TradAO: A Visual Analytics System for Trading Algorithm Optimization

Ka Wing Tsang, Haotian Li *
HKUST

Fuk Ming Lam †
ALGOGENE

Yifan Mu, Yong Wang ‡
HKUST

Huamin Qu §
HKUST

Figure 1: *TradAO* assists users in exploring the whole optimization process of a trading algorithm and evaluating its detailed performances. A) The algorithm evolution view explores how a trading algorithm evolves along with the configuration of different parameters. B) The parameter correlation view examines if a trading algorithm instance is overfitting. C) The trading residual view helps traders confirm the performance consistency of a trading algorithm instance more efficiently. D) The cash usage view facilitates monitoring the cash usage along different trading periods. E) The trading history view allows users to explore trading order information along the whole trading period.

ABSTRACT

With the wide applications of algorithmic trading, it has become critical for traders to build a winning trading algorithm to beat the market. However, due to the lack of efficient tools, traders mainly rely on their memory to manually compare the algorithm instances of a trading algorithm and further select the best trading algorithm instance for the real trading deployment. We work closely with industry practitioners to discover and consolidate user requirements and develop an interactive visual analytics system for trading algorithm optimization. Structured expert interviews are conducted to evaluate *TradAO* and a representative case study is documented for illustrating the system effectiveness. To the best of our knowledge, previous financial data visual analyses have mainly aimed to assist investment managers in investment portfolio analysis but have neglected the need of traders in developing trading algorithms for portfolio execution. *TradAO* is the first visual analytics system that assists users in comprehensively exploring the performances of a trading algorithm with different parameter settings.

---

*kwtsangae, haotian.li@connect.ust.hk.

†tonylam@algogene.com.

‡ymuaa, ywangct@connect.ust.hk; Y. Wang is the corresponding author.

§huamin@cse.ust.hk.

1 INTRODUCTION

Algorithmic trading, has been transforming the nancial market by utilizing computer algorithms in the trading over the recent two decades [16]. The time of doubling the total trading volume of stocks in Dow Jones Industrial Average has accelerated from 7.5 years to 2.9 years in the recent decades [16]. Due to the increasing level of market fragmentation, both the trade execution cost and monitoring cost across markets are rising significantly [32]. Such a cost increase promotes algorithmic trading.

Building a winning trading algorithm to beat the market has become critical for traders to conduct successful trading. To evaluate the viability of a trading algorithm, traders need to conduct backtestings, where traders test the algorithm against historical data and observe its simulation performance. Backtesting results are regarded as a predictive proxy for the potential future performance of an algorithm [30]. To build a winning trading algorithm, traders need to tune different algorithm parameters, conduct rounds of backtesting, and further compare and evaluate algorithm performances. Traders repeat this work ow whenever a new set of trading instruction parameters and model variables is used. Such a process is often time-consuming and tedious [24]. Due to the lack of efficient tools, traders encounter multiple pain points in optimizing trading algorithms. First, it is challenging to compare *algorithm instances* along the whole development process of a trading algorithm. An algorithm instance refers to a particular set-up of parameter combination.

Traders can only do manual comparisons based on their memory. Second, the work efficiency of formulating inner parameters and variables of a trading algorithm is low and subjected to traders experience and intuition. Third, it is inconvenient for traders to trace the transaction details of a trading algorithm in execution.

To address these challenges, we propose *TradAO*, a novel visual analytics system to assist traders in exploring the trading algorithm optimization process and evaluating its detailed performances. Specifically, *TradAO* offers five well-coordinated views: *Algorithm Evolution View* to overview a trading algorithm intuitively, *Parameter Correlation View* to investigate the relationship between variables used in an algorithm instance, *Trading Residual View* to examine the trading model performance consistency of an algorithm instance, *Cash Usage View* to monitor the cashflow stability and *Trading History View* to evaluate the buying and selling pattern. We designed the system with an experienced quantitative researcher who is also a co-author of this paper. We conducted three structured expert interviews with domain experts to assess the usefulness and usability of *TradAO*. The major research contributions of this paper are summarized as follows:

- A novel visual analytics system called *TradAO*, allowing traders to explore and compare different stages of a trading algorithm along the development cycle;

- Preliminary evaluations on the usefulness and usability of *TradAO* through structured interviews with domain experts, including both traders and system developers.

2 RELATED WORK

The related work of this paper can be categorized into two groups: *trading models* and *financial data visualization*.

Extensive research has been conducted on algorithmic trading and various trading models have been proposed, for instance, *Pairs Trading* [9, 18], *Moving Average* [17, 22], *Linear Regression* [10, 25], *Neural Networks* [14, 21] and *Sentiment Analysis* [6, 20]. Trading algorithms built on these trading models might involve complicated trading rules in an attempt to achieve a higher prediction accuracy. In this paper, we select three commonly used foundational trading models, which are relatively easier to comprehend yet quantitatively enough as being an entry point to evaluate our system, for demonstration. *Pairs Trading* is a common market neutral strategy that opens long and short position for two securities simultaneously to capture the price differential [9]. *Moving Average* usually applies to univariate time series for studying the relationship between a variable and its lagged terms [22], while *Multiple Linear Regression Model* is used for modelling multivariate time series [25].

Visual analytics methods have been employed to assist financial industry practitioners in facilitating their investment decision makings. For example, FinanVis [23] is proposed to explore evolution patterns from financial news. WireVis [7] helps anti-money laundry specialist in bank to detect account suspicious transactions. FinVis [27] assists retail investors in doing personal finance planning. In addition to visualization systems assessing the overall situation in the market [3, 11, 15, 31, 33], previous visualization systems also focused on portfolio analysis [8] for portfolio managers [31]. However, no visualization system exists for assisting traders in developing trading algorithms to the best of our knowledge.

3 TASK ANALYSIS

We worked closely with four professional domain experts (D1, E1, E2, E3) from the financial field during the development of *TradAO*. D1 is a quantitative researcher from ALGOGENE[1] and also a co-author of this paper. ALGOGENE is an online trading algorithm development cloud platform. E1 and E2 are a fund manager and a trader from a multi-asset management company and an investment

bank respectively. E3 is a system interface developer in an investment bank. We conducted structured interviews with each expert and collected their feedback on the tasks and challenges of optimizing trading algorithms. We further grouped the trading algorithm optimization tasks into three categories: *strategy overview*, *algorithm instance inspection* and *algorithm instance assessment*.

Strategy overview tasks offer a full picture of the whole trading algorithm optimization cycle and enable inter-comparisons among algorithm instances with different parameter settings.

T1. Explore how a trading algorithm evolves over time. Developing an effective trading algorithm involves multiple rounds of backtesting with different parameter settings. Traders are interested to gain a convenient overview of the model instances with different settings and their performances.

T2. Identify how the effectiveness of a trading algorithm changes under new parameter settings. The changes of model parameter settings can significantly influence the effectiveness of different trading algorithms. A common method to evaluate their effectiveness is to compare different performance metrics.

Algorithm instance inspection tasks focus on the relationships among different parameters for a particular trading algorithm and enable inner-comparisons among those parameters.

T3. Examine if a trading algorithm is overfitting. A trading algorithm can be overfitting, which makes the algorithm achieve an ideal performance under a chosen time range in the backtesting simulation environment, but may perform badly on other time ranges. Traders often need to check if a trading algorithm is overfitting by investigating the correlation between different parameters.

T4. Verify if the trading algorithm is robust. A trading algorithm may be derived from economy theory or mathematical models. To evaluate the appropriateness and robustness of an algorithm, traders rely on residual analysis to validate the underlying assumptions.

Algorithm instance assessment tasks reveal the viability of an algorithm instance with specific parameter settings in terms of helping traders identify trading patterns in the real application. Trading pattern refers to the transaction information such as the transaction time, frequency, price and volume, which can result in the fluctuation of the portfolio value over a trading period.

T5. Measure how well a trading algorithm optimizes the usage of investment capital. To maintain the fund as a self-financing portfolio for cash flow stability, it is crucial to strike a balance between leveraging positions and reserving cash buffer. Traders will record the daily investment balance to monitor cash flows.

T6. Assess the performances of a trading algorithm by comparing them with the market index. Traders also evaluate a trading algorithm's effectiveness through its buying and selling patterns. A significant discrepancy between its trading pattern and the market trend can guide the subsequent refinement of a trading algorithm.

Regarding *T2*, we select a set of performance metrics based on our discussion with domain experts. Specifically, there are six types of metrics that are often chosen by domain experts as a quick proxy to evaluate the overall performance of an algorithm instance, including Activeness, Consistency, Prediction, Profitability, Recovery and Robustness. Activeness delineates how frequently an algorithm execute transactions; Consistency means the stability of an algorithm in making profits; Prediction represents the prediction accuracy of an algorithm; Profitability is the overall return of an algorithm; Recovery assesses how fast an algorithm recovers from previous losses; Robustness measures the stress tolerance level of an algorithm. Specifically, we propose using nine measures belonging to the six types to evaluate the overall performance of a trading algorithm instance: Mean Annualized Return (Yield) [5] and Maximum Drawdown Percentage (MD) [12] under the profitability category measure the profitability of an algorithm instance; Annualized Sharpe Ratio (Sharpe) [4] and Annualized Sortino Ratio (Sortino) [28] under the consistency category assess the risk adjusted

return; Maximum Drawdown Duration (MaxDD) [12] and Average Drawdown Duration (AvgDD [12]) under the recovery category delineate the recovery time from loss. 99% 1-day Return Value-at-Risk (VaR99) [1] and Annualized Return Volatility (Vol) [26] under the robustness category indicate the performance volatility; and Win Rate (WinRate) [29] under the prediction category represents the prediction accuracy.

4 VISUAL DESIGN

TradAO is a web-based application consisting of five main views: algorithm evolution view (Figure 1(A)), parameter correlation view (Figure 1(B)), trading residual view (Figure 1(C)), cash usage view (Figure 1(D)), and trading history view (Figure 1(E)). The algorithm evolution view provides a quick overview of the performances and parameter settings of different algorithm instances. The parameter correlation view and the trading residual view then return the detailed parameter information of any selected algorithm instance on demand. Meanwhile the cash usage view and the trading history view display the trading patterns of the selected algorithm instance. In this paper, we analyzed the backtesting records collected by AL-GOGENE platform. These backtesting records were imported into *TradAO* through REST APIs. The trading period of an algorithm is often several months or even several years, while its trading frequency can be daily, hourly or even higher frequency [2]. There is no fixed rule in deciding the backtesting trading period, as it depends on traders' own preferences and the availability of backtesting platform's historical data. *TradAO* enables linked analysis across different views to facilitate convenient exploration of such records and help traders quickly find a good trading algorithm with appropriate parameter settings. The detailed visualization designs will be introduced in this section.

4.1 Algorithm Evolution View

We provide the algorithm evolution view for users to explore how a trading algorithm evolves along with the configuration of different parameters before deployment (**T1**).

The evolution of a trading algorithm is visualized as a tree diagram that consists of linked sequential glyphs. Each glyph represents a trading algorithm instance, i.e., a trading algorithm with a specific parameter setting configured by a trader. As shown in Figure 1(A1), each glyph consists of a outer ring and two inner radar charts. Each segment of the outer ring represents a trading algorithm parameter. The color of the segment encodes the relative value of an algorithm parameter compared with the value of the same parameter used in other trading algorithm instances over the whole algorithm development. A darker color corresponds to a higher parameter value. The inner radar chart visualizes the performance of an algorithm instance with respect to the six measure categories mentioned in section 3. Customized methods are used to normalize and standardize the characteristic scores into a scale from 0 to 100 for each dimension. The orange star plot shows the performance of the current algorithm instance, while the blue star plot indicates the performance of its parent algorithm instance. Parent algorithm instance refers to the algorithm instance that the current algorithm instance is derived from by further tuning its parameters. Users can make a quick comparison on the performance of two consecutive instances at a glance (**T2**). The root glyph represents the trading algorithm instance with initial parameter configurations and the subsequent glyphs represent other algorithm instances that are further developed.

The algorithm evolution view enables rich interactions. First, when a user hovers over a segment on the outer ring or any spoke on the radar chart, a tooltip will display both the corresponding parameter name and value. Second, a relative mode offers users another perspective to overview the whole trading strategy and compare two consecutive trading algorithms through a different color scheme denoting the change of value of variables. The greener the segment

Figure 2: Performance comparison of two trading algorithms. The higher the left five performance metric values (Yield, Sharpe, WinRate, Sortino, VaR99) on the parallel coordinate are, the better the trading algorithm performs. The lower the right four performance metric values (MD, AvgDD, MaxDD, Vol) are, the better the algorithm performs.

is, the higher amount of the variable value increased in a particular trading algorithm compared to its parent. While the redder the segment is, the higher amount of the value decreased. Third, when users click a particular glyph, an expanded view, consisting of a parallel coordinate and glyphs showing the current algorithm instance and its parent, will be overlaid in Evolution View, as shown in Figure 2. Users can evaluate the effectiveness of the selected trading algorithm instance and compare with other algorithm instances performance via the parallel coordinate. Similar to the color encoding of the star plot, the orange line in the parallel coordinate represents the selected algorithm instance while the blue line represents its parent. Grey lines representing all other algorithm instances are shown for benchmarking purposes. When users click on a particular glyph, further details of that trading algorithm instance will be be shown in all the other views simultaneously.

4.2 Parameter Correlation View

The parameter correlation view helps users examine if a trading algorithm is overfitting (**T3**). As shown in Figure 1(B), it is a 3×3 grid, which can be categorized into three parts: lower right cells, diagonal cells and upper left cells. The lower right cells shows three scatter plots that visualize the correlation between each pair of parameters used in both the selected algorithm instance (the orange scatter plot) and its parent algorithm instance (the blue scatter plot). The diagonal cells show univariable distribution histograms. The horizontal and vertical axes encode the value and occurrence probability of a variable used in the selected algorithm. The upper left cells show line charts to indicate the trends of correlation values over time. All the line charts vertical scale are normalized to $[-1, 1]$. The green line in the middle denotes 0. Such a visualization provide users with deep insights into the correlation between parameters. The scatterplots in the bottom right offer traders a quick overview of the correlation, while the line charts enable a detailed observation of the correlation magnitude change over time.

4.3 Trading Residual View

The trading residual view helps traders confirm the performance consistency of a particular trading algorithm more efficiently (**T4**). As shown in Figure 1(C), the trading residual view consists of a scatter plot in the left and a histogram in the right. The scatterplot, where the horizontal axis represents the ordinal sequence of model residuals and vertical axis represents the residual value, shows the statistical properties of residuals, helping traders determine the confidence level of generated trading signals at different time points. The vertical histogram displays the probability distribution of residuals, where its horizontal axis is the discrete probability density and vertical axis is the domains of residual value.

4.4 Cash Usage View

The cash usage view (Figure 1(D)) facilitates monitoring the cash usage along different trading periods. It is a common practice

that traders use leverage to increase the potential investment return. Leverage [19] is an investment approach through using borrowed capital. Liquidity [13] is an important indicator to evaluate the viability of a trading algorithm (**T5**). Traders can use the cash usage view to track both the net asset value (NAV) and remaining cash amount (Available Cash) along the whole trading period, which are represented by a blue line and a green line with dynamic colors, respectively. The dotted grey line represents the initial amount of capital for users reference. Two benchmark lines warn users against the liquidity shortage. The orange dotted line represents the *warning level* to indicate a potential liquidity risk, while the red dotted line indicates a significant liquidity risk (*danger level*). When the remaining cash amount falls below the warning level or the danger level, the corresponding line becomes orange or red to alert users. Users can click on the top right gear button to interactively adjust the benchmark levels according to their own risk tolerance levels with the two sliders. Users can also use the brush and zoom function to drill into more cash usage details at a specific period and the trading history view will be updated simultaneously, helping to identify the discrepancy among the net asset value, cash flow and inventory flow.

4.5 Trading History View

The trading history view (Figure 1(E)) allows users to explore trading order information (e.g., the price and volume of a transaction) along the whole trading period. Users can use the trading history view to identify outlier transactions (**T5**). In Figure 1(E1), horizontal axis represents the trading period of a trading algorithm. The left vertical axis represents the price of financial instruments and the right vertical axis represents the inventory level involved in transactions. The height of the ellipse represents the net volume transacted in a single trading day, while the center position of the ellipse is the average price of all transactions in a single trading day. Two green horizontal lines indicate the highest and lowest buying price when there are buying transactions. Two red horizontal lines indicate the highest and lowest selling price. The height of the shading in the background encodes the daily outstanding inventory amount. Green shading indicates a positive amount of inventory. Red shading refers to a negative amount of inventory, meaning a short position.

Users can also use this view to compare its trading patterns with the market trends of various financial instruments or market indices in Figure 1(E2) (**T6**). The Drop-down button on the top of the right panel enables users to interactively select displayed market charts.

5 EVALUATION

We conducted structured interviews with three domain experts to evaluate the usefulness and usability of *TradAO*. Three domain experts (i.e., E1, E2 and E3), who participated in our task analysis (Section 3), are further interviewed to collect their feedback. Each interview lasted for about 45 minutes and contained three sections, an introduction session to explain the project scope, the dataset and the visual encoding (10 mins), a task session to observe how interviewees use the system (20 mins), and a comment session to collect feedbacks in terms of system usability, effectiveness and visualization designs (15 mins).

5.1 Overall Feedback on *TradAO*

The overall system was confirmed to be useful for facilitating experts daily research and analysis tasks. All experts appreciated the functionality of the algorithm evolution view in particular. E1 said that *It can help reduce a significant amount of workload in tracing my (trading algorithm development) changes, analyzing algorithm details and comparing different (algorithm instance) performance at the same time*. E2 confirmed that he would have overlooked some details and built a less preferable trading algorithm without the help of *TradAO*. In the task session, interviewees were asked to perform three practical tasks within a limited time period to examine the

system effectiveness: (1) select the best trading algorithm instance out of the given algorithm pool; (2) identify problematic algorithm configurations which may lead to a significant performance gap between two consecutive instances; and (3) suggest potential issues for the algorithm instances. All the experts can finish all tasks within the given time and they confirmed the effectiveness of *TradAO*.

All the three experts said that the visualization designs are intuitive and easy to understand. They appreciated the cross-view interactions, for example, enabling inline brushing in the cash usage view and the trading history view helps users quickly narrow down into the time periods of interest.

5.2 Case Study: Identify the Problematic Algorithm Configurations

We report a case found by E2 during our expert interviews to further demonstrate the usefulness of *TradAO*. To finish the second task, E2 first explored the evolution of a trading strategy based on *Pairs Trading Model* as shown in Figure 1(A) (**T1**). E2 said that the tree-based visualization in the algorithm evolution view greatly reduces his workload in narrowing down the scope of the instance candidates.

E2 noticed that $\beta 1$ has the largest hexagon-like shape radar chart among all trading algorithm instances in Figure 1(A) and confirmed that it was one of the optimal algorithms in terms of the highest Yield and Win Rate, as shown in Figure 2. E2 found an upward NAV curve and green Available Cash curve in the cash usage view which indicated high profitability and cash flow stability (**T5**). E2 speculated that $\beta 1$ may be a good algorithm instance for deployment after a quick review on model parameters shown by its outer ring.

E2 was also interested to examine the descendent algorithm instances of $\beta 1$. He made a quick comparison through the radar charts in $\beta 2$ and noticed a significant performance gap between $\beta 1$ and $\beta 2$ (**T2**). To further explore the reasons behind this gap, he continued to explore the trading history view and spotted a significant change from a sizeable pink shading area to a green shading on NSXUSD (a financial instrument) in August 2019 as shown in Figure 1(E1). This reflected a substantial momentum change in holding inventory level from short position to long position. SPXUSD was traded reciprocally during the same period. E2 further checked Figure 1(E2) to confirm if these trade actions match the corresponding market direction (**T6**). The percentage decrease in NSXUSD market index is greater than the drop in SPXUSD market index. E2 regarded $\beta 2$ as an ineffective instance and investigated into its parameter details in the parameter correlation view as shown in Figure 1(B) (**T3**). E2 observed the line charts and identified a highly fluctuated correlation value pattern between Coeff_1 and Diff_thre. E2 cross-checked the trading algorithm robustness of $\beta 2$ with the trading residual view as shown in Figure 1(C) (**T4**) and observed a non-random cyclic pattern. E2 speculated that $\beta 2$ is statistically non-stationary and probably ineffective to generate good performance.

6 CONCLUSION

We propose a novel visual analytics system, *TradAO*, for helping traders explore and compare different trading algorithm instances to find an appropriate trading algorithm. Expert interviews with professional traders and trading algorithm developers provide support for the usefulness and usability of *TradAO*. However, our approach is not without limitations. First, *TradAO* suffers from scalability issues. For instance, when traders built and backtest the same trading algorithm for too many times, it will be difficult to display all the algorithm instances in the algorithm evolution view due to the space limit. Second, *TradAO* only provides a finite set of pre-selected performance metrics and may not satisfy different requirements of users under all conditions. Third, our current qualitative evaluation approach could be supplemented with an additional quantitative analysis to measure the system effectiveness in terms of time spent for task completion under a control set-up. In future work, we would like to further improve *TradAO* by handling the above limitations.

REFERENCES

[1] P. Abad, S. Muela, and C. Lopez. A comprehensive review of value at risk methodologies. *The Spanish Review of Financial Economics*, 12, 01 2013. doi: 10.1016/j.srfe.2013.06.001

[2] I. Aldridge. *High-frequency Trading*. Wiley, 2013.

[3] J. Alsakran, Y. Zhao, and X. Zhao. Visual analysis of mutual fund performance. In *2009 13th International Conference Information Visualisation*, pp. 252–259, 2009.

[4] G. O. Aragon and W. E. Ferson. Portfolio performance evaluation. *Foundations and Trends in Finance*, 2(2):83–190, 2007. doi: 10.1561/0500000015

[5] M. T. Assoc. *CMT Level I 2016: An Introduction to Technical Analysis*. John Wiley & Sons, 2015.

[6] S. W. Chan and M. W. Chong. Sentiment analysis in financial texts. *Decision Support Systems*, 94:53 – 64, 2017. doi: 10.1016/j.dss.2016.10.006

[7] R. Chang, M. Ghoniem, R. Kosara, W. Ribarsky, J. Yang, E. Suma, C. Ziemkiewicz, D. Kern, and A. Sudjianto. Wirevis: Visualization of categorical, time-varying data from financial transactions. In *2007 IEEE Symposium on Visual Analytics Science and Technology*, pp. 155–162, 2007.

[8] C. Csallner, M. Handte, O. Lehmann, and J. Stasko. Fundexplorer: supporting the diversification of mutual fund portfolios using context treemaps. In *2003 IEEE Symposium on Information Visualization*, pp. 203–208, 2003.

[9] A. Deshpande and B. R. Barmish. A general framework for pairs trading with a control-theoretic point of view. In *2016 IEEE Conference on Control Applications (CCA)*, pp. 761–766, 2016.

[10] T. Dielman, T. J. Nantell, and R. L. Wright. Price effects of stock repurchasing: A random coefficient regression approach. *Journal of Financial and Quantitative Analysis*, 15(1):175189, 1980. doi: 10.2307/2979025

[11] T. Dwyer and D. R. Gallagher. Visualising changes in fund manager holdings in two and a half-dimensions. *Information Visualization*, 3(4):227244, Dec. 2004. doi: 10.1057/palgrave.ivs.9500081

[12] L. Goldberg and O. Mahmoud. Drawdown: From practice to theory and back again. *Mathematics and Financial Economics*, 11, 09 2016. doi: 10.1007/s11579-016-0181-9

[13] T. Hendershott, C. M. Jones, and A. J. Menkveld. Does algorithmic trading improve liquidity? *The Journal of Finance*, 66(1):1–33, 2011.

[14] W. Huang, K. K. Lai, Y. Nakamori, S. Wang, and L. Yu. Neural networks in finance and economics forecasting. *International Journal of Information Technology & Decision Making*, 06(01):113–140, 2007. doi: 10.1142/S021962200700237X

[15] D. Keim, T. Nietzschmann, N. Schelwies, J. Schneidewind, T. Schreck, and H. Ziegler. A spectral visualization system for analyzing financial time series data. In *Eurographics/IEEE-VGTC Symposium on Visualization*, pp. 195–202, 2006. doi: 10.2312/VisSym/EuroVis06/195-202

[16] A. A. Kirilenko and A. W. Lo. Moore's law versus murphy's law: Algorithmic trading and its discontents. *Journal of Economic Perspectives*, 27(2):51–72, May 2013. doi: 10.1257/jep.27.2.51

[17] F. Klinker. Exponential moving average versus moving exponential average. *Mathematische Semesterberichte*, 58(1):97–107, 2011. doi: 10.1007/s00591-010-0080-8

[18] C. Krauss. Statistical arbitrage pairs trading strategies: Review and outlook. *Journal of Economic Surveys*, 31(2):513–545, 2017. doi: 10.1111/joes.12153

[19] M. S. Long and I. B. Malitz. Investment patterns and financial leverage. In *Corporate Capital Structures in the United States*, pp. 325–352. National Bureau of Economic Research, Inc, 1985.

[20] S. Mehtab and J. Sen. A robust predictive model for stock price prediction using deep learning and natural language processing. *Econometric Modeling: Capital Markets - Forecasting eJournal*, 2019.

[21] A. H. Moghaddam, M. H. Moghaddam, and M. Esfandyari. Stock market index prediction using artificial neural network. *Journal of Economics, Finance and Administrative Science*, 21(41):89 – 93, 2016. doi: 10.1016/j.jefas.2016.07.002

[22] M. Nakano, A. Takahashi, and S. Takahashi. Generalized exponential moving average (ema) model with particle filtering and anomaly detection. *Expert Systems with Applications*, 73:187 – 200, 2017. doi: 10.1016/j.eswa.2016.12.034

[23] N. V. T. Nguyen, V. T. Nguyen, V. Pham, and T. Dang. Finanviz: Visualizing emerging topics in financial news. In *2018 IEEE International Conference on Big Data (Big Data)*, pp. 4698–4704, 2018.

[24] J. Ni and C. Zhang. An efficient implementation of the backtesting of trading strategies. In *Parallel and Distributed Processing and Applications*, pp. 126–131, 2005.

[25] N. J. Park, K. M. George, and N. Park. A multiple regression model for trend change prediction. In *2010 International Conference on Financial Theory and Engineering*, pp. 22–26, 2010.

[26] L. Pastor and R. F. Stambaugh. Are stocks really less volatile in the long run? *The Journal of Finance*, 67(2):431–477, 2012.

[27] S. Rudolph, A. Savikhin, and D. S. Ebert. Finvis: Applied visual analytics for personal financial planning. In *2009 IEEE Symposium on Visual Analytics Science and Technology*, pp. 195–202, 2009.

[28] P. Srivastava and S. Mazhar. Comparative analysis of sharpe and sortino ratio with reference to top ten banking and finance sector mutual funds. *International Journal of Management Studies*, V:93, 10 2018. doi: 10.18843/ijms/v5i4(2)/10

[29] E. Thorp. The kelly criterion in blackjack, sports betting, and the stock market. *Handbook of Asset and Liability Management*, 1, 12 2008. doi: 10.1016/B978-044453248-0.50015-0

[30] T. Wiecki, A. Campbell, J. Lent, and J. Stauth. All that glitters is not gold: Comparing backtest and out-of-sample performance on a large cohort of trading algorithms. *The Journal of Investing*, 25(3):69–80, 2016. doi: 10.3905/joi.2016.25.3.069

[31] X. Yue, X. Shu, X. Zhu, X. Du, Z. Yu, D. Papadopoulos, and S. Liu. Bitextract: Interactive visualization for extracting bitcoin exchange intelligence. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):162–171, 2019.

[32] H. Zhou and P. S. Kalev. Algorithmic and high frequency trading in Asia-Pacific, now and the future. *Pacific-Basin Finance Journal*, 53(C):186–207, 2019. doi: 10.1016/j.pacfin.2018.10.

[33] H. Ziegler, M. Jenny, T. Gruse, and D. A. Keim. Visual market sector analysis for financial time series data. In *2010 IEEE Symposium on Visual Analytics Science and Technology*, pp. 83–90, 2010.