# Effect of labeling algorithms on financial performance metrics

Tomislav Kovačević*, Sven Goluža†, Andro Merćep‡ and Zvonko Kostanjčar§
*University of Zagreb, Faculty of Electrical Engineering and Computing*
*Laboratory for Financial and Risk Analytics*
*Unska 3, 10000 Zagreb, Croatia*
Email: *tomislav.kovacevic@fer.hr, †sven.goluza@fer.hr, ‡andro.mercep@fer.hr, §zvonko.kostanjcar@fer.hr

***Abstract*—Machine learning models are increasingly common in predicting financial market movements. Unlike some other research areas, labels of financial time series are not unambiguously determined, which is why multiple labeling algorithms were proposed. The effect of a particular labeling algorithm on a trading strategy is often overlooked as most existing research uses only one type of labels to develop a machine learning model used for trading. However, different labeling algorithms lead to different generalization errors that may impede financial performance of a strategy. This paper examines the relationship between the classification performance of a model and the financial performance of a strategy based on the same model. The relationship is examined for two commonly used labeling algorithms: fixed-time horizon and triple-barrier method. Although the results for both labeling algorithms confirm a statistically significant correlation between classification and financial performance, the correlation coefficient itself has a low value.**

***Index Terms*—financial time series, stock prediction, machine learning, labeling algorithms**

## I. Introduction

The widespread application of machine learning models (especially supervised learning) in various scientific disciplines has led to the increasing use of such models for predicting market movements [1]. To date, research mostly addressed the question of whether supervised learning models can perform better than standard statistical approaches in predicting market trend, returns, and volatility [2], [3]. The attention has been paid to the selection and comparison of models and different feature combinations, while less attention has been paid to the financial time series labeling. However, unlike some other applications of machine learning models, where the dependent variable is selected by an expert in the field and is unambiguous, in financial time series it can be defined in different ways, and the labeling is done algorithmically. Although several labeling algorithms have been proposed [4]–[6], the impact of a particular algorithm on a trading strategy is often opaque, as misclassification of different labels can affect trading performance differently. In this paper, we aim to investigate the relationship between the classification performance of a model and the financial performance of a strategy based on the same model.

The rest of the paper is organized as follows: In Section II, we briefly explain the machine learning model, the feature generating process, the labeling algorithms, and the data used to conduct experiments. In Section III we explain the experimental setup – how we train the model, and how we backtest the machine learning based trading strategy. The results are presented in Section IV, where we also provide a discussion and interpretation of the results.

## II. Methodology

### A. Model

Gradient Boosting is a popular machine learning algorithm for structured tabular data because it can be trained quickly and often outperforms even deep learning models [7]. As with any supervised learning method, the goal is to minimize a loss function $L(y, f(\mathbf{x}))$ given a training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$. In the gradient boosting model, the hypothesis function $f$ is defined as the weighted sum of the outputs of $M$ weak learners $F_i$ belonging to the same model class (such as shallow decision trees), which can be formed as:

$$f(\mathbf{x}) = \sum_{i=1}^{M} \beta_i F_i. \tag{1}$$

The boosting algorithm is trained by sequentially fitting additive models, where each $F_i$ is a binary classifier. After training a weak learner $F_i$ on the training set, we increase the weight of the training examples that were misclassified by $F_i$. The next weak learner $F_{i+1}$ is then trained on the newly weighted training set and the procedure is repeated designated number of times. Once all classifiers have been trained, their predictions are combined using weighted majority voting as in (1). Gradient boosting models are trained in a similar manner by training a base learner $F_i$ on the gradient of the loss function of an existing model. The pseudocode for the training procedure is shown in Algorithm 1 [8]. Our trading strategy is based on Extreme Gradient Boosting model (XGBoost), a very efficient and widely used implementation of gradient-boosted trees [9].

### B. Features

Model features are divided into two categories: return-based and technical indicator-based. The former includes features generated using returns calculated from the closing price data described in paragraph II-D. The technical indicators category includes features generated using technical indicators from open-high-low-close (OHLC) price series [10]; these features

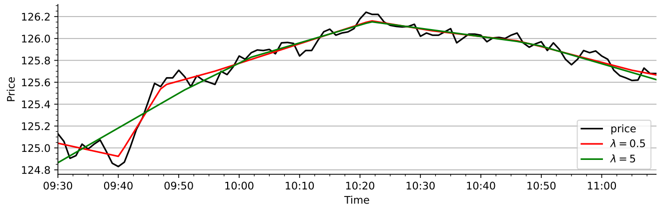

Fig. 1. $\ell_1$ trend filtered price series for two different $\lambda$ values.

are described in detail in Table I. Each row represents several features computed for raw and filtered time series with specific underlying parameters and lag operators. The $\ell_1$ trend filter is used to obtain a piecewise linear representation of an original time series as denoising price series previously proved beneficial [11], [12]. The trend estimate is chosen to minimize the weighted sum objective function:

$$L = \frac{1}{2}\sum_{t=1}^{n}(y_t - x_t)^2 + \lambda \sum_{t=2}^{n-1} |x_{t-1} - 2x_t + x_{t+1}|, \tag{2}$$

where $y_t$ is an element of the original time series at timestamp $t$, $x_t$ is an element of the filtered time series at timestamp $t$, and $\lambda \geq 0$ is the regularization parameter used to control the number of piecewise linear estimates. When $\lambda \to 0$, the representation converges to the original time series, while the representation converges to a single affine function when $\lambda \to \infty$ [13]. An example of $\ell_1$ trend filtered price series for two different $\lambda$ values is shown in Fig. 1. This combination of return-based and technical indicator-based features, along with lag operator and price series filtering yields 384 input features in total.

**Algorithm 1:** Gradient boosting [8].

```
Initialize $f_0(\mathbf{x}) = \arg\min_F \sum_{i=1}^{N} L(y_i, F(\mathbf{x}_i))$;
for $m$ = $1$ : $M$ do
    Compute the gradient residual:
        $$r_{im} = -\left[\frac{\partial L(y_i, f(\mathbf{x_i}))}{\partial f(\mathbf{x_i})}\right]_{f(\mathbf{x_i}) = f_{m-1}(\mathbf{x_i})}$$
    Train a weak learner $F_m$ using $r_{im}$ as labels:
        $$F_m = \arg\min_F \sum_{i=1}^{N}(r_{im} - F(\mathbf{x}_i))^2$$
    Compute weights:
        $$\beta_m = \arg\min_\beta \sum_{i=1}^{N} L(y_i, f_{m-1}(\mathbf{x}_i) + \beta F(\mathbf{x}_i))$$
    Update $f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + \beta_m F_m(\mathbf{x})$
end
Return $f(\mathbf{x}) = f_M(\mathbf{x})$
```

### C. Labels

The two frequently used labeling algorithms – fixed-time horizon method and triple-barrier method – are described in [15]. Both yield two-class labels $C \in \{0, 1\}$. These labels have the following meaning: positive class (label 1) represents long position and negative class (label 0) represents close position. This defines the trading strategy, where a change in model output $0 \to 1$ triggers a "buy" action, while a change $1 \to 0$ triggers a "sell" (of an existing position). See Fig. 2. for examples of labels obtained using these two algorithms. Both labeling algorithms are explained in detail along with their corresponding parameters in the following paragraphs.

*1) Fixed-time horizon method (FTHM):* Let $\boldsymbol{p} = \{p_0, p_1, \ldots\}$ be a price series (e.g. close prices). Future return for a fixed-time horizon sized $h$ is defined as:

$$r_t = \frac{p_{t+h} - p_t}{p_t} \tag{3}$$

The labels are then obtained by thresholding future return using some predefined level $\tau_{\text{long}}$:

$$l_t = \begin{cases} 1 & \text{if } r_t > \tau_{\text{long}}, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

This is the most commonly used method in the literature [14]. By setting a threshold for future return, machine learning models predict the trend of market dynamics (upward or downward). The modification of the proposed algorithm can be made by introducing additional threshold $\tau_{\text{short}}$, and therefore the strategy can enter both long and short positions.

*2) Triple-barrier method (TBM):* FTHM has several known problems, such as class imbalance and generally poor statistical properties [15]. TBM is proposed as an alternative solution. It dynamically determines the class threshold by taking into account previous price volatility.

At timestamp $t$, using look-back period sized $b$ we can calculate previous price volatility $\sigma_t$. Then, we can define take-profit price (or upper barrier) as a function of previous volatility $\sigma_t$ and upper barrier multiplier $U$:

$$p_{\text{TP}} = p_t \cdot (1 + U \cdot \sigma_t). \tag{5}$$

If we hit the take-profit price first, the label $l_t$ is set to 1. In contrast, we define stop-loss price (or lower barrier) as a function of previous volatility $\sigma_t$ and lower barrier multiplier $L$:

$$p_{\text{SL}} = p_t \cdot (1 - L \cdot \sigma_t), \tag{6}$$

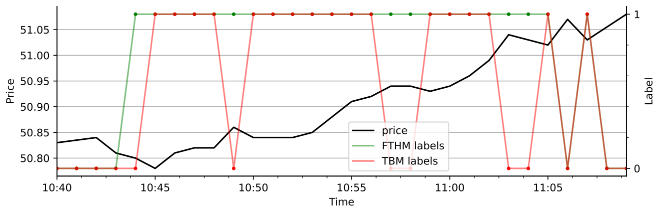

Fig. 2. Labels obtained using fixed-time horizon method and triple-barrier method for the same price series.

TABLE I
INPUT FEATURES USED FOR MODEL TRAINING.

| Feature | Category | Parameters |
|---|---|---|
| Return | return-based | {1,2,3,4,5,15,30}-minute returns |
| Variance | return-based | {1,2,3,4,5,15,30}-minute returns;<br>30 minute look-back period |
| Summed Exponential Decaying Return | tehnical indicator | {1,2,3,4,5,15,30}-minute returns;<br>30 minute look-back period;<br>0.9 decay factor |
| High-Low Spread | tehnical indicator | - |
| Commodity Channel Index (CCI) | tehnical indicator | {10, 15, 30} minute look-back period |
| Average Directional Movement Index (ADX) | tehnical indicator | {10, 15, 30} minute look-back period |
| Relative Strength Index (RSI) | tehnical indicator | {10, 15, 30} minute look-back period |
| Chande Momentum Oscillator (CMO) | tehnical indicator | {10, 15, 30} minute look-back period |
| Moving Average Convergence Divergence (MACD) | tehnical indicator | {10 ,15, 30} minute fast period;<br>{20, 30, 45} minute slow period;<br>9 minute signal period |
| Stochastic Oscillator (STOCH) | tehnical indicator | 30 minute look-back period;<br>{5, 15, 15} minute fast K-period;<br>{3, 5, 8} minute fast D-period;<br>{3, 5, 10} minute slow K-period;<br>{3, 5, 8} minute slow D-period |
| Ultimate Oscillator (ULTOSC) | tehnical indicator | {5, 7, 1} minute 1st period;<br>{15, 14, 30} minute 2nd period;<br>{30, 28, 45} minute 3rd period |
| Normalized Average True Range (NATR) | tehnical indicator | {10, 15, 30} minute look-back period |
| Accumulation/Distribution (AD) | tehnical indicator | 120 minute look-back period |
| Chaikin Oscillator (ADOSC) | tehnical indicator | 120 minute look-back period |
| On Balance Volume (OBV) | tehnical indicator | 120 minute look-back period |

and if we hit the stop-loss price fist, we set $l_t = 0$. Multipliers $U$ and $L$ are used to manage the risk of individual positions. Finally, if neither take-profit nor stop-loss prices are hit during the period of $h$ timestamps, then we calculate the future return using (3), and assign the label using (4). Note that TBM is path-dependant as it takes into account price dynamics in period $[t, t+h]$. An example of TBM barriers is presented in Fig. 3.

### D. Data

We used S&P 500 minute OHLC and volume data from 2018 to 2020. All pre-market and after-hours trading data were removed from the data, leaving only prices within regular trading hours. Features for a timestamp are calculated using a rolling window with a look-back period of 120 minutes; this means that no trading can occur during the first two trading hours of the day. Additionally, no trading was allowed during the last 30 minutes of the trading day due to the high volatility caused by the tendency of intraday traders to close their positions before the market closes.

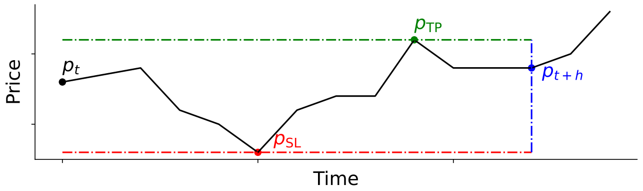

Fig. 3. The considering timestamp price is marked as $p_t$, the take-profit price (upper barrier) is marked as $p_{\text{TP}}$, the stop-loss price (lower barrier) is marked as $p_{\text{SL}}$, while the vertical barrier is marked as $p_{t+h}$.

## III. EXPERIMENTS

As previously discussed, we labeled the historical data using FTHM and TBM. In order to get a theoretical sense of how model accuracy affects financial performance of a strategy, we decided to simulate multiple models of varying accuracy for each labeling method. Predictions of those models were then used for simulated trading, which provided a financial metric that corresponds to the simulated model accuracy. This experimental setup allows us to assess the impact of accuracy we wanted to examine. Then, to confirm the simulation results, we trained the XGBoost model using six weeks of data as the train set, and use the following two weeks as the test set. An additional time series split (three validation folds) is performed on the train set to find the optimal model hyperparameters. Once trained, the model is used for trading for two weeks. After that period the model is retrained using the same procedure (six weeks for training, two weeks for testing). Such an approach in model development and deployment is quite common in both practice and literature [3]. An example of rolling window model training/testing can be inspected in Fig. 4.

The main classification metric we have used is accuracy score. It is used both for optimizing hyperparameters on three validation splits and for measuring classification performance on two-week test periods. On the other hand, the financial performance of a trading strategy is measured only during the test period in which we measure the two-week return. After measuring both financial and classification performance during test periods, we can examine the correlation between them. We compared theoretical and empirical results for labels obtained with both FTHM and TBM.

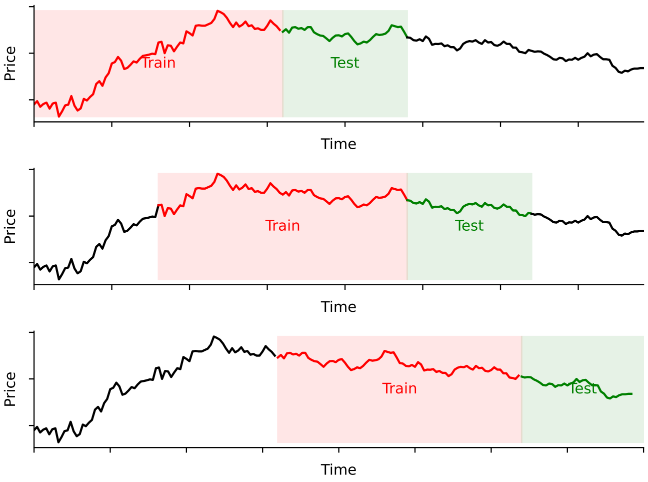

Fig. 4. Time series train/test split example for three different models.

## IV. Results and discussion

The simulated results for the labels obtained with FTHM and TBM show that both the average and median returns for TBM are slightly higher than FTHM. It is apparent that the measures of central tendency of the returns increases almost linearly with accuracy, with the slope being slightly higher for TBM labels. This implies that, on average, a trading strategy developed using TBM is expected to have a higher return on investment than one based on FTHM. Simulation results are displayed in Fig. 5. As for the model training, with a total of 104 weeks of S&P 500 stock minute data, we get 48 two-week test periods – which means 48 different models are used for trading for each stock. This process is repeated for each of the two labeling algorithms. Their parameters are as follows: $\tau_{\text{long}} = 2 \cdot 10^{-4}$ and $h = 5$ for FTHM, and $\tau_{\text{long}} = 2 \cdot 10^{-4}$, $h = 5$, $b = 20$, $U = 2$ and $L = 2$ in case of TBM. In this experimental setting and considering simulation results, one might expect a strong linear relationship between the classification performance of the model and the financial performance of a strategy based on the same model. The results shown in Fig. 6, however, suggest otherwise. For FTHM labels, the correlation between accuracy and financial return is statistically significant at the $\alpha = 0.01$ level of significance. Although statistically significant, the value of the correlation coefficient is rather low and equals $0.1404$. The average financial return is $0.9\%$, and the average accuracy is $0.5600$. Similar results can be observed for TBM labels. Again, the correlation coefficient is statistically significant, albeit low and equal to $0.1525$ (which is still a higher value than that of FTHM). Also, the average financial return of $1.22\%$ is slightly higher than before, and the same is true for the average accuracy of $0.5821$. Empirical results of a XGBoost model are in line with the theoretical simulation, suggesting that a trading strategy based on TBM labels is more resistant the model prediction errors.

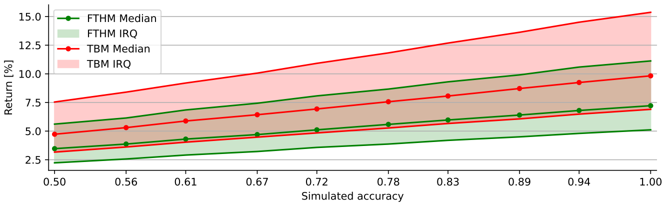

Fig. 5. Median returns of FTHM and TBM-based trading strategy for simulated model prediction accuracy. The width of the envelopes around the curves represents the interquartile range (IRQ) for each accuracy level from $0.5$ to $1$.

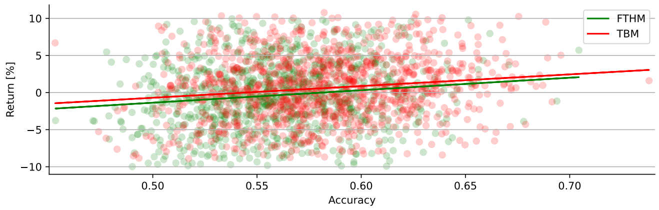

Fig. 6. Two-week test period returns of a trading strategy based on an XGBoost model trained on historical minute data of S&P500 stocks labeled with FTHM and TBM.

## V. Conclusion

In this paper, we studied the relationship between the model's classification performance and the financial performance of a strategy based on the same model. We chose two frequently used algorithms to label financial time series: the fixed-time horizon method and the triple-barrier method. For the labels obtained with these two algorithms, we retrained and tested the model every two weeks for S&P 500 stocks over the period from 2018 to 2020. Each model was trained using the previous six weeks of price data. In the test period, we measured the accuracy of the XGBoost model and the realized return of a trading strategy based on the output of the same model. With a total of 48 models trained for each label type per stock, we find that the linear correlation between accuracy and financial return is rather low, but statistically significant. The results show that the triple-barrier method achieves slightly larger average accuracy, and a better average return than fixed-time horizon method in both theoretical and empirical case. This suggests that a trading strategy based on the output of the model trained on TBM seems more resistant to errors that the model makes. Since the parameters of the labeling algorithm were constant, future work should investigate which parameters of the labeling algorithm are most appropriate in the context of machine learning generalization error, which could make the trading strategy more robust considering model errors and therefore somewhat increase the correlation coefficient.

## Acknowledgment

This work was supported in part by the Croatian Science Foundation under Project 5241, and European Regional Development Fund under Grant KK.01.1.1.01.0009 (DATACROSS).

## References

[1] Dixon, M., Halperin, I. & Bilokon, P. Machine learning in Finance, Springer, 2020

[2] Kumbure, M., Lohrmann, C., Luukka, P. & Porras, J. Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems With Applications*. **197** pp. 116659 (2022), https://doi.org/10.1016/j.eswa.2022.116659

[3] Kolm, P., Turiel, J. & Westray, N. Deep Order Flow Imbalance: Extracting Alpha at Multiple Horizons from the Limit Order Book. *Econometric Modeling: Capital Markets - Portfolio Theory EJournal*. (2021), http://dx.doi.org/10.2139/ssrn.3900141

[4] Wu, D., Wang, X., Su, J., Tang, B. & Wu, S. A Labeling Method for Financial Time Series Prediction Based on Trends. *Entropy*. **22** (2020), https://doi.org/10.3390/e22101162

[5] Huerta, R., Corbacho, F. & Elkan, C. Nonlinear support vector machines can systematically identify stocks with high and low future returns. *Algorithmic Finance*. **2** pp. 45-58 (2013), https://dx.doi.org/10.2139/ssrn.1930709

[6] Zhu, M., Philpotts, D. & Stevenson, M. The benefits of tree-based models for stock selection. *Journal Of Asset Management*. **13** pp. 437-448 (2012), https://doi.org/10.1057/JAM.2012.17

[7] Shwartz-Ziv, R. & Armon, A. Tabular data: Deep learning is not all you need. *Information Fusion*. **81** pp. 84-90 (2022), https://doi.org/10.1016/j.inffus.2021.11.011

[8] Murphy, K. Probabilistic Machine Learning: An introduction. (MIT Press,2022), probml.ai

[9] Chen, T. & Guestrin, C. XGBoost. *Proceedings Of The 22nd ACM SIGKDD International Conference On Knowledge Discovery And Data Mining*. (2016), https://doi.org/10.1145/2939672.2939785

[10] Murphy, J. Technical analysis of the financial markets: A comprehensive guide to trading methods and applications, Penguin, 1999

[11] Ouahilal, M., Mohajir, M.E., Chahhou, M. et al. A novel hybrid model based on Hodrick–Prescott filter and support vector regression algorithm for optimizing stock market price prediction. J Big Data 4, 31 (2017). https://doi.org/10.1186/s40537-017-0092-5

[12] Babu, C. & Reddy, B. A moving-average filter based hybrid ARIMA–ANN model for forecasting time series data. *Applied Soft Computing*. **23** pp. 27-38 (2014), https://doi.org/10.1016/j.asoc.2014.05.028

[13] Kim, S., Koh, K., Boyd, S. & Gorinevsky, D. $\ell_1$ Trend Filtering. *SIAM Review*. **51**, 339-360 (2009), https://doi.org/10.1137/070690274

[14] Jiang, W. Applications of deep learning in stock market prediction: Recent progress. *Expert Systems With Applications*. **184** pp. 115537 (2021), https://doi.org/10.1016/j.eswa.2021.115537

[15] De Prado, M. Advances in financial machine learning, John Wiley & Sons, 2018