EPFL

Machine Learning in Finance Project FIN-413

Deep-Learning backed Market Predictions

Alexandre Huou (SCIPER 342227)
Loris Tran (SCIPER 341214)
Marius Pecaut (SCIPER 330684)
Mahmoud Dokmak (SCIPER 301995)
Zayed Kriem (SCIPER 311046)

Financial Engineering
Group ICGANG

Prof. Malamud

Contents

1 Introduction

The quick development of machine learning models raises the question of whether it is possible to predict stock returns by analyzing historical data and several indicators. In this project, we explore this question by leveraging both structured data, such as asset returns, and unstructured textual information extracted from earnings calls and 10-K reports. These textual sources may be relevant, as they are written or spoken by company insiders and contain forward-looking insights and sentiment that may influence investor behavior.

As part of our indicator data, we used the Quarterly Compustat Firm Characteristics dataset. However, this dataset contains a significant number of missing values, which required careful preprocessing before it could be used effectively in our models.

Earnings call transcripts and 10-K filings provide rich qualitative information and were processed using Natural Language Processing (NLP) techniques to extract sentiment features. These features were then incorporated into our return prediction models.

Our project is structured as follows. In Section 2, we describe in detail how we cleaned and preprocessed the Compustat dataset. In Section 3, we present our NLP pipeline for extracting sentiment features from the textual datasets. We also describe an attempt to use an LSTM model to predict future returns based on past information. 4 While the model demonstrated some promise on individual stock series, it failed to generalize due to data sparsity of the provided datasets. Finally, we present our main approach: a comparative analysis of a range of models that assign weights to firm characteristics and sentiment features in order to predict stock returns.

2 Data Preprocessing and Feature Engineering

2.1 Data Preprocessing

Our project required integrating quarterly financial data from Compustat with monthly stock returns to build effective predictive models. The Compustat dataset presented significant challenges due to its sparsity, with many columns having coverage below 50%. We have a table with coverage of some of the columns here, with other indicators ranging from 60% to below 20%

Table 1: Some Financial Variables in Compustat Dataset

Variable	Coverage (%)	Description
gvkey	100.0	Global company key identifier
datadate	100.0	Date of the financial data (fiscal period end)
cusip	99.91	CUSIP identifier
epspxy	70.47	Earnings Per Share (Diluted) Excluding Extraordinary Items
oiadpy	71.95	Operating Income After Depreciation
niy	73.53	Net Income
saley	73.47	Net Sales/Revenue
capxy	66.11	Capital Expenditures
dvy	65.72	Cash Dividends
cogsy	72.41	Cost of Goods Sold

We began by filtering the Compustat dataset to include only companies present in our returns dataset, using CUSIP identifiers to link them. This process reduced our dataset to only those with corresponding return data, ensuring that our subsequent modeling efforts would focus exclusively on relevant entities. Financial statements are released quarterly, while stock returns are observed monthly. To address this, we implemented a forward-fill approach using ffill from pandas. For each company, we created a monthly grid spanning their data history and populated it with the most recently available financial information at each point in time with ffil, ensuring we don't have access to future information for present time.

2.2 Feature Engineering

We engineered several financial indicators that capture some aspects of company performance and financial health. These features were designed to provide signals that might correlate with market sentiment and future stock performance.

The most notable feature we made was to use several moving averages (Both Simple and Exponential Moving Averages, with 3-month, 6-month, and 12-month windows) as targets instead of the raw returns. This approach was necessary because the raw returns were oscillating too much to be predictable.

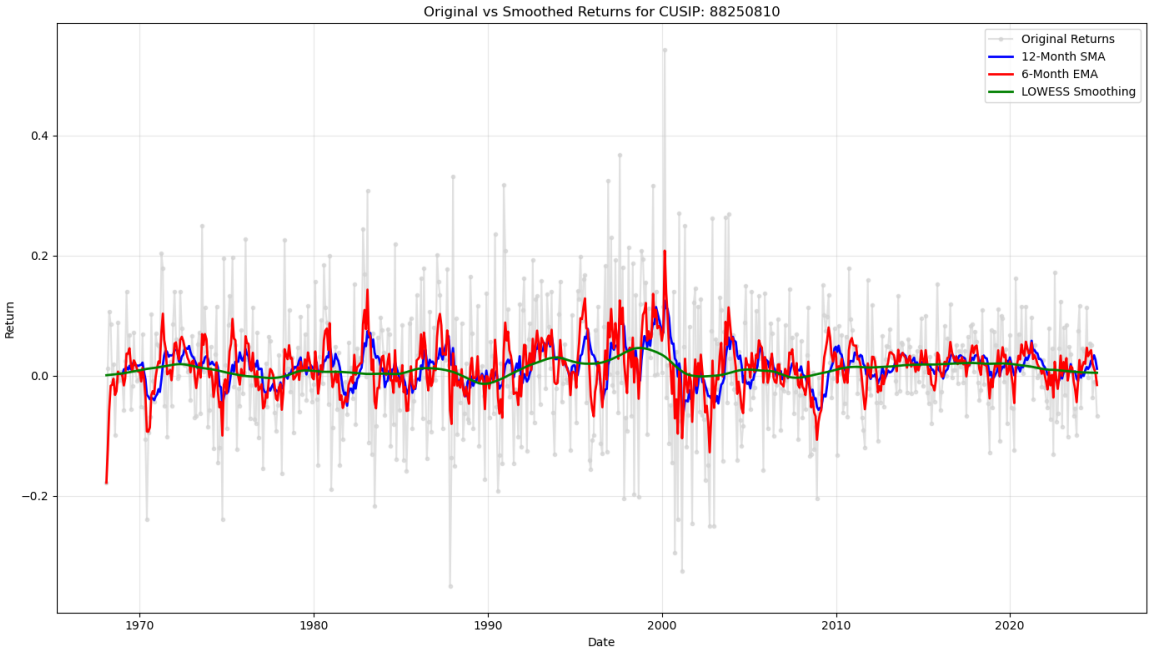

Figure 1: Moving averages vs Raw returns for a particular CUSIP

In particular, as can be seen in (Figure 2) Exponential Moving average tended to keep the trends of oscillation in returns, while smoothing them enough to be predictable.

We also created features for changes in some of the Compustat Metrics to have the evolution trends of some of the indicators:

Earnings Growth measures the year-over-year change in net income, capturing the company's profitability trend. This metric is particularly important as earnings performance often drives investor sentiment. We calculated it as:

Earnings Growth $= \frac{NI_t - NI_{t-12}}{|NI_{t-12}|}$

Revenue Growth tracks the year-over-year change in company sales to identify growth trends, which can indicate business expansion or contraction. This provides insight into the company's market performance independent of cost management. The formula used was:

Revenue Growth $= \frac{SALE_t - SALE_{t-12}}{|SALE_{t-12}|}$

Capital Expenditure Growth examines the year-over-year change in capital expenditures, helping identify companies that are investing in future growth or reducing their capital investments:

CAPEX Growth $= \frac{CAPX_t - CAPX_{t-12}}{|CAPX_{t-12}|}$

Repurchase Intensity calculates stock repurchases as a percentage of sales to measure stock buybacks. This can indicate management's view that the stock is undervalued or that they lack better investment opportunities:

Repurchase Intensity $= \frac{PRSTKC_t}{SALE_t}$

3 Sentiment Analysis

For our deep learning part, we turn our attention to formal sources of information such as corporate communications and regulatory filings. The idea is that such documents are written by company executives and should provide insights into performance, strategic direction, and perceived risks. By extracting sentiment from this language, we aim to understand how the tone and wording used internally can influence investor expectations.

To evaluate whether qualitative information improves return prediction, we incorporated two textual sources: **10-K filings** and **earnings call transcripts**. Unlike structured financial metrics, sentiment from these documents can reflect management's tone, confidence, and forward-looking language, elements that may affect market perception and future returns.

- **10-K Reports (MD&A Section):** The Management Discussion and Analysis section offers structured commentary on company performance and risk. Sentiment here reflects long-term strategic tone and may signal future expectations.
- **Earnings Calls:** These transcripts offer real-time insight, particularly during the Q&A session, where management's unscripted responses can expose sentiment shifts or uncertainty, potentially affecting short-term returns.

We first cleaned the textual data by removing user mentions, URLs, hashtags, and metadata headers to improve language detection and model performance.

For sentiment analysis, we evaluated multiple NLP models. We tested `StephanAkkerman/FinTwitBERT` [1], a BERT-based model trained on financial tweets at first. Although effective for social media analysis, its performance was suboptimal on formal corporate documents due to style and tone mismatch.

We therefore adopted `ProsusAI/finbert` [2], a financial-domain BERT model trained on professional texts. FinBERT classifies sentiment as *positive*, *neutral*, or *negative*. We mapped these to numerical values (1, 0, -1) and weighted each by its associated confidence score, resulting in a polarity-adjusted sentiment score.

Tables 2 and 3 present the sentiment distributions for each document type.

Table 2: Sentiment Distribution — Earnings Calls

Sentiment Category	Count (%)	Average Confidence
Positive	1,728,592 (13.9%)	0.76
Neutral	9,875,256 (79.1%)	0.84
Negative	872,048 (7.0%)	0.76
Total	12,475,896	–

Table 3: Sentiment Distribution — 10-K MD&A Section

Sentiment Category	Count (%)	Average Confidence
Positive	220 (2.6%)	0.75
Neutral	7,946 (95.5%)	0.91
Negative	156 (1.9%)	0.81
Total	8,322	–

The distribution shows a strong skew toward neutral sentiment, particularly in 10-K filings, reflecting the conservative tone of formal corporate communication. This class imbalance may influence downstream model performance.

For each earnings call, we computed a transcript-level sentiment score as the weighted mean of all sentence-level scores, capturing both tone and confidence. We then aligned these scores with financial data using `pd.merge_asof()` based on reporting dates. To create a unified qualitative feature, we averaged the earnings call and 10-K sentiment scores when both were available. This gave us a consolidated dataset with both quantitative factors and sentiment metrics suitable for predictive modeling.

4 Predictive Models

Project Idea Given how competitive ML-backed trading is, we figured from the start that any method that would reveal itself to be efficient and provide an edge would need to be a creative one. This is the reason why initially our efforts were inspired by the paper *QWGA: Quantum Walk Gated Architecture for Time Series Forecasting* [3], which proposes a novel neural architecture for predictive modeling on time series. The model, built on the innovative merged usage of computer vision, quantitative gates and time series showed competitive performance on a variety of sequential prediction tasks. We trained a single model per stock, and in the first implementation using the YahooFinance library, the model managed to accurately predict 'TSLA', 'AAPL', 'MSFT' and most of the big names (60-70% accuracy depending on the stock and the split). However when trying to use this codebase on our own stocks, the accuracy plummeted down to 5%. After some debugging, we realized that all of our stocks actually had less than 400 datapoints, as opposed to yhfin (8000 per stock).

The sparsity of the data made us realize that training a Deep Learning model per stock would not be feasible due to the over-parametrization (D>>N) problem. As such we decided to use Deep Learning to extract a sentiment score from the 10K reports, and use this sentiment as another feature to some more "classical" ML models.

4.1 Model Benchmarking and Experimental Evaluation

For each experiment, we isolated the data for a single CUSIP. The dataset consisted of lagged monthly returns and various firm-level predictors such as earnings, growth rates, and balance sheet metrics. We used a train/test split preserving temporal structure, with performance evaluated on the test set using the following metrics: **R^2 score**: Proportion of variance explained, **MAE (Mean Absolute Error)**: Average magnitude of errors, **RMSE (Root Mean Squared Error)**: Sensitive to larger errors

Each model was tuned via `GridSearchCV`, with careful hyperparameter specification to ensure fairness across comparisons.

4.2 Models Tested

We implemented and evaluated the following models on a randomized (time-coherent) 8-fold cross validation set. It should be noted that the parameters for eahc of these models have been chosen after running a hyperparameter tuning program in order to guarantee them all to be on equal footing.

1. **Linear Regression (OLS)**: Served as a baseline with no regularization. This model surprisingly performed well for most CUSIPs, especially when features were already well-filtered and not strongly collinear

2. **ElasticNet**: Combined L1 and L2 regularization. Performance was highly sensitive to the α parameter. When $\alpha = 0$, ElasticNet behaved like OLS and achieved results close to Linear Regression. However we found the regularization to be too strict which impeded the model from making good steps.

3. **Random Forest Regressor**: A strong ensemble method that improved on Decision Trees by reducing variance. Provided robust results but missed most of the extreme values, thus making us lose out on the high risk, high reward opportunities.

4. **Gradient Boosting Regressor**: Often yielded the best performance among tree-based methods, though tuning was computationally expensive due to this method's structure.

5. **XGBoost Regressor**: Emerged as the best overall model in our initial benchmark. After careful hyperparameter tuning, it consistently achieved the highest R^2 scores, with the best configuration reaching $R^2 \approx 0.693$ on a held-out test set.

6. **HuberRegressor**: Got the most inconsistent results, staying in the negative R2 zones, made us realize that due to the stock's variable amount of available data, some methods would be overfitting no matter what

7. **Support Vector Regression (SVR)**: Captured some nonlinear trends in the returns. However, the overall R^2 score remained relatively low, indicating limited benefit from its kernel-based structure on this dataset.

8. **KNeighborsRegressor**: The model failed to generalize meaningful patterns in the returns series, likely due to the relatively high dimensionality of the input space or the low signal-to-noise ratio.

9. **HistGradientBoostingRegressor**: The model was able to extract non-linear patterns and interactions in the data, and perform effective regularization. Though it did not match the performance of OLS or GradientBoostingRegressor in this particular case, HGBR remains a strong contender due to its speed and scalability in large structured datasets.

Table 4: Performance Comparison of Regression Models on Selected CUSIP

Model	**R^2**	**MAE**	**RMSE**
Linear Regression (OLS)	0.692	0.039	0.046
ElasticNet	-0.070	0.080	0.110
Random Forest Regressor	0.681	0.052	0.071
Gradient Boosting Regressor	0.710	0.044	0.064
Huber Regressor	-0.710	0.142	0.167
Support Vector Regressor (SVR)	0.030	0.080	0.109
KNeighbors Regressor	-0.030	0.080	0.110
HistGradientBoosting Regressor	0.630	0.050	0.080

4.3 Observations

- **XGBoost** consistently outperformed all linear and tree-based baselines, both in terms of R^2 and error metrics. This reinforces the strength of gradient-boosted ensembles in structured tabular data.

- **ElasticNet with** $\alpha = 0$ was very close in performance to Linear Regression, confirming that regularization was unnecessary (or even detrimental) for the filtered feature set used.

- **SVR** was underwhelming, despite its theoretical capacity to capture nonlinearities. It tended to regress toward the mean and failed to capture higher-order relationships in the feature space.

- **Model selection was highly CUSIP-dependent**. Some identifiers favored sparse linear models, while others benefited from more complex trees.

4.4 Individual Walkthrough: Gradient Boosting

In this subsection, we provide a detailed analysis of the performance of the `GradientBoostingRegressor` model, one of the most effective predictors in our experiments. Gradient Boosting builds an ensemble of weak learners—typically shallow decision trees—sequentially, where each subsequent model corrects the residual errors of the combined ensemble of previous models. This iterative refinement enables the model to capture complex non-linear relationships within the data.

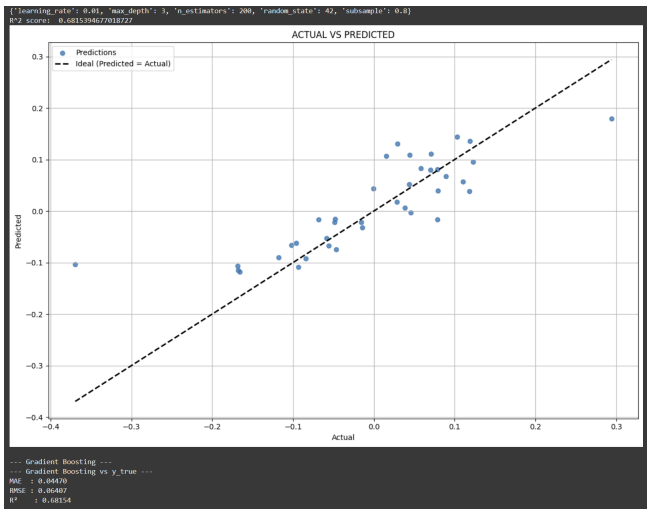

Figure 2: Gradient Boosting optimal parameters fit

On this run, Gradient Boosting achieved an R^2 score of 0.68, an MAE of 0.044, and an RMSE of 0.064, positioning it among the top three models tested. These values suggest a good balance between underfitting and overfitting, as well as strong generalization capabilities on unseen data. The model's ability to handle both feature interactions and non-linear effects makes it especially well-suited for financial time series prediction, where traditional linear models often struggle to fully capture the dynamics involved.

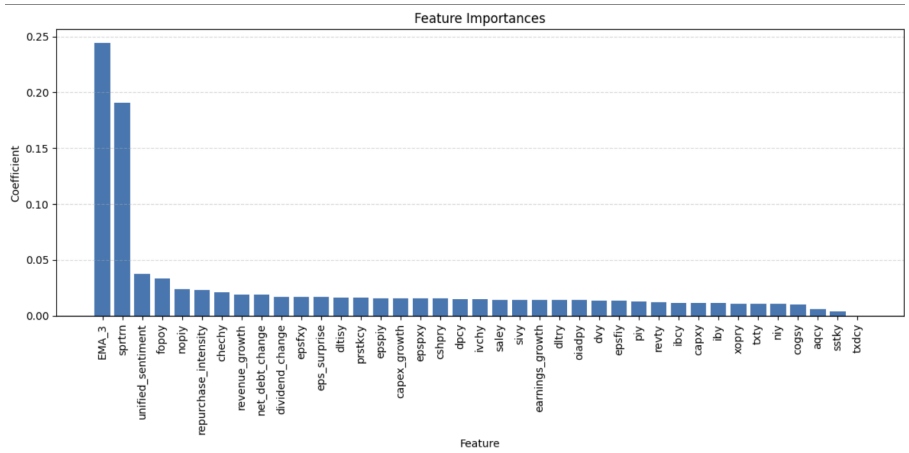

Figure 3: Model Weight Repartition of Predictors for a given CUSIP

Furthermore, GradientBoost is one of the models that has the best weight usage repartition, basing itself strongly on `EMA_3` but still using other features such as `sprtrn`, `fopoy` and `unified_sentiment`

5 Trading Strategy

The strategy integrates return predictions with news sentiment to construct a monthly long-short equity portfolio. We employ a state-of-the-art forecasting model as the primary signal, refined by financial news sentiment to reduce false positives and improve the Sharpe ratio, as shown in prior studies like [4].

Positions are initiated only when quantitative forecasts and qualitative sentiment align: going long on stocks with strong positive forecasts and non-negative sentiment, and shorting those with strongly negative forecasts combined with negative sentiment. Conflicting signals, such as a bullish prediction accompanied by negative sentiment result in skipped trades or reduced weights due to uncertainty.

This integration leverages multiple data sources, aligning with evidence that news analytics enhance the predictive power of quantitative models. For example, [5] showed significant performance improvements when sentiment was incorporated into LSTM-based forecasts.

Thus, sentiment serves as a quality filter, increasing confidence in model signals and decreasing false positives. The long-short structure captures alpha from individual stocks while hedging broader market movements, enabling a potentially market-neutral, alpha-focused approach.

6 Trading Strategy

6.1 Entry Rules

We define clear criteria for taking positions each month. A **long** position is initiated when predicted returns rank in the top 20% across all stocks and sentiment is positive, combining high expected returns with bullish sentiment. A **short** position is taken when predicted returns rank in the bottom 20% and sentiment is negative. When model predictions and sentiment signals conflict, we implement **no trade** as these discrepancies indicate low-confidence signals.

6.2 Exit Rules

Positions are reviewed monthly in alignment with our prediction horizon. By default, all positions are closed at month-end and reassessed based on updated signals. A **stop-loss** mechanism closes positions intra-month if losses exceed -10%. Additionally, **news-driven exits** occur mid-month if substantial sentiment shifts are detected. This systematic approach ensures portfolio construction always reflects the latest model and sentiment signals.

6.3 Risk Management

Volatility-Based Position Sizing

To improve risk-adjusted performance, we implement volatility-scaled position sizing so that each stock contributes more equally to portfolio risk. Positions are sized inversely proportional to their recent volatility, using a 6-month historical standard deviation of returns as our volatility proxy. For each selected stock, we begin with an equal weight, then scale it by 1/Volatility before normalizing all weights to maintain target total exposure. This approach ensures each position contributes approximately equally to overall portfolio variance and prevents high-volatility stocks from dominating risk. Additionally, this dynamic sizing allows the portfolio to adapt as market volatility changes, automatically reducing position sizes when volatility rises.

Diversification and Correlation Limits

We enforce strict diversification rules to avoid concentrated bets that could result in large losses. Beyond limiting exposure to any single stock or sector, we actively monitor correlations among selected positions. When many selected stocks exhibit high correlations, we recognize that the portfolio's effective risk is higher than it appears and adjust accordingly. In cases where two stocks have a correlation of 0.9, we avoid treating them as separate equal-weighted positions, instead halving their weights or removing one entirely to prevent redundancy. This approach ensures genuine diversification through uncorrelated exposures rather than merely spreading capital across nominally different securities.

7 Conclusion

Market prediction is arguably the most exciting, mysterious and lucrative part of finance. Given ML's recent advancements it should be no surprise that it is frequently used in order to make such predictions. However, one quickly realizes that if there were to be such a model it would not be public knowledge, since otherwise it would lose its edge anyways.

Once one couples the fact that markets return might sometimes been perceived as almost "random", the struggle and competitiveness present in this field become apparent and explain the lack of mind-blowing results (barely 10% above the baseline).

However we believe the paper using QGAF is one of these non-conventional approaches that have solid scientific backing and could lead to a concrete improvement in our results which is why we'd like to explore in the future the idea of using these concepts again with data augmentation as yet another feature for our more traditional ML models to use.

8 External Libraries

Our data processing pipeline relied on the following Python libraries:

- **pandas (1.3.5)**: Core data manipulation and analysis
- **numpy (1.21.5)**: Numerical operations and handling of missing values
- **matplotlib (3.5.1)** and **seaborn (0.11.2)**: Data visualization
- **pathlib (1.0.1)**: File path handling for data import and export
- **Scikit-Learn (Latest)**: ML methods and classes
- **plotly (Latest)**: Extra library for visualisation

A Entry Rules

Model Prediction	**FinBERT Sentiment**	**Trading Action**
Among highest predicted returns	Positive	Go Long
High predicted return	Neutral sentiment	Go Long
High predicted return	Negative sentiment	No Trade
Among lowest predicted returns	Negative	Short Sell
Low/negative predicted return	Neutral sentiment	Short Sell
Low/negative predicted return	Positive sentiment	No Trade

Table 5: Entry Rules Combining Model and Sentiment Signals

References

[1] Yanzhao Zou and Dorien Herremans. "PreBit — A multimodal model with Twitter FinBERT embeddings for extreme price movement prediction of Bitcoin". In: *Expert Systems with Applications* 233 (Dec. 2023), p. 120838. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2023.120838. URL: http://dx.doi.org/10.1016/j.eswa.2023.120838.

[2] Dogu Araci. *FinBERT: Financial Sentiment Analysis with Pre-trained Language Models*. 2019. arXiv: 1908.10063 [cs.CL]. URL: https://arxiv.org/abs/1908.10063.

[3] Yung-Hsu Yang Tsai et al. "QWGA: Quantum Walk Gated Architecture for Time Series Forecasting". In: *arXiv preprint arXiv:2310.07427* (2023).

[4] Cem Kirtac and Nicola Germano. "Sentiment Trading with Large Language Models". In: (2024).

[5] Wei Zeng and Hao Jiang. "Financial Sentiment Analysis using FinBERT in Predicting Stock Movement". In: (2023).