thai culture. Finally, I would like to thank all the members of the Kasetsart University Computer Science Department for their warm welcome and their efforts to help us settle in.

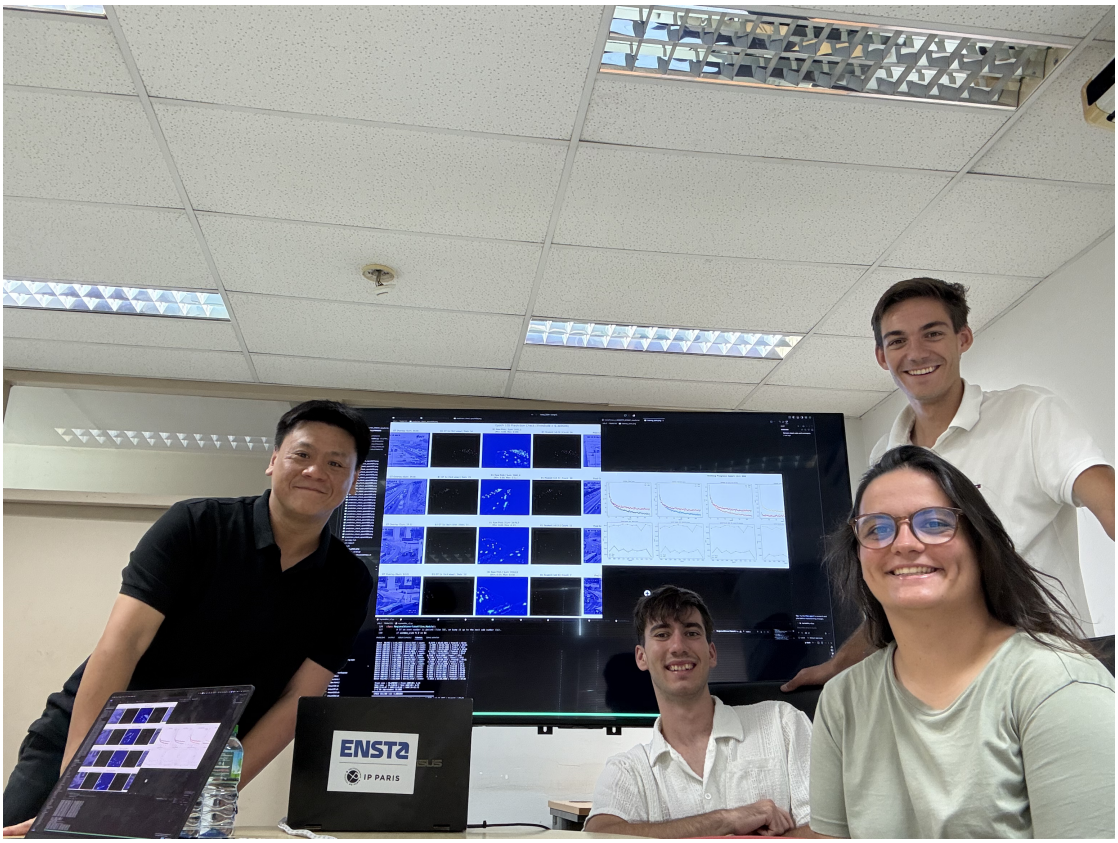


Figure 1: Pictures of my supervisor Piya and my colleagues Juliette and Florian.

Contents

Abstract . . . 3
1 Introduction . . . 3
2 Internship Context . . . 4
2.1 Department and Research Team . . . 4
2.2 Supervision . . . 4
2.3 Course of the Internship . . . 5
3 State of the Art . . . 5
3.1 Finance Journals and Machine Learning Journals . . . 5
4 Two Machine Learning Oriented Articles: Daily Returns from Price History . . . 6
4.1 Abolmakarem et al. (2024): Engineered and Derivative Indices . . . 6
4.2 Prasetyo et al. (2025): LSTM and Inflation Data . . . 7
5 Gu, Kelly and Xiu (2020) with Public Data . . . 10
5.1 The Article . . . 10
5.2 Data . . . 10
5.2.1 Universe and Sources . . . 10
5.2.2 Canonicalization . . . 11
5.2.3 Choice of the 21 Characteristics . . . 12
5.2.4 Panel Raw . . . 12
5.2.5 Panel Ranked . . . 14
5.3 Models, Protocol and Metrics . . . 15
5.3.1 Models . . . 15
5.3.2 Expanding-Window Protocol . . . 16
5.3.3 Evaluation Metrics . . . 17
5.4 Results of the Models . . . 17
5.5 A Closer Look at OLS3 . . . 18
5.5.1 The OLS3 Model . . . 18
5.5.2 The Three Variables . . . 18
5.5.3 Out-of-Sample Performance . . . 19
5.5.4 Correlation Between the Ranked Features and the Target . . . 19
5.5.5 What the Model Learns . . . 20
5.5.6 Predicted Versus Realized Returns . . . 21
5.6 Checking for Data Leakage . . . 22
5.7 Discussion and Limits . . . 24
6 Conclusion . . . 25
Glossary . . . 25
Bibliography . . . 27
Appendices . . . 28
Evaluation Form Completed by the Supervisor . . . 29

Abstract

This report presents my second-year internship at the Computer Engineering department of Kasetsart University, in Bangkok, whose goal was to study to what extent machine learning can predict the future returns of stocks.

I first reproduced two articles that predict daily returns from price history alone (Abolmakarem et al., 2024; Prasetyo et al., 2025): in both cases, my out-of-sample R^2 stays close to zero or negative.

I then attempted to reproduce the article of Gu, Kelly and Xiu (2020) with public data only (Yahoo Finance, SEC XBRL filings and the Kenneth French Data Library), on about 500 S&P500 stocks, 21 characteristics and nine yearly test windows from 2017 to 2026. The best model, OLS3, reaches an out-of-sample R^2 of about 1.6 %, but a constant forecast alone already reaches 1.50 %: its three characteristics only add about 0.1 percentage point, mostly through the size effect.

These results show that the signal contained in stock data is thin compared with the noise of returns, and that a strict out-of-sample protocol and leakage checks are essential before trusting a predictive performance.

Keywords: machine learning, empirical asset pricing, stock return prediction, data leakage.

1 Introduction

Finance covers the activities related to the allocation of resources over time, the management of risk, and the search for investment opportunities. This report focuses on market finance, and more precisely on stocks. A stock represents a fraction of a company's capital. Its price varies with supply and demand, which depend on the company's results, investor expectations, the general economic situation, or interest rates. The return on a stock measures the gain or loss generated by holding it over a given period, through the change in its price and, when taken into account, the dividends paid.

Two kinds of data describe a stock. Market data, such as historical prices and trading volumes, make it possible to build variables such as past returns, volatility, or momentum. Accounting data, taken from the balance sheets and income statements that listed companies in the United States publish through the Securities and Exchange Commission (SEC), measure their size, valuation, profitability, or debt level.

Machine learning is a field of artificial intelligence whose algorithms learn relationships from data: instead of following manually defined rules, a model adjusts its parameters to a set of observations, then uses what it has learned to predict[1] a value on new data. Its main families (regularized linear regressions, decision trees, random forests, boosting methods, and neural networks) differ in their ability to represent nonlinear relationships, to select relevant variables, and to capture interactions between them. These properties make machine learning attractive in finance, where candidate variables are numerous and their links with returns can be complex, unstable, and nonlinear. Predicting returns nevertheless remains difficult, because financial data are very noisy, markets evolve constantly, and unpredictable events weigh on prices.

The initial goal of this internship was to create a new method based on Artificial Intelligence (AI) in order to improve existing approaches for predicting certain financial market values. This topic falls within a context where financial data is increasingly abundant, accessible, and diverse. Asset prices, trading volumes, economic indicators, and variables derived from technical analysis are all sources of information that can be exploited to better understand market dynamics.

[1] The term "predict" is used here in the statistical sense: it refers to estimating a variable of interest from observed data and relationships learned within a database.

Predicting financial quantities, such as the future return of a stock, has many applications. It can notably contribute to decision support for investors, the improvement of trading strategies, portfolio construction, or better risk assessment. However, obtaining reliable predictions remains a significant technical challenge. Financial markets are characterized by high uncertainty, a high level of noise, relationships that are sometimes unstable over time, and great sensitivity to economic, political, or geopolitical events. These factors make it difficult to extract a genuinely usable signal from the available data.

In this context, many research works seek to improve the accuracy, robustness, and generalization ability of predictive models applied to financial markets. The main challenge of this internship was therefore to reduce prediction errors for certain financial variables, particularly returns, while keeping a method robust enough to remain relevant on real, noisy, and evolving data.

The initial goal of this internship was to propose a new machine learning method improving existing approaches for predicting stock returns. Before improving these approaches, I first had to reproduce and evaluate them. This report therefore addresses a preliminary question: to what extent can machine learning predict the future returns of stocks by combining market data with SEC accounting data?

After presenting the internship context and the problem statement, this report describes the state of the art, then the reproduction of three reference articles: two that predict daily returns from price history alone [1, 2], and the article of Gu, Kelly, and Xiu [3], reproduced with public data only. For this third article, I did not stop at reimplementing the models: I analyzed one of them, OLS3, in more detail, to understand where its performance comes from and to check that it is not produced by a data leak.

2 Internship Context

I carried out my second-year internship in a research laboratory of the Computer Engineering department at Kasetsart University (KU) in Bangkok, Thailand.

2.1 Department and Research Team

This department brings together the university's computer engineering activities. It has two main activities: teaching and research. For my part, I was in a research laboratory, surrounded by PhD students and researchers. Even though my supervisor was also a professor alongside his research work, I was not involved in that activity.

This work is of interest to the research team, whose work mainly focuses on the study and development of sensors, while also showing an interest in applications of AI in fields related to finance. Indeed, some members of the team already use AI methods to estimate the value of real estate properties based on their surroundings, notably to help banks assess loan amounts. My role was to bring a fresh perspective to this problem, to propose a new approach, and then to evaluate it and compare it with existing methods, drawing on the skills acquired during my engineering education.

2.2 Supervision

My internship supervisor is Piya Limcharoen. He holds a PhD in computer engineering and works in several fields, including AI, Machine Learning (ML), and Deep Learning (DL). He is currently working on the processing of raw sensor data for AI-based robotics. The goal is to improve the accuracy of these sensors.

2.3 Course of the Internship

A research project generally unfolds in several stages. The first two are closely linked and consist of becoming familiar with existing work on the topic.

- **State of the Art**: study of the scientific literature in order to identify work already carried out on the topic. This step makes it possible to understand existing approaches, to avoid duplicating work already done, and to define a starting point for the research.
- **Implementation of Existing Methods**: reproduction and analysis of the methods proposed in previous work in order to ensure they are properly understood, to evaluate their performance, and to identify their possible limitations.
- **Development of a New Method**: research and experimentation phase aimed at proposing improvements to existing approaches and obtaining better performance.
- **Writing a Scientific Article**: formalization and communication of the results obtained in order to share them with the scientific community and to allow other researchers to continue or improve this work.

The state of the art was a substantial step in this research internship, as it required assimilating both the vocabulary of machine learning and that of market finance. This phase lasted about four weeks. The first week was devoted to choosing a feasible topic. The initial topic was meant to focus on sensor data, but the difficulty in finding usable datasets led to redirecting the work toward a financial application of artificial intelligence. The following three weeks were devoted to a more in-depth reading of the literature, in order to clarify the scope of the topic, the methods to be implemented, and the reference articles to be studied in detail.

3 State of the Art

3.1 Finance Journals and Machine Learning Journals

This reading showed that the literature on machine learning applied to finance does not form a homogeneous body of work. A useful distinction contrasts articles published in finance journals with those published in machine learning, or artificial intelligence journals, as these two communities formulate the scientific problem, evaluate results, and define what constitutes a contribution in different ways. In finance journals, machine learning is generally used as a tool in service of financial questions: measuring risk premia, predicting returns, out-of-sample evaluation, transaction costs, or portfolio construction. The article by Gu, Kelly, and Xiu, published in *The Review of Financial Studies*, is a good illustration of this approach [3]. The authors compare several families of models for an *empirical asset pricing* problem and show that trees and neural networks achieve the best results, notably because they capture nonlinear interactions between financial variables. Along the same lines, Leippold, Wang, and Zhou study return predictability in the Chinese market and place liquidity and transaction costs at the center of their analysis [4].

Conversely, in machine learning or intelligent systems journals, the emphasis is more directly on predictive performance, the choice of input variables, model architecture, and error metrics. The article by Abolmakarem et al., published in *Intelligent Systems with Applications*, for example proposes a multi-stage approach based on indices built from historical series, then uses *multi-layer perceptron* neural networks to predict the next day's price for four FTSE 100 stocks [1]. Similarly, Prasetyo, Witanti, and Hadiana use a Long Short-Term Memory (LSTM) model to predict stock returns from inflation, price history, and computed returns, with an evaluation based on Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE)

[2]. In this type of work, the main contribution often lies in the forecasting protocol or in the comparison of models, rather than in the economic interpretation of the result.

For this report, this distinction is structuring, since my work lies at the intersection of these two approaches: it draws on machine learning tools, but evaluates them on a financial task where out-of-sample robustness, the absence of data leakage, and the economic interpretation of returns are decisive.

4 Two Machine Learning Oriented Articles: Daily Returns from Price History

The first article I tried to reproduce was that of Abolmakarem, Abdi, Khalili-Damghani, and Didehkhani (2024), *"A multi-stage machine learning approach for stock price prediction: Engineered and derivative indices"*, published in *Intelligent Systems with Applications*.

The second article I tried to reproduce was that of Prasetyo, Nur Faid and Witanti, Wina and Hadiana, Asep Id (2025), *"Forecasting stock returns using long short-term memory (LSTM) model based on inflation data and historical stock price movements"*, published in *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*.

The third article I tried to reproduce was that of Gu, Shihao and Kelly, Bryan and Xiu, Dacheng (2020), *"Empirical asset pricing via machine learning"*, published in *The Review of Financial Studies*.

4.1 Abolmakarem et al. (2024): Engineered and Derivative Indices

This article is machine learning oriented. The research paper find his originality in creating engineered indices that could be better to train machine learning models on.

My Implementation I followed the article's protocol almost identically and I ran this protocol on CAC40 stocks (LVMH, TotalEnergies, Sanofi, Danone, Orange, Imerys) over the 2000-2020 period, just as the article does on 4 FTSE 100 stocks over the same period.

I wanted to apply the method from the research article to CAC40 stocks to apply that method to stocks that I already know. Though, I should have exactly followed the method of the research article as using MATLAB instead of python and selecting the same stock in order to make better comparisons. Still, even doing exactly the same method results in different results due to randomness in the training part and predicting a daily return with an R^2 of 0.9999 based solely on price history, without any accounting or fundamental data, is incompatible with the near-random nature of the market: such a level of precision is, in practice, still impossible to obtain honestly on this type of problem.

My Results On LVMH, for the *Ret* target (daily return), the best model selected on validation achieves the following performance:

Configuration	R^2 train	R^2 validation	R^2 test
15 neurones, 500 epochs	0.045	-0.012	-0.040
25 neurones, 500 epochs	0.060	-0.009	-0.194

Table 1: Performance obtained on LVMH / Ret with causal alignment.

R^2 values close to zero, even negative: a daily return is virtually unpredictable from price history alone, which is consistent with the efficient market hypothesis.

The Results Reported by the Article Yet the article reports, for this same type of target and with an almost identical protocol, extraordinary performance:

- *Ret*: $R^2 > 0.9992$ for the 4 stocks (BP: 0.99992; BHP: 0.99999; GSK: 0.99982; VOD: 0.99999), including on the test set.
- *Delta*: $R^2 > 0.9998$ for the 4 stocks.
- *PKHe / TRUDe*: $R^2 > 0.999$ as well.
- *Trend* (classification): accuracy of 96 to 98%.

Analysis of the Gap: My Data Leakage Test It is this gap between my results, realistic but mediocre, and those of the article, nearly perfect, that led me to test the hypothesis of a data leak. I compared, on LVMH / Ret (25 neurons, 500 epochs), a causal alignment (input variables from day t predicting the return from t to $t+1$) with an alignment where the input variables are shifted one day forward, and therefore partially derived from the closing price at $t+1$, the very price used to compute the return to be predicted:

Split	R^2 without leakage (causal)	R^2 with leakage
Train	0.376	0.999
Validation	-0.970	0.993
Test	-2.221	0.664

Table 2: Comparison of performance with and without intentional data leakage.

A simple one-day shift is enough to bring the R^2 from very negative to nearly perfect (0.99), which is exactly the order of magnitude of the scores reported in the article. This strongly reinforces the hypothesis that the performance reported by the article is explained by a similar data leak rather than by any real predictive power of the model. Even though this is not a proof of the wrongness of the article, I decided to try implementing the second article without loosing more time (after having hardly tried to convince my tutor to warn the journal to review again the article as their implementation).

4.2 Prasetyo et al. (2025): LSTM and Inflation Data

The second article I tried to reproduce was that of Prasetyo, Witanti, and Hadiana (2025), *"Forecasting Stock Returns Using Long Short-Term Memory (LSTM) Model Based on Inflation Data and Historical Stock Price Movements"*, published in *Jurnal RESTI*.

This article is also machine learning oriented. This research find his originality on adding inflation data in the data used to train his model.

My Implementation I followed the article's protocol: downloading the price of the BBCA.JK stock (Bank Central Asia, Indonesia) from Yahoo Finance and the Indonesian annual inflation rate, temporal merging of the two sources (the inflation value for a given year is applied to all days of that same year, as described in the article), computing the daily return $SR_t = \frac{AdjClose_t - AdjClose_{t-1}}{AdjClose_{t-1}} \times 100$, an 80%/20% train/test time split, scaling (*MinMaxScaler* fitted on the training set only), and building sequences of 30 time steps over 7 variables (Open, High, Low, Close, Adj Close, inflation rate, SR). The model uses exactly the best configuration reported by the article: two LSTM layers of 50 units, dropout of 0.2, Dense(1) output, RMSprop optimizer, 200 epochs, batch size of 8.

My Results The metrics obtained on the test set are as follows:

Configuration	MSE	RMSE	MAE	MAPE	R^2
RMSprop, batch 8, 200 epochs	1.85	1.36	1.01	93 263 735 %	≈ 0

Table 3: Performance obtained on BBCA / SR with the configuration taken from the article.

Examining the predictions directly, the model actually collapsed to a nearly constant value (mean ≈ 0.091, standard deviation ≈ 0.0014), whereas the actual target ranges from -10% to $+17\%$ (standard deviation $\approx 1.85\%$). The correlation between prediction and reality is essentially nil (≈ -0.008) and the R^2 is essentially zero ($\approx -1.6 \times 10^{-5}$). The training loss curve confirms this diagnosis: it remains nearly flat over the 200 epochs (from 3.88 to 3.84), which corresponds to the loss level achieved by simply predicting the historical mean return. The aberrant MAPE is explained by division by daily returns very close to zero, a known flaw of this metric on this type of series, independent of the model's quality.

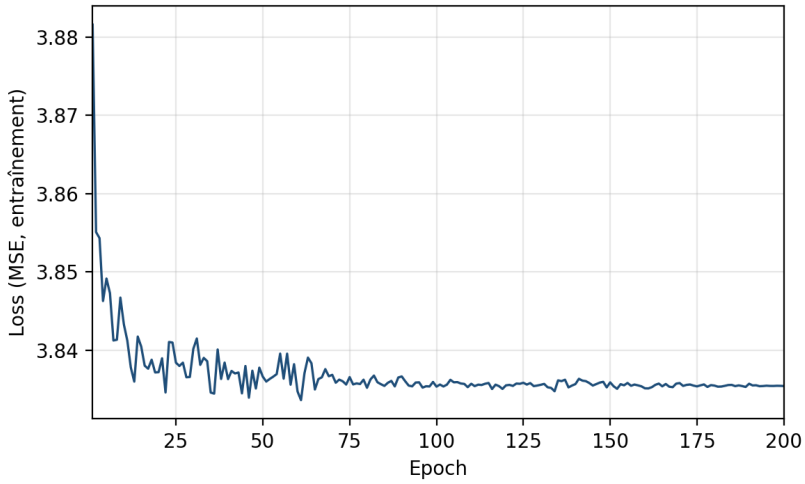


Figure 2: Training loss curve (MSE) of the LSTM on BBCA / SR, RMSprop configuration, batch 8, 200 epochs. The loss remains nearly flat, a sign that the model does not learn any signal beyond the historical mean return.

The Results Reported by the Article For this same configuration (RMSprop, batch 8, 200 epochs), the article reports:

- RMSE = 0.0167
- MSE = 0.0002
- MAE = 0.0129
- MAPE = 25.89%

The article does not, however, clearly specify whether these values are expressed on the percentage scale used to define the target (Equation 1 of the article) or on an equivalent decimal scale, which strongly changes the interpretation of the result.

Analysis of the Gap Two readings are possible. If the article's RMSE is expressed on the same percentage scale as my own target, then an average error of 0.0167 percentage points on a daily return again constitutes a nearly perfect and implausible performance, consistent with the finding already established for the first article: such a level of precision is incompatible with the near-random nature of stock returns. If, on the other hand, this RMSE is expressed on a decimal scale, the article's result becomes plausible again and is closer to my own order of magnitude.

Later in the internship, I nonetheless returned to this article to settle both this implementation doubt and the scale ambiguity discussed above, re-implementing the full pipeline and computing both an out-of-sample and an in-sample R^2 for each hyperparameter, against a constant-zero benchmark rather than the sample mean. The decimal-scale reading turns out to be the better-supported one: dividing my own RMSE by 100 gives values between 0.0134 and 0.0142 across the 24 configurations, close to the article's reported 0.0167, whereas the percentage-scale reading would leave my results roughly two orders of magnitude off.

Out-of-sample R^2 never exceeds $+0.015$ and is negative for 19 of the 24 configurations, and even the best configuration only improves on a trivial "always predict the historical mean" baseline by about 1.5 % in RMSE. Longer training makes this worse rather than better: for configurations trained over 300 epochs, in-sample R^2 climbs as high as 0.32 while the corresponding out-of-sample R^2 stays at or below zero. This overfitting rather than genuine learning once again illustrates the difficulty of obtaining, without data leakage or fundamental information, genuine predictive ability on daily stock returns using only historical stock data.

Critical Reading of the Article In my first implementation, I trained the models but was unable to identify any meaningful predictive signal from historical price data. However, I could not be completely certain that this result was reliable, since my implementation itself might have contained an error that I had failed to identify.

Nevertheless, I decided to move on to another research article rather than remain stuck on a study for which I had little confidence or intuition. Several observations motivated this decision.

First, I had already experienced working with an article whose methodology turned out to be questionable, so I knew that published research could sometimes contain inconsistencies or methodological weaknesses.

Second, I noticed an apparent inconsistency in the definition of the target variable. The article defines the feature

$$SR_t = \frac{AdjClose_t - AdjClose_{t-1}}{AdjClose_{t-1}} \times 100$$

so that stock returns are expressed as percentages, while later referring to the predicted return $\hat{R}_t$ as a decimal value. This difference in scale was not clearly explained.

Third, the article states that "adding inflation as a macroeconomic factor [...] improves stock return prediction accuracy." However, the authors do not appear to present a corresponding experiment without inflation. A comparison between models trained with and without this variable would have provided stronger empirical support for such a conclusion.

Finally, the article gave me the impression of being strongly result-oriented. Its main objective seemed to be achieving better predictive performance on a limited set of stocks and comparing these results with those reported in other research papers, even when those studies considered different stocks or datasets. This approach is relatively typical of machine-learning-oriented research, where performance metrics often play a central role. Our supervisor also encouraged us to improve the model according to a clearly defined performance metric, but I wanted the experimental methodology and the interpretation of the results to remain sufficiently rigorous and convincing.

5 Gu, Kelly and Xiu (2020) with Public Data

5.1 The Article

The third article I tried to reproduce was that of Gu, Shihao and Kelly, Bryan and Xiu, Dacheng (2020), *"Empirical asset pricing via machine learning"*, published in *The Review of Financial Studies*.

Contrary to the two previous article, this article is more finance oriented and not only use historical stock data but accounting data too. This research paper compare different machine learnings methods applied to stock prediction, in order to measure equity risk premiums in different ways. This article is an empirical study meaning that the conclusions are evidence-driven, its rely directly on what the collected data shows, rather than on personal beliefs or abstract reasoning. The two previous articles, on another hand, showed to be result-driven and are likely to have made some wrong experiences or wrong interpretation.

5.2 Data

5.2.1 Universe and Sources

The article studies all the stocks listed on the three main U.S. exchanges (NYSE, AMEX and NASDAQ) from March 1957 to December 2016: almost 30 000 stocks in total, and more than 6 200 in an average month, small companies included. My universe is much narrower: it is made of the 503 tickers that currently make up the S&P 500, that is, about 500 of the largest U.S. companies, some of them having several share classes. 471 of these tickers end up in the final panel, which covers January 2005 to April 2026 and contains on average about 450 stocks per month from 2009 onward. Since these are today's constituents, the universe leaves out the companies that have left the index or disappeared; this survivorship bias is discussed in Section 5.7.

Collecting the exact same dataset was not possible. Indeed, the article relies on two private datasets :

- The Center for Research in Security Prices (CRSP) dataset for returns, prices, volumes, market data, and certain monthly characteristics
- Compustat dataset for fundamental accounting data

My university has not subscribed to these two private datasets and I had to find a way to collect as much data as I could to be the closest I can from the article. To replace those databases, I relied on Yahoo Finance for prices, volumes, dividends and splits and I relied on SEC EDGAR for accounting fundamentals via XBRL filings. Each company have to submit public financial reports with the eXtensible Business Reporting Language (XBRL) format, in accordance with the rules of the securities and exchange commission (SEC). Those filling are archived in a public database called EDGAR (Electronic Data Gathering, Analysis, and Retrieval). The risk-free rate and certain market factors were obtained using the Kenneth French Data Library. Once the data has been collected, we need to turn the raw sources into our dataset containing the predictor variables and the target.

The Securities and Exchange Commission (SEC) is the U.S. federal agency in charge of regulating financial markets. Every listed company must file periodic reports with it, in particular an annual report (Form 10-K) and quarterly reports (Form 10-Q), which are made public on EDGAR. Since the SEC progressively made it mandatory between 2009 and 2011, the financial statements of these reports are also filed in XBRL, a machine-readable format in which each figure is associated with a standardized accounting tag (for instance `us-gaap:StockholdersEquity`), a period and a filing date.

Our SEC data were collected through the *companyfacts* Application Programming Interface (API) of the SEC, which gathers in a single JavaScript Object Notation (JSON) file all the XBRL figures ever reported by a given company. This makes the collection simple and reproducible, but I am aware that this file does not contain all the data. For example, Tyson Foods (TSN) has two classes of shares, Class A and Class B, and reports its number of shares outstanding separately for each class, never as a single total. Since *companyfacts* only keeps the figures that describe the company as a whole, the number of shares outstanding of TSN is simply absent from it. The XBRL figures themselves also contain errors: some values are reported with a wrong scale[2], some companies create their own tag instead of using the standard one[3], and more generally XBRL filings are known to contain a significant number of errors[4].

A few of these cases, such as TSN, were corrected individually. However, trying to understand the XBRL format in detail and to fix every existing problem had already cost me a lot of time, and would have taken far too long for this internship. I therefore kept the *companyfacts* data, knowing that they remain an imperfect substitute for the Compustat data used in the article.

Despite this step was really long, some SEC data problems remain and some might have never been identified.

5.2.2 Canonicalization

The goal of this layer is to produce stable tables that can be joined later with Yahoo Finance, Fama-French and SEC in the panel assembly step in order to produce the raw predictor variables.

The main goal of this layer is to turn the raw sources into tables that can be joined cleanly:

- Yahoo is normalized into a monthly table (close, adj_close) and a daily table (close, adj_close, volume, high, low), the latter feeding the liquidity/risk features (retvol, maxret, dolvol, turn, std_turn, ill, zerotrade, baspread, beta, idiovol);
- Fama-French is parsed into a monthly table and a daily table (mkt_rf, smb, hml, rf), with decimal conversion; the daily table feeds beta/idiovol;
- SEC shares outstanding and the stockholders_equity / revenues / eps_basic / secured_debt_current / secured _debt_noncurrent fundamentals are retained as filing-level facts;
- An availability lag is applied when assembling each panel (allowing several lag specifications without re-running canonicalization)

In order to produce the selected raw predictor variables, we need to gather the following accounting data from the SEC database, in addition to the number of shares outstanding:

- **stockholders_equity** : the book value of the company, that is, what remains for the shareholders once all debts are subtracted from the assets (assets minus liabilities). It is a balance sheet figure, measured at a given date
- **revenues** : the total amount of sales made by the company over a period, before any expense is deducted. It is an income statement figure, measured over a period; only annual values are kept
- **eps_basic** : the earnings per share, that is, the net income of the period divided by the average number of shares outstanding. Only quarterly values are kept

[2]The SEC itself warned filers about such scaling errors on the number of shares outstanding, for instance 555,555,000 shares tagged as 555,555 [5].

[3]Debreceny et al. (2011) [6]

[4]About one quarter of the first XBRL filings of the largest U.S. companies contained figures that did not add up to the sum of their components [7].

- **secured_debt_total**, **secured_debt_current** and **secured_debt_noncurrent** : the secured debt, that is, the debt guaranteed by an asset (a collateral) that the lender can seize if the company does not repay. The total is sometimes reported directly, sometimes split into a current part (due within one year) and a noncurrent part (due later)

Each of these data points is therefore used at least once in the calculation of the raw predictor variables.

5.2.3 Choice of the 21 Characteristics

In the research paper, there is 920 raw predictor variables obtained by doing a Kronecker product of the stock characteristics with the macro variables. Though, in order to make an achievable implementation, knowing my limited access to all the same data, I only selected the top-20 most influential stock characteristics in the NN3 model showed in the Figure 4 of the article (Variable importance by model) plus the book market variable used in the OLS3 model of the article [5]. This raw predictor variables are specified as raw because we ranked them, in a future step, in order to keep value between -1 and 1.

The calculation of those stock characteristics first need the data to be canonicalized.

5.2.4 Panel Raw

Once the raw data has been canonicalized, the next step is to create the dataset, that means computing all the features and the target, for each stock and each month. The result of this step is the panel raw: a single table with one row per stock and per month, which contains the date, the ticker, the 21 features and the target. In what follows, i denotes a stock, m a month and d a trading day, and the following notations are used:

- $AC_{i,d}$ and $C_{i,d}$: the Yahoo Finance closing prices of day d, respectively adjusted for splits and dividends and unadjusted; $AC_{i,m}$ and $C_{i,m}$ denote their values on the last trading day of month m;
- $V_{i,d}$, $H_{i,d}$ and $Lo_{i,d}$: the volume, the highest price and the lowest price of day d;
- $R_{i,m} = AC_{i,m}/AC_{i,m-1} - 1$ and $R_{i,d} = AC_{i,d}/AC_{i,d-1} - 1$: the monthly and daily returns, and $DV_{i,d} = C_{i,d} \times V_{i,d}$: the daily dollar volume;
- $S_{i,m}$, $E_{i,m}$ and $Sales_{i,m}$: the number of shares outstanding, the stockholders' equity and the annual revenues reported to the SEC, and $ME_{i,m} = C_{i,m} \times S_{i,m}$: the market capitalization;
- RF, MKT, SMB and HML: the risk-free rate, the market excess return, the size factor and the value factor of the Kenneth French Data Library, at a monthly or daily frequency.

An accounting figure is not known by the market at the end of the period it describes, but only once it has been filed with the SEC. The SEC data are therefore taken point-in-time: a fact only becomes usable $L = 30$ days after its filing date, and in month m the most recent fact available at the end of the month is retained.

For a stock i in month m, the target is the excess return of the following month

$$r_{i,m+1} = R_{i,m+1} - RF_{m+1} = \frac{AC_{i,m+1}}{AC_{i,m}} - 1 - RF_{m+1}.$$

The features are the following, grouped by family.

Momentum. These features are past returns over different horizons. Except `mom1m`, they skip the most recent month(s), in order to separate the medium-term momentum effect from the

[5] This model is explained later in the report.

short-term reversal effect. `indmom` is the average `mom6m` of the stocks of the same industry $ind(i)$ as stock i, weighted by market capitalization, the industries being the 48 Fama-French industries obtained from the Standard Industrial Classification (SIC) code of each company.

$$\text{mom1m}_{i,m} = \frac{AC_{i,m}}{AC_{i,m-1}} - 1, \qquad \text{mom6m}_{i,m} = \frac{AC_{i,m-1}}{AC_{i,m-6}} - 1,$$
$$\text{mom12m_skip1m}_{i,m} = \frac{AC_{i,m-1}}{AC_{i,m-12}} - 1, \qquad \text{mom36m}_{i,m} = \frac{AC_{i,m-13}}{AC_{i,m-36}} - 1,$$
$$\text{chmom}_{i,m} = \text{mom6m}_{i,m} - \text{mom6m}_{i,m-6},$$
$$\text{indmom}_{i,m} = \frac{\sum_{j \in ind(i)} ME_{j,m}\, \text{mom6m}_{j,m}}{\sum_{j \in ind(i)} ME_{j,m}}.$$

Size and valuation. `mvel1` measures the size of the company, `bm` (book-to-market) and `sp` (sales-to-price) compare its accounting value and its revenues with its market value.

$$\text{mvel1}_{i,m} = \ln\left(ME_{i,m}\right), \qquad \text{bm}_{i,m} = \frac{E_{i,m}}{ME_{i,m}}, \qquad \text{sp}_{i,m} = \frac{Sales_{i,m}}{ME_{i,m}}.$$

Liquidity. These features measure how easily the stock is traded during month m: dollar volume, turnover and its volatility, Amihud illiquidity, share of days without any trade, and bid-ask spread.

$$\text{dolvol}_{i,m} = \ln\Big(\sum_{d \in m} DV_{i,d}\Big), \qquad \text{ill}_{i,m} = \underset{d \in m}{\text{mean}}\left(\frac{|R_{i,d}|}{DV_{i,d}}\right),$$
$$\text{turn}_{i,m} = \underset{d \in m}{\text{mean}}\left(\frac{V_{i,d}}{S_{i,m}}\right), \qquad \text{std_turn}_{i,m} = \underset{d \in m}{\text{std}}\left(\frac{V_{i,d}}{S_{i,m}}\right),$$
$$\text{zerotrade}_{i,m} = \frac{\#\{d \in m : V_{i,d} = 0\}}{\#\{d \in m : V_{i,d} \text{ is known}\}}, \qquad \text{baspread}_{i,m} = \underset{d \in m}{\text{mean}}\left(spread_{i,d}\right).$$

Since Yahoo Finance does not provide bid and ask quotes, the daily spread is estimated from the highest and lowest prices of two consecutive days, with the estimator of Corwin and Schultz:

$$\beta_d = \left[\ln\frac{H_{i,d}}{Lo_{i,d}}\right]^2 + \left[\ln\frac{H_{i,d+1}}{Lo_{i,d+1}}\right]^2, \qquad \gamma_d = \left[\ln\frac{\max(H_{i,d}, H_{i,d+1})}{\min(Lo_{i,d}, Lo_{i,d+1})}\right]^2,$$
$$\alpha_d = \frac{\sqrt{2\beta_d} - \sqrt{\beta_d}}{3 - 2\sqrt{2}} - \sqrt{\frac{\gamma_d}{3 - 2\sqrt{2}}}, \qquad spread_{i,d} = \max\left(0,\ \frac{2\left(e^{\alpha_d} - 1\right)}{1 + e^{\alpha_d}}\right).$$

Risk. `retvol` and `maxret` are the volatility and the maximum of the daily returns of the month. `beta` is the sensitivity of the stock to the market, estimated over the 252 trading days ending at the end of month m (at least 200 observations). `idiovol` is the volatility of the part of the return that the three Fama-French factors do not explain, estimated on the days of month m only (at least 15 observations).

$$\text{retvol}_{i,m} = \underset{d \in m}{\text{std}}\left(R_{i,d}\right), \qquad \text{maxret}_{i,m} = \underset{d \in m}{\max}\left(R_{i,d}\right),$$
$$\text{beta}_{i,m} = \frac{\text{Cov}\left(R_{i,d} - RF_d,\ MKT_d\right)}{\text{Var}\left(MKT_d\right)}, \qquad \text{idiovol}_{i,m} = \underset{d \in m}{\text{std}}\left(\widehat{\varepsilon}_{i,d}\right),$$

where $\widehat{\varepsilon}_{i,d}$ are the residuals of the regression

$$R_{i,d} - RF_d = a_i + b_{1,i}\, MKT_d + b_{2,i}\, SMB_d + b_{3,i}\, HML_d + \varepsilon_{i,d}, \qquad d \in m.$$

Accounting signals. `nincr` counts the earnings increases of the company. Let $EPS_{i,q}$ be the successive quarterly earnings per share available at the end of month m, q being the most recent one, each new value published being treated as a new quarter. `nincr` counts how many times in a row, going back one year at a time, the earnings per share of a quarter are strictly higher than those of the same quarter one year earlier:

$$\text{nincr}_{i,m} = \max\left\{k \geq 0 \ : \ EPS_{i,q-4j} > EPS_{i,q-4(j+1)} \text{ for } j = 0, \ldots, k-1\right\}.$$

`securedind` indicates whether the company reports secured debt, from the three SEC tags SD^{tot} (total secured debt), SD^{ncur} (noncurrent part) and SD^{cur} (current part):

$$\text{securedind}_{i,m} = \begin{cases} 1 & \text{if } SD^{tot}_{i,m} > 0 \text{ or } SD^{ncur}_{i,m} > 0 \text{ or } SD^{cur}_{i,m} > 0, \\ 0 & \text{if at least one of them is reported and none is positive,} \\ \text{missing} & \text{if none of them has ever been reported.} \end{cases}$$

5.2.5 Panel Ranked

Once the features are all computed into the raw panel, we need to normalize them so that all the features keep the same order of value. In the raw panel, the scales are indeed very different from one feature to another. Most of them are also strongly skewed, with a few extreme values. Given as they are to a model, the features with the largest values or the most extreme outliers would dominate the estimation.

As in the article, each feature is therefore replaced, month by month, by its cross-sectional rank rescaled to $[-1, 1]$:

$$\text{rank}_{i,m} = \frac{2\,(r_{i,m} - 1)}{n_m - 1} - 1,$$

where $r_{i,m}$ is the rank of stock i among the n_m stocks for which the feature is available in month m. Only the features are ranked: the target is kept as an excess return in decimals, since it is the quantity that the models predict and on which they are evaluated. The result is the *panel ranked*, which has the same rows and columns as the panel raw.

Figure 3 shows the number of stocks available each month and the share of them for which each variable can be computed. The momentum only needs prices and is available for almost every stock from 2005 onward. On the other hand, `mvel1` and `bm` need SEC data in XBRL format, which was progressively made mandatory between 2009 and 2011: their coverage is essentially zero before 2009, then rises very quickly before slowly converging toward 100 %. This is why the backtest starts in January 2009.

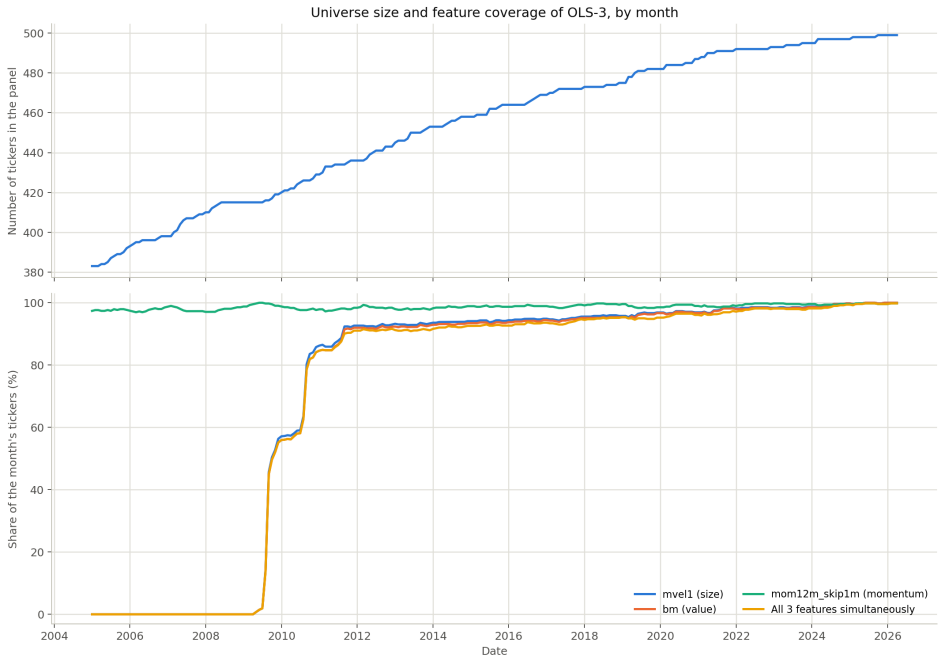


Figure 3: Top: number of tickers in the panel each month. Bottom: share of these tickers for which each OLS3 variable is available. The coverage of `mvel1` and `bm` follows the adoption of XBRL filings by the SEC.

5.3 Models, Protocol and Metrics

5.3.1 Models

The article trains different models, from the simplest linear regressions to deep neural networks:

- **OLS**: ordinary least squares linear regression
- **OLS3**: ordinary least squares linear regression using three characteristics, size, book-to-market and momentum, which serves as a simple benchmark;
- **PLS**: Partial Least Squares
- **PCR**: Principal Components Regression
- **ENet**: Elastic Net regression
- **GLM**: Generalized Linear Model
- **RF**: Random Forest
- **GBRT**: Gradient Boosted Regression Trees
- **NN1 to NN5**: feed-forward neural networks with one to five hidden layers

Several of these models (OLS, OLS3, ENet, GLM and GBRT) also exist in a "+H" version, in which the squared loss is replaced by the Huber loss, which is less sensitive to extreme returns.

5.3.2 Expanding-Window Protocol

All the models are trained and evaluated with the same protocol, adapted from the article. The sample is split in time, with expanding windows (Figure 4). It starts in January 2009, when the SEC data in XBRL format become available. In the first iteration, the first 30 % of the months are used for training, the following 20 % for validation and the next 12 months for the test. At each following iteration, the whole split moves forward by 12 months: the training set grows by 12 months, while the validation set keeps the same length and is rolled forward by 12 months, so that it always covers the most recent months before the test set. This gives 9 iterations whose test sets cover August 2017 to April 2026.

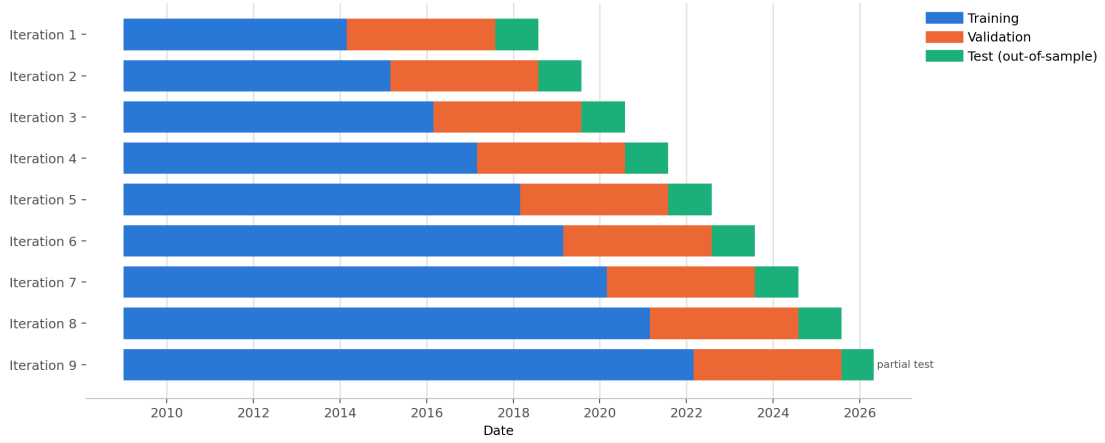


Figure 4: Expanding-window schedule: the training set (blue) grows by one year at each iteration, and is followed by 41 months of validation (orange) and 12 months of test (green), both rolled forward by one year at each iteration.

Each of the three sets has a precise role:

- the **training set** is used to estimate the parameters of the model: the coefficients of a regression, the trees of a forest or the weights of a neural network;
- the **validation set** is used to choose the hyperparameters, which are not learned from the training data: the penalty of the Elastic Net and of the GLM, the number of components of PLS and PCR, the depth of the trees of the Random Forest and of GBRT, or the moment at which the training of a neural network is stopped (early stopping). For each candidate value, the model is estimated on the training set and evaluated on the validation set, and the value giving the best validation performance is kept. The models without hyperparameters, such as the ordinary least squares regressions, do not use it;
- the **test set** is never used to build the model: it only measures its out-of-sample performance.

The models are thus re-estimated once a year, like an investor who would update the model each year with the newly available data. The predictions of the 9 test sets are finally pooled to measure the performance of each model over the whole test period.

The article uses the same principle over a much longer history: 18 years of training (1957–1974), 12 years of validation (1975–1986) and 30 years of test (1987–2016), the validation window also keeping a fixed length, of 12 years, and being rolled forward each year. With only about 17 years of usable data, I had to use much shorter windows.

5.3.3 Evaluation Metrics

All the models are evaluated with the same metrics, computed from their predictions $\widehat{r}_{i,m+1}$ on the test sets. Their hyperparameters are chosen by minimizing the mean squared error on the validation set, which amounts to maximizing the R^2_{oos} defined below on this set.

The main metric is the out-of-sample R^2 of the article, computed on the 9 test sets pooled together and on each test window:

$$R^2_{oos} = 1 - \frac{\sum_{(i,m)\in\text{test}} \left(r_{i,m+1} - \widehat{r}_{i,m+1}\right)^2}{\sum_{(i,m)\in\text{test}} r^2_{i,m+1}}.$$

Unlike the usual R^2, the denominator does not subtract the mean: the benchmark is a forecast of zero, because the historical mean of individual stock returns is too noisy to be a good benchmark. A positive R^2_{oos} means that the model does better than predicting a zero excess return.

I also report the RMSE and the MAE, in percent per month:

$$\text{RMSE} = \sqrt{\underset{(i,m)\in\text{test}}{\text{mean}} \left(r_{i,m+1} - \widehat{r}_{i,m+1}\right)^2}, \qquad \text{MAE} = \underset{(i,m)\in\text{test}}{\text{mean}} \left|r_{i,m+1} - \widehat{r}_{i,m+1}\right|.$$

5.4 Results of the Models

Table 4 gathers the out-of-sample performance of the linear models and of the neural networks, over the 9 test windows pooled together (August 2017 – April 2026). The last two columns recall the R^2_{oos} reported by the article [3], for all the stocks and for the 1 000 largest ones, which are closer to my universe.

	My results			Article: R^2_{oos} (%)	
Model	R^2_{oos} (%)	RMSE (%)	MAE (%)	All	Top 1 000
OLS3	1.62	9.19	6.70	0.16	0.31
PLS	1.40	9.35	6.77	0.27	−0.14
ENet+H	1.30	9.35	6.77	0.11	0.25
GLM+H	1.36	9.35	6.77	0.19	0.14
NN1	0.42	9.39	6.81	0.33	0.49
NN2	−2.67	9.54	6.93	0.39	0.62
NN3	−5.21	9.66	7.06	0.40	0.70
NN4	−45.79	11.37	8.45	0.39	0.67
NN5	−34.79	10.93	8.26	0.36	0.64

Table 4: Pooled out-of-sample performance of each model (RMSE and MAE in percent per month), compared with Table 1 of the article.

The linear models (OLS3, PLS, ENet+H and GLM+H) all reach an R^2_{oos} between 1.3 and 1.6 %, and the simplest one, OLS3, is the best. This level is close to that of a constant forecast, about 1.5 % (Section 5.5.5): the characteristics add little to what the average return of the universe already provides. This is also why all these values are much higher than those of the article, without meaning that the models are better. The neural networks with two hidden layers or more have a negative R^2_{oos}, which generally gets worse with depth, down to −46 % for NN4; only NN1 stays slightly positive. This ranking is the opposite of the article's, in which the neural networks do best. A likely explanation is the size of my sample: with about 450 stocks per month over 17 years, instead of more than 6 000 over 60 years, the most flexible models have far fewer observations to learn from and mostly fit noise. The rest of this part therefore focuses on OLS3.

5.5 A Closer Look at OLS3

Among my results, the best one is OLS3: over the whole out-of-sample period, it reaches a pooled out-of-sample R^2 of about 1.6 %. Rather than simply reporting this number, I wanted to understand where it comes from: what the model actually computes, what the data it receives look like, which of its three variables carries the signal and, given what I had found with the first article, whether this result could be produced by a data leak.

5.5.1 The OLS3 Model

In the article, OLS3 is an ordinary linear regression restricted *a priori* to three stock characteristics: size, value and momentum. These are the most documented and most robust return predictors of the empirical literature, and using only three of them leaves very little room for overfitting. OLS3 is therefore not meant to be a competitive model but a reference point: a richer model is only interesting if it does better than OLS3.

For a stock i in month m, the predicted excess return of the following month is

$$\widehat{r}_{i,m+1} = \beta_0 + \beta_{mvel1}\,\mathrm{mvel1}_{i,m} + \beta_{bm}\,\mathrm{bm}_{i,m} + \beta_{mom}\,\mathrm{mom}_{i,m},$$

where mom stands for the `mom12m_skip1m` variable. The regression is *pooled*: all the (stock, month) observations of the training set are stacked and regressed together, without any stock-specific or month-specific fixed effect. A single set of four coefficients is thus estimated per training window and reused as is for every stock of the test period.

I also trained the OLS3+H variant of the article, in which the squared loss is replaced by the Huber loss

$$H(x;\xi) = \begin{cases} x^2 & \text{if } |x| \le \xi, \\ 2\xi|x| - \xi^2 & \text{if } |x| > \xi, \end{cases}$$

which is quadratic for small residuals and linear for large ones, and therefore less sensitive to extreme returns. Following the Internet Appendix of the article, ξ is not tuned on the validation set but set to the 99.9th percentile of the absolute residuals of a first OLS fit on the training set. This threshold only affects about 0.1 % of the training observations, so the two models are almost indistinguishable: the pooled R^2_{oos} goes from 1.615 % (OLS3) to 1.622 % (OLS3+H), with the same RMSE (9.49 %) and MAE (6.85 %). The coefficient and performance figures below come from the OLS3+H pipeline, but their squared-loss counterparts are visually identical.

5.5.2 The Three Variables

The three characteristics are computed from public sources only, month by month and stock by stock:

$$\begin{aligned} \mathrm{mvel1}_{i,m} &= \ln\left(C_{i,m} \times S_{i,m}\right), \\ \mathrm{bm}_{i,m} &= \frac{E_{i,m}}{C_{i,m} \times S_{i,m}}, \\ \mathrm{mom}_{i,m} &= \frac{AC_{i,m-1}}{AC_{i,m-12}} - 1, \end{aligned}$$

where $C_{i,m}$ is the unadjusted Yahoo Finance closing price of the last trading day of month m, $AC_{i,m}$ the same price adjusted for splits and dividends, $S_{i,m}$ the number of shares outstanding and $E_{i,m}$ the stockholders' equity, both taken from the SEC. `mvel1` is the logarithm of the market capitalization (size), `bm` the book-to-market ratio (a high value designates a "value" stock, a low value a "growth" stock) and `mom12m_skip1m` the return accumulated over the eleven months preceding the most recent one.

OLS3 does not receive the raw characteristics but their monthly cross-sectional ranks from the panel ranked, where -1 corresponds to the weakest stock of the month on a variable and $+1$ to the strongest.

5.5.3 Out-of-Sample Performance

Figure 5 shows the performance of each test window. The R^2_{oos} is positive on 7 of the 9 windows. The two negative windows are August 2018 – July 2019 (about $-1\,\%$) and August 2021 – July 2022 (about $-3.6\,\%$), and the best one is August 2020 – July 2021 ($+8.2\,\%$), a year of strong market rebound after the Covid crash. The RMSE varies a lot from one window to another (from 6.9 to 11.3 %) and is highest in the most turbulent periods.

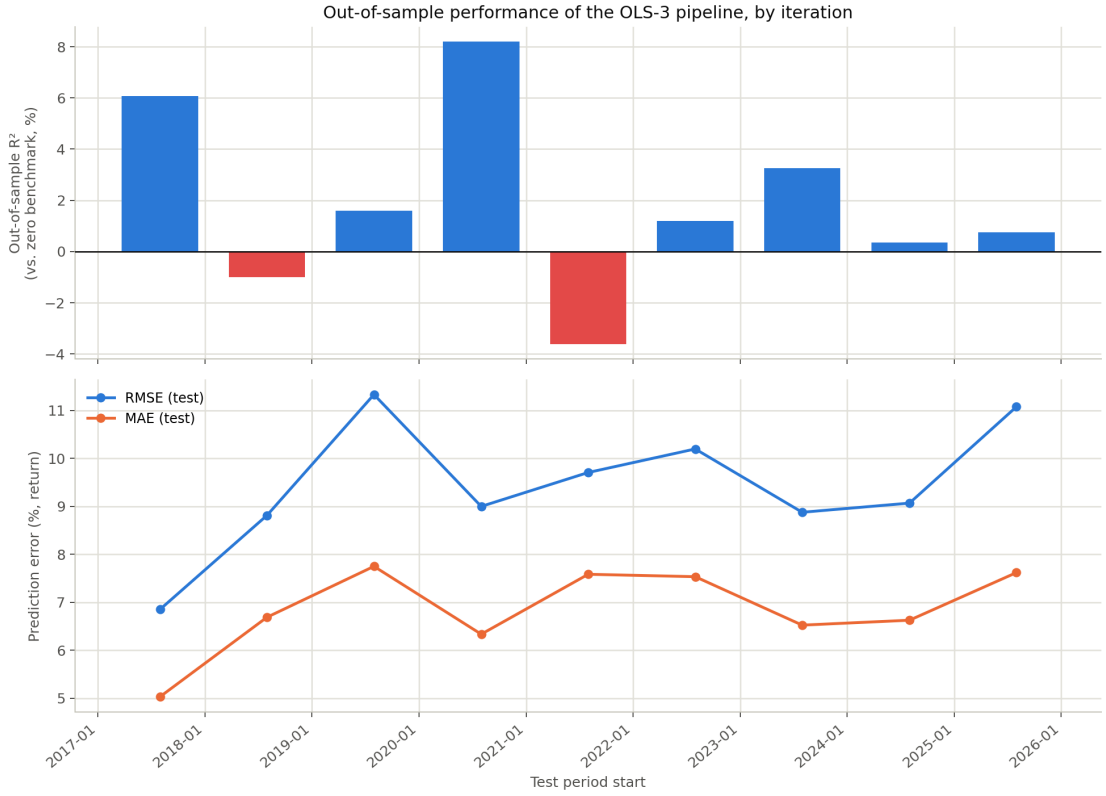


Figure 5: Out-of-sample R^2 (top, blue if positive and red if negative) and RMSE/MAE (bottom) of each test window.

5.5.4 Correlation Between the Ranked Features and the Target

In order to understand better the data I had and their influence on the model, I first calculated the correlation between the ranked features and the target (Figure 6). Between the features, the strongest correlation links `mvel1` and `bm` (-0.26): the largest companies tend to have a low book-to-market, which is economically expected since large capitalizations are often growth stocks, valued well above their book value. Value and momentum are slightly anti-correlated (-0.18), which is consistent with two distinct and complementary investment styles, while size and momentum are almost uncorrelated (0.03). These correlations are low enough for the three coefficients to be estimated without any collinearity problem.

The most important information of this figure is however in its last row: the correlation of each feature with the target lies between -0.04 and 0.00. Taken individually, none of the three variables has a visible linear relationship with next month's return, which already suggests that the R^2_{oos} of OLS3 can only be modest.

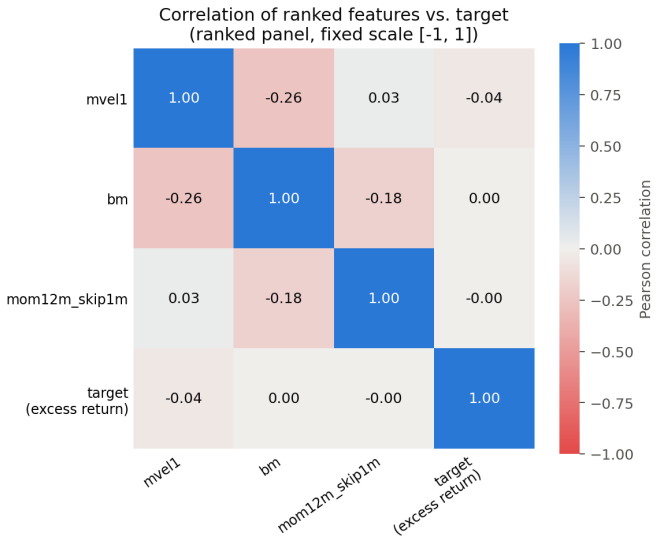


Figure 6: Pearson correlations between the three ranked features and the target (excess return of the following month).

5.5.5 What the Model Learns

Figure 7 shows the three coefficients estimated at each of the nine iterations. They tell three different stories.

- `mvel1` (size): the coefficient is always negative and remarkably stable, between -0.0070 and -0.0079. The smallest companies of the universe receive a higher predicted return: this is the size effect. Between the smallest and the largest stock of a month, it creates a gap of about $2 \times 0.0075 \approx 1.5$ percentage points of predicted monthly return.
- `bm` (value): the coefficient is always negative too, but 2.5 to 13 times smaller than the size coefficient and much less stable (from -0.0006 in the first iteration to -0.0029 around 2021, then back to about -0.0013). Its sign is the opposite of the classical value effect, which is consistent with the well-known underperformance of value stocks during the 2010s.
- `mom12m_skip1m` (momentum): the coefficient oscillates around zero and changes sign from one iteration to another (between -0.0011 and $+0.0009$). It is the least reliable of the three signals.

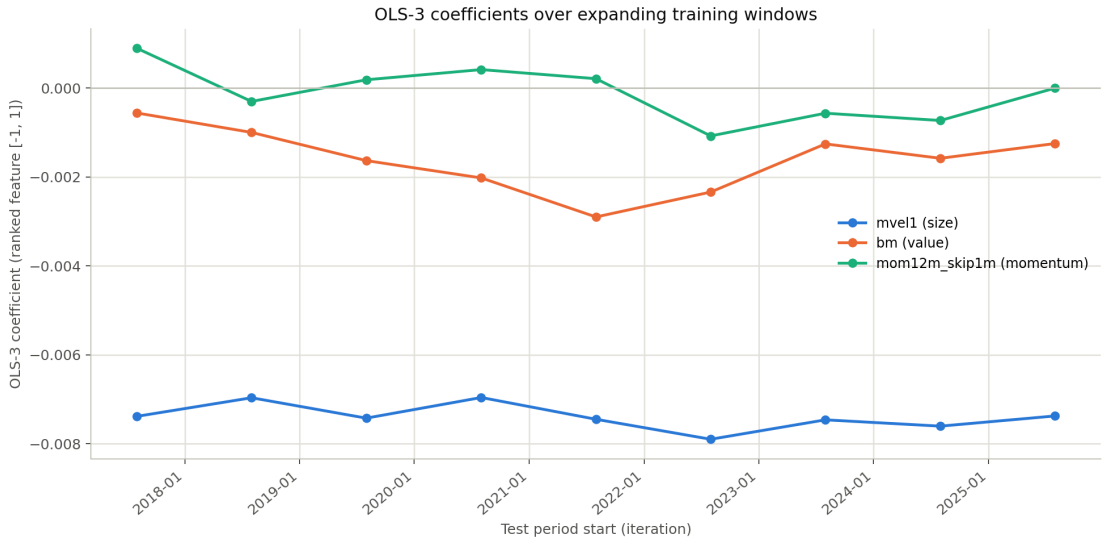


Figure 7: OLS3 coefficients of the three ranked features, for each expanding training window.

Table 5 decomposes one prediction of the first iteration, for the ticker `A`. Each term is simply the product of a coefficient and the rank of the stock, and their sum exactly reproduces the prediction written by the pipeline. This example shows a point that holds for all the predictions: the intercept (1.80 %) is much larger than the contributions of the three characteristics, which only move the prediction by less than a tenth of a percentage point here.

Term	Coefficient	Rank	Contribution
Intercept	0.017998	—	0.017998
`mvel1`	−0.007647	0.114094	−0.000872
`bm`	−0.000593	−0.307865	+0.000182
`mom12m_skip1m`	0.000715	0.393548	+0.000281
Sum = prediction			**0.017589**

Table 5: Decomposition of an OLS3 prediction (squared loss, iteration 1, ticker `A`). The realized excess return that month was −0.008935, hence a residual of −0.026523.

Since the ranks are centered on zero, the intercept is almost equal to the average excess return of the training sample, between 1.4 and 1.8 % per month depending on the iteration. This gave me an important key to read the R^2_{oos}. I recomputed it on the 49 754 test predictions with, as the only forecast, the intercept of each iteration: this constant forecast alone already reaches $R^2_{oos} = 1.50\,\%$, compared with 1.615 % for the full OLS3. In other words, most of the performance of OLS3 comes from the fact that it predicts a positive average return rather than zero. The three characteristics only add about 0.1 point of R^2, almost entirely through `mvel1`.

5.5.6 Predicted Versus Realized Returns

Figure 8 shows the test predictions. It is, in my opinion, the most telling figure about OLS3. To read it, recall that the features are ranks on $[-1, 1]$: a coefficient β_x is read as a sensitivity to the *position* of the stock in the cross-section of the month, and the largest effect a variable can have on its own is $2|\beta_x|$, the gap between the two most extreme stocks of the month. The predicted returns stay confined to a narrow band, between about 0.5 and 2.7 %, which is consistent with

these maximum effects given the coefficients above, whereas the realized returns spread from less than $-20\,\%$ to more than $+25\,\%$. The perfect-prediction line $y = x$ is almost flat compared with the cloud of points. By construction, OLS3 can only produce low-variance forecasts: it captures a real but economically thin signal, and never the bulk of the variance of the realized returns.

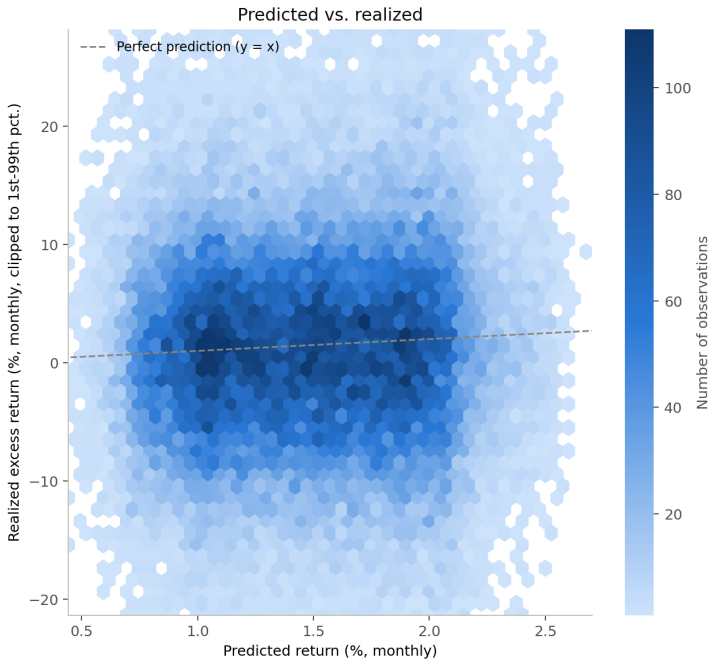


Figure 8: Predicted versus realized excess returns (density of observations; realized returns clipped to the 1st–99th percentiles). The dashed line is the perfect prediction $y = x$.

5.6 Checking for Data Leakage

Given what I had found with the first article, I wanted to check that this result was not inflated by information coming from the future. I carried out two tests.

The first test concerns the SEC availability lag. I rebuilt the whole raw panel, then the ranked panel, and retrained the whole backtest for 11 values of the lag L, from +30 days (the value used in production) down to -120 days. A negative lag makes an accounting figure available *before* its actual filing date, which simulates a leak. Only the variables built from SEC data are affected: the share count, hence `mvel1`, and the stockholders' equity, hence `bm`; prices, momentum and the target are unchanged. If the production pipeline benefited from a leak through the SEC data, making these data available even earlier should increase the R^2_{oos}. Figure 9 shows the opposite: the pooled R^2_{oos} slightly and almost monotonically decreases, from $1.622\,\%$ at +30 days to $1.562\,\%$ at -120 days. At the same time, the average `bm` coefficient loses about $37\,\%$ of its magnitude (from -0.00161 to -0.00103): the signal carried by `bm` weakens when its availability is moved earlier, it does not strengthen.

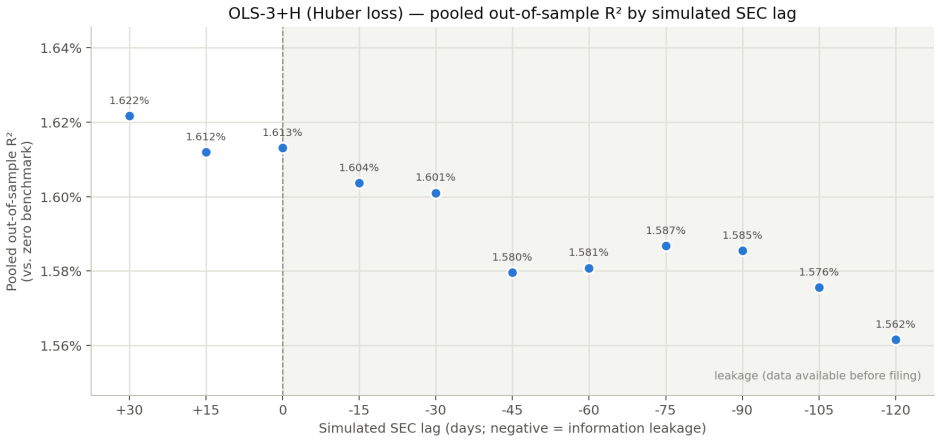


Figure 9: Pooled out-of-sample R^2 of OLS3+H as a function of the simulated SEC availability lag. Negative lags (shaded area) make accounting data available before their filing date.

The second test deliberately injects a leak into the features, in order to see what a real leak would look like. Each variable can be shifted by K months into the future,

$$\text{leaked_value}(i,t) = \text{true_value}(i,t+K),$$

where $K = 0$ means no leak. I tested all the $7^3 = 343$ combinations of $K \in \{0,\dots,6\}$ for the three variables. To keep the computation fast, this test uses a single train/validation/test split instead of the full backtest, which is why its no-leak reference (1.365 %) differs from the pooled value above.

The distribution of the 343 values of R^2 (Figure 10) is clearly bimodal: 98 combinations lie between 0.8 and 1.9 %, around the reference, and the 245 others lie between 5.5 and 6.7 %, with nothing in between. These two numbers are exactly 2×49 and 5×49: the first group gathers the combinations in which the momentum is shifted by 0 or 1 month, the second those in which it is shifted by 2 months or more, whatever the leaks of the two other variables.

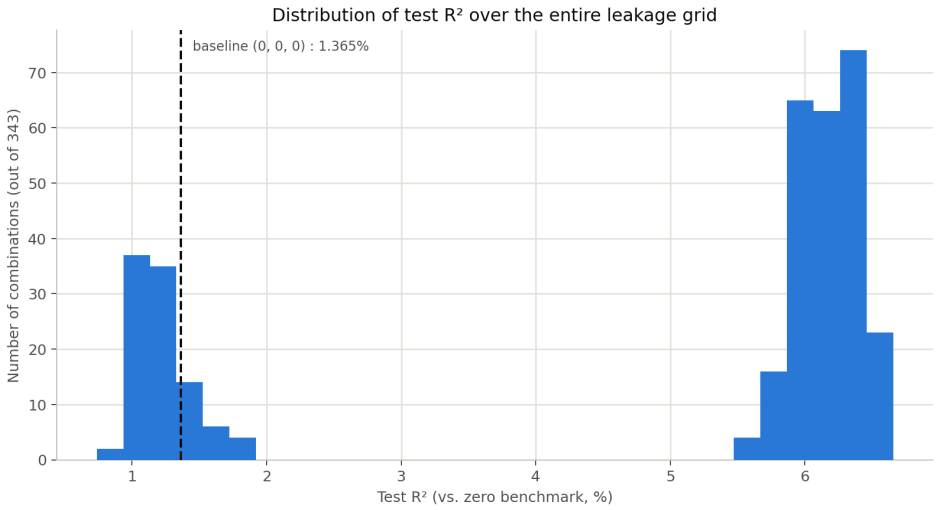


Figure 10: Distribution of the test R^2 over the 343 leak combinations. The dashed line is the no-leak reference.

Figure 11 isolates the effect of each variable, the two others being kept without leak. When the momentum is shifted by $K = 2$ months, the R^2 jumps from 1.36 to 6.37 % and its coefficient is multiplied by about 33 (from +0.0009 to +0.030). The `bm` coefficient already becomes much more negative at $K = 1$ (from −0.0006 to −0.0078), but the R^2 barely moves (at most 1.83 %). The `mvel1` coefficient moves toward zero and the R^2 even decreases slightly.

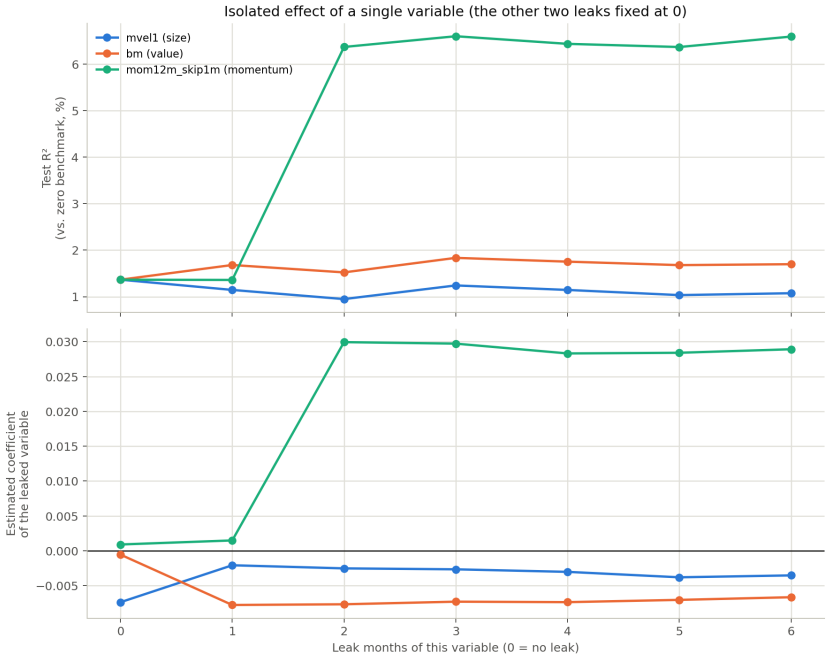


Figure 11: Effect of leaking a single variable by K months, the two others being kept without leak: test R^2 (top) and estimated coefficient of the leaked variable (bottom).

These thresholds can be explained directly by the formulas of the variables. The target of month t is the return realized during month $t+1$. A value of `mvel1` or `bm` taken at $t+1$ contains the closing price $C_{i,t+1}$, that is, the price at the end of the very month whose return is predicted: this is why their coefficients change as early as $K = 1$. However, the rank of a stock on size or book-to-market mostly reflects persistent differences between companies, which a single month of price variation hardly changes, so the R^2 barely benefits from it. For the momentum, because of the skipped month, the value taken at $t+1$ is $AC_{i,t}/AC_{i,t-11} - 1$ and does not yet contain the return of month $t+1$; the value taken at $t+2$, $AC_{i,t+1}/AC_{i,t-10} - 1$, does. The jump appears exactly at $K = 2$, and it is large because the momentum is itself a sum of monthly returns, so adding the return of the target month strongly changes the rank of the stock.

These two tests are complementary. The second one shows that a real leak of the target month information is immediately visible, with an R^2 multiplied by about 5. The first one shows that nothing of this kind happens in the production pipeline, where moving the SEC data earlier only has an effect below 0.1 point of R^2, and in the direction opposite to a leak. Within the limits of these tests, the result of OLS3 is therefore not explained by a leak in the construction of the features.

5.7 Discussion and Limits

This analysis changed the way I read the result of OLS3. Its R^2_{oos} of 1.6 % mostly comes from its intercept, that is, from predicting a positive average return rather than zero; the three char-

acteristics add only about 0.1 point, almost entirely through the size effect. The model does not suffer from a data leak in its features, but it only produces a thin signal compared with the noise of monthly returns.

This result must not be directly compared with the 0.16 % reported by the article for OLS3+H, because the setting is very different. The article uses all the stocks of the CRSP database (nearly 30 000 companies) and a test period of 30 years (1987–2016), whereas I use about 500 large companies and a test period of 9 years (2017–2026), with Yahoo Finance and SEC data instead of CRSP and Compustat. Moreover, my universe is made of the *current* constituents of the S&P 500, that is, companies that are known today to have survived and grown. This survivorship bias, is a bias that the original article mention to avoid, is a form of look-ahead that the two tests above cannot detect. I cannot measure it with my data, but it may explain part of the high average return of the universe, and therefore part of the R^2 obtained by the intercept alone.

6 Conclusion

This internship set out to measure to what extent machine learning can predict the future returns of stocks. The two articles that predict daily returns from price history alone could not be reproduced: my out-of-sample R^2 stays close to zero or negative, and a simple one-day shift of the input variables is enough to reach the near-perfect scores reported by the first one. With the public reproduction of Gu, Kelly and Xiu, on the S&P 500 stocks and 21 characteristics, the linear models reach an R^2_{oos} of 1.3 to 1.6 % and OLS3 is the best, while the deeper neural networks do worse than a zero forecast. However, a constant forecast alone already reaches 1.50 %: the characteristics only add about 0.1 %, mostly through the size effect, and the leakage tests show that this result is not produced by a shift in time in the construction of the features. On my data, machine learning therefore predicts returns only to a very small extent, and the main lesson of this work is methodological: a strict out-of-sample protocol is needed before trusting any predictive performance.

Despite numerous challenges, ranging from unsuitable publications to the complexities of data collection and interpretation, this internship enabled me to gain knowledge in machine learning and finance. Reading research papers allowed me to familiarize myself with artificial intelligence; I am now able to understand the pros and cons of various models, the implications of overfitting, and ways to minimize methodological bias. On the financial side, I deepened my understanding of corporate financial indicators and their market correlations. I also gained insight into the inner workings of the research world and the interpersonal dynamics within that community. I recognize that research does not always aim for immediate results and that certain environments—such as the French research landscape—prioritize publication quality. Consequently, and despite a disappointing experience, I cannot yet say whether a career in research is the right path for me.

Glossary

AI Artificial Intelligence. 3, 4

AMEX American Stock Exchange. 10

API Application Programming Interface. 11

CAC Cotation Assistée en Continu. 6

CRSP Center for Research in Security Prices. 10

DL Deep Learning. 4

EDGAR Electronic Data Gathering, Analysis, and Retrieval. 10

ENet Elastic Net. 15

FTSE Financial Times Stock Exchange. 5

GBRT Gradient Boosted Regression Trees. 15

GLM Generalized Linear Model. 15

JSON JavaScript Object Notation. 11

KU Kasetsart University. 4

LSTM Long Short-Term Memory. 5

MAE Mean Absolute Error. 8

MAPE Mean Absolute Percentage Error. 5

ML Machine Learning. 4

MSE Mean Squared Error. 8

NASDAQ National Association of Securities Dealers Automated Quotations. 10

NYSE New York Stock Exchange. 10

OLS Ordinary Least Squares. 15

PCR Principal Components Regression. 15

PLS Partial Least Squares. 15

RF Random Forest. 15

RMSE Root Mean Squared Error. 5

SEC Securities and Exchange Commission. 3

SIC Standard Industrial Classification. 13

XBRL eXtensible Business Reporting Language. 10

Bibliography

[1] Shaghayegh Abolmakarem et al. "A multi-stage machine learning approach for stock price prediction: Engineered and derivative indices". In: *Intelligent Systems with Applications* 24 (2024), p. 200449. DOI: 10.1016/j.iswa.2024.200449.

[2] Nur Faid Prasetyo, Wina Witanti, and Asep Id Hadiana. "Forecasting stock returns using long short-term memory (LSTM) model based on inflation data and historical stock price movements". In: *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)* 9.3 (2025), pp. 453–464. DOI: 10.29207/resti.v9i3.6422.

[3] Shihao Gu, Bryan Kelly, and Dacheng Xiu. "Empirical asset pricing via machine learning". In: *The Review of Financial Studies* 33.5 (2020), pp. 2223–2273. DOI: 10.1093/rfs/hhaa009.

[4] Markus Leippold, Qian Wang, and Wenyu Zhou. "Machine learning in the Chinese stock market". In: *Journal of Financial Economics* 145.2 (2022), pp. 64–82. DOI: 10.1016/j.jfineco.2021.08.017.

[5] U.S. Securities and Exchange Commission, Division of Economic and Risk Analysis. *Scaling and Tagging Errors for Entity Common Stock Shares Outstanding and Common Stock Shares Outstanding*. Oct. 6, 2022. URL: https://www.sec.gov/newsroom/whats-new/osd-announcement-2210-dqreminder-entitycommonstocksharesoutstanding (visited on 09/26/2026).

[6] Roger S. Debreceny et al. "Flex or Break? Extensions in XBRL Disclosures to the SEC". In: *Accounting Horizons* 25.4 (2011), pp. 631–657. DOI: 10.2308/acch-50068.

[7] Roger Debreceny et al. "Does it add up? Early evidence on the data quality of XBRL filings to the SEC". In: *Journal of Accounting and Public Policy* 29.3 (2010), pp. 296–306. DOI: 10.1016/j.jaccpubpol.2010.04.001.

Appendices

Appendix 1: Evaluation Form Completed by the Supervisor

ENSTA | INSTITUT POLYTECHNIQUE DE PARIS

RAPPORT D'EVALUATION
ASSESSMENT REPORT

Merci de retourner ce rapport par courrier ou par voie électronique en fin du stage à :
At the end of the internship, please return this report via mail or email to:

ENSTA – Bureau des stages - 2 rue François Verny - 29806 BREST cedex 9 – FRANCE
☎ 00.33 (0) 2.98.34.87.70 / stages.brest@ensta.fr

I - ORGANISME / *HOST ORGANISATION*

NOM / *Name*_____ Kasetsart University_____

Adresse / *Address*_____Department of Computer Engineering, Kasetsart University 50 Ngamwongwan Rd, Chatuchak Bangkok 10900 Thailand_____

Tél / *Phone (including country and area code)*_____Tel. (+66) 0-2797-0999 ext 1403-04 1450___

Nom du superviseur / *Name of internship supervisor*
_____Piya Limcharoen_____
Fonction / *Function*_____Lecture_____

Adresse e-mail / *E-mail address*_____piya.lim@ku.th_____

Nom du stagiaire accueilli / *Name of intern* Clément Sartre

II - EVALUATION / *ASSESSMENT*

Veuillez attribuer une note, en encerclant la lettre appropriée, pour chacune des caractéristiques suivantes. Cette note devra se situer entre **A (très bien)** et **F (très faible)**
*Please attribute a mark from **A (excellent)** to **F (very weak).***

.1 MISSION / *TASK*

❖ La mission de départ a-t-elle été remplie ? A **B** C D E F
Was the initial contract carried out to your satisfaction?

❖ Manquait-il au stagiaire des connaissances ? oui/*yes* **non/*no***
Was the intern lacking skills?

Si oui, lesquelles ? / *If so, which skills?* _____

.2 ESPRIT D'EQUIPE / *TEAM SPIRIT*

❖ Le stagiaire s'est-il bien intégré dans l'organisme d'accueil (disponible, sérieux, s'est adapté au travail en groupe) / *Did the intern easily integrate the host organisation? (flexible, conscientious, adapted to team work)*
A **B** C D E F

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here*_____
_____

Version du 05/04/2019

.3 COMPORTEMENT AU TRAVAIL / *BEHAVIOUR TOWARDS WORK*

Le comportement du stagiaire était-il conforme à vos attentes (Ponctuel, ordonné, respectueux, soucieux de participer et d'acquérir de nouvelles connaissances) ?
Did the intern live up to expectations? (Punctual, methodical, responsive to management instructions, attentive to quality, concerned with acquiring new skills)?

A **B** C D E F

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* _____

_____

.4 INITIATIVE – AUTONOMIE / *INITIATIVE – AUTONOMY*

Le stagiaire s'est –il rapidement adapté à de nouvelles situations ? A B C D E F
(Proposition de solutions aux problèmes rencontrés, autonomie dans le travail, etc.)

Did the intern adapt well to new situations? A **B** C D E F
(eg. suggested solutions to problems encountered, demonstrated autonomy in his/her job, etc.)

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* _____

_____

.5 CULTUREL – COMMUNICATION / *CULTURAL – COMMUNICATION*

Le stagiaire était-il ouvert, d'une manière générale, à la communication ? A **B** C D E F
Was the intern open to listening and expressing himself /herself?

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* _____

_____

.6 OPINION GLOBALE / *OVERALL ASSESSMENT*

❖ La valeur technique du stagiaire était : A **B** C D E F
Please evaluate the technical skills of the intern:

III - PARTENARIAT FUTUR / *FUTURE PARTNERSHIP*

❖ Etes-vous prêt à accueillir un autre stagiaire l'an prochain ?

Would you be willing to host another intern next year? **oui/*yes*** non/*no*

Fait à _____, le _____
In _Kasetsart University, Bangkok__, on__Starting from April (same period as this year)__

Signature Entreprise Piya Digitally signed by Piya Limcharoen DN: CN=Piya Limcharoen, O=CPE KU, OU=CPE KU, E=piya.lim@ku.th Reason: I am the author of this document Location: Date: 2026.09.08 09:26:14+07'00' Foxit PDF Editor Version: 2026.1.3 Signature stagiaire
Company stamp _____ *Intern's signature*

Merci pour votre coopération
We thank you very much for your cooperation