A Fast Heuristic for Segmented Technical Analysis to Support Trading Operations

Yang Sun*
California State University, Sacramento, Sacramento, California, USA

Sohil Varshney
Oak Ridge High School, El Dorado Hills, California, USA

Technical analysis is a valuable dimension for understanding stock price movements and trends to facilitate short-term trading operations. Support and resistance trend lines serve as the foundation of technical analysis for supporting trading decisions. Price oscillators can help determine where one may be within a trend. While it is desirable that support and resistance lines be drawn in a segmented fashion when a long time series is present, traditionally this is done manually. In this research we develop a heuristic algorithm to automatically determine 1) the number of segments and 2) the split points in a multi-objective segmentation problem in order to generate segmented support and resistance lines. Our experiments based on the stock prices of the 100 largest companies listed on the Nasdaq Exchange show that the algorithm is computationally efficient and practically effective for supporting a large number of trading decisions for both teams involved in trading operations at financial firms as well as individuals, or serving as an embedded part of an automated trading system.

* Corresponding Author. E-mail address: suny@csus.edu

I. INTRODUCTION

Operations management has become increasingly critical among the success factors for financial institutions as trading operations have developed and evolved from data-entry and mass labor-intensive processes to highly automated processes (Van den Brink, 2003). To support trading operations, fundamental and technical analyses are the two main methodologies for analyzing the financial markets and asset valuations. Fundamental analysis is used to support long term investment strategies and confidences (Lee, Sun and Varshney, 2016), whereas technical analysis is considered an operational decision tool for supporting short-term trading. Technical analysis serves as a valuable dimension to understanding stock price movements and trends, especially when they diverge substantially from fundamental data and explanations that include both macro and micro variables. The year 2018 serves as a good example of the value of technical analysis: the United States stock markets witnessed the worst October since October 2008 (Great Recession), the worst December since 1931, the worst Christmas Eve ever on record, and a negative year across all asset classes – all in spite of the best economic growth of 3.5% in more than a decade, the best employment market in 50 years, and record consumer and business confidence levels in 20 years. These call for a fast analytical tool for data segmentation, turning point detection, and short-term trend projection to support frequent trading decisions when noisier non-stationary time series data are present.

Among classical technical analysis tools, horizontal support and resistance levels are the easiest to eyeball and draw; however the support for profitable trading decisions from these horizontal lines is limited. Support and resistance trend lines with rising or falling trends serve better as the foundation of technical analysis for trading decisions. Price oscillators can help determine where we may be within a trend channel and serve as a catalyst point preparing us for a trading action that can be identified when the stock approaches the trend line. Factors such as sentiment, flows, and seasonality can help us use history to forecast investor behavior. It is therefore desirable that support and resistance lines be drawn in a segmented fashion when a long time series is present. Segmented time series analysis is widely used in economic and social studies such as public health administration to detect intervention effects and interruptive behaviors (Cheong et al., 2011, Taljaard et al., 2014, Bernal, Cummins, and Gasparrini, 2017); however, traditionally the segmentation is often done manually in financial technical analysis (Salcedo, 2004). In this research we develop a heuristic algorithm to automatically determine 1) the number of segments and 2) the split points, in order to generate segmented support and resistance lines using an optimization approach.

We study the effect of the algorithm on the stock prices of the 100 largest companies listed on the Nasdaq Exchange from July 2018 to March 2019. Our research adds to the literature by developing a fast algorithm for solving the time series segmentation problem as a multi-objective nonlinear optimization problem in an automated process, to enable more refined real-time trading decisions based on segmented trend lines that distill the information derived from the algorithm and combine with technical analysis to improve the odds of success. Computational studies show that the algorithm is efficient and effective for supporting a large amount of trading decisions.

This paper is organized as follows. Section II reviews the relevant background and literature. Section III models this analytical problem as a multi-objective nonlinear optimization problem. Section IV introduces the heuristic algorithm for solving the problem. Section V fine-tunes the algorithm using an empirical approach and demonstrates the efficiency and effectiveness of the algorithm in computational studies. Concluding remarks and suggestions for future research are presented in Section VI.

II. LITERATURE REVIEW

There is extensive literature and research spanning technical analysis, trading strategies, patterns and algorithms, profitability of trading rules, and efficacy of short term investor price discovery as well as returns across various asset classes and markets, both in the U.S. as well as internationally. Among the seminal papers, the following were most widely cited. Brock, Lakonishok, and LeBaron (1992) test two of the simplest and most popular trading rules—moving average and trading range break—by utilizing the Dow Jones Index from 1897 to 1986. They extend standard statistical analysis through the use of bootstrap techniques. Overall, their results provide strong support for the technical strategies. They show that buy signals consistently generate higher returns than sell signals, and further, the returns following buy signals are less volatile than returns following sell signals, and further, the returns following buy signals are less volatile than returns following sell signals. Moreover, returns following sell signals are negative, which is not easily explained by any of the currently existing equilibrium models.

Marshal, Cahan, and Cahan (2008) investigate whether intraday technical analysis is profitable in the U.S. equity market. They report that surveys of market participants indicate that they place more emphasis on technical analysis (and less on fundamental analysis) the shorter the time horizon. Using two bootstrap methodologies, they find that that none of the 7,846 popular technical trading rules they test are profitable after data snooping bias is taken into account and fail to find evidence that the market is inefficient over this time horizon.

Gradojevic and Gencay (2013) show that from the market microstructure perspective, technical analysis can be profitable when informed traders make systematic mistakes or when uninformed traders have predictable impacts on price. However, chartists face a considerable degree of trading uncertainty because technical indicators such as moving averages are essentially imperfect filters with a nonzero *phase shift*. Consequently, technical trading may result in erroneous trading recommendations and substantial losses. This paper presents an uncertainty reduction approach based on fuzzy logic that addresses two problems related to the uncertainty embedded in technical trading strategies: market timing and order size. The results of high-frequency exercises show that ‘fuzzy technical indicators’ dominate standard moving average technical indicators and filter rules.

Manahov, Hudson, and Gebka (2014) investigate how high frequency trading affects technical analysis and market efficiency by using a special adaptive form of the Strongly Typed Genetic Programming (STGP)-based learning algorithm. They compare the STGP performance with that of parametric and non-parametric models and document experiments that suggest that the STGP forecasting technique significantly outperforms the traditional econometric models. They find evidence that the excess returns are both statistically and economically significant, even when appropriate transaction costs are taken into account.

Allen and Karjalainen (1999) use a genetic algorithm to learn technical trading rules for the S&P 500 index using daily prices from 1928 to 1995. After transaction costs, the rules do not earn consistent excess returns over a simple buy-and-hold strategy in the out-of-sample test periods. The rules are able to identify periods as in the index when daily returns are positive and volatility is low and out when the reverse is true. These latter results can largely be explained by low-order serial correlation in stock index returns.

A survey by Cohen, Kudryavtsev and Hon-Snir (2011) shows that support and resistance lines are the most commonly used technical analysis tools when investors make buy/sell decisions. Zapranis and Tsinaslanidis (2012) use a rule-based mechanism to identify local horizontal support and resistance levels. Daily adjusted closing prices from July 1990 to July 2010 for 232 stocks from Nasdaq and 501 stocks from NYSE are used for the classical bounce test and the resultant success rate of 60.99% is consistent with earlier studies by Curcio et al. (1997) and Osler (2000).

Salcedo (2004) indicates that price channel charts based on rising or falling support and resistance lines can successfully support profitable trades if they can be developed in a segmented fashion with appropriate breakpoints to catch the changes in trends, as such charts are very easy for investors to read to identify buy/sell opportunities. However, while it is relatively easy to draw segmented support and resistance trend lines manually, to the best of our knowledge the literature lacks a systematic approach to generating segmented price channels. This research contributes to the literature by proposing a novel method for drawing trend line-based price channels algorithmically, so that a large number of charts can be quickly generated to support investors' decisions or generate trading signals based on trading strategies and logic for large investment portfolios.

Stock prices are by nature time series data. In the data mining literature, time series segmentation is a well-studied algorithmic problem. Keogh et al. (2004) survey time series segmentation articles that include Fourier transformation, wavelets, symbolic mapping and piecewise linear representation with various applications in physics, biology and medicine, signal and image processing, and finance. Piecewise linear representation is the most commonly used approach. Oliver, Baxter and Wallace (1998) use a maximum likelihood approach to detect behavioral change in time series data with a fixed number of change points. With a similar approach, Guralnik and Srivastava (1999) use an iterative algorithm to determine if the segments should be further partitioned. Bryant and Duncan (1994), Duncan and Bryant (1999), Ge and Smyth (2000) and Ge and Smyth (2001) use recursive dynamic programming algorithms to detect change points of linear patterns with the maximum likelihood approach. Hébrail et al. (2010) use dynamic algorithms to segment functional data into a given number of clusters represented by piecewise constants. Chung et al. (2004) and Chen et al. (2013) use genetic algorithms to determine split points of a time series where the fitness of the solution is defined based on vertical distances between critical points. Ahmed et al. (2010) indicate that the machine learning algorithm *Classification and Regression Trees* (CART) can be used to segment time series where each splitting leads to a significant reduction in the mean squared error.

Time series segmentation problems mentioned above all pursue solutions using a sole criterion as the effectiveness measure, and the algorithms are relatively inefficient. This paper provides original contributions to the literature by:

- Proposing a multi-objective non-linear optimization version of the time series segmentation problem
- Proposing an extremely fast greedy heuristic algorithm to solve the time series segmentation problem efficiently and draw segmented support and resistance trend lines automatically
- Integrating an empirical approach to tuning the algorithm and validating the effectiveness of the solutions with inputs from investment experts in order to obtain robust support for massive short-term trading operations involving large investment portfolios

III. PROBLEM STATEMENT

Table 1 summarizes the mathematical notation used in this paper. The time series segmentation problem in data mining literature can be presented in various technical ways. For example, the problem can be stated as producing the best piecewise linear representation for a giving time series T that minimizes total error (often presented as the sum of squared errors). Certain constraints can also be given, e.g., that the segmentation needs to be done with exactly m segments, or the maximum error of any individual segment cannot exceed a given threshold. The linear representation of a segment can be simply the straight line connecting the first and last data points in the segment, or the linear regression line for the segment. The former can have lines of consecutive segments connecting to each other, while the latter often has disjointed lines, although the method can be adjusted to make consecutive lines connect. In this research, we use the disjointed piecewise linear regression version since it is unnecessary to represent financial time series with connected lines. It is also unnecessary to predetermine the number of segments for short-term forecasting and decision support purposes.

TABLE 1. NOTATION.

Notation	Definition
T	The given time series to be segmented
n	The number of data points in T
$t_1, t_2, \dots, t_n$	Data points in T
m	The number of segments for T
$S_1, S_2, \dots, S_m$	The segments for T
$s_1, s_2, \dots, s_{m-1}$	The split points (the last data point in each segment)
β_{s0}	Intercept of the support trend line of the last segment
β_{s1}	Slope of the support trend line of the last segment
β_{r0}	Intercept of the resistance trend line of the last segment
β_{r1}	Slope of the resistance trend line of the last segment
SSE	Total sum of squared errors from all segments
$CumSSE$	Cumulative total sum of squared errors
x	The cost threshold in the segmentation algorithm
i, j	Indexes used in the algorithm
K	Total number of time series in the portfolio

We define the problem as a multi-objective nonlinear optimization problem with a combination of technical and empirical senses.

3.1. Decision variables

- The number of linear regression segments m for the given time series T with n data points.
- Split points $s_1, s_2, \dots, s_{m-1}$ in time series T.
- A fine-tuned threshold x that is used as the stopping criterion in the segmentation algorithm.

3.2. Objectives

- Produce the best piecewise linear regression representations for all K time series in the trading portfolio that appropriately show the short-term trends that can lead to satisfactory trading decisions based on the judgment of a group of trading experts with the following technical objectives for each stock:
  1) Minimizing the sum of squared errors in a least square sense
  2) Minimizing the number of segments m

Note that objectives 1) and 2) are conflicting objectives in this multi-objective optimization problem. For instance, when n is even, it is straightforward to show that having a total number of $n/2$ segments, where each pair of consecutive data points form a segment, will carry zero error; however, this contradicts the minimal m objective and would be useless for showing trends. On the other hand, technically the minimal m is always one; however, this could carry a large amount of error and would not support trading operations very well in most cases. While trading experts are often able to tell what the best segments are for each stock by recognizing patterns in the time series plot, an automated process would require the optimization algorithm to be robust so that appropriate segmentations can be done efficiently for a large amount of time series to support massive, frequent trading decisions. The cost threshold x is used to fine-tune the algorithm to help trade-off the two objectives in the most appropriate way for all stocks in the portfolio based on the judgment of the trading experts in an empirical sense.

IV. HEURISTIC ALGORITHM FOR SEGMENTATION

The goal is to automatically detect both the number of segments and the choices of split points in the time series segmentation problem, and this is computationally challenging given that the problem can be reduced to a partition problem with a large solution search space if n is large. Standard machine learning algorithms such as CART can be used to partition the time series data and fit a simple regression line within each partition as the partitioning can be represented as a decision tree (Loh, 2011). The algorithm is inefficient due to its recursive nature and is less effective for the purpose of supporting massive, frequent trading decisions. In this research we propose a fast greedy heuristic for solving the time series segmentation problem that can be less accurate but very robust in supporting quick trading decisions when the investment portfolio is large.

For the given time series *T* with *n* data points $t_1, t_2, \ldots, t_n$, the pseudo algorithm for producing the best segmentation is expressed as follows.

Step 1. Data Cleansing.
If *n* is an odd number,
    delete the first (oldest) data point. $n \leftarrow n$ - 1 and re-index *T* with indexes 1 through *n*;
    otherwise, keep the original time series *T* with indexes 1 through *n*.
Step 2. Initial Segmentation.
Divide T into $m = \frac{n}{2}$ segments where each pair of consecutive data points form a segment, i.e., $S_1 = (t_1, t_2), S_2 = (t_3, t_4), \ldots, S_m = (t_{n-1}, t_n)$. At this point the regression line in a segment is simply the connection of the two data points and the current *SSE* is zero. *CumSSE* $\leftarrow$ *SSE*.
Step 3. Merging neighborhood segments.
j = 0
Do while (m > 1)
    j += 1
    For *i* = 1 to $m - 1$
        Merge S_i and S_{i+1} and produce the linear regression line for the merged segmentation.
        Record the *SSE* increase, SSE_i, as a result of this merge.
    End of the *i* loop.
    Choose the *i*th merge that has the lowest SSE_i to implement and record the resulting *SSE*. *CumSSE* $\leftarrow$ *CumSSE* + *SSE*.
    If $j > 1$ and $SSE_i <= x * CumSSE$,
        record *j* as the best number of merges and current segmentation as the best solution to replace the initial solution or the last best solution;
        otherwise, keep the last best merge as the stopping point and the last best segmentation solution as the best solution.
    $m \leftarrow m - 1$.
Step 4. Solution Output
Output the recorded best segmentation solution.

While the condition *SSE > x*CumSSE* can be used as a stopping criterion, the algorithm is implemented to the last possible merge producing only one segment for research purpose. Given that the time complexity of the algorithm is less than $O(n^2)$, this sacrifice of efficiency can be ignored.

The threshold x represents the maximum allowed cost increase when implementing the next merge. It is straightforward to show that, a large x will potentially lead to more merges resulting in fewer segments and a small x will do the opposite. In addition, because *CumSSE* increases as the number of merges j increases, the overall tendency of this algorithm is to produce more merges or fewer segments unless a merge generates a very large increase in total error. Computational experiments are implemented in this research to fine-tune the x threshold so that the algorithm is able to robustly produce the appropriate number of segments for all stocks in an investment portfolio that would generate support and resistance trend lines that are simple and clear, and capable of recommending satisfactory short-term trading decisions (buy, hold, or sell) based on the empirical judgment of trading experts.

V. COMPUTATIONAL STUDY

The testing data set in this research contains daily adjusted close prices of the 100 largest companies listed on the Nasdaq Exchange from July 2018 to March 2019. This provides us with 100 time series, each with 188 data points (nine months of data). For each stock we produce a segmentation solution on the price time series and plot the support and resistance trend lines of the last segment. The support and resistance trend lines are automatically generated using nonlinear programming in a constrained least squared sense. The nonlinear programming models are presented in Table 2.

TABLE 2. OPTIMIZATION MODELS FOR SUPPORT AND RESISTANCE TREND LINES.

For the support trend line:	For the resistance trend line:
Minimize: $\sum_{i=1}^{n}\left(t_i - (\beta_{s0} + \beta_{s1} i)\right)^2$ Subject to: $t_i - (\beta_{s0} + \beta_{s1} i) \le 0, \forall i$	Minimize: $\sum_{i=1}^{n}\left(t_i - (\beta_{r0} + \beta_{r1} i)\right)^2$ Subject to: $t_i - (\beta_{s0} + \beta_{s1} i) \ge 0, \forall i$

The generated chart is presented to a group of five trading experts and each of them is asked to choose one of the following answers independently:

- Yes, the projection appropriately shows where we are in the price trend and can support a good buy, hold or sell decision on the next trading day;
- or, No, the projection does not appropriately show where we are in the price trend and can lead to a bad buy, hold or sell decision on the next trading day.

All surveyed experts have over 15 years of successful experience in trading operations that include a senior investment officer at one of the world's largest pension fund as well as senior investment strategists at large banks and mutual funds. Note that new data points from April 2019 are available when the generated support and resistance trend lines are presented to the experts. The experts can easily validate their judgments to provide robust inputs.

As an example, Figure 1 shows the non-segmented trend lines for a given stock. The movement range would hint for a sell decision. Figure 2 shows how the segmentation is done after using the proposed algorithm. Figure 3 shows how *CumSSE* increases as the number of merges increases when the algorithm is implemented. The best solution is obtained before the increase in *CumSSE* gets too large. Figure 4 shows the new trend lines on the last (most recent) segment of the time series based on the automatically produced segmentation that are presented to the experts. The new trend lines would suggest a hold decision and potentially a buy decision in a few days. New data points show that holding this stock for one more month would generate another 10% gain in a relative short term. The experts agree that the automatically generated segmentation and trend lines match with what they would have done manually and would appropriately support short-term decision making based on price oscillators, i.e., where we are in a channel trend. Manually drawing support and resistant trend lines would not be feasible for a large portfolio with massive data for decision making purposes. On the other hand, the automatic segmentation produced by the algorithm can easily be embedded in a decision support system to generate trading recommendations and signals for short term trading operations

.

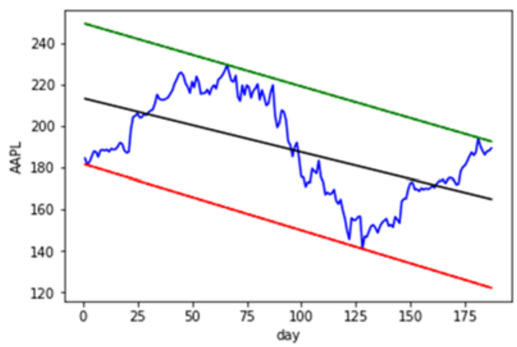

FIGURE 1. NON-SEGMENTED TREND LINES FOR AAPL, JULY 2018 – MARCH 2019.

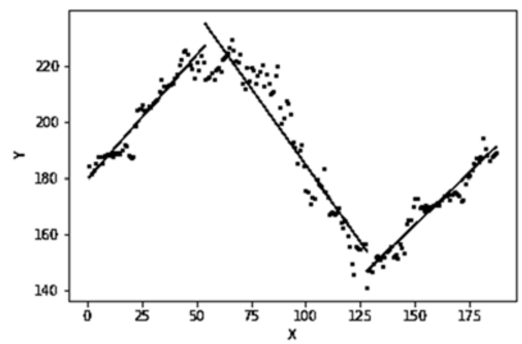

FIGURE 2. SEGMENTED LINEAR REGRESSION FOR AAPL, JULY 2018 – MARCH 2019.

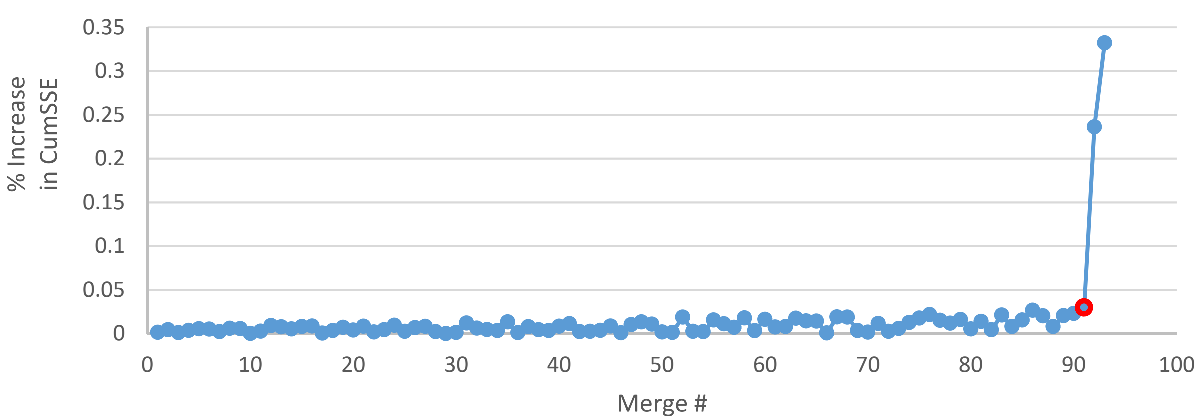

FIGURE 3. INCREASE IN *CUMSSE* AS MERGE NUMBER INCREASES.

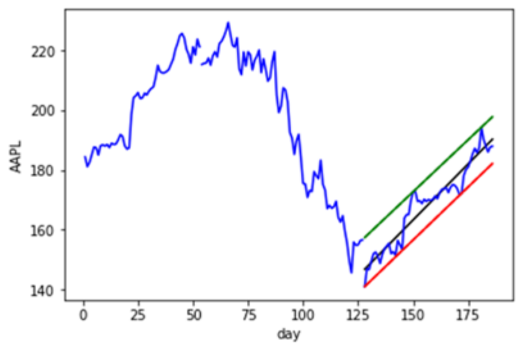

FIGURE 4. TREND LINES OF THE MOST RECENT SEGMENT FOR AAPL, JULY 2018 – MARCH 2019.

Computational experiments using the proposed algorithm are done in two phases. In Phase I, two logistic regression models are fitted using designed *x* points given that the response is binomial. At each design point, five stocks are randomly selected from the portfolio. The segmented trend lines are produced for each stock and presented to the five experts for their Yes or No answers and therefore each design point gets 25 inputs. The first set of experiments has *x* in the range of 0.005 and 0.040 and the logistic regression model shows that in this range, increasing *x* will lead to a better segmentation with fewer segments that is more likely to provide a successful prediction. The second set of experiments has *x* in the range of 0.040 and 0.080. As *x* increases in this range, the performance of the algorithm quickly gets worse since too few segments could lead to an unsuccessful prediction. The analysis shows that an *x* between 0.035 and 0.04 should provide a robust performance with a success rate of approximately 0.97 for the given portfolio. The performance of the algorithm is not sensitive to the change of *x* as long as *x* is chosen in the neighborhood of the optimal value. This result is validated using the unselected stocks in the portfolio as testing data in Phase II. Results of the experiments are summarized in Figure 5 and Table 3.

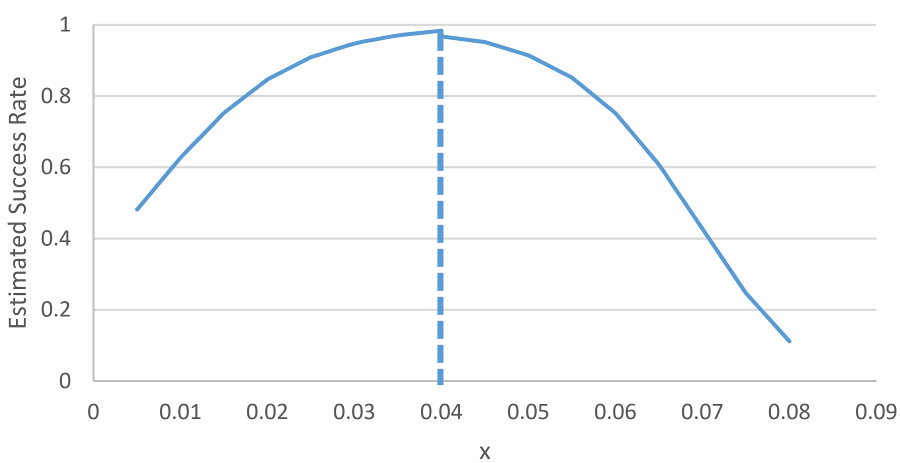

FIGURE 5. ESTIMATED SUCCESS RATE OF THE ALGORITHM BASED ON *X* VALUES.

TABLE 3. SUMMARY OF COMPUTATIONAL EXPERIMENTS.

Phase	*x* Setting	Number of cases (number of stocks × number of experts)	Estimated Success Rate
I	0.005	5 × 5	0.48
	0.01	5 × 5	0.63
	0.015	5 × 5	0.75
	0.02	5 × 5	0.84
	0.025	5 × 5	0.91
	0.03	5 × 5	0.95
	0.035	5 × 5	0.97
	0.04	10 × 5	0.97
	0.045	5 × 5	0.95
	0.05	5 × 5	0.91
	0.055	5 × 5	0.85
	0.06	5 × 5	0.75
	0.065	5 × 5	0.61
	0.07	5 × 5	0.43
	0.075	5 × 5	0.25
	0.08	5 × 5	0.11
II	0.03	5 × 5	0.96
	0.035	5 × 5	1.0
	0.04	5 × 5	0.96

The same data set is also tested using the largely available CART algorithm from the Python community for time series splitting and segmented regression using a decision tree approach (Pedregosa et al., 2011). While the splitting can be optimized, the maximum depth of the tree that controls the maximum number of segments needs to be predefined. Experiments with a maximum depth of 3 (or a maximum of 8 segments) result in very much the same outputs as our proposed algorithm; however it takes a personal desktop with a 3.4GHz quad-core Intel Core i7 processor with 6MB shared L3 cache and 16GB 1600MHz DDR3L onboard memory 30 to 40 seconds to analyze one stock. For a portfolio of 100 stocks, the estimated computational time is about 1 hour that is considered too long for supporting massive operational trading decisions. One the other hand, it takes our proposed algorithm, also coded in Python, only a few seconds to finish the same 100-stock job using the same device since the time complexity of the algorithm is very low. In addition to automatically producing charts with support and trend lines or other plots in a segmented way, decision logics can also be coded to generate trading suggestions or signals for large investment portfolio.

VI. CONCLUSION

In this research we propose a fast heuristic algorithm for a multi-objective segmentation problem that can be fine-tuned to produce segmented technical analysis such as support and resistance trend lines to support short-term trading decisions facing large investment portfolios with massive time series data. In this research we are able to document and illustrate that with a relatively good choice of x, the proposed algorithm can be very successful in short-term decision support with massive time series data in a computationally efficient and practically effective way. Results are validated by trading experts and the algorithm can be integrated into decision support systems for trading operations teams at financial firms as well as individuals to use, or embedded into automated trading systems. Our paper also suggests future research should be directed to further exploring the integration of segmented technical analysis and trading strategies and logic in trading decision support systems, and to promoting further investigation into high-frequency trading.

Acknowledgement: The authors would like to trank the financial experts who participated in the survey study.

REFERENCE

Van den Brink, G.J., *Banking/Trading - Operations Management,* Palgrave Macmillan, 2003

Lee, J.J., Sun, Y. and Varshney, S.B., "A Weakening Chain? Impact of China's Economic Slowdown on Global Supply Chain Management of US Multinational Enterprises: A Capital Market's Perspective", Journal of Supply Chain and Operations Management, 14(1), 2016, 68-84.

Cheong, S.A., Fornia, R.P., Lee, G.H.T., Kok, J.L., Yim, W.S., Xu, D.Y. and Zhang, Y., "The Japanese Economy in Crises: A Time Series Segmentation Study", *Economics*, 6(5), 2012, 1–81.

Taljaard, M., McKenzie, J.E., Ramsay, C.R. and Grimshaw, J.M., "The Use of Segmented Regression in Analysing Interrupted Time Series Studies: an Example in Pre-hospital Ambulance Care", *Implementation Science*, 9, 2014.

Bernal, J.L., Cummins, S., and Gasparrini, A., "Interrupted Time Series Regression for the Evaluation of Public Health Interventions: a Tutorial", *International Journal of Epidemiology*, 46(1), 2017, 348–355.

Salcedo, Y., "Channel Surfing for Profits", *Futures*, 33(4), 2004, 62-64.

Brock, W., Lakonishok, J. and LeBaron, B., "Simple Technical Trading Rules and the Stochastic Properties of Stock Returns", *Journal of Finance*, 47(5), 1992, 1731-1764.

Marshal, B., Cahan, R. and Cahan, J., "Does Intraday Technical Analysis in the U.S. Equity Market Have Value?", *Journal of Empirical Finance*, 15(2), 2008, 199-210.

Gradojevic, N. and Gencay, R., "Fuzzy Logic, Trading Uncertainty, and Technical Trading", *Journal of Banking and Finance*, 37(2), 2013, 578-586.

Manahov, V., Hudson, R. and Gebka, B., "Does High Frequency Trading Affect Technical Analysis and Market Efficiency? And If So, How?", *Journal of International Financial Markets, Institutions and Money*, 28, 2014, 131-157.

Allen, F. and Karjalainen, R., "Using Genetic Algorithms to Find Technical Trading Rules", *Journal of Financial Economics*, 51(2), 1999, 245-271.

Cohen, G., Kudryavtsev, A. and Hon-Snir, S., "Stock Market Analysis in Practice: Is It Technical or Fundamental?", *Journal of Applied Finance and Banking*, 1(3), 2011, 125-138.

Zapranis, A. and Tsinaslanidis, P.E., "Identifying and Evaluating Horizontal Support and Resistance Levels: an Empirical Study on US Stock Markets", *Applied Financial Economics*, 22(19), 2012, 1571–1585

Curcio, R., Goodhart, C., Guillaume, D. and Payne, R., "Do Technical Trading Rules Generate Profits? Conclusions from the Intra-day Foreign Exchange Market", *International Journal of Finance and Economics*, 2(4), 1997, 267–80.

Osler, C.L., "Support for Resistance: Technical Analysis and Intraday Exchange Rates", *FRBNY Economic Policy Review*, 6(2), 2000, 53–68.

Keogh, E., Chu, S., Hart, D. and Pazzani, M., "Segmenting Time Series: A Survey and Novel Approach", *Data Mining in Time Series Databases - Series in Machine Perception and Artificial Intelligence*, 57, 2004, 1-21

Oliver, J.J., Baxter, R.A. and Wallace, C.S., "Minimum message length segmentation", *Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Melbourne, Australia, 1998, 222-233

Guralnik, V. and Srivastava, J., "Event detection from time series data", *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, California, 1999, 33-42.

Bryant, G.F. and Duncan, S.R., "A solution to the segmentation problem based on dynamic programming", *Proceedings of the 3rd IEEE Conference on Control and Applications*, Vol. 2, 1994, 1391-1396.

Duncan, S.R. and Bryant, G.F., "A new algorithm for segmenting data from time series", *Proceedings of the 35th IEEE Conference on Decision Control*, Vol. 3, 1996, 3123-3128.

Ge, X. and Smyth, P., "Deformable Markov model templates for time-series pattern matching", *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Boston, Massachusetts, 2000, 81-90.

Ge, X. and Smyth P., "Segmental Semi-Markov Models for Endpoint Detection in Plasma Etching", *IEEE Transactions on Semiconductor Engineering*, 2001, 1-5.

Hébrail, G., Hugueney, B., Lechevallierc, Y. and Rossi, F., "Exploratory Analysis of Functional Data via Clustering and Optimal Segmentation", *Neurocomputing*, 73(7), 2010, 1125-1141.

Chung, F.L., Fu, T.C., Ng, V., and Luk., R.W.P., "An Evolutionary Approach to Pattern-based Time Series Segmentation", *IEEE Transactions on Evolutionary Computation*, 8(5), 2004, 471-489.

Chen, C., Tseng, V., Yu, S., and Hong, H., "Time Series Pattern Discovery by a PIP-based Evolutionary Approach", *Soft Computing*, 17(9), 2013, 1699-1710.

Ahmed, N., Atiya, A., Gayar, N., and El-Shishiny, H.. "An Empirical Comparison of Machine Learning Models for Time Series Forecasting", *Econometric Reviews*, 29(5-6), 2010, 594-621.

Loh, W.Y., *Classification and Regression Trees*, John Wiley & Sons, Inc., 2011

Pedregosa F., Varoquaux, B., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, 12, 2011, 2825-2830.