Market Movement Prediction Algorithm Selection by Metalearning

A. V. P. M. Bandeira, G. M. Ferracioli, M. R. dos Santos, A. C. P. L. F. de Carvalho

University of São Paulo, Brazil
{avalentim98,ferracioligabriel,mmrsantos}@usp.br, andre@icmc.usp.br

Abstract. The prediction of market price movement is an essential tool for decision-making in trading scenarios. However, there are several candidate methods for this task. Metalearning can be an important ally for the automatic selection of methods, which can be machine learning algorithms for classification tasks, named here classification algorithms. In this work, we present an empirical evaluation of the metalearning application for the classification algorithms selection in the market movement prediction task. Different setups and metrics were evaluated for the meta-target selection. Cumulative return was the metric that achieved the best meta and base-level results. According to the experimental results, metalearning was a competitive selection strategy for predicting market price movement.

CCS Concepts: • **Computing methodologies** → **Machine learning**; • **Theory of computation** → **Design and analysis of algorithms**.

Keywords: market movement, metalearning, stock market, machine learning

1. INTRODUCTION

The stock market, the gathering of buyers and sellers of stocks, has been an important activity for centuries. It can be defined as an environment where it is possible to buy and sell fractions of a company. Since it works with the supply and demand principle, the price of each stock (fraction of a specific company) varies over time. Predicting this variation has always been a challenging problem that has taken much time from experts in this area. The most common way to represent market fluctuation is through a time series. With the advent of machine learning, predicting behavior in many sectors became possible, and it can also be applied to the stock market [Jordan and Mitchell 2015].

Predicting market price movement is difficult when it is based on price alone. The most used approach in the literature seeks to characterize the market movement as a binary classification problem: the stock price will either fall or rise. Classical time series prediction methods, such as, Autoregressive Integrated Moving Average (ARIMA), have accuracy ranging around 50% [Wen et al. 2019]. New variables were explored in the literature to add relevant information regarding the asset to be traded, such as Natural Language Processing (NLP) and Technical Analysis, to improve the predictive performance.

Any binary classification algorithm can be used to classify price movement in the market. Thus, there are many possibilities to consider. The choice of the best algorithm by evaluating the predictive performance of all available approaches has a high computational cost and demands a good knowledge on business model and machine learning algorithms [Prudêncio and Ludermir 2004]. Metalearning is a set of methods that have been successfully used as an alternative to the costly work of algorithm selection. It differs from traditional manual selection machine learning algorithms by using the experience gained from past machine learning tasks to obtain better predictive performance faster and more efficiently in future tasks [Brazdil et al. 2008].

Past works have tried to extract technical analysis features and select the best subset of them [Ni et al. 2011; Lee 2009; Huang et al. 2008; Lin et al. 2013]. Some state-of-the-art experiments worry about how much income the model can get [Zhang et al. 2018; Charkha 2008]. To the best of our knowledge, metalearning has not been used to solve this problem. The most similar work [Dong et al. 2021] tries to do a dynamic recommendation using a neural network. What we propose here also dynamically deals with the market movement. This work's main objective is to apply metalearning to select algorithms for predicting price movement in the financial market.

The main contribution of this work is to propose an automatic algorithm selection approach through metalearning that can bring benefits to trading market operations, not restricted to the financial market but extendable to other markets, for example, the electricity market. Another relevant analysis consists of considering different metrics for the selection of algorithms, having achieved the best results in terms of financial simulation when considering the financial return accumulated at the meta-level.

This paper is structured as follows. Section 2 describes the materials and methods for the proposed approach; Section 3 presents the results and evaluation insights; Section 4 presents the conclusions and future research directions.

2. EXPERIMENTS

In this section, we explain the development process of the recommendation system, from selecting data sources to the base-level and meta-level evaluation methods. All experiments are publicly available at a GitHub repository [1].

2.1 Data acquisition

The data used in this experiment is gathered from two main indexes of the stock market and a general list of stocks, which contain an amount of time-series compound by the days of trades, the prices of open, close, high, and low, and the volume of trades.

Although all the data used came from the global stock market, the list of selected companies was obtained with three different strategies:

- **S&P 500 (abbreviation of Standard & Poor's 500 Index) [Tsaih et al. 1998]**: an index that represents a specific list of stocks in the market. In this case, a weighted index of about 500 of the most relevant stock companies in the USA. The selection is not only tied to the top companies because there are other criteria of S&P Dow Jones Indices, the company that maintains the index.
- **Wilshire 5000 [Haugen and Baker 1991]**: differently from the S&P 500 index, the Wilshire 5000 tracks the whole stock market in the US. The index company criteria inclusion is that the stock must be publicly traded and the headquarters in the USA. There is no stock limit in the index, despite its name. In 1998, the number of companies in the index was about 7500 and currently is under 3500. The index companies can be reviewed monthly, and there is a possibility to adjust.
- **Use all stocks available**: with a web crawler algorithm is possible to get a list of all companies that are listed on any stock website. The page we used was Stock Monitor [2], and all stocks listed on this site were used in this work, despite being companies with good or bad performance in the market. This can be helpful for metalearning because it can promote diversity of data. This list has about 5100 stock names.

[1] Project Repository: github.com/ferracioli/Market-Movement-Prediction-Algorithm-Selection-by-Metalearning
[2] Stock Monitor URL: https://www.stockmonitor.com

After getting the stocks list, the methodology is the same for the three variations. We first iterate by reading each stock name and using the Yahoo Finance tool [3] to get the dataset of the company. Some of the listed names can not be found while searching the databases. There is only one restriction: select only stocks with at least 800 rows in their time series. That is the minimal data quantity we fixed for a good result. If the stock has more than 1500 rows, we get only the 1500 most recent rows. This strategy was set to select only the time series of the last period. Our final rate of success databases obtained with the whole market list was near 70%, and the sectors with the most stock names discarded were Healthcare and Financial Services since there are a lot of recent companies that were not considered in the minimal row quantity. The reproduction of this step may not result in the same list of companies because some of them may differ according to time. With the stock lists from different sources, we concatenated them into an unique dataset and removed duplicates as a strategy to improve the diversity of companies during the learning step.

2.2 Experiment Design

There are several possibilities of trade logic to be constructed: compare the difference between the opening and closing price on the same day, the closing of one day and the opening of the next day, or closing prices between two days. For this work, we chose one that seemed more interesting: define the market tendency as the difference between the closing price between this and the previous day. This affects how the experiment will be constructed because up and down trends can be predicted with classification models.

Figure 1 presents the complete architecture of the experiment, showing the process of generating the metadata, starting with the stocks list, and going until the recommendation system.

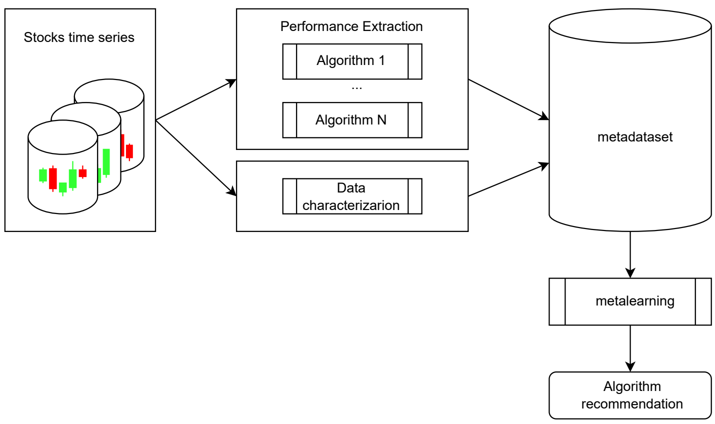

Fig. 1: Metalearning based recommendation system architecture

As shown in Figure 1, two steps are needed to generate the metadata: performance extraction and data characterization.

The data characterization step comprehends extracting relevant information related to the algorithm's performance called metafeatures. We combined two metafeatures sets implemented on python libraries:

—**PyMFE [Alcobaça et al. 2020]**: is a python package for extracting various types of metafeatures for the classification task. In this work we extract 55 General (simple information, like the number

[3]Yahoo Finance URL: https://finance.yahoo.com

of attributes) and Statistical (properties of the data distribution) metafeatures. For this last one we can cite Pillai's trace [Pillai 1955], used as test statistic in multivariate analysis, and Skewness, which is the degree of asymmetry observed in a probability distribution, calculated as the third statistical moment.

—**Catch22 [Lubba et al. 2019]**: a high-performance library that extracts 22 canonical time series metafeatures. As examples we can cite the time intervals between successive extreme events above (or below) the mean and the proportion of successive differences exceeding 0.04 standard deviation. Further explanation of each of the features can be obtained from its reference.

We apply eight different algorithms during base-level learning. A ranking system selects the best algorithm for that specific stock. The name of the company and its sector are also stored for data analysis reasons. Since the time series of each stock is relatively large, we are using a time series split of size four (this split results in 5 sub-groups of time series and using the sliding-window strategy) and the mean of the result of each split. The algorithms used in the base-level are Random Forest (RF) [Ho 1995], Decision Tree (DT) [Breiman et al. 2017], Extreme Gradient Boosting (XGB) [Chen and Guestrin 2016], K-Nearest Neighbors (KNN) [Cover and Hart 1967], Support-Vector Machine (SVM) [Vapnik 1998], Naive Bayes (NB) [McCallum et al. 1998], Adaptative Boosting (ADA) [Freund and Schapire 1997] and Logistic Regression (LogReg) [Cox 1958].

For each time series, we extract the performances by running each base-learner and calculating performance metrics. In this work, two metrics were used to define the meta-target, balanced accuracy, and cumulative return:

—**Balanced Accuracy**: built-in *SKLearn* metric, is an option to deal with imbalanced datasets. It is defined as the average of recall obtained in each class.

—**Cumulative Return**: The cumulative return uses a specific function that predicts if the stock is in a trend of High or Down movement. If the market goes up, it buys one unit of the stock and sells the next day. Using the same logic, if the market is going down, it operates in short, selling stock and repurchasing it the next day. The cumulative return is the sum of all the operations done in the whole period. A negative cumulative return means the algorithm lost money. In that way, it is hard to compare cumulative return between different stocks because it is related to the volatility of the stock. So, this is only useful for comparing different algorithms applied in the same stock.

We can build our metadata by combining the data obtained in data characterization with the data from performance extraction. As stated before, we are dealing with a classification problem, so one must select a target for the metalearning step. We have two target options: Balanced Accuracy and Cumulative Return, data obtained during the performance extraction. We can only test one target type at a time, so two versions of the metadata will be tested.

Metalearner is a machine learning algorithm applied to metadata for meta-knowledge extraction. That is, it will be able to recommend a promising model from past tasks. This work used four metalearners options: KNN, SVM, RF, and DT.

We can use the meta-knowledge obtained by metalearners to verify a new stock, applying the same functions of metafeature extraction and trying to recommend the most promising algorithm. Cross-validation is used in these 80% of dataset for training, and we use the 20% of data for the testing. Using the Balanced Accuracy, we verify if its value is greater than 1/8 (12,5%), the maximal randomness, which means the probability to choose an algorithm at random given the number of classes.

To add statistical support, we apply a *Mann-Whitney-Wilcoxon test [Mann and Whitney 1947] two-sided with Bonferroni correction [Dunn 1961]* (which is the nonparametric counterpart of the Student's t-test for independent samples), comparing all the possible pairs of base-level models distributions, considering the following hypothesis:

H_0: The distributions of both populations are identical.

vs

H_1: The distributions of both populations are not identical.

3. RESULTS AND DISCUSSION

The section shows the performance of each variation of our metadata. Since we are using two variations of the metadata, there will be a comparison between them, with a discussion about on which occasion each one is more relevant than the other.

3.1 Meta-level

We analyze the meta-level data by looking at whether the metadata components can differentiate their models based on the extracted metafeatures. As the first step in this exploration, we plotted Figure 2 and Figure 3, which show the distribution of the base-level learners using respectively the Balanced Accuracy and Cumulative Return as meta-targets.

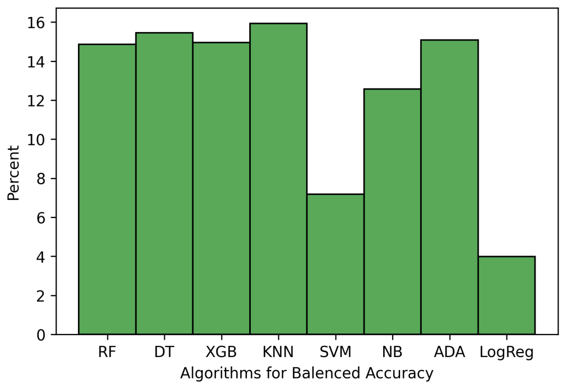

Fig. 2: Distribution of algorithms over the metadata with Balanced Accuracy as target

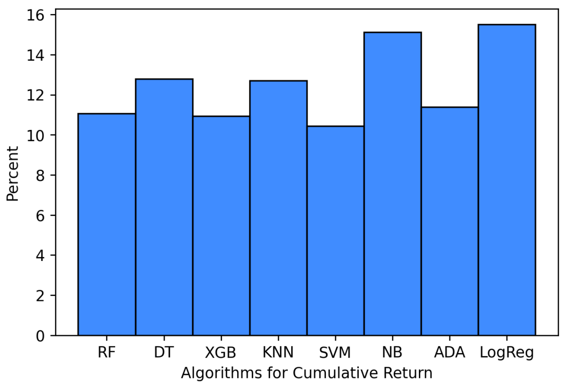

Fig. 3: Distribution of algorithms over the metadata with Cumulative Return as target

There is a considerable unbalance of learning algorithms distribution in Figure 2 with a small presence of SVM and LogReg, and the models are more equally distributed in Figure 3. Unbalance in the distribution can be a problem during the recommendation step because it may tend to the most frequent models to classify an unknown stock time series.

As a next step in verifying the data unbalance and metadata behavior, we test how accurate recommendations are for both meta-targets. So, doing a 10-fold of the data with ten repetitions, we trained the four metalearners to see how good are the recommendation performances.

Figure 4 and Figure 5 show the performance of the four metalearners in Balanced Accuracy. In the first case, the recommendation target was Balanced Accuracy, and for the second, the Cumulative Return.

In Figure 4, with 5% of significance level, we can spot the difference between most of the models for the Balanced Accuracy metric, but they are still at the same level of performance, i.e., although the algorithms have different learning mechanisms, they have essentially the same result. Moreover, all of them have a median of about 0.135, which is a poor result considering the quantity of metadata that was extracted. A possible reason for this result to be near the random guess (0.125) is that the balanced accuracy should not be a good choice of target for this task. Another possible explanation is that the unbalanced model distribution results in the metalearners making wrong recommendations.

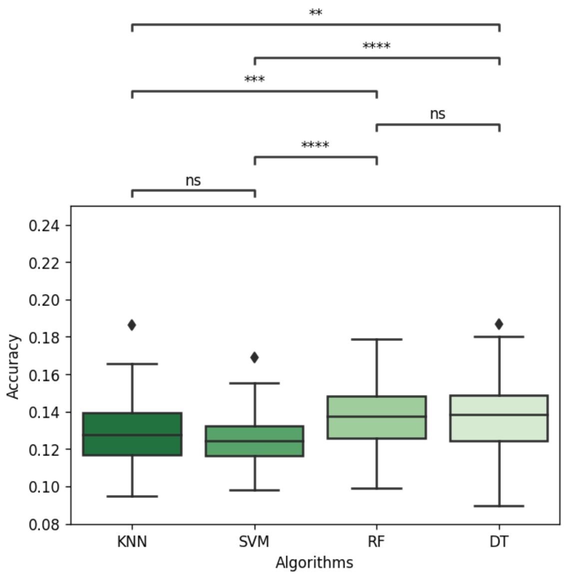

Fig. 4: Recommendation based on Balanced Accuracy metric

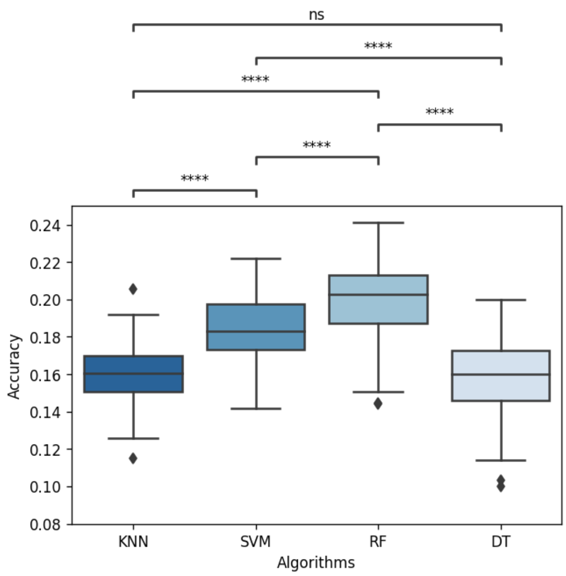

Fig. 5: Recommendation based on Cumulative Return metric

In the hypothesis test, 'ns' means nonsignificant, and each '*' means a decrease in the p-value for the test.

Focusing now on Figure 5, according to the test results, with 5% of significance level, there is no significant difference between the distributions of the pair *KNN v.s. Decision Tree*, which is not true in the other cases, where we see that the distributions are different and Random Forest proved to be the best metalearner.

Besides that, the plot shows that even the worst algorithm has the metric median above the probability of randomly choosing the best algorithm for one specific stock. This means that the recommendation using the metadataset for Cumulative Return can outperform a random choice of algorithm. Although the accuracy seems low in this multi-class classification problem, it doesn't necessarily means a significant loss in the cumulative return, considering that, in most cases, the performance of the best algorithm is not that far from the second and third places, and sometimes a different seed or hyperparameter could close this gap. This shows that it is more promising to apply metalearning with a focus on cumulative return rather than balanced accuracy as a meta-target, as it has a better result in terms of recommendation. To confirm this choice, we analyze metalearning at the base-level.

3.2 Base-level

The base-level evaluation compares the eight models used in the metadataset construction with our recommendation of models that we call MetaMM (Meta Market Movement). There are two variations of this system, BAC (recommendation focused on balanced accuracy meta-target) and CRet (based on the cumulative return meta-target). Our last objective is to see if the MetaMM recommendations can outperform the profit of the best base-level model. Figure 6 shows the performances of cumulative return for the ten different trading models.

Based on the Figure 6 results, the MetaMM (CRet) strategy was the one with the best performance, becoming more efficient than the best base-level model (SVM). Besides that, the MetaMM (BAC) result was an inferior performance. The only model that ended the test with loss was DT, and the others were capable of getting a positive result.

One last piece of knowledge we can extract from the metadata is that high balanced accuracy does

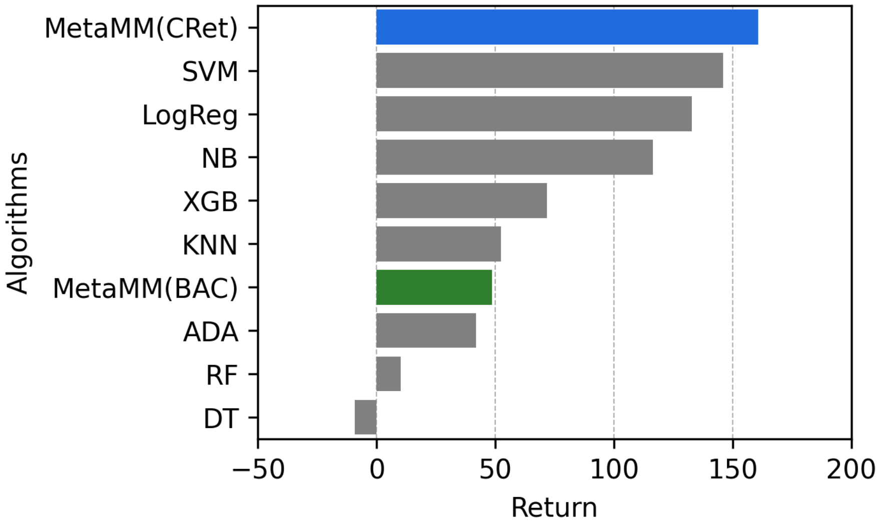

Fig. 6: Comparison of Returns between recommendations and other models

not imply a high cumulative return since the performance of these two metrics is relatively different. We saw with the Figures that the metadata results with balanced accuracy as the target were inferior, not showing any advantage compared with more straightforward strategies or focusing on cumulative return. So in practice, if we had to choose only one metric, the cumulative return probably would be the best option because protecting assets is more important than having a greater accuracy rate for the financial market.

4. CONCLUSION

The selection of machine learning algorithms in the market movement prediction task is essential in decision-making in stock market trading scenarios. From metalearning, it was possible to recommend promising models for the task, reducing the computational cost and the need for specialist knowledge to select models in the market movement prediction task. An important finding of this analysis was that the meta-target based on cumulative return presented the best results at the meta-level and the base-level, being the best choice for representing the performance in this task. In future research directions, we propose to test other metafeature approaches and explore relationships in which each classification algorithm benefits specific types of time series.

Acknowledgments

This study was partially funded by grant 2019/10012-2, São Paulo Research Foundation (FAPESP), CEPID-CeMEAI Process 2013/07375-0, USP Research Office (PIPAE project), and CNPq.

REFERENCES

Alcobaça, E., Siqueira, F., Rivolli, A., Garcia, L. P. F., Oliva, J. T., and de Carvalho, A. C. P. L. F. Mfe: Towards reproducible meta-feature extraction. *Journal of Machine Learning Research* 21 (111): 1–5, 2020.

Brazdil, P., Carrier, C. G., Soares, C., and Vilalta, R. *Metalearning: Applications to data mining.* Springer Science & Business Media, 2008.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. *Classification and regression trees.* Routledge, 2017.

Charkha, P. R. Stock price prediction and trend prediction using neural networks. In *2008 First International Conference on Emerging Trends in Engineering and Technology.* IEEE, pp. 592–594, 2008.

Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. pp. 785–794, 2016.

Cover, T. and Hart, P. Nearest neighbor pattern classification. *IEEE transactions on information theory* 13 (1): 21–27, 1967.

Cox, D. R. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)* 20 (2): 215–232, 1958.

Dong, S., Wang, J., Luo, H., Wang, H., and Wu, F.-X. A dynamic predictor selection algorithm for predicting stock market movement. *Expert Systems with Applications* vol. 186, pp. 115836, 2021.

Dunn, O. J. Multiple comparisons among means. *Journal of the American statistical association* 56 (293): 52–64, 1961.

Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* 55 (1): 119–139, 1997.

Haugen, R. A. and Baker, N. L. The efficient market inefficiency of capitalization–weighted stock portfolios. *The journal of portfolio management* 17 (3): 35–40, 1991.

Ho, T. K. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE, pp. 278–282, 1995.

Huang, C.-J., Yang, D.-X., and Chuang, Y.-T. Application of wrapper approach and composite classifier to the stock trend prediction. *Expert Systems with Applications* 34 (4): 2870–2878, 2008.

Jordan, M. I. and Mitchell, T. M. Machine learning: Trends, perspectives, and prospects. *Science* 349 (6245): 255–260, 2015.

Lee, M.-C. Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Systems with Applications* 36 (8): 10896–10904, 2009.

Lin, Y., Guo, H., and Hu, J. An svm-based approach for stock market trend prediction. In *The 2013 international joint conference on neural networks (IJCNN)*. IEEE, pp. 1–7, 2013.

Lubba, C. H., Sethi, S. S., Knaute, P., Schultz, S. R., Fulcher, B. D., and Jones, N. S. catch22: Canonical time-series characteristics. *Data Mining and Knowledge Discovery* 33 (6): 1821–1852, 2019.

Mann, H. B. and Whitney, D. R. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, 1947.

McCallum, A., Nigam, K., et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*. Vol. 752. Citeseer, pp. 41–48, 1998.

Ni, L.-P., Ni, Z.-W., and Gao, Y.-Z. Stock trend prediction based on fractal feature selection and support vector machine. *Expert Systems with Applications* 38 (5): 5569–5576, 2011.

Pillai, K. S. Some new test criteria in multivariate analysis. *The Annals of Mathematical Statistics*, 1955.

Prudêncio, R. B. and Ludermir, T. B. Meta-learning approaches to selecting time series models. *Neurocomputing* vol. 61, pp. 121–137, 2004.

Tsaih, R., Hsu, Y., and Lai, C. C. Forecasting s&p 500 stock index futures with a hybrid ai system. *Decision support systems* 23 (2): 161–174, 1998.

Vapnik, V. *Statistical Learning Theory*. NY: Wiley, 1998.

Wen, M., Li, P., Zhang, L., and Chen, Y. Stock market trend prediction using high-order information of time series. *Ieee Access* vol. 7, pp. 28299–28308, 2019.

Zhang, J., Cui, S., Xu, Y., Li, Q., and Li, T. A novel data-driven stock price trend prediction system. *Expert Systems with Applications* vol. 97, pp. 60–69, 2018.