KeAi
CHINESE ROOTS
GLOBAL IMPACT

Available online at www.sciencedirect.com

ScienceDirect

The Journal of Finance and Data Science 8 (2022) 86–104

http://www.keaipublishing.com/en/journals/jfds/



Performance attribution of machine learning methods for stock returns prediction☆

Stéphane Daul*, Thibault Jaisson, Alexandra Nagy

Pictet Asset Management, Acacias 60, 1211 Geneva, Switzerland

Received 15 December 2021; revised 28 February 2022; accepted 5 April 2022
Available online 14 April 2022

Abstract

We analyze the performance of investable portfolios built using predicted stock returns from machine learning methods and attribute their performance to linear, marginal non-linear and interaction effects. We use a large set of features including price-based, fundamental-based, and sentiment-based descriptors and use model averaging in the validation procedure to get robust out-of-sample predictions. We find that the superiority of regression trees and neural networks comes from two points: their strong regularization mechanism and their capacity to capture interaction effects. The non-linear component of the marginal predictions on the other hand has no predictive power. Thanks to our methodology, we manage to isolate and study in detail the interaction component. We find that it has significative long term performance independent from the linear modeling and is stable through time.
© 2022 The Authors. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Keywords: Machine learning; Return prediction; Performance attribution; Cross sectional returns; Lasso; Boosted trees; Neural networks

1. Introduction

Machine learning is a domain of data science essentially concerned by the prediction of outcomes based on realized data. It has had an enormous success in many non-financial fields and is now increasingly applied to finance. In contrast to pure statistical modeling the effort is concentrated more on the prediction power and less on the understanding of the outcome.

1.1. Related work

Recently many research articles (Rasenkhschaffe and Jones[1]; Gu et al[2]; Choiet al[3]; Leung et al[4]; Li et al[5]) have focused on the prediction of stock returns based on their features. Table 1 summarizes the basic settings of all these

☆ We are grateful to Rémy Cottet, Reda Jürg Messikh and two anonymous referees for their useful comments. The responsibility for the information and views expressed in this research article lies entirely with the authors.
* Corresponding author.
E-mail addresses: sdaul@pictet.com (S. Daul), tjaisson@pictet.com (T. Jaisson), anagy@pictet.com (A. Nagy).
Peer review under responsibility of KeAi.

https://doi.org/10.1016/j.jfds.2022.04.002
2405-9188/© 2022 The Authors. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Table 1
Basic settings of recent research on stock returns prediction.

Reference	Coverage	Date range	Average # stocks per month	Number of features
Rasenkhschaffe and Jones[1]	World	1994–2016	5900	194
Gu et al[2]	US	1957–2016	6200	94 (financial) + 9 (macro)
Choi et al[3]	World	1957–2017	3000	12
Leung et al[4]	World	1991–2018	3300	20
Li et al[5]	SP 500	1992–2020	500	16

studies. They all find that trees and neural networks are the most efficient predictive methods and outperform (generalized) linear models. These two approaches allow for marginal non-linearity and interaction between features.

With the applications of non-linear machine learning techniques to virtually all applied sciences in recent years it has become paramount to be able to interpret the output of such models, see for example Patir et al[6] and references therein. Because of the need to understand where trades are coming from, financial machine learning models have been particularly analyzed by such methods. For example, De Prado[7] presents many feature importance methods while Li et al[5] splits the predicted return in three components, a linear component, a marginal non-linear component and an interaction component.

1.2. Contributions and findings

Our contributions to the field of stock returns prediction with features are three-fold. First, we take an investor viewpoint by concentrating on highly liquid international stocks and also extend the features to include sentiment-based descriptors in addition to price and fundamental-based descriptors. Second, we choose the validation procedure while keeping in mind that our ultimate goal is to build long short portfolios based on the return predictions. Third, we modify the performance attribution framework from Li et al[8] to handle very large sample sizes and large number of features.

We find that boosted trees and neural networks make use of the features homogeneously and are superior to linear models as shown in Rasenkhschaffe and Jones[1]; Gu et al[2]; Choiet al[3]; Leung et al[4]; Li et al[5] The performance attribution of these methods shows that a large part of the prediction is captured by a linear component that is similar across the different approaches. This linear component is close to the one obtained by linear models with careful regularization, combining a L_1 constraint with model averaging. We consider regularization not only as the addition of a penalty to the objective function but as the whole process of adding information or modifying the learning algorithm to prevent overfitting. We also find that the non-linear marginal component has no significant predictive power. This is consistent with the results of Gu et al[2] which show that generalized linear models which allow for such marginal non linearities do not outperform linear models. Finally, we show that the interaction term between features increases the forecasting performance of the models. When isolating this interaction term, we find a signal which is independent from the linear models and which has positive performance and is stable through time.

1.3. Organization of the article

The paper is organized as follows. In Section 2 we define our data universe and the features that we use. In Section 3, we present the methodological tools used to fit our models, performance attribution and feature importance. The empirical results are shown in Section 4. We conclude on how to apply this in practice in Section 5.

2. Data

2.1. Base universe

We perform all our analysis on a proprietary universe of liquid stocks that we build with DataStream from 1990 to 2021 in 23 developed countries. The forecast horizon and data frequency is monthly. The starting universe had about 7600 stocks on average per month. We keep only the largest stocks that sum to 80% of the market cap. This final universe contains between 1200 and 2000 stocks per month on which the median daily liquidity goes from 10M USD at the beginning of the period to 50M USD at the end. The high liquidity of this universe makes investment strategies

based on the signals developed here closer to implementation compared to analysis performed on tens of thousands of micro stocks. Refer to Appendix A for basic description of this universe.

2.2. Input features

For all the stocks in our universe (in total we have more than 600'000 stocks × months pairs in our universe), we compute $K = 205$ features that we split in three categories:

- 33 *price-based features* that are computed from price changes and traded volumes. For example: the past return of the stock, its volatility, its beta, liquidity, liquidity surprise or activity (traded volume over market capitalization). The data used to compute these features comes from DataStream and is lagged by at least one business day.
- 30 *fundamental-based features* that are computed from company reported fundamental data. For example: book to price, earnings to price, return on assets, asset growth or sales growth. We also include the change in these features over the last year. Note that these features are vanilla ones, we do not include sophisticated financial ratios which were recently studied. The data used to compute these features comes from Worldscope and is lagged by three months.
- 142 *sentiment-based features* on sell side analyst data. For example: we compute the average recommendation and price targets of companies, the forward earnings to price ratio and return on assets, as well as the number of analysts who have revised upward and downward their earnings predictions. The data used to compute these features comes from IBES and is lagged by one business day.

The complete description of features is given in Appendix B.1.

We also split features based on their half-life (defined as the time it takes for their autocorrelation to go below 0.5). When the half life is smaller than three months we categorize the feature as fast, and slow otherwise. We have 75 fast signals and 130 slow signals. The exhaustive list in given in Appendix B.2.

Each month we then rank the features cross sectionally (to get a score between 0 and 1) and center them between -0.5 and $+ 0.5$. When we do not have data, we set the feature to 0.

2.3. Targets

Let us note the monthly realized total return by $r_{i,t+1}$ where the months are indexed by $t = 1, \ldots, T$ and the stocks by $i = 1, \ldots, N_t$. In this study we choose to eventually build investable portfolios based on stock specifics. We believe it is difficult to time market based on stocks specifics only, hence we want to avoid sector or market timing bias. On the other hand, we want to capture all traditional factor exposure of the stocks. Therefore we will not predict $r_{i,t+1}$ as such, but the stock-specific residual return defined as

$$r^*_{i,t+1} = r_{i,t+1} - \beta_{i,t} r_{M,t+1}, \tag{1}$$

where the performance of the market is $r_{M,t+1}$, and the stock beta $\beta_{i,t}$ is estimated using one year of past daily performances. This specific return is then centered by sector, divided by its volatility $\sigma_{i,t}$ (estimated using one year of past daily total returns),

$$\tilde{r}_{i,t+1} = \frac{r^*_{i,t+1} - \bar{r}_{\text{sector},t+1}}{\sigma_{i,t}} \tag{2}$$

and trimmed for outliers (below the 1% and above the 99% quantiles) at each date. This will consist of our target and the sectors we consider are the GICS sectors.

3. Methodology

3.1. Prediction models

The prediction of the scaled stock-specific return in a general form can be written as

$$\widehat{r}_{i,t+1} = \Phi(z_{i,t}), \tag{3}$$

where $z_{i,t}$ is a vector of K features for stock i at time t. The function Φ does not depend on i or t, and is the same for all sectors.

3.1.1. Linear model with penalization

Our first model is the simple linear regression model estimated using ordinary least square (OLS)

$$\Phi(z_{i,t}, \theta) = z_{i,t}^{'}\theta, \tag{4}$$

with the parameters θ. This model does not allow for any non-linear effects or interactions between features. It is well-known to have poor out-of-sample prediction power when the number of features is large and the signal to noise ratio is low.

One can regularize this model by adding a penalty term to the original loss function. The choice of this penalty is not unique, here we choose to add the popular L_1 penalization to the loss function:

$$\sum_{i,t}\left(\tilde{r}_{i,t+1} - z_{i,t}^{'}\theta\right)^2 + \lambda\sum_{k}|\theta_k|, \tag{5}$$

with the constraint parameter λ. The implementation of the penalized regression is based on the LARS algorithm from Efron et al[9] and the procedure to optimize the hyperparameter λ is described in Section 3.2. This final model will be described as LASSO in the rest of the article.

3.1.2. Boosted regression trees

The first machine learning method allowing non linearity and interactions between the features that we consider are the regression trees. Formally, the prediction of a simple tree with L leaves (or partitions of the data) is given by

$$\Phi(z_{i,t}, \kappa, \{C_1, \ldots, C_L\}) = \sum_{\ell}\kappa_\ell \mathbf{1}_{\{z_{i,t}\in C_\ell\}} \tag{6}$$

where C_ℓ is one of the L partitions of the data. The constant κ_ℓ associated to C_ℓ is the sample average of the outcomes of the data in the partition. The training of a regression tree consists of finding the partitions that minimize the mean squared error. Trees are very flexible and give much freedom to the data, and hence are prone to overfitting. Different regularization methods have been introduced to increase the out-of-sample power and we use gradient boosted regression trees. This is an ensemble regularization which combines forecasts from many simple trees, see for example Hastie et al[10] for more details. This method is very greedy in computation time and for our purposes we turn to the recently developed light GBM algorithm from Ke et al.[11] This algorithm allows also further regularization such as bagging or imposing a minimum data per leaves.

The list of hyperparameters and the range we considered are given in Appendix C.1 while the procedure to optimize the hyperparameters is described in Section 3.2. This model will be described as LGBM in the remaining of the article.

3.1.3. Neural networks

The second non-linear method that we study is the widely used feed-forward neural network (Rasekhschaffe and Jones).[a] A feed-forward neural network consists of multiple layers. The first layer takes the vector of K features $z_{i,t}$ as input and its last layer returns the prediction. In between we have H hidden layers. The model is the composition of $H + 2$ layers of form

[a] See for example Dixon[12] for an introduction.

$$\sigma^{(\ell)}\left(W^{(\ell)}x + b^{(\ell)}\right), \tag{7}$$

where x is the output of the previous layer (or the input for the first layer), the matrix $W^{(\ell)}$ and the scalar $b^{(\ell)}$ are the parameters of the layer and $\sigma^{(\ell)}$ a non-linear activation function. In our case, we use the *ReLU* function. The training of our models will be done using the mean-squared error as the loss function and the Adam optimizer Kingma et al[13] with default parameters of PyTorch. To prevent overfitting we will use early stopping and ensemble averaging as the regularization technique.

The list of hyperparameters and the range we considered are given in Appendix C.2 while the procedure to optimize the hyperparameters is described in Section 3.2. This model will be described as NN in the remaining of the article.

3.2. Training-validation-prediction

All of our models have hyperparameters which need to be tuned using one or more validation sets that are distinct from the training and prediction sets. Let us note however, that the most common validation procedures have not been developed for financial applications. Such data has the property that it is non stationary, it is indexed by a cross-sectional and a time component, and most importantly it is characterized by a low signal-to-noise ratio.

In our approach, to tune the hyperparameters we only select the dates for the training/validation sets and then include the whole cross section for each chosen time. We could choose to take the validation set chronologically just after the training set. However the period of validation could be totally different due to the non-stationarity nature of the data. That's why we choose to train our models on 120 months of data, from which we exclude randomly 20 months for the validation set so that they are not potentially all in a different regime. Furthermore, the choice of the validation loss function is important. A natural choice would be to use the training loss function, which in our case is the mean-squared error. However, we use the rank correlation of the predictions as it is a more robust statistic and closer to our long-short portfolio construction. Finally, and importantly, to further increase regularization (out-of-sample predcition power), we will use an ensemble of hyperparameters instead of one set only. The sets are obtained by combining all possible values for each hyperparameter. For example, in the case of LGBM we have 900 possible sets of hyperparameters (see C.1). We then select the 20% sets of hyperparameters with the best rank correlations on the validation dates. The final prediction will be the average of models trained with these hyperparameters sets. In the case of LGBM, this means averaging 180 models. This procedure has been applied to both non-linear methods (LGBM and NN) as well as the LASSO method.

3.3. Feature importance

In order to invest based on non-linear machine learning signals, investors need to get a sense of what features contribute the most to the predicted expected return. In the literature various methods are used to estimate feature importance, see De Prado[7] for an introduction. Some of them are dependent on the given machine learning methods, such as mean decrease in impurity for trees, Hastie et al[10] while other approaches are agnostic of the method. In this research, we use the reduction in mean-squared error (MSE) in the training sample as the measure of feature importance, as this approach can be applied irrespective of the chosen machine learning method. We first estimate the MSE with all features, then for each feature we set its value to its mean in all the samples, while keeping the others unchanged, and calculate the MSE on the new dataset. The feature importance is then proportional to the decrease in MSE. We finally normalize all feature importances such that they sum to 1.

3.4. Linear, non-linear and interaction attribution

In order to understand if and how non-linear models bring investment value compared to simpler and more interpretable linear models, we need to be able to decompose the output of the model between its linear and non-linear components. We follow the framework introduced in Li et al.[8]

The prediction is first decomposed using marginal prediction, defined for feature ℓ as

$$\Phi^{(\ell)}_{\text{marginal}}(z^{\ell}) = \mathbf{E}_{z_{\setminus\ell}}[\Phi(z)] \tag{8}$$

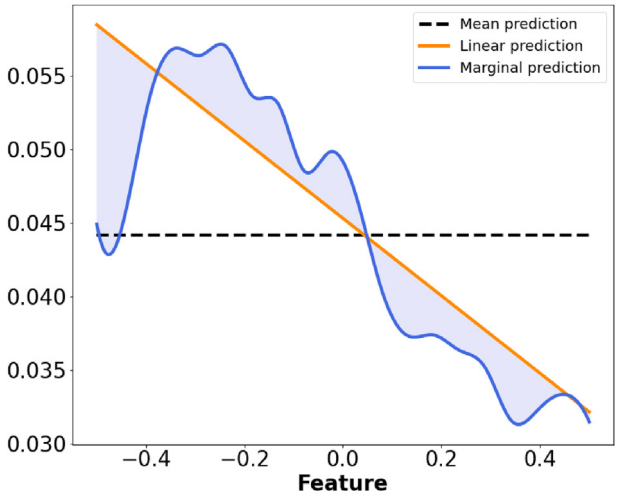


Fig. 1. The empirical linear and marginal prediction of a selected feature (MOMENTUM_5D) using LGBM. The marginal prediction is an interpolation calculated using 50 points and the linear prediction is an OLS fit to these values.

where $\mathbf{E}_{z\setminus\ell}$ is the expected value of the prediction over all features except feature ℓ. This decomposition has the important property that if the model is additive

$$\Phi(z) = \sum_{\ell} \psi^{(\ell)}(z^{\ell}), \tag{9}$$

then the obtained marginals are (up to a constant)

$$\Phi^{(\ell)}_{\text{marginal}}(z^{\ell}) = \psi^{(\ell)}(z^{\ell}). \tag{10}$$

We define the interaction attribution as the remaining part

$$\Phi_{\text{interaction}}(z) = \Phi(z) - \sum_{\ell} \Phi^{(\ell)}_{\text{marginal}}(z^{\ell}). \tag{11}$$

This term will be zero if the model is additive. In case we use a non-additive predictive model (like regression trees or neural networks), we will obtain a non-zero interaction that will have no prediction power in case the true underlying model is additive.

Empirically, the marginal prediction for feature ℓ is calculated as

$$\Phi^{(\ell)}_{\text{marginal}}(z^{\ell}) = \frac{1}{\sum_t N_t} \sum_{i,t} \Phi\left(z^1_{i,t}, \ldots, z^{\ell}, \ldots, z^K_{i,t}\right). \tag{12}$$

We fix a value for z^{ℓ} and average over all predictions using the $N \times T$ realizations of the other $(K-1)$ features.

Our sample size is large (more than 200'000) and we have many features. It is computationally too costly to evaluate this function for all sample realizations $\{z^{\ell}_{i,t}\}$. Hence instead of evaluating it for each of the 200'000 values, we will evaluate this function for only 50 points equally spaced between the min and the max of $\{z^{\ell}_{i,t}\}$ (implying a speedup of 4000). Since the features have been ranked, this will naturally span their phase space. To get a value for a z^{ℓ} different from this grid we use a standard spline interpolation.[b]

The linear prediction $\Phi^{(\ell)}_{\text{linear}}$ is obtained as the OLS fit on these 50 points minus the average of all marginal predictions, and the marginal non-linear prediction is then obtained as

$$\Phi^{(\ell)}_{\text{non-linear}}(z^{\ell}) = \Phi^{(\ell)}_{\text{marginal}}(z^{\ell}) - \Phi^{(\ell)}_{\text{linear}}(z^{\ell}). \tag{13}$$

[b] The results are similar when taking piece-wise linear interpolation or more than 50 points.

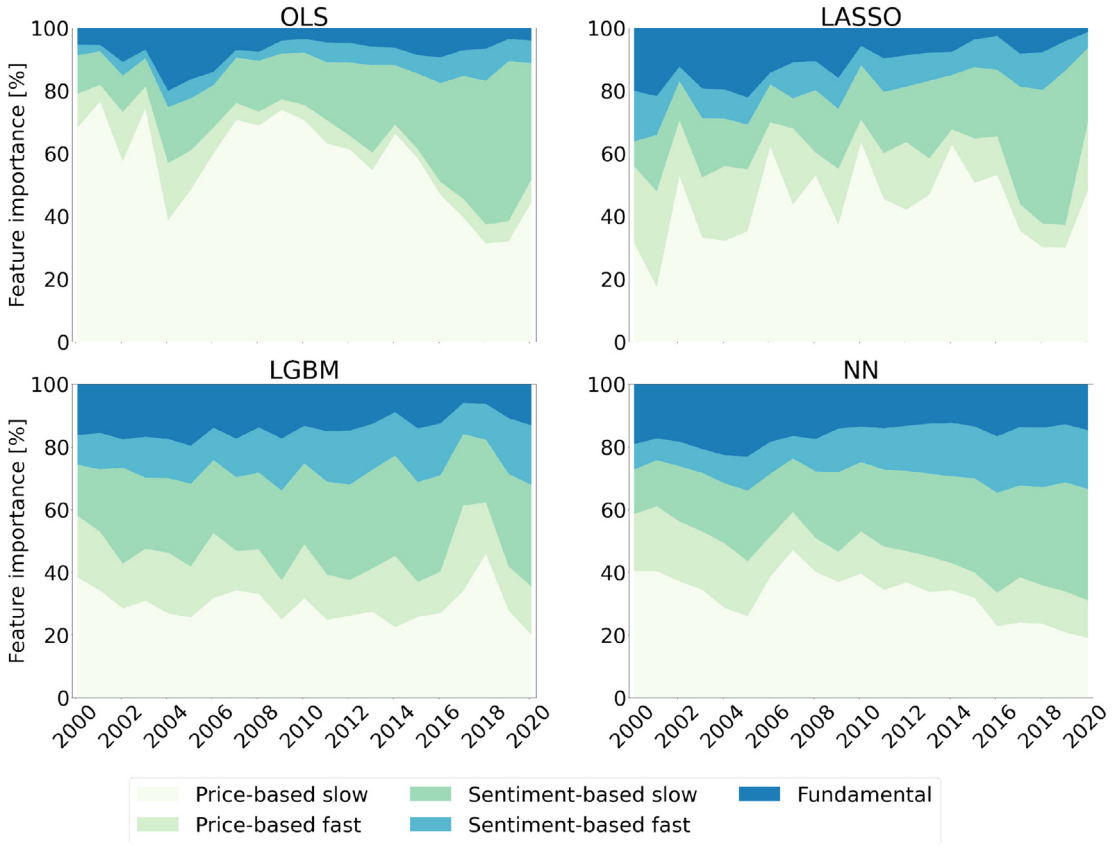


Fig. 2. Time evolution of the sum of feature importance for the different categories of features. Each point is labelled by the last date of the training set.

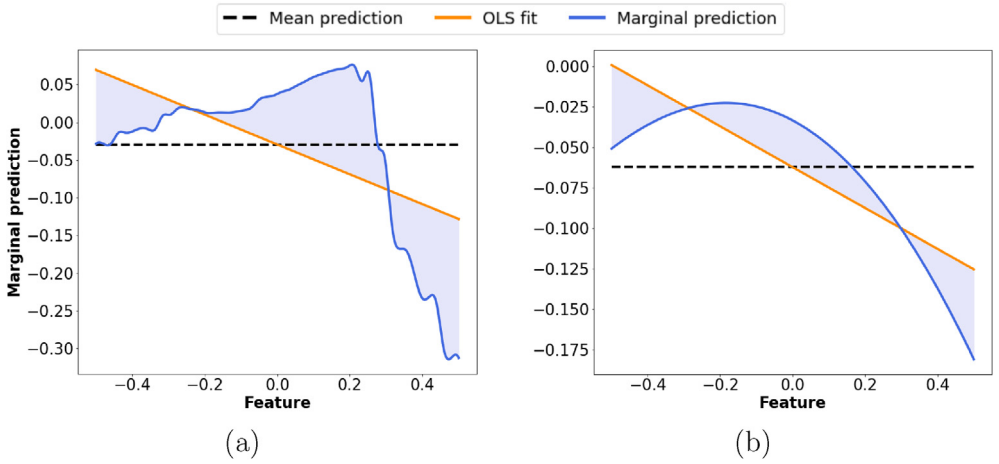


Fig. 3. The marginal components of the prediction of the strongest non-linear features for (a) LGBM and (b) NN (BETA_3M and BETA_1M respectively). The difference between the marginal prediction and the OLS fit is the marginal non-linear effect.

This is illustrated in Fig. 1.

The performance attribution can eventually be written as

$$\Phi(z_{i,t}) = \sum_{\ell} \Phi^{(\ell)}_{\text{linear}}(z^{\ell}_{i,t}) + \sum_{\ell} \Phi^{(\ell)}_{\text{non-linear}}(z^{\ell}_{i,t}) + \Phi_{\text{interaction}}(z_{i,t}). \tag{14}$$

4. Empirical results

As mentioned earlier we use data from 1990 to 2021. We use 120 months for training and validation, then keep the model fixed for the prediction of the following 12 months. Every sets are then rolled by one year yielding 21 years of predicted returns used in our analysis.

The first analysis will focus on the feature importance evolution. Then we will concentrate on the whole cross section of stocks return predictions. And finally, we study backtests of investable long-short portfolios.

4.1. Feature importance

Feature importance has been calculated for each of the 21 years of analysis. The individual ranking of the features by importance are varying through time. Hence we show in Fig. 2 the evolution of the sum of all feature importance ratios belonging to each category through time. As described above, the features are categorized as price-based, sentiment-based, and fundamental-based, and further divided in slow and fast. We notice that the non-linear methods span more homogeneously through all features.

The fast price-based features are part of the most important features. Looking at Table 2 we see that 4% of the features are fast price based, when in Table 3 we see that, for example for LGBM, the average ratio of fast price-based feature importance is 15% (almost 4 times more) which is the highest relative ratio compared to the other categories of features.

4.2. Stock return predictions

We report results for the whole cross section of stocks predictions. Many research articles use the out-of-sample R^2 to evaluate predictability. We however consider the rank correlation of the predictions with the realized targets as it is less sensitive to outliers. Further to account for non-stationarity we do not estimate rank correlation for the whole $T \times N$ sample, but estimate for each month the rank correlation and average it through time. Table 4 shows the results for the various methods and their attribution. We see that boosted trees and neural networks are better than the linear models, as found by (Rasenkhschaffe and Jones[1]; Gu et al[2]; Choi et al[3]; Leung et al[4]; Li et al[5])

The linear component of both models has the strongest prediction power. The interaction component has prediction power as well, while the marginal non-linear component predicts the targets very poorly. We also notice that the linear component and even more the interaction component are more stable through time compared with the linear component as can be seen by the lower standard deviation of the monthly rank correlation in Table 4.

The fact that the non-linear component has no predictive power was not expected when studying the marginal predictions payout. Indeed we clearly see a non-linear effect in Fig. 3 where we show the marginal predictions for the strongest non-linear components. This behaviour was also previously found in Gu et al[2] or Li et al.[5]

Table 2
Relative weight of each different category of the 205 features.

Category	Relative weight
Price-based (slow)	12%
Price-based (fast)	4%
Sentiment-based (slow)	35%
Sentiment-based (fast)	34%
Fundamental	15%

Table 3
Average feature importance over time for the different categories of the features for each of the methods.

Feature category	OLS	LASSO	LGBM	NN
Price-based (slow)	54%	43%	30%	32%
Price-based (fast)	7%	15%	16%	14%
Sentiment-based (slow)	22%	21%	27%	24%
Sentiment-based (fast)	5%	9%	14%	13%
Fundamental	8%	11%	14%	16%

Finally we show in Fig. 4 the rank correlation map between the LASSO predictions and the various components of the LGBM and NN predictions. We see that for each non-linear methods their components are uncorrelated. We also notice that boosted trees and neural networks capture a similar linear component. This is illustrated by a rank correlation of 0.82 (or 0.85) for LGBM (resp. NN) linear prediction with the LASSO predictions. To some extent the interaction component also capture similar effect as their rank correlation is 0.49.

4.3. Long-short portfolios

We now use the stock predictions and analyze investable portfolios. The models were trained on targets that are scaled sector beta-neutral returns. We will construct long-short portfolios based on these predicted performances. The performances are ranked and we select each month the highest decile for the long portfolio and the lowest decile for the short portfolio. The stocks are equally weighted in each sector and the long exposures are adjusted such that each long-short sector is beta neutral. The gross exposure of each sector is then proportional to its allocation in the base universe.

The 21 year simulations are shown in Fig. 5 on a log scale, and basic statistics are given in Table 5. In order to compare each backtest, the gross exposure has been adjusted each month with an ex-ante volatility estimation to target an annualized volatility of 10%. This yields a gross exposure around 2. In line with previous results, the best portfolio is obtained with the neural network, yielding a high Sharpe ratio of 1.73.

To account for costs we calculate the monthly two-way turnover,

$$\mathrm{TO}_t = \sum_i \left| w_t^i - w_{t-1}^i \right|, \tag{15}$$

and consider a fixed trading costs of 10 bps per unit of this turnover. In Table 5, we see that the best strategy has then still a Sharpe ratio of 1.40. Further we notice that the turnover of the non-linear machine learning methods is not higher than the linear methods.

We now consider only the three attribution components of each non-linear machine learning methods. The gross exposure of all portfolios has not been adjusted and is fixed at 2. In Fig. 6 we show the backtest curves of the long-short portfolios on a log scale and in Table 6 we show the statistics of these backtests. We see that the portfolios based solely

Table 4
Average and standard deviation of the monthly rank correlation between the realized out-of-sample targets and the predictions. In the case of LGBM and NN we also show the results by components. Each month the rank correlation has been calculated cross-secionnaly and averaged through time.

Attribution	Model	Rank correlation	
		Average [%]	Standard deviation [%]
Linear	OLS	3.64	8.77
	LASSO	4.30	10.08
	LGBM	4.51	9.71
	NN	4.16	8.88
Non-linear	LGBM	−0.60	6.03
	NN	−0.32	6.24
Interaction	LGBM	2.04	5.09
	NN	1.78	5.17
All	LGBM	4.39	8.71
	NN	4.48	8.35

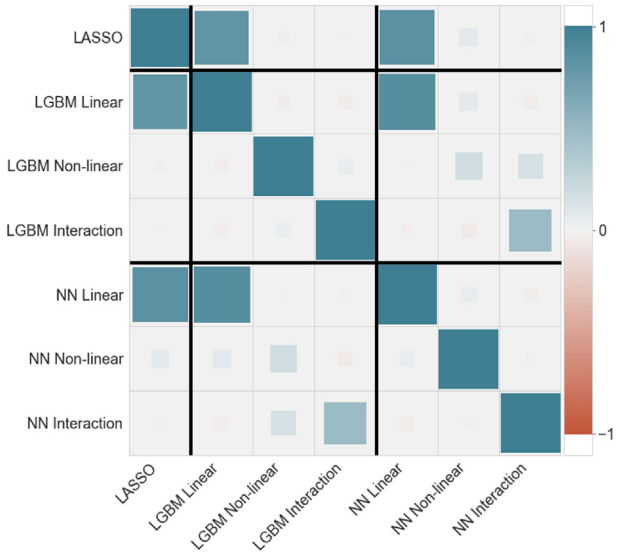


Fig. 4. Map of the rank correlation between the LASSO predictions and each components of the LGBM and NN predictions. The linear components of LGBM and NN are both very correlated to the LASSO predictions. The interaction component of the LGBM and NN are also correlated.

on the interaction component has an important Sharpe ratio of 1.02 for LGBM and 0.90 for NN, while the non-linear based portfolios perform negatively.

Finally, we notice in Fig. 5 that all long-short portfolios are quite resilient in economic crisis except during the COVID pandemic crisis in 2021. In that period the portfolios had a significant net exposure due to beta-sector neutralization which induced a strong negative performance. However the long-short portfolio built using only the interaction component is positive during that crisis as seen in Fig. 6 for LGBM and NN.

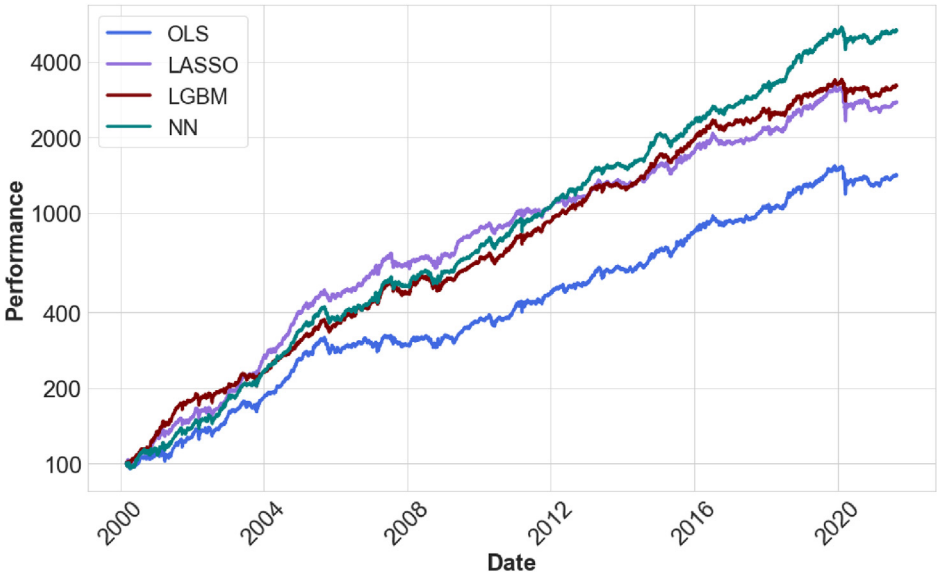
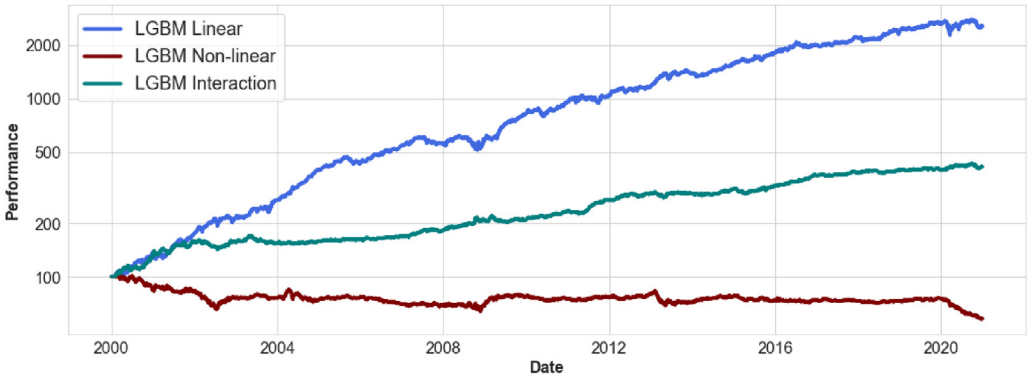


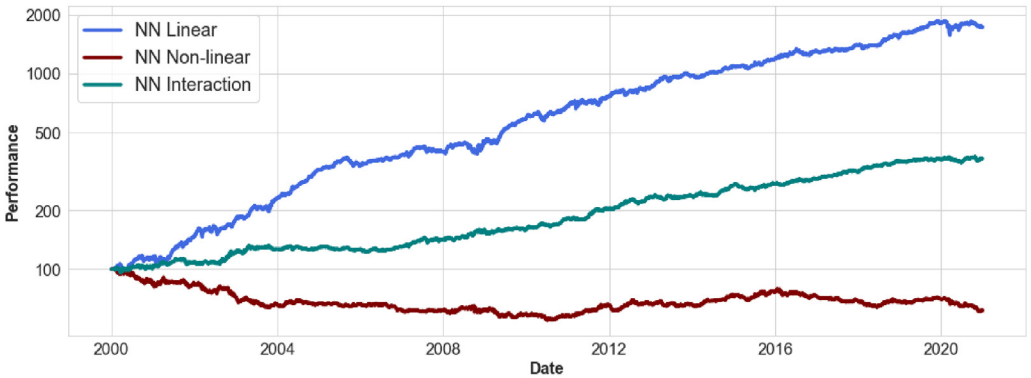
Fig. 5. Backtest curves without costs of long-short portfolios using each method. The gross exposure of the long-short portfolios have adjusted ex-ante to target an annualized volatility of 10%. The scale of the y axis is logarithmic.

Table 5
Main statistics of the long-short portfolio backtests shown in Fig. 5 for each method. The turnover is the monthly two-way turnover as defined in Eq. (15).

Method	Annualized return	Annualized volatility	Average turnover	Sharpe ratio	
				No costs	With costs
OLS	12.25%	10.73%	240%	1.14	0.82
LASSO	15.50%	10.84%	216%	1.43	1.17
LGBM	16.03%	10.59%	240%	1.51	1.20
NN	18.30%	10.56%	234%	1.73	1.40



(a)



(b)

Fig. 6. Backtest curves without costs of long-short portfolios using each components of the predictions for LGBM (a) and NN (b). The gross exposure of the long-short portfolios are set ex-ante to 2. The scale of the y axis is logarithmic.

Table 6
Main statistics of the long-short portfolio backtests shown in Fig. 6 for both methods and each components. The turnover is the monthly two-way turnover as defined in Eq. (15).

	Annualized return	Annualized volatility	Sharpe ratio	Average turnover
LGBM				
Linear	15.28%	10.55%	1.45	236%
Non-linear	−2.13%	8.08%	−0.26	247%
Interaction	6.68%	6.56%	1.02	252%
NN				
Linear	13.43%	9.93%	1.35	236%
Non-linear	−1.90%	7.78%	−0.24	164%
Interaction	6.14%	6.84%	0.90	219%

5. Conclusion

In this article, we presented a liquid dataset of international equities to which we assigned 205 price-based, fundamental-based and sentiment-based features over the last 31 years. We applied linear (OLS and LASSO) and non-linear (boosted trees and neural networks) models to predict the idiosyncratic part of the stock returns over the next month. To tune hyperparameters, we applied a careful validation procedure which takes into account the specific structure of financial data and our final aim to build investment strategies. As found in the recent literature, we confirm that even on a liquid universe, non linear methods (and in particular neural networks) achieve the best Sharpe ratio in backtests.

To understand where this outperformance is coming from, we applied a performance attribution method which splits the return prediction into three components: linear, marginal non-linear and interactions. We adapted it to our case where we have a large number of features and samples. We found that the outperformance of non-linear methods can be attributed to their strong regularization yielding a similar linear component for the boosted trees and the neural networks. This linear prediction is very close to the one obtained with linear models to which we apply regularization (L_1 constraint and model averaging). More importantly we show that these machine learning methods have no predictive power associated to marginal non-linearity, rather they capture return predicting interactions between features.

From an investor viewpoint, this interaction term is particularly interesting. Indeed, it is a signal which has shown persistent return predictability through time and is independent from the linear combination of base features. Furthermore it does not have higher turnover than the linear component.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Universe description

Table A.9
Monthly average of number of stocks, size and weights in the reference universe from 1990 to 2021.

Country	Average number of stocks	Average stock size (Bio USD)	Average weight (%)
Australia	63	8.7	2.2
Austria	19	3.6	0.2
Belgium	14	13.6	0.7
Canada	102	8.4	3.2
Denmark	19	8.5	0.4
Finland	14	11.9	0.5
France	61	23.9	4.6
Germany	36	22.7	3.3
Hong Kong	74	11.9	3.3
Ireland	8	6.9	0.2
Israel	41	2.2	0.3
Italy	33	12.8	1.6
Japan	365	8.6	16.4
Netherlands	13	28.5	1.7
New Zealand	17	1.7	0.1
Norway	22	5.8	0.4
Portugal	10	5.0	0.2
Singapore	43	5.3	0.9
Spain	22	19.0	1.5
Sweden	29	8.8	1.0
Switzerland	20	38.9	2.6
United Kingdom	96	21.1	8.1
United States	486	24.7	46.5

Appendix B. Features

Appendix B1. Features definition

Table B.9
Definition of all 205 features.

Category	Number	Mnemonics example	Description
Price	9	MOMENTUM_D5	Return over the last (1, 2) weeks, (1, 2, 3, 6) months and (1, 2, 4) years with a lag of one business day. Labels are all given in days.
Price	6	BETA_3	Stock beta over the last (3, 6) months and (1, 2, 3, 5) years with a lag of 3 business days computed with rolling time steps of 3 business days with a equally weighted kernel. Labels are all given in months.
Price	6	VOL_3	Stock volatility over the last (3, 6) months and (1, 2, 3, 5) years with a lag of 3 business days computed with rolling time steps of 3 business days with a equally weighted kernel. Labels are all given in months.
Price	6	ACTIVITY_3	Stock activity defined as the average daily volume over the last (3, 6) months and (1, 2, 3, 5) years with a 3 business days lag divided by the market capitalization 3 business days before. Labels are all given in months.
Price	5	SURPRISE_3	Average daily volume over the last (3, 6) months and (1, 2, 3) years with a 3 business days lag divided by the 5 years average daily volume. Labels are all given in months.
Fundamental	2	DY and DYD1	Dividend yield computed with a one quarter lag (as for all fundamental based features) and its 4 quarter variation.
Fundamental	2	NPM	Net profit margin and its 4 quarter variation.
Fundamental	2	ROA	Return on assets and its 4 quarter variation.
Fundamental	2	E2P	Earnings to price and its 4 quarter variation.
Fundamental	2	P2B	Price to book and its 4 quarter variation.
Fundamental	4	DTA	Debt to assets and long term debt to assets and its 4 quarter variation.
Fundamental	2	ROE	Return on equity and its 4 quarter variation.
Fundamental	2	ROIC	Return on invested capital and its 4 quarter variation.
Fundamental	2	CF2S	Cash flow to sales and its 4 quarter variation.
Fundamental	4	AGR and SGR	Asset and Sales 5Y growth and its 4 quarter variation.
Fundamental	2	E2IE	EBIT to interest expense and its 4 quarter variation.
Fundamental	2	CGS2S	Cost of goods sold to sales and its 4 quarter variation.
Fundamental	2	OI2C	Operating income to total capital and its 4 quarter variation.
Sentiment	7	ANALYST_MOM-ENTUM_EPS_5	Number of upward minus number of downward fiscal year 1 and 2 earnings per share (EPS) revisions over the last (1, 2) weeks, (1, 2, 3, 6, 12) months with a one business day lag divided by the total number of revisions over the last year. Labels are all given in days.
Sentiment	77	ANALYST_MOM-ENTUM_SAL_5, ANALYST_MOM-ENTUM_BPS_5, etc	Same definition replacing EPS by sales (SAL), book value per share (BPS), cash flow per share (CPS) dividend per share (DPS), entreprsise value (ENT), net asset value (NAV), net debt (NDT), net income (NET), operating profit (OPR), return on assets (ROA), return on equity (ROE). Labels are all given in days.
Sentiment	14	ANALYST_MOM-ENTUM_REC_5 and ANALYST_MOM-ENTUM_PRT_5	Same definition replacing EPS by recommendations (REC) and price targets (PRT). Labels are all given in days.
Sentiment	1	ANALYST_ME-AN_REC	Average analyst recommendation with a one week lag.
Sentiment	1	ANALYST_ME-AN_IMP_RET	Average analyst implied return (price target divided by the current stock priceminus one) with a one week lag.

Table B.9 (*continued*)

Category	Number	Mnemonics example	Description
Sentiment	1	ANALYST_EP-S_GROWTH	Implied earnings per share (EPS) growth between fiscal year 1 and 3 with a one week lag.
Sentiment	11	ANALYST_SA-L_GROWTH	Same definition replacing EPS by sales (SAL), book value per share (BPS), cash flow per share (CPS), dividend per share (DPS), entreprsise value (ENT), net asset value (NAV), net debt (NDT), net income (NET), operating profit (OPR), return on assets (ROA), return on equity (ROE).
Sentiment	3	ANALY-ST_FY1_ROA	Median predicted return on assets for fiscal year 1, 2 and 3 with a one week lag.
Sentiment	3	ANALY-ST_FY1_ROE	Median predicted return on equity for fiscal year 1, 2 and 3 with a one week lag.
Sentiment	3	ANALY-ST_FY1_E2P	Median predicted earning for fiscal year 1, 2 and 3with a one week lag over the current price with a one week lag.
Sentiment	18	ANALY-ST_FY1_S2P	Same definition replacing E2P by sales (S2P), net income (N2P), cash flow (C2P), assets (A2P), dividends (D2P) and book value (B2P).

Appendix B2. Features category

Table B.10
Category of each of the 205 features.

Category	Mnemonics
Price - Slow	ACTIVITY_12
Price - Slow	VOL_60
Price - Slow	ACTIVITY_24
Price - Slow	SURPRISE_24
Price - Slow	VOL_12
Price - Slow	BETA_6
Price - Slow	SURPRISE_6
Price - Slow	BETA_60
Price - Slow	MOMENTUM_D1040
Price - Slow	BETA_36
Price - Slow	BETA_12
Price - Slow	ACTIVITY_60
Price - Slow	MOMENTUM_D130
Price - Slow	MOMENTUM_D520
Price - Slow	BETA_24
Price - Slow	ACTIVITY_36
Price - Slow	MOMENTUM_D260
Price - Slow	VOL_6
Price - Slow	VOL_24
Price - Slow	VOL_36
Price - Slow	ACTIVITY_6
Price - Slow	SURPRISE_12
Price - Slow	SURPRISE_60
Price - Slow	SIZE
Price - Fast	VOL_3
Price - Fast	SURPRISE_3
Price - Fast	MOMENTUM_D40
Price - Fast	MOMENTUM_D22

(*continued on next page*)

Table B.10 (*continued*)

Category	Mnemonics
Price - Fast	MOMENTUM_D65
Price - Fast	MOMENTUM_D5
Price - Fast	BETA_3
Price - Fast	MOMENTUM_D10
Price - Fast	ACTIVITY_3
Fundamental	DY
Fundamental	SGRD1
Fundamental	NPMD1
Fundamental	CCS2S
Fundamental	DYD1
Fundamental	ROICD1
Fundamental	AGRD1
Fundamental	LTDTA
Fundamental	OI2CD1
Fundamental	ROE
Fundamental	E2EI
Fundamental	NPM
Fundamental	SGR
Fundamental	CF2S
Fundamental	OI2C
Fundamental	DTAD1
Fundamental	LTDTAD1
Fundamental	P2BD1
Fundamental	ROAD1
Fundamental	ROED1
Fundamental	E2PD1
Fundamental	DTA
Fundamental	ROA
Fundamental	CCS2SD1
Fundamental	AGR
Fundamental	E2P
Fundamental	E2EID1
Fundamental	P2B
Fundamental	CF2SD1
Fundamental	ROIC
Sentiment - Slow	ANALYST_ROA_GROWTH
Sentiment - Slow	ANALYST_MOMENTUM_BPS_180
Sentiment - Slow	ANALYST_FY3_EPS2P
Sentiment - Slow	ANALYST_MOMENTUM_BPS_365
Sentiment - Slow	ANALYST_FY3_SAL2P
Sentiment - Slow	ANALYST_FY1_CPS2P
Sentiment - Slow	ANALYST_ME-AN_IMP_RET
Sentiment - Slow	ANALYST_FY3_NET2P
Sentiment - Slow	ANALYST_MOMENTUM_PRT_180
Sentiment - Slow	ANALYST_FY1_ROE
Sentiment - Slow	ANALYST_MOMENTUM_OPR_365
Sentiment - Slow	ANALYST_FY3_DPS2P
Sentiment - Slow	ANALYST_FY3_ROE
Sentiment - Slow	ANALYST_NAV_GROWTH
Sentiment - Slow	ANALYST_MOMENTUM_EPS_365
Sentiment - Slow	ANALYST_FY1_ROA
Sentiment - Slow	ANALYST_FY1_NET2P
Sentiment - Slow	ANALYST_MOMENTUM_NDT_365
Sentiment - Slow	ANALYST_MOMENTUM_REC_365
Sentiment - Slow	ANALYST_MOMENTUM_NET_180
Sentiment - Slow	ANALYST_NET_GROWTH
Sentiment - Slow	ANALYST_FY2_EPS2P

Table B.10 (*continued*)

Category	Mnemonics
Sentiment - Slow	ANALYST_FY1_NAV2P
Sentiment - Slow	ANALYST_MOMENTUM_DPS_365
Sentiment - Slow	ANALYST_MOMENTUM_ROA_180
Sentiment - Slow	ANALYST_SAL_GROWTH
Sentiment - Slow	ANALYST_FY2_NAV2P
Sentiment - Slow	ANALYST_MOMENTUM_PRT_365
Sentiment - Slow	ANALYST_FY2_NET2P
Sentiment - Slow	ANALYST_FY2_CPS2P
Sentiment - Slow	ANALYST_FY3_ROA
Sentiment - Slow	ANALYST_MOMENTUM_NAV_180
Sentiment - Slow	ANALYST_FY3_BPS2P
Sentiment - Slow	ANALYST_MOMENTUM_ROE_180
Sentiment - Slow	ANALYST_MOMENTUM_NET_365
Sentiment - Slow	ANALYST_FY2_SAL2P
Sentiment - Slow	ANALYST_FY1_EPS2P
Sentiment - Slow	ANALYST_MOMENTUM_CPS_180
Sentiment - Slow	ANALYST_NDT_GROWTH
Sentiment - Slow	ANALYST_FY3_ENT2P
Sentiment - Slow	ANALYST_FY1_SAL2P
Sentiment - Slow	ANALYST_MOMENTUM_DPS_180
Sentiment - Slow	ANALYST_FY2_ROE
Sentiment - Slow	ANALYST_FY1_ENT2P
Sentiment - Slow	ANALYST_OPR_GROWTH
Sentiment - Slow	ANALYST_DPS_GROWTH
Sentiment - Slow	ANALYST_MOMENTUM_NAV_365
Sentiment - Slow	ANALYST_FY2_ENT2P
Sentiment - Slow	ANALYST_ENT_GROWTH
Sentiment - Slow	ANALYST_MOMENTUM_CPS_365
Sentiment - Slow	ANALYST_FY1_DPS2P
Sentiment - Slow	ANALYST_MOMENTUM_SAL_365
Sentiment - Slow	ANALYST_CPS_GROWTH
Sentiment - Slow	ANALYST_FY3_NAV2P
Sentiment - Slow	ANALYST_EPS_GROWTH
Sentiment - Slow	ANALYST_MOMENTUM_REC_180
Sentiment - Slow	ANALYST_MOMENTUM_SAL_180
Sentiment - Slow	ANALYST_BPS_GROWTH
Sentiment - Slow	ANALYST_FY2_ROA
Sentiment - Slow	ANALYST_MOMENTUM_ENT_180
Sentiment - Slow	ANALYST_FY1_BPS2P
Sentiment - Slow	ANALYST_FY2_DPS2P
Sentiment - Slow	ANALYST_MOMENTUM_NDT_180
Sentiment - Slow	ANALYST_FY3_CPS2P
Sentiment - Slow	ANALYST_MEAN_REC
Sentiment - Slow	ANALYST_ROE_GROWTH
Sentiment - Slow	ANALYST_MOMENTUM_OPR_180
Sentiment - Slow	ANALYST_FY2_BPS2P
Sentiment - Slow	ANALYST_MOMENTUM_EPS_180
Sentiment - Slow	ANALYST_MOMENTUM_ROA_365
Sentiment - Slow	ANALYST_MOMENTUM_ENT_365
Sentiment - Slow	ANALYST_MOMENTUM_ROE_365
Sentiment - Fast	ANALYST_MOMENTUM_DPS_90
Sentiment - Fast	ANALYST_MOMENTUM_CPS_60
Sentiment - Fast	ANALYST_MOMENTUM_SAL_60
Sentiment - Fast	ANALYST_MOMENTUM_PRT_7
Sentiment - Fast	ANALYST_MOMENTUM_BPS_60
Sentiment - Fast	ANALYST_MOMENTUM_OPR_7
Sentiment - Fast	ANALYST_MOMENTUM_EPS_90

(*continued on next page*)

Table B.10 (*continued*)

Category	Mnemonics
Sentiment - Fast	ANALYST_MOMENTUM_ENT_30
Sentiment - Fast	ANALYST_MOMENTUM_NDT_60
Sentiment - Fast	ANALYST_MOMENTUM_PRT_90
Sentiment - Fast	ANALYST_MOMENTUM_ROA_14
Sentiment - Fast	ANALYST_MOMENTUM_SAL_7
Sentiment - Fast	ANALYST_MOMENTUM_DPS_30
Sentiment - Fast	ANALYST_MOMENTUM_NET_90
Sentiment - Fast	ANALYST_MOMENTUM_ROE_30
Sentiment - Fast	ANALYST_MOMENTUM_ENT_60
Sentiment - Fast	ANALYST_MOMENTUM_ROA_60
Sentiment - Fast	ANALYST_MOMENTUM_BPS_7
Sentiment - Fast	ANALYST_MOMENTUM_ROE_14
Sentiment - Fast	ANALYST_MOMENTUM_CPS_30
Sentiment - Fast	ANALYST_MOMENTUM_ROA_90
Sentiment - Fast	ANALYST_MOMENTUM_ENT_14
Sentiment - Fast	ANALYST_MOMENTUM_NDT_30
Sentiment - Fast	ANALYST_MOMENTUM_EPS_14
Sentiment - Fast	ANALYST_MOMENTUM_NDT_90
Sentiment - Fast	ANALYST_MOMENTUM_REC_60
Sentiment - Fast	ANALYST_MOMENTUM_ENT_90
Sentiment - Fast	ANALYST_MOMENTUM_NET_60
Sentiment - Fast	ANALYST_MOMENTUM_ROE_7
Sentiment - Fast	ANALYST_MOMENTUM_ROE_90
Sentiment - Fast	ANALYST_MOMENTUM_NAV_14
Sentiment - Fast	ANALYST_MOMENTUM_CPS_7
Sentiment - Fast	ANALYST_MOMENTUM_NAV_90
Sentiment - Fast	ANALYST_MOMENTUM_NAV_60
Sentiment - Fast	ANALYST_MOMENTUM_BPS_14
Sentiment - Fast	ANALYST_MOMENTUM_SAL_30
Sentiment - Fast	ANALYST_MOMENTUM_REC_14
Sentiment - Fast	ANALYST_MOMENTUM_ENT_7
Sentiment - Fast	ANALYST_MOMENTUM_NAV_30
Sentiment - Fast	ANALYST_MOMENTUM_PRT_30
Sentiment - Fast	ANALYST_MOMENTUM_DPS_7
Sentiment - Fast	ANALYST_MOMENTUM_ROE_60
Sentiment - Fast	ANALYST_MOMENTUM_EPS_7
Sentiment - Fast	ANALYST_MOMENTUM_DPS_14
Sentiment - Fast	ANALYST_MOMENTUM_ROA_30
Sentiment - Fast	ANALYST_MOMENTUM_REC_90
Sentiment - Fast	ANALYST_MOMENTUM_OPR_60
Sentiment - Fast	ANALYST_MOMENTUM_CPS_90
Sentiment - Fast	ANALYST_MOMENTUM_DPS_60
Sentiment - Fast	ANALYST_MOMENTUM_REC_7
Sentiment - Fast	ANALYST_MOMENTUM_OPR_14
Sentiment - Fast	ANALYST_MOMENTUM_NAV_7
Sentiment - Fast	ANALYST_MOMENTUM_ROA_7
Sentiment - Fast	ANALYST_MOMENTUM_REC_30
Sentiment - Fast	ANALYST_MOMENTUM_SAL_90
Sentiment - Fast	ANALYST_MOMENTUM_BPS_30
Sentiment - Fast	ANALYST_MOMENTUM_BPS_90
Sentiment - Fast	ANALYST_MOMENTUM_PRT_14
Sentiment - Fast	ANALYST_MOMENTUM_PRT_60
Sentiment - Fast	ANALYST_MOMENTUM_NET_30
Sentiment - Fast	ANALYST_MOMENTUM_SAL_14
Sentiment - Fast	ANALYST_MOMENTUM_NET_14
Sentiment - Fast	ANALYST_MOMENTUM_NDT_7
Sentiment - Fast	ANALYST_MOMENTUM_NDT_14
Sentiment - Fast	ANALYST_MOMENTUM_OPR_30

Table B.10 (*continued*)

Category	Mnemonics
Sentiment - Fast	ANALYST_MOMENTUM_EPS_30
Sentiment - Fast	ANALYST_MOMENTUM_EPS_60
Sentiment - Fast	ANALYST_MOMENTUM_CPS_14
Sentiment - Fast	ANALYST_MOMENTUM_NET_7
Sentiment - Fast	ANALYST_MOMENTUM_OPR_90

Appendix C. Hyperparameters sets

Appendix C1. Light GBM

Table C.11
Hyperparameter values for LGBM. The 900 possible sets are obtained by combining each values.

Hyperparameter	Values
Number of leaves	10, 20, 30, 40, 50
Number of trees	100, 200, 300, 400, 500, 600, 700, 800, 900, 1000
Learning rate	0.001, 0.01, 0.05
Bagging ratio	80%, 100%
Minimum data per leaves	1%, 2%, 4%

Appendix C2. Feed-Forward Neural Network

Table C.12
Hyperparameter values for NN. The 90 possible sets are obtained by combining each values.

Hyperparameter	Values
Number of layers	1, 2, 3
Number of units in first layer	N = 8, 16, 32, 48, 64
Number of units in second layer	$N/2$
Number of units in third layer	$N/4$
Learning rate	0.001, 0.005, 0.01
Ensemble size	4
Batch size	10’000
Number of epochs	100
Number of patience steps for early stopping	10, 100

References

1. Rasekhschaffe K, Jones R. Machine learning for stock selection. *Financ Anal J*. 2019;75(3).
2. Gu S, Kelly B, Xiu D. Empirical asset pricing via machine learning. *Rev Financ Stud*. 2020;33(5).
3. Choi D, Jiang W, Zhang C. *Alpha Go Everywhere: Machine Learning and International Stock Returns*. 2022. Available at: SSRN 3489679.
4. Leung E, Lohre H, Mischlich D, Shea Y, Stroh M. The promises and pitfalls of machine learning for predicting stock returns. *J Financial Data Sci*. 2020;3.
5. Li Y, Simon Z, Turkington D. Investable and interpretable machine learning for equities. *The Journal of Financial Data Science*. 2022;4(1):54–74.
6. Patir R, Singhal S, Anantaram C, Goyal V. Interpretability of black box models through data-view extraction and shadow model creation. In: *International Conference on Neural Information Processing*. Springer; 2020:378–385.
7. De Prado ML. *Advances in Financial Machine Learning*. John Wiley & Sons; 2018.
8. Li Y, Turkington D, Yazdani A. Beyond the black box: an intuitive approach to investment prediction with machine learning. *J Financial Data Sci*. 2020;2(1).

9. Efron B, Hastie T, Johnstone I, Tibshirani R. Least angle regression. *Ann Stat*. 2004;32(2).
10. Hastie T, Tibshirani R, Friedman JH, Friedman JH. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. vol. 2. Springer; 2009.
11. Ke G, Meng Q, Finley T, Wang T, et al. Lightgbm: a highly efficient gradient boosting decision tree. *Adv Neural Inf Process Syst*. 2017;30.
12. Dixon MF, Halperin I, Bilokon P. *Machine Learning in Finance*. vol. 1170. Springer; 2020.
13. Kingma DP, Ba J, Adam. *A Method for Stochastic Optimization*. 2014. arXiv preprint arXiv:1412.6980.