

Sentiment-driven prediction of financial returns: a Bayesian-enhanced FinBERT approach

Raffaele Giuseppe Cestari and Simone Formentin

Abstract—Predicting financial returns accurately poses a significant challenge due to the inherent uncertainty in financial time series data. Enhancing prediction models’ performance hinges on effectively capturing both social and financial sentiment. In this study, we showcase the efficacy of leveraging sentiment information extracted from tweets using the FinBERT large language model. By meticulously curating an optimal feature set through correlation analysis and employing Bayesian-optimized Recursive Feature Elimination for automatic feature selection, we surpass existing methodologies, achieving an F1-score exceeding 70% on the test set. This success translates into demonstrably higher cumulative profits during backtested trading. Our investigation focuses on real-world SPY ETF data alongside corresponding tweets sourced from the StockTwits platform.

I. INTRODUCTION

Predicting the sign of financial asset returns with precision remains a challenging task ([1]). Given the human-centric nature of finance, textual data related to financial topics represents valuable information that can enhance prediction models. In the literature, Natural Language Processing (NLP) techniques have been extensively employed for sentiment extraction, employing deep learning models [2], [3] as well as lexicon-based approaches [4]. Notably, the VADER (Valence Aware Dictionary and sEntiment Reasoner) algorithm [5] has been utilized for sentiment classification, demonstrating its efficacy in predicting price movements, as exemplified in the case of Microsoft stock.

A comprehensive review of sentiment analysis techniques in the financial domain is provided by [6]. Gupta and coauthors in [7] emphasize the importance of sentiment data volume in enhancing prediction accuracy, highlighting financial tweets sourced from platforms like Twitter (now X) or StockTwits as primary data sources [8], [9].

In this study, we investigate the influence of both model selection and input features on prediction quality using real StockTwits tweets associated with SPY ETF financial time series. We begin with a feature set outlined in [3] and employ an automatic feature selection algorithm, Bayesian-optimized

This paper is partially supported by FAIR (Future Artificial Intelligence Research) project, funded by the NextGenerationEU program within the PNRR-PE-AI scheme (M4C2, Investment 1.3, Line on Artificial Intelligence), by the Italian Ministry of Enterprises and Made in Italy in the framework of the project 4DDS (4D Drone Swarms) under grant no. F/310097/01-04/X56 and by the PRIN PNRR project P2022NB77E “A data-driven cooperative framework for the management of distributed energy and water resources” (CUP: D53D23016100001), funded by the NextGeneration EU program.

The authors are with Department of Electronics, Information, and Bioengineering, Politecnico di Milano, Milano, Italy. Email to: raffaelegiuseppe.cestari@polimi.it.

Recursive Feature Elimination (BO-RFE), to extract the most informative subset of features. This process combines Recursive Feature Elimination (RFE) with Bayesian Optimization (BO) to iteratively converge on the optimal feature set, evaluated based on the F1-score. Additionally, we conduct a correlation analysis to elucidate the dynamic impact of sentiment, as classified by FinBERT [10], on returns. This analysis enriches the set of regressors and aids in top-feature selection.

We compare three architectures, employing the same prediction model but varying in input features and training window size:

- *Literature*: Benchmark architecture as described in [3].
- *BO-RFE-2* (2 regressors): Literature architecture with a reduced set of input features obtained from BO-RFE.
- *BO-RFE-5* (5 regressors): Literature architecture with a reduced set of input features obtained from BO-RFE, augmented with additional correlation-based regressors.

Our findings demonstrate that:

- Utilizing BO-RFE/correlation-based feature selection with the same prediction model (Support Vector Machine, SVM) yields superior results compared to the benchmark [3] in terms of F1-score (above 70%) and simulated profits in real trading days.
- Correlation analysis significantly contributes to feature selection by capturing slow sentiment dynamics.
- Sentiment-based features play a crucial role, as evidenced by the shift from 3 out of 10 sentiment features in the benchmark to 4 out of 5 in our proposed BO-RFE-5 architecture.
- Reducing features to essential elements enables a shorter training window, resulting in better results with less data and ensuring model reactivity.

The remainder of this paper is organized as follows: Section II presents the characteristics of the dataset, followed by Section III detailing the FinBERT classification of tweets and sentiment index computation. Section IV outlines the automatic feature selection procedure, while Section V discusses the correlation analysis and the enriched set of candidate optimal regressors. Section VI describes the prediction model structure, and Section VII presents the numerical simulations and results, including return sign prediction metrics on the test set (Section VII-A) and simulated profits in actual trading days (Section VII-B).

II. THE DATASET

The data on which this article is based are offered by [3]. The sources of information are:

- The financial time series of the SPDR S&P 500 ETF (SPY), a fund replicating the S&P 500 index including the IT, finance, energy, and TLC sectors (the 500 largest US companies). It implements automatic share balancing based on stock value.
- A collection of 3,261,867 tweets gathered through StockTwits API, tagged SPY, containing the users' sentiment information on the same SPY asset.

The temporal horizon is from 2020-03-23 to 2022-03-14. To avoid redundancy, please refer to [3] for further details.

III. TWEETS CLASSIFICATION AND SENTIMENT SCORING

FinBERT, introduced by Araci et al. [10], is a specialized variant of the Google BERT (Bidirectional Encoder Representations from Transformers) model [11], specifically tailored for financial text analysis. BERT itself is a sophisticated language model built upon the transformer architecture, featuring 12 encoders with 12 bidirectional self-attention heads, constituting either 110 million parameters or, in its larger variant, 24 encoders with 16 bidirectional self-attention heads, amounting to 340 million parameters.

FinBERT is fine-tuned using a substantial corpus of 4.9 billion textual tokens extracted from corporate and analyst reports, as well as earnings call transcripts, notably leveraging the *Financial Phrasebank* corpus [12]. This fine-tuning process enables FinBERT to discern financial sentiment within textual inputs. The model classifies input text into one of three categories: positive, negative, or neutral, reflecting the underlying financial implications conveyed in the text. Positive sentiment typically aligns with positive market trends, while negative sentiment may indicate market downturns, and neutral sentiment suggests market indifference.

In the context of analyzing StockTwits tweets related to the SPY ETF, depicted in Figure 1, FinBERT exhibits an imbalance in classification. Approximately 80% of tweets are classified as neutral, while roughly 8% are identified as carrying discernible financial sentiment, encompassing both positive and negative sentiments. This outcome aligns with expectations, given the nature of publicly available social media data, where genuine financial insights are relatively scarce. Nevertheless, even neutral classifications hold significance, as they reflect market sentiment and hype. Additionally, the volume of tweets on a particular day serves as valuable information in itself.

To quantify sentiment, akin to the approach proposed in [13], we compute a sentiment score S_t using

$$S_t = \frac{P_t - N_t}{T_t}, \quad (1)$$

where P_t represents the number of positive tweets, N_t signifies the number of negative tweets, and T_t denotes the total number of tweets (including those labeled as neutral) on day t . This sentiment score provides a concise metric to gauge overall sentiment trends within the analyzed dataset.

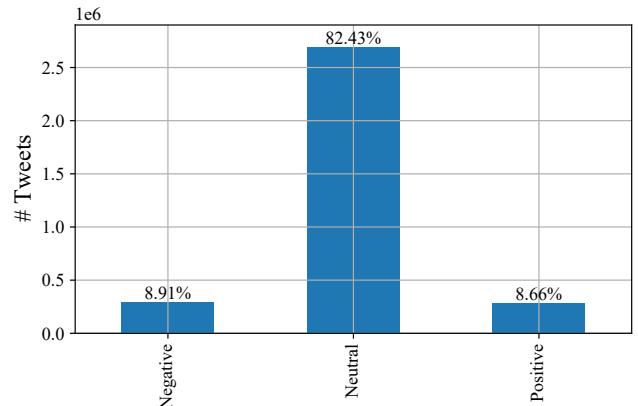


Fig. 1. FinBERT sentiment tweets classification.

IV. BAYESIAN-OPTIMIZED RECURSIVE FEATURE ELIMINATION

This section delineates the Bayesian-optimized Recursive Feature Elimination (BO-RFE) methodology. This automated algorithm enables the extraction of an optimal subset of features from a feature collection, ensuring maximal informativeness on the dataset while adhering to a predefined cost function.

Recursive Feature Elimination (RFE) operates with two configurable parameters: the target number of features to retain from the original set and the choice of kernel algorithm for feature selection. It is essential that the kernel algorithm produces a linear ranking of features; therefore, algorithms like SVM with radial basis function kernels are unsuitable for RFE due to their non-linear feature ranking.

The workflow of the algorithm is as follows: after selecting the kernel, it is trained, and feature importance is assessed based on the linear ranking. The least important features are iteratively removed, and the kernel algorithm is re-trained with the remaining features until the desired number of features is achieved.

Our approach merges RFE with Bayesian Optimization (BO). RFE enables the identification of the optimal number of features γ and the set of dominant features Γ , given a target feature count and a kernel model. γ , a hyperparameter for RFE, is tuned alongside the parameter set Θ of the kernel model. BO automatically tunes γ (and thereby Γ) and Θ . The objective function for BO is the F1-score obtained on a reduced test set spanning 30 days ($T^{test}BO$), using the same prediction model employed in the simulation as outlined in the literature [3], after training on 222 days ($T^{train}BO$). Bayesian Optimization is a widely recognized hyperparameter tuning approach; for detailed insights, please consult [14].

Remark: This procedure occurs within the training/validation set described in Section VII. The values of γ_k and Θ_k will maximize, along the BO iterations $k = 1, \dots, K$, the F1-score of the return sign prediction, eventually converging at the best γ_K , Γ_K and Θ_K .

Algorithm 1 Bayesian-optimized Recursive Feature Elimination Algorithm (BO-RFE)

```

1: for  $k = 1, \dots, K$  do
2:    $\gamma_k, \Gamma_k = RFE(\Theta_k)$ .
3:   for  $\tau = 1, \dots, T_{BO}^{test}$  do
4:      $\hat{y}_\tau = SVM(\gamma_k, \Gamma_k)$ 
5:   end for
6:   F1-score([ $y_1, \dots, y_{T_{BO}^{test}}$ ], [ $\hat{y}_1, \dots, \hat{y}_{T_{BO}^{test}}$ ])
7: end for
8: Return  $\gamma_K, \Gamma_K$ 
```

Algorithm 1 delineates the BO-RFE process. Here, k denotes the BO iteration index, while τ represents the relative time index within the reduced test set. y_τ and \hat{y}_τ signify the true and predicted return signs at time τ , respectively.

The RFE kernel is a *random forest classifier*, with the optimally identified hyperparameter Θ representing the *number of trees* within the forest. Random forest classifiers are established machine learning models wherein the predicted class is determined through a democratic election based on the predictions from Θ_K trees within the forest. Further insights into random forest classifiers can be found in [15].

For BO convergence, we set K to 50, which proves to be sufficient. The initial feature set size is 10, consistent with the feature set utilized in [3], summarized in Table II.

The iterative procedure yields an optimal number of trees $\Theta_K = 5$, an optimal number of features $\gamma_K = 2$, and the corresponding optimal feature set $\Gamma_K = [R_{t-1}, S_t \text{ pre-market}]$. The achieved F1-score on the reduced test set T_{BO}^{test} stands at 75%.

V. CORRELATION ANALYSIS

In this section we present the correlation analysis that allowed us to prove the meaningfulness of additional candidate regressors to the subset identified by the *BO-RFE* algorithm. Table I shows the Pearson correlation coefficient between the number of negative N_t , neutral n_t and positive P_t FinBERT-classified StockTwits tweets and SPY return R_t and traded volume V_t , at the same day t . We highlight that negative tweets carry higher information content with respect to positive and neutral tweets. This reflects the interpretable behaviour that in negative scenarios the number of financially negative tweets increases, describing market hype-to-sell and ultimately leads to increased traded stock volumes and lower returns. This analysis motivates our choice of including in the regressors set the number of negative FinBERT-classified tweets.

Pearson [-]	# Tweets (-) N_t	# Tweets (=) n_t	# Tweets (+) P_t
R_t	-0.06	-0.04	0.035
V_t	0.64	0.57	0.55

TABLE I

PEARSON CORRELATION COEFFICIENT BETWEEN NUMBER OF NEGATIVE, NEUTRAL AND POSITIVE TWEETS (FINBERT LABELED) AND RETURN AND STOCK VOLUME, AT THE SAME TIME INSTANT.

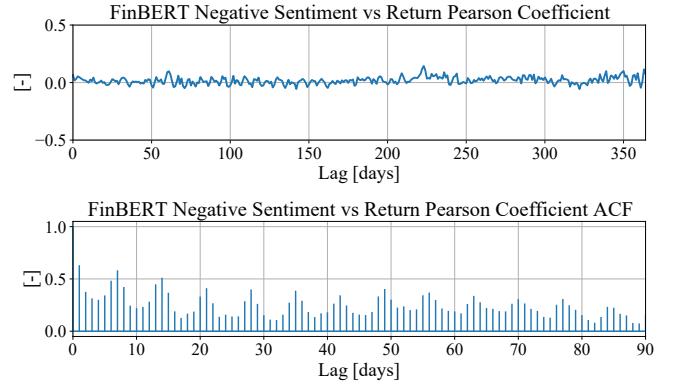


Fig. 2. Upper panel: FinBERT negative sentiment vs return Pearson correlation coefficient for different time lags. Lower panel: Pearson autocorrelation function.

Figure 2 upper panel shows the Pearson correlation coefficient between FinBERT negative sentiment and SPY return for different time lags. In the lower panel, the Pearson autocorrelation function (ACF) is shown. Both upper and lower panel depicts a periodic behaviour in the Pearson coefficient for various lags. In particular the ACF reflects a weekly seasonality in the correlation between sentiment and returns with periodic peaks. This allows us to conclude that impact of sentiment information on return has a slow dynamics with weekly period. This analysis, jointly with the results in Table I lead us to the choice of the additional set of regressors:

- S_{t-7} intra-market, to capture the seasonal behaviour of sentiment index in intra-market time.
- S_{t-7} post-market, to capture the seasonal behaviour of sentiment index in post-market time.
- N_{t-7} , to capture the seasonal behaviour of number of negative tweets.

These 3 additional regressors, together with the ones identified through the BO-RFE procedure, constitute the set we propose as best configuration for the return prediction problem of SPY ETF.

For sake of compactness, we summarize in Table II the feature sets for the 3 strategies: F_{t-1} represents a synthetic

Strategy	Literature	BO-RFE - 2	BO-RFE - 5
R_{t-1}	yes	yes	yes
S_t pre-market	yes	yes	yes
S_{t-1} intra-market	yes	no	no
S_{t-7} intra-market	no	no	yes
S_{t-1} post-market	yes	no	no
S_{t-7} post-market	no	no	yes
F_{t-1} [6 financial features]	yes	no	no
N_{t-7}	no	no	yes

TABLE II
FEATURE SETS FOR Literature, BO-RFE-2 AND BO-RFE-5 ALGORITHMS.

notation for the 6 financial features used exclusively in *Literature* startegy, taken at day $t - 1$ with the goal of

predicting the return sign at time t . The financial features used in literature are: opening, closing, high and low prices, stock volume and trade value. For additional details see [3].

VI. THE PREDICTION MODEL

The prediction model is the same used in [3] and it is used both in the BO-RFE algorithm 1 and during online return sign predictions. It consists in a support vector machine (SVM) with the combination of synthetic minority oversampling technique (SMOTE, see [16]) to address class imbalance (negative return sign days are around 300 whereas positive days are around 420) and bagging (a procedure that performs bootstrapping on original data and trains L classifiers in parallel to eventually generate a democratic ensemble classifier for return sign prediction, resembling the random forest classifier concept, see [17]).

The SVM (SVC if we consider classification) training problem is the following.

$$\min_{w,b,\xi} \left\{ \frac{1}{2} w^T \cdot w + C \sum_{i=1}^n \xi_i \right\} \quad (2a)$$

$$y_i(w^T \cdot \phi(x_i) + b) \geq 1 - \xi_i \quad (2b)$$

$$\xi_i \geq 0 \quad \forall i = 1, \dots, n \quad (2c)$$

$$x_i, w \in R^P \quad \forall i = 1, \dots, n \quad (2d)$$

$$b \in R, \xi_i \in R^n \quad \forall i = 1, \dots, n \quad (2e)$$

Where P is the number of features, n is the number of samples, x_i is the i -th input of P features among the n samples, w are the model weights, b is the bias, ξ is a slack variable for weakly constraining the hyperplanes, C and $\phi(x_i)$ (kernel) are SVC hyperparameters. Following [18], the machine implements the idea that input vectors are non-linearly mapped to a very high-dimension feature space. In this space a separating hyperplane is constructed, allowing to distinguish samples among different target classes exploiting non-linear relationships which otherwise would have been hidden. The model degrees of freedom depend on C , the penalty weight of the slack variables, and $\phi(x_i)$, the kernel of the SVM. In this application, we strictly stick with the hyperparameter choices $C = 1$ and *radial basis function* kernel, carried on in [3], to provide a fair comparison for the same SVM model. We highlight that the number of model parameters (in particular model weights w) depends on the number of features P . For this reason, a lower number of features as in the case of *BO-RFE-2* and *BO-RFE-5* ultimately leads to a lower amount of model parameters, thus requiring a smaller training dataset.

VII. EXPERIMENTAL RESULTS

We perform the numerical simulations with real StockTwits and SPY ETF data from 2020-03-24 to 2022-02-13. With the purpose of aligning StockTwits tweet data (available each day) and the financial return data (available only during work days) we carried on a linear interpolation during weekend days on financial data. This allows us to

capture the textual data available in the weekends that can effectively affect the market during the work days, due to financial news made available during saturday and sunday. However, during the online trading simulation the trading actions (buy/sell) will be applied exclusively during work days. All computations are carried out on an Intel Core i7-8750H with 6 cores, at 2.20 GHz (maximum single core frequency: 4.10 GHz), with 16 GB RAM. Code is implemented in Python 3.7.11. Training and validation sets cover the 84% of the data. Test set is the remaining 16%. Training and validation sets are merged since training is done in a moving window (with size W) fashion. Table III collects each strategy setting. The only hyperparameter tuned for each strategy is the moving window size W . For each strategy, we carried on a sensitivity analysis on the training/validation sets for different window sizes, picking up the best window W for each strategy. *Literature* best window size confirms the range described in [3] where the authors tested the window range between 232 and 252. Indeed, the identified best window range for *Literature* is 240. For *BO-RFE-2* and *BO-RFE-5* strategies instead, the best window size is 40 and 210, respectively. This window size reduction depends on the lower number of regressors involved in BO-RFE strategies. Following SVM model structure, a lower number of features ultimately implies a lower number of parameters and hence a lower optimal number of data required in training. This allows us to conclude that *BO-RFE-5* strategy requires 1 month less of training data with respect to its literature benchmark. This is an interesting point for an online application of the algorithm as it requires less training data (and therefore a shorter warm-up period), moreover, it guarantees higher reactivity properties given the shorter required past horizon.

Strategy	Literature	BO-RFE - 2	BO-RFE - 5
Window W [days]	240	40	210
# Features	10	2	5
# Financial Features	7	1	1
# Sentiment Features	3	1	4

TABLE III
STRATEGY SETTINGS - HYPERPARAMETERS AND NUMBER OF (FINANCIAL AND SENTIMENT) FEATURES.

We remark that the contribution of sentiment-based features is relevant in *BO-RFE* strategies. Indeed, they constitute the 50% of the features in *BO-RFE-2* and the 80% in *BO-RFE-5*, differently from *Literature* where the sentiment features are only the 30% of the total. Given the promising results of *BO-RFE*, this proves the huge impact of sentiment-based information, processed through FinBERT transformer architecture, to generate meaningful regressors for financial return series prediction.

A. RETURN SIGN PREDICTION

Table IV collects accuracy, precision, recall and F1-score for the 3 strategies in the test set of 80 trading days (110 days, including non-trading weekends). *BO-RFE-5* achieves

outstanding performances with an accuracy (e.g., percentage of correctly labeled data, $accuracy = \frac{TP+TN}{TP+TN+FP+FN}$) of the 64.1%, a precision (e.g., percentage of correctly labeled data over the total labeled, for each class, $precision = \frac{TP}{TP+FP}$) above 63%, a recall (e.g., percentage of correctly labeled data over the total number of data belonging to the same class, $recall = \frac{TP}{TP+FN}$) above 78% and a F1-score (e.g., synthetic measure of precision and recall, $F1\text{-score} = \frac{2 \cdot precision \cdot recall}{precision + recall}$) above 70%. Figure 3 shows the test set

Strategy	Literature	BO-RFE - 2	BO-RFE - 5
Accuracy	0.564	0.512	0.641
Precision	0.583	0.541	0.634
Recall	0.667	0.619	0.785
F1-score	0.622	0.577	0.702

TABLE IV

CLASSIFICATION METRICS IN TEST SET FOR LITERATURE, BO-RFE - 2 AND BO-RFE - 5 ARCHITECTURES.

F1-score, for 8 batches of 10 trading days (2 weeks including weekends). *BO-RFE-5* outstands *Literature* and *BO-RFE-2*. Its scores show higher mean and quartiles, demonstrating a consistent improvement over all the test set. *Literature* scores confirm those described in [3]. These results demonstrate that

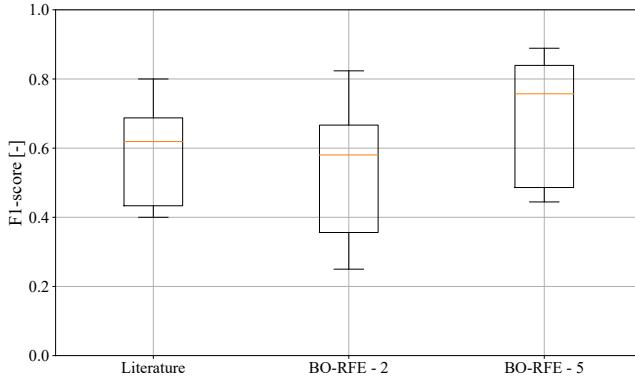


Fig. 3. F1-score on 8 batches of 10 trading days each in test set.

the careful choice of features, considering the most significant ones identified in the literature and a correlation analysis that evaluates the potential contribution of past regressors, is decisive in prediction quality. The results in particular prove that the contribution of the regressors S_{t-7} intra-market, S_{t-7} post-market, and N_{t-7} is significant. Compared to the BO-RFE-2 feature set, BO-RFE-5 regressors significantly improve performance, demonstrating the information content of the sentiment information at lag 7 and the slow dynamics of user social media interactions on the financial asset. Furthermore, it confirms the informativeness of using not only the sentiment index S_t described in equation (1) but also the number of negative FinBERT-classified tweets in predicting returns.

B. TRADING SIMULATION

In this section we apply the strategies in a trading simulation. We show that, for the same return sign prediction model,

the profit magnitude depends on the quality of the feature selection. The trading strategy is the same described in [1] and it is as follows: if the predicted sign is positive (negative), the automatic trading algorithm will purchase (short-sell) SPY ETF for a value of 10.000 \$. If the actual sign matches the predicted one, the trader has a profit; otherwise, the trader incurs in a loss. No transaction costs are considered. Figure 4, upper panel, shows the cumulative profit time

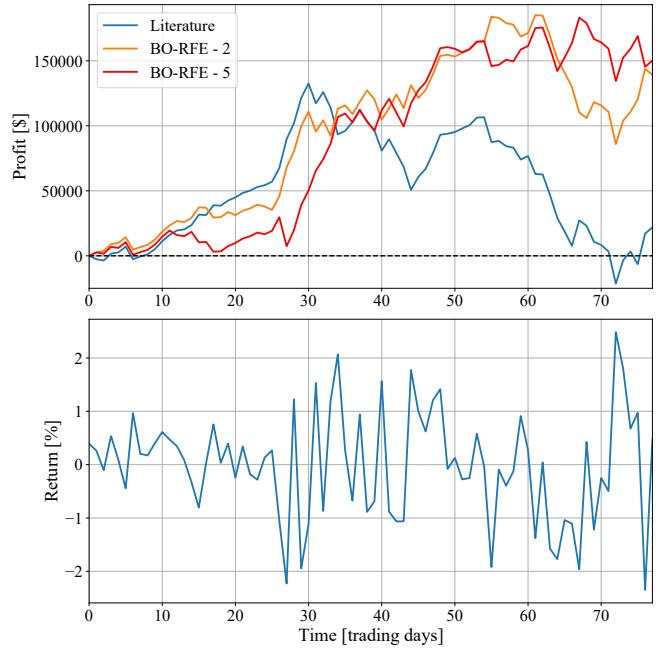


Fig. 4. Upper panel: cumulative profit time series in test set. Lower panel: SPY return.

series obtained with the different trading algorithms in the test set. Before day 30, return absolute value is bounded (as it is clear in the lower panel) below 1%. The strategies have lower profit and *Literature* outperforms. From day 30 onward, the return absolute value is higher (with higher oscillations around the mean, e.g., higher volatility). This allows the strategies to cumulate profit, proportionally to their accuracy in predicting the return sign. This ultimately leads to outstanding performance for *BO-RFE-5*. The ranking shows that both the strategies *BO-RFE-2* and *BO-RFE-5* have higher profit. This depends on their higher accuracy earned in the more volatile period, whereas the *Literature* model accuracy is "wasted" in the beginning of the simulation horizon, where the returns are lower and, eventually leads to a lower cumulative profit, slightly above 10.000 \$ in the test set considered.

VIII. CONCLUSIONS

In this study, we explore the influence of feature selection on the performance of return sign prediction for the SPY, an ETF tied to the S&P 500 index, using the same prediction model. Employing an automated feature selection technique, BO-RFE, we combine the Recursive Feature Elimination (RFE) procedure with Bayesian Optimization (BO) to extract

an optimal subset of features from those identified in existing literature.

We conduct a correlation analysis to unveil the sluggish, weekly seasonality-driven dynamics of sentiment effects on returns, particularly focusing on sentiments labeled as negative by the LLM FinBERT. Integrating this insight as additional regressors, we demonstrate how the choice of input features significantly impacts prediction quality and, consequently, achievable profits with different trading strategies on a test set. Notably, the *BO-RFE-5* solution exhibits a substantial performance enhancement.

Our findings underscore the predominance of sentiment-based features among the optimal set, *highlighting the significant influence of textual information on financial assets*. While the state-of-the-art solution comprises 3 out of 10 sentiment-based features, our *BO-RFE-5* method increases this ratio to 4 out of 5. Additionally, *the selection of “essential” features leads to model simplification, resulting in a smaller optimal training window and, consequently, enhanced responsiveness and shorter warm-up periods*.

Future endeavors will involve validating our proposed methodology across a broader testing horizon and exploring its applicability to different financial assets.

REFERENCES

- [1] R. G. Cestari, F. Barchi, R. Busetto, D. Marazzina, and S. Formentin, “Hawkes-based cryptocurrency forecasting via limit order book data,” *arXiv preprint arXiv:2312.16190*, 2023.
- [2] Y. Xu and V. Keselj, “Stock prediction using deep learning and sentiment analysis,” in *2019 IEEE international conference on big data (big data)*. IEEE, 2019, pp. 5573–5580.
- [3] J.-X. Liu, J.-S. Leu, and S. Holst, “Stock price movement prediction based on stocktwits investor sentiment using finbert and ensemble svm,” *PeerJ Computer Science*, vol. 9, p. e1403, 2023.
- [4] P. Koukaras, C. Nousi, and C. Tjortjis, “Stock market prediction using microblogging sentiment analysis and machine learning,” in *Telecom*, vol. 3, no. 2. MDPI, 2022, pp. 358–378.
- [5] C. Hutto and E. Gilbert, “Vader: A parsimonious rule-based model for sentiment analysis of social media text,” in *Proceedings of the international AAAI conference on web and social media*, vol. 8, no. 1, 2014, pp. 216–225.
- [6] X. Man, T. Luo, and J. Lin, “Financial sentiment analysis(fsa): A survey,” in *2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*, 2019, pp. 617–622.
- [7] R. Gupta and M. Chen, “Sentiment analysis for stock price prediction,” in *2020 IEEE conference on multimedia information processing and retrieval (MIPR)*. IEEE, 2020, pp. 213–218.
- [8] A. Mittal and A. Goel, “Stock prediction using twitter sentiment analysis,” *Stanford University, CS229*, vol. 15, p. 2352, 2012.
- [9] V. S. Pagolu, K. N. Reddy, G. Panda, and B. Majhi, “Sentiment analysis of twitter data for predicting stock market movements,” in *2016 international conference on signal processing, communication, power and embedded system (SCOPES)*. IEEE, 2016, pp. 1345–1350.
- [10] D. Araci, “Finbert: Financial sentiment analysis with pre-trained language models,” *arXiv preprint arXiv:1908.10063*, 2019.
- [11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [12] P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala, “Good debt or bad debt: Detecting semantic orientations in economic texts,” *Journal of the Association for Information Science and Technology*, vol. 65, no. 4, pp. 782–796, 2014.
- [13] J. Z. G. Hiew, X. Huang, H. Mou, D. Li, Q. Wu, and Y. Xu, “Bert-based financial sentiment index and lstm-based stock return predictability,” *arXiv preprint arXiv:1906.09024*, 2019.
- [14] I. Dewancker, M. McCourt, and S. Clark, “Bayesian optimization for machine learning: A practical guidebook,” *arXiv preprint arXiv:1612.04858*, 2016.
- [15] A. Parmar, R. Katariya, and V. Patel, “A review on random forest: An ensemble classifier,” in *International conference on intelligent data communication technologies and internet of things (ICICI) 2018*. Springer, 2019, pp. 758–763.
- [16] A. Fernández, S. García, F. Herrera, and N. V. Chawla, “Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary,” *Journal of artificial intelligence research*, vol. 61, pp. 863–905, 2018.
- [17] M. Skurichina and R. P. Duin, “Bagging for linear classifiers,” *Pattern Recognition*, vol. 31, no. 7, pp. 909–930, 1998.
- [18] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, pp. 273–297, 1995.