Novel Data-Driven Fuzzy Algorithmic Volatility Forecasting Models with Applications to Algorithmic Trading

A. Thavaneswaran
Department of Statistics
University of Manitoba
Winnipeg, Canada
thavane@umanitoba.ca

You Liang
Department of Mathematics
Ryerson University
Toronto, Canada
you.liang@ryerson.ca

Zimo Zhu
Department of Statistics
University of Manitoba
Winnipeg, Canada
zhuz3456@myumanitoba.ca

Ruppa K. Thulasiram
Department of Computer Science
University of Manitoba
Winnipeg, Canada
tulsi@cs.umanitoba.ca

***Abstract*—The explosion of algorithmic trading has been one of the most prominent trends in the finance industry. In this paper, two strategies for algorithmic trading such as Bollinger bands and the simple moving average (SMA) crossover strategy are studied in the fuzzy settings. The commonly used Bollinger bands trading strategy assumes that the difference between an asset's price and its SMA is normally distributed. However, it is shown that a data-driven t distribution is more appropriate to model the difference between an asset's price and its SMA. A novel data-driven fuzzy Bollinger bands strategy is proposed for algo trading. A good strategy should have a good algo return on investment with low algo volatility. Therefore, forecasting algo volatility and identifying an appropriate distribution of algo returns play a crucial role in algo trading. Sharpe Ratio (SR) is a measure of average algo return earned in excess of the risk-free rate per unit of algo volatility. For a class of SMA crossover strategies with varying window sizes, fuzzy estimates of SR are computed based on various risk measures including the data-driven volatility estimate (DDVE). SR fuzzy forecasts are computed using two recently proposed volatility forecasting models such as data-driven exponentially weighted moving average (DD-EWMA) and data-driven neuro volatility models. The main reason of using the fuzzy approach is to provide α-cuts (interval forecasts) of the SR. An empirical application on a set of widely traded technology stocks shows that the proposed models deliver forecasts of SR with small errors.**

***Index Terms*—Algorithmic trading, Data-driven fuzzy Bollinger bands, Algo volatility, Fuzzy Sharpe ratio**

I. Introduction

Algorithmic trading uses algorithms defined by sets of instructions (or rules) to automatically monitor stock prices and place trades. Algorithmic trading is taking over the world of financial trades, and the days of manual trading are coming to an end. On a typical trading day, computers account for 50% to 60% of market trades, according to Art Hogan, chief market strategist for B. Riley FBR (refer to the website money.cnn.com/2018/02/06/investing/wall-street-computers-program-trading/index.html). When the market is extremely volatile, they can make up 90% of trades. Most common algorithmic trading strategies follow trends in simple moving averages (SMA) and related technical indicators based on Bollinger bands. A comprehensive introduction of algorithmic trading can be found in [2] and [3]. Most trades are initiated based on the occurrence of desirable trends, which are easy and straightforward to implement through algorithms. In this paper, the short-term and long-term SMA trend indicators are used. A SMA is an average of the past adjusted closing prices $P_t, t = 1, 2, \ldots, D$. A SMA crossover strategy applies moving average crossovers to enter buy (long) or sell (short) positions. The trend following position indicator (1/-1) at each trading time t is summarized as follows:

- Long position if short-term SMA $\geq$ long-term SMA, and the position is entered as 1
- Short position if short-term SMA $<$ long-tem SMA, and the position is entered as -1

Adjusted closing prices can be converted to simple returns as $R_t = (P_t - P_{t-1})/P_{t-1}$. Algo return A_t is calculated as return R_t multiplied by the corresponding position for each t. Assume that daily algo return A_t has mean μ_A and standard deviation σ_A then the unconditional daily Sharpe ratio (SR) is defined as

$$Daily.SR = \frac{E(A_t - r_f/N)}{\sqrt{Var(A_t - r_f/N)}} = \frac{\mu_A - r_f/N}{\sigma_A}, \quad (1)$$

where r_f is the annual risk-free rate and N is the number of trading periods in a year. For daily data, $N = 252$; monthly data, $N = 12$; quarterly data, $N = 4$. The higher the SR, the better the reward to risk ratio. That is, a high SR indicates that an asset generates high returns without taking excessive risk. If a strategy has a high algorithmic volatility (which is usually the standard deviation of the algo return over the period of trading), then the strategy has a high risk factor even if the return is high at the end. A good strategy should have a good algo return on investment with low algo volatility. One example of the influence of high volatility is that the "flash crash" of May 2010, which wiped billions from U.S. stock markets in less than an hour. Therefore, estimating and forecasting algo volatility and identifying an appropriate distribution of algo returns play a crucial role in algorithmic trading. In this paper, a recently proposed data-driven volatility estimate (DDVE) and a data-driven exponentially weighted

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

moving average (DD-EWMA) volatility forecast are used to study SR estimation and forecasting.

In [10], the sign correlation of a random variable X with mean μ is defined as

$$\rho_X = \text{Corr}(X-\mu, \text{sign}(X-\mu)). \tag{2}$$

In applications we use the sample sign correlation $\hat{\rho}_X$ to estimate ρ_X. For any symmetric distribution with finite mean μ and variance σ^2, the sign correlation ρ_X is given by

$$\rho_X = \frac{\text{E}|X-\mu|}{\sigma}.$$

Therefore, $E|X-\mu|/\rho_X$ is an unbiased estimator of volatility σ. If X follows a Student's t distribution with sign correlation ρ_X and finite variance, the corresponding degrees of freedom (d.f.) ν can be computed by solving the following equation (see [10]):

$$2\sqrt{\nu-2} = (\nu-1)\rho_X \text{Beta}\left[\frac{\nu}{2}, \frac{1}{2}\right]. \tag{3}$$

Following [10], the data-driven algo volatility estimator in terms of algo returns $A_1, \cdots, A_n$, is given as

$$\hat{\sigma}_A = \frac{1}{n}\sum_{t=1}^{n}\frac{|A_t-\bar{A}|}{\hat{\rho}_A}, \tag{4}$$

where $\hat{\rho}_A$ is the sample sign correlation of A_t. The asymptotic variance of the data-driven algo volatility estimator $\hat{\sigma}_A$ is

$$\left(\frac{1-\rho_A^2}{\rho_A^2}\right)\frac{\sigma_A^2}{n}, \tag{5}$$

which depends only on the algo variance σ_A^2 and the sign correlation ρ_A. The conventional volatility estimator, sample standard deviation s_n, is the square root of the estimated sample variance s_n^2, with the asymptotic variance

$$\frac{(\kappa+2)}{4}\frac{\sigma_A^2}{n},$$

where κ is the excess kurtosis. Empirical studies (see [10]) have shown that some financial data follows certain heavy-tailed distributions such as Student's t with d.f. less than four and with theoretically infinite kurtosis. For example, technology stocks are typically volatile, and may demonstrate obvious price fluctuations. It is shown in [10] that the asymptotic variance of the data-driven volatility estimator $\hat{\sigma}_A$ is smaller than that of the sample standard deviation estimator s_n. That is $\hat{\sigma}_A$ is more suitable to estimate volatility of returns with large kurtosis.

Volatility forecasting with applications is one of the most active and successful areas of research in time-series econometrics and economic forecasting. However, their application to algorithmic trading has not yet been widely studied. [10] proposed a novel DD-EWMA volatility forecasting model to forecast the volatility directly and obtained value-at-risk (VaR) forecasts. [12] was the first one to propose a novel direct data-driven neuro predictive model for conditional volatility and to study the fuzzy VaR forecasts. Let the conditional mean and the conditional variance of algo return A_t be

$$E(A_t|\mathcal{F}_{t-1}) = \mu_t, Var(A_t|\mathcal{F}_{t-1}) = \sigma_t^2, t = 1, \cdots, n,$$

where $\mathcal{F}_{t-1}$ is the past data up to time $t-1$. Following [10], the algo volatility forecasting model can be written as

$$\hat{\sigma}_{t+1} = (1-\alpha)\,\hat{\sigma}_t + \alpha\frac{|A_t-\bar{A}|}{\hat{\rho}_A}, \quad 0<\alpha<1. \tag{6}$$

This volatility model is data-driven in the sense that the optimal value of the smoothing constant α is obtained by minimizing the one-step ahead forecast error sum of squares (FESS) and the sample sign correlation $\hat{\rho}_A$ is used to identify the conditional distribution of A_t.

There has been a growing interest in combining randomness and fuzziness to solve option pricing problems in finance (see [7] and [9] for details). However, many proposed fuzzy methods remain difficult to use in practice and hence, there is a need for data-driven approachs to fit the fuzzy models for real data. Neural network (NN) is one of the most common methods to approximate a multivariate nonlinear function. [11] discussed fuzzy option pricing using data-driven feed forward NN volatility model. Superiority of the fuzzy forecasting method over the minimum mean square forecasting had been demonstrated for fuzzy coefficient (linear as well as nonlinear) time series models in [8]. [12] applied their neuro volatility model to forecast VaR with actual financial data. In this paper, a data-driven fuzzy Bollinger bands strategy is proposed for algo trading using DDVEs in (4). The estimate in (4) is also used to calculate fuzzy estimates of SR for a class of SMA crossover strategies with varying window sizes. Data-driven volatility forecasting model (6) is used to study the DD-EWMA rolling fuzzy forecasts of SR. Moreover, the data-driven neuro volatility model is used to study the rolling neuro fuzzy forecasts of SR. The main reason of using the fuzzy approach is to provide α-cuts for SR forecasts.

The remainder of the article is organized as follows. Section II discusses data-driven fuzzy Bollinger bands using algo volatility estimates in equation (4). Section III discusses the fuzzy estimates of SR using various risk measures including algo volatility estimates in (4). SMA crossover strategies usually use ad hoc procedures to choose the window sizes. Window size selection for SR estimates using real data is discussed in section III. Moreover, SR forecasts are calculated based on DD-EWMA forecasting and neuro volatility forecasting models. Algorithms are provided to illustrate our models and methods. Section IV presents the empirical results, and section V provides conclusions. R Markdown file can be download from the Github repository: github.com/datasciencecodeshare/Data-Driven-Fuzzy-Volatility-Forecasting-Algorithmic-Trading.

II. Data-Driven Fuzzy Bollinger Bands

Bollinger bands, introduced by John Bollinger in the early 1980s, are well known in the trading community. Bollinger bands include three different lines: upper, middle, and lower band. The purpose of these bands is to give a relative definition of high and low. In theory, prices are high at the upper band and they are low at the lower band. The middle line is a D-period SMA at each t:

$$Middle_{t,D} = SMA_{t,D} = \frac{\sum_{i=t-D+1}^{t} P_i}{D}.$$

In general, a trading strategy can be constructed such that traders open a position when the middle SMA line is nearing the lower band and close a position when the middle SMA line reaches the upper one. The standard Bollinger bands formula sets the lower and upper bands as two sample standard deviations below and above the middle SMA. Assuming normality, standard Bollinger bands are defined as

$$(Lower_{t,D}, Upper_{t,D}) = SMA_{t,D} \pm 2s_{t,D},$$

where

$$s_{t,D} = \sqrt{\frac{\sum_{i=t-D+1}^{t}(P_i - SMA_{t,D})^2}{D-1}}.$$

In the following Algorithm 1, we will illustrate how to use the sign correlation of the difference $P_t - SMA_{t,D}$ to determine an appropriate distribution and construct the data-driven fuzzy Bollinger bands. The α-cuts of the data-driven Bollinger bands formula using t fuzzy numbers is given by

$$\overline{P_t(\alpha)} = (Lower_{t,D}, Upper_{t,D}) = SMA_{t,D} \pm t_{\alpha/2,df.est}\hat{\sigma}_{t,D}$$

where

$$\hat{\sigma}_{t,D} = \sum_{i=t-D+1}^{t} \frac{|P_i - SMA_{t,D}|}{\hat{\rho}_{res}}$$

and $\hat{\rho}_{res}$ is the sample sign correlation of the difference $P_t - SMA_{t,D}$. The estimated d.f., df.est, is obtained by solving (3) for ν. Algorithm 1 is used to calculate fuzzy Bollinger bands.

Algorithm 1 Data-driven fuzzy Bollinger bands

Require: Adjusted closing stock prices $P_t, t = 1, \ldots, n$, moving average window size D, α-level (for fuzzy α-cuts)

1: $DSMA_t \leftarrow SMA(P_t, D)$ {Calculate D-period SMA}
2: $res_t \leftarrow P_t - DSMA_t$ {Calculate residuals}
3: $\hat{\rho}_{res} \leftarrow$ Corr$(res_t - \bar{res}, \text{sign}(res_t - \bar{res}))$ {Calculate sample sign correlation of residuals}
4: $df.est \leftarrow$ Solve $2 * \sqrt{\nu - 2} = (\nu - 1) * \hat{\rho}_{res} *$ Beta $\left[\frac{\nu}{2}, \frac{1}{2}\right]$ for ν {Estimate the d.f. of the conditional t-distribution of P_t}
5: $vol.cal \leftarrow function(y)\ mean(abs(y - mean(y)))/\hat{\rho}_{res}$ {Define a function for rolling volatility estimates}
6: $vol.sma_t \leftarrow rollapply(P_t, width = D, FUN = vol.cal, by.column = TRUE, fill = NA, align = \text{“}right\text{”})$ {Calculate D-day rolling volatility estimates}
7: $lower.bound \leftarrow DSMA_t - qstd(1 - \alpha/2, nu = df.est) * vol.sma_t$ {Calculate the lower band}
8: $upper.bound \leftarrow DSMA_t + qstd(1 - \alpha/2, nu = df.est) * vol.sma_t$ {Calculate the upper band}

Consider adjusted closing prices of a set of stocks: Advance Auto Parts, Inc. (AAP), Apple Inc. (AAPL), International Business Machines Corporation (IBM), Microsoft Corporation (MSFT), Alphabet Inc. (GOOG) and Amazon.com, Inc. (AMZN). The full dataset runs from 2010-01-02 to 2019-12-13. It follows from Table I that t-distribution instead of normal distribution ($\hat{\rho}_{res}$ = 0.79) is appropriate to model res_t. Especially, t-distributions with d.f. less than four are more appropriate for res_t for the volatile stocks such as AAPL, MSFT and AMZN.

TABLE I
SUMMARY STATISTICS OF RESIDUALS FOR ALL ASSETS: 2010-01-04 - 2019-12-13

Assets	Mean	SD	κ	Lag 1 sample acf			$\hat{\rho}_{res}$	df.est
				res_t	$\lvert res_t \rvert$	res_t^2		
AAP	0.444	5.238	3.158	0.915	0.862	0.854	0.710	4.074
AAPL	0.923	4.666	3.780	0.926	0.881	0.862	0.689	3.622
MSFT	0.488	1.658	4.411	0.866	0.815	0.723	0.690	3.638
GOOG	4.002	22.747	3.367	0.895	0.831	0.760	0.717	4.282
IBM	0.152	3.850	1.803	0.915	0.834	0.856	0.761	7.267
AMZN	6.251	35.305	8.379	0.905	0.878	0.780	0.612	2.824

SD: standard deviation; κ: excess kurtosis; res_t: residuals; res_t^2: daily squared residuals; $|res_t|$: absolute residuals; acf: autocorrelation; $\hat{\rho}_{res}$: sample sign correlation; df.est: estimated d.f. from $\hat{\rho}_{res}$

Therefore, the normality assumption for standard Bollinger bands is not valid and the values of the bands are under-estimated for all the stocks, especially for AAPL, MSFT and AMZN. The following Figure 1 visualizes the fuzzy Bollinger bands trading strategy for AMZN using the window size $D = 20$ for the whole trading period in the upper plot. If we drag the time slide bar to certain periods for further information shown in the lower plot, the certain trading period with detailed information for each stock is visualized and then summarized in Table II. The visualization of the fuzzy Bollinger bands trading strategy for other stocks will be found in the R Markdown file. It is shown in Figure 1 and Table II that the data-driven 0.01-cut Bollinger bands (purple) and 0.05-cut Bollinger bands (green) are wider than standard Bollinger bands (blue). Standard Bollinger bands constructed with the normality assumption underestimate the values of upper and lower bands and affects the trading signals. Data-driven fuzzy Bollinger bands are more realistic, and can be used as trading strategies especially for stock prices with heavy-tailed distributions such as AAPL, MSFT and AMZN. It is a collection of α-cuts, including standard Bollinger bands as a special case when $\alpha = 0.05$ and $\hat{\rho}_{res} = 0.79$.

TABLE II
FUZZY BOLLINGER BANDS (α-CUTS)

Asset	Date	Price	SMA	Standard BB	0.05-cut of BB	0.01-cut of BB
AAP	18-12-31	169	153	(124, 182)	(115, 191)	(91, 215)
AAPL	18-11-27	172	192	(162, 222)	(157, 228)	(132, 253)
MSFT	19-06-24	137	129	(119, 139.8)	(117, 142)	(108, 151)
GOOG	18-04-06	1007	1072	(951, 1194)	(922, 1221)	(829, 1315)
IBM	18-10-26	118	132	(113, 151)	(110, 153)	(100, 163)
AMZN	18-12-31	1502	1566	(1334, 1798)	(1276, 1857)	(1018, 2114)

BB:Bollinger bands

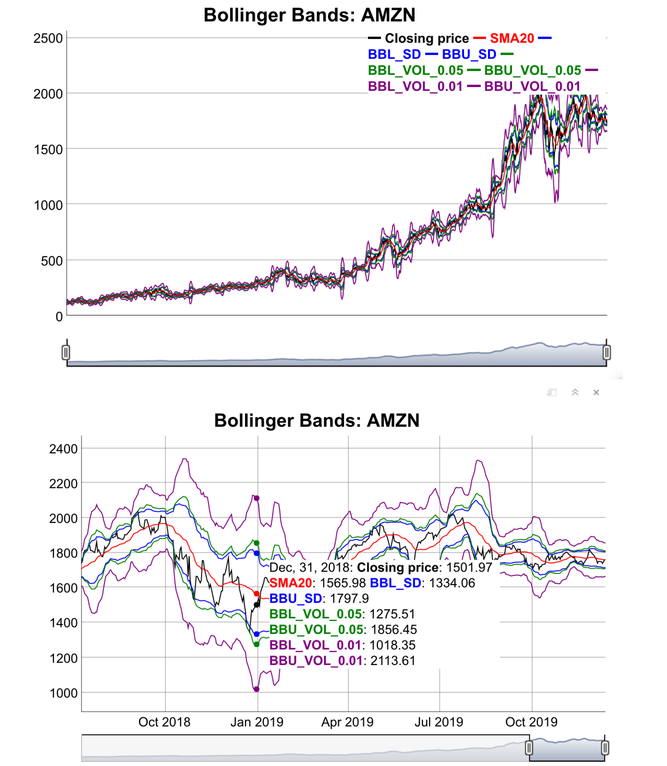

Fig. 1. Data-driven Bollinger bands: closing price (black), SMA20 (red), standard Bollinger bands (blue), 0.05-cut Bollinger bands (green), 0.01-cut Bollinger bands (purple)

III. Data-Driven Sharpe Ratio Estimates and Forecasts

A. Moving Average Strategies

Moving average crossover strategies apply two separate moving averages with varying lengths for indications of one of the averages crosses over or under the other. Bullish signals occur when the shorter moving average crossing above the longer moving average, and bearish signals are sent by the shorter moving average crossing below the longer moving average. The SMA crossover trading strategy is described in Algorithm 2. The algorithm buys a unit of share of an asset if the shorter SMA crosses above the longer SMA and sells a unit of share vice versa.

Consider adjusted closing prices of the set of stocks: AAP, AAPL, IBM. MSFT, GOOG and AMZN from 2015-01-02 to 2019-12-13. The SMA crossover strategy with short-term window size 9 and long-term window size 24 is applied to all stocks. The following Figure 2 visualizes this strategy for AMZN. If we drag the time slide bar to the recent period for buying versus selling, the green shaded area from 2019-02-07 to 2019-03-01 indicates selling while the pink shaded area from 2019-03-01 to 2019-05-15 indicates buying/holding. The visualization of SMA trading strategy for other stocks will be discussed in the R Markdown file.

Algorithm 2 SMA crossover trading strategy

Require: Adjusted closing stock price $P_t, t = 0, \ldots, n$, short-term moving average window size S, long-term moving average window size L

$R_t \leftarrow (P_t - P_{t-1})/P_{t-1}, t = 1, \cdots, n$ {Calculate returns}
$SSMA_t \leftarrow SMA(P_t, S)$ {Calculate short-term SMA}
$LSMA_t \leftarrow SMA(P_t, L)$ {Calculate long-term SMA}
if $SSMA_t \geq LSMA_t$ **then**
    $Position_t \leftarrow 1$
else
    $Position_t \leftarrow -1$
end if
$AlgoReturn \leftarrow R_t * Position_t$ for each t {Calculate algo returns}
return $AlgoReturn$

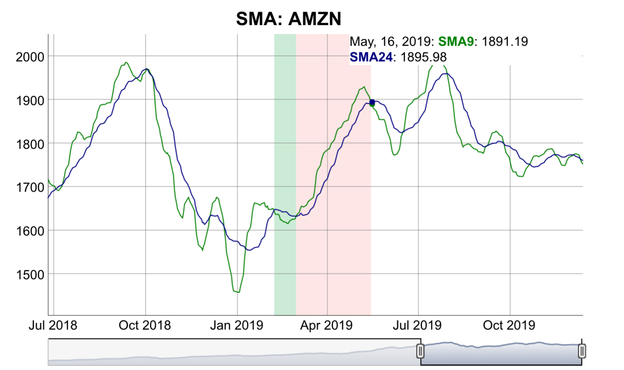

Fig. 2. SMA crossover strategy with shaded regions for buying versus selling

B. Data-Driven Sharpe Ratio Estimates

Using a SMA crossover strategy, we study the data-driven SR estimates as well as various SR estimates based on the algo returns. The daily SR is calculated using (1). In practice, the annualized SR is more convenient for comparison. The annualized SR is computed by dividing the annualized mean excess return by the annualized volatility of excess return. The simple interest formula is used to calculate annualized SR from the daily one. Let N be the number of trading days in a year, then the annualized SR is calculated as

$$Annualized.SR = \frac{N(\mu_A - r_f/N)}{\sqrt{N}\sigma_A} = \sqrt{N} Daily.SR, \quad (7)$$

where μ_A and standard deviation σ_A are the mean and standard deviation of daily algo returns, respectively. Equation (7) is used to compute the annualized SR estimates from daily volatility estimates obtained by using the DDVE $\hat{\sigma}_A$ given in (4) for σ_A. Moreover, the α-cuts of the estimate of σ_A using the asymptotic variance given in equation (5) can be written as

$$\overline{\sigma_A(\alpha)} = (LL^{\sigma_A}, UL^{\sigma_A}) = \hat{\sigma}_A \pm cv_\alpha \sqrt{\frac{(1-\hat{\rho}_A^2)\hat{\sigma}_A^2}{n\hat{\rho}_A^2}}, \quad (8)$$

where cv_α is the critical value of level α. Then, the α-cuts of annualized SR can be written as

$$\left(\frac{\sqrt{252}(\bar{A}_t - r_f/252)}{UL^{\sigma_A}}, \frac{\sqrt{252}(\bar{A}_t - r_f/252)}{LL^{\sigma_A}}\right) \tag{9}$$

Algorithm 3 implements the above method.

Algorithm 3 SR fuzzy estimates using DDVE

Require: Algo return $A_t, t = 1, \cdots, n$ from Algorithm 2, annual risk-free interest rate r_f

1: $\hat{\rho}_A \leftarrow \text{Corr}(A_t - \bar{A}, \text{sign}(A_t - \bar{A}))$ {Calculate sample sign correlation}

2: $\hat{\sigma}_A \leftarrow mean(|A_t - \bar{A}|/\hat{\rho}_A)$ {Compute DDVE of algo returns}

3: α-cuts of $\sigma_A \leftarrow (LL^{\sigma_A}, UL^{\sigma_A}) = \hat{\sigma}_A \pm cv_\alpha * \sqrt{(1-\hat{\rho}_A^2) * \hat{\sigma}_n^2/(\hat{\rho}_A^2 * n)} = (LL^{\sigma_A}, UL^{\sigma_A})$ {Compute α-cuts of algo volatility σ_A based on $\hat{\sigma}_A$}

4: $Daily.SR \leftarrow (\bar{A}_t - r_f/252)/\hat{\sigma}_A$ {Compute daily SR}

5: $Annualized.SR \leftarrow \sqrt{252} * Daily.SR$ {Compute annualized SR}

6: α-cuts of Annualized.SR $\leftarrow \sqrt{252} * (\bar{A}_t - r_f/252)/UL^{\sigma_A}, \sqrt{252} * (\bar{A}_t - r_f/252)/LL^{\sigma_A}$ {Compute α-cuts of annualized SR}

7: **return** $\hat{\sigma}_A$, α-cuts of σ_A, Annualized.SR, α-cuts of Annualized.SR

C. Data-Driven Window Size Selection

We compute and interpret the effect of short-term and long-term SMA window sizes on SR estimates. Annualized SR estimates are computed using DDVEs for long-term window sizes: 20, 40, 60 and 200. For each long-term window size, the short-term window size ranges from 1 to the long-term window size. It is shown in Figure 3 for AMZN with long-term window size 20, SR stays stable (mean reverting) after the short-term window size 10. That is, if we use the long-term window as 20 then the corresponding data-driven short-term window size can be selected as 10.

D. Dynamic Data-Driven Rolling Sharpe Ratio Forecasts

Significant sample autocorrelation of the absolute algo returns shown in Table III suggests that time varying volatility models are more appropriate for algo returns. The following Algorithm 4 is used to compute the daily data-driven EWMA volatility forecasts. Algorithm 5 is used to compute the daily data-driven neuro volatility forecasts. Equation (7) is used to compute the annualized SR forecasts using these two daily rolling volatility forecasts in Algorithm 6.

In Algorithm 4, based on past observations of algo returns, we compute the sample sign correlation $\hat{\rho}_A$ and observed algo volatility $Z_t = |A_t - \bar{A}|/\hat{\rho}_A$. The optimal smoothing constant α is determined by minimizing the one-step ahead FESS. Using the optimal α, we calculate the smoothed value S_t recursively, and calculate the last optimal smoothed value as the one-day-ahead volatility forecast. The first l observations is used to calculate the initial smoothed value S_0, and root

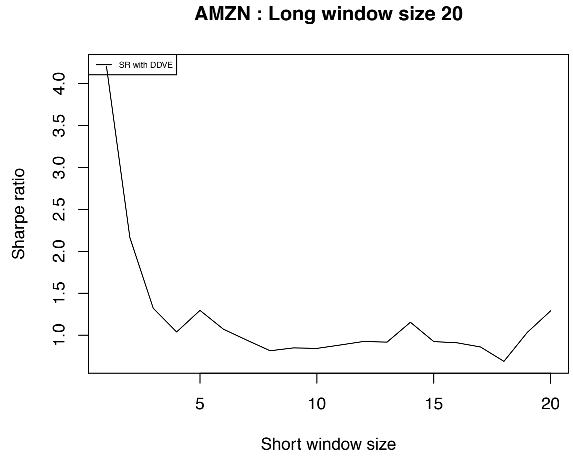

Fig. 3. Annualized SR estimates vs. short-term window sizes

mean square error (RMSE) of volatility forecasts is calculated as

$$\sum_{t=l+1}^{k}(Z_t - S_{t-1})^2/(k-l).$$

In Algorithm 5, observed algo volatilities are also computed based on the sample sign correlation of algo returns. Then, the R function nnetar from the R package forecast is used to calculate neuro volatility forecasts. A rolling window approach in Algorithm 6 is used to calculate two different daily and annualized SR forecasts based on DD-EWMA algo volatility forecasts or neuro algo volatility forecasts. For each one of the rolling windows, the algo volatility forecast and daily SR forecast are calculated by DD-EWMA (Algorithm 4) and neuro volatility (Algorithm 5). The α-cuts of daily SR forecasts can be calculated by daily SR forecasts from all rolling windows as

$$\begin{aligned}\overline{Daily.SR(\alpha)} &= (LL^{Daily.SR}, UL^{Daily.SR}) \\ &= mean(Daily.SR_i) \pm cv_\alpha sd(Daily.SR_i).\end{aligned}$$

The α-cuts of annualized SR forecasts can be calculated by daily SR forecasts as

$$\overline{Annualized.SR(\alpha)} = (\sqrt{N}LL^{Daily.SR}, \sqrt{N}UL^{Daily.SR}).$$

IV. Empirical Study

Consider closing stock prices of the set of stocks: AAP, AAPL, IBM. MSFT, GOOG and AMZN. The full dataset runs from 2015-01-02 to 2019-12-13. Algorithm 2 calculates daily algo returns A_t using the SMA crossover strategy with short-term window size 9 and long-term window size 24. The summary statistics of A_t for all six assets are listed in Table III. For all stocks, the sample sign correlation $\hat{\rho}_A$ of algo returns is less than 0.79. We model the distribution of A_t as a t distribution with mean μ_A and volatility σ_A. The d.f. of the t distribution is determined by the sample sign correlation

Algorithm 4 Dynamic DD-EWMA algo volatility forecasts

Require: Algo return $A_t, t = 1, \cdots, k$ from Algorithm 2

1: $\hat{\rho}_A \leftarrow \text{Corr}(A_t - \bar{A}, \text{sign}(A_t - \bar{A}))$
2: $Z_t \leftarrow |A_t - \bar{A}|/\hat{\rho}_A$ {Compute observed algo volatility}
3: $S_0 \leftarrow \bar{Z}_l$ {Initial volatility forecast using first l observations}
4: $\alpha \leftarrow (0.01, 0.3)$ *by* 0.01{Set a range for α}
5: $S_t \leftarrow \alpha * Z_t + (1 - \alpha) * S_{t-1}, t = 1, \ldots, k$
6: $\alpha_{opt} \leftarrow \min_\alpha \sum_{t=l+1}^{k} (Z_t - S_{t-1})^2$ {Determine optimal α by minimizing FESS}
7: $S_t \leftarrow \alpha_{opt} * Z_t + (1 - \alpha_{opt}) * S_{t-1}, t = 1, \ldots, k$
8: $\hat{\sigma}^{A,DD} \leftarrow S_k$ {Calculate one-step-ahead DD EWMA algo volatility forecast based on k observations}
9: $RMSE \leftarrow \sum_{t=l+1}^{k} (Z_t - S_{t-1})^2/(k - l)$ {Compute minimum RMSE}
10: **return** $\alpha_{opt}, \hat{\sigma}^{A,DD}, RMSE$

Algorithm 5 Dynamic data-driven algo neuro volatility forecasts

Require: Algo return $A_t, t = 1, \cdots, k$ from Algorithm 2

1: $\hat{\rho}_A \leftarrow \text{Corr}(A_t - \bar{A}, \text{sign}(A_t - \bar{A}))$
2: $Vol_t \leftarrow |A_t - \bar{A}|/\hat{\rho}_A$ {Compute observed algo volatility}
3: $Vol.nnet \leftarrow nnetar(Vol)$ {Compute data-driven neuro algo volatility using R function nnetar}
4: $\hat{\sigma}^{A,NN} \leftarrow forecast(Vol.nnet, h = 1)\$mean$ {Compute one-step-ahead neuro volatility forecast using R function forecast}
5: **return** $\hat{\sigma}^{A,NN}$

$\hat{\rho}_A$ from data. It shows that all the algo returns have a t-distribution with d.f less than 4. Moreover, the absolute algo returns $|A_t|$ are significantly autocorrelated, which indicates the volatility clustering.

TABLE III
SUMMARY STATISTICS OF DAILY ALGO RETURN FOR ALL ASSETS: 2015-01-02-2019-12-13

Assets	Mean	SD	κ	Lag 1 sample acf			$\hat{\rho}_A$	df.est
				A_t	$\|A_t\|$	A_t^2		
AAP	0.0005	0.020	19.975	-0.030	0.094	0.032	0.635	2.99
AAPL	0.0008	0.016	3.778	-0.001	0.157	0.098	0.706	3.97
MSFT	0.0003	0.015	6.612	-0.097	0.184	0.114	0.683	3.52
GOOG	-0.0005	0.015	14.495	-0.007	0.114	0.042	0.673	3.38
IBM	0.0007	0.013	6.938	-0.028	0.125	0.045	0.686	3.58
AMZN	0.0014	0.018	8.713	-0.012	0.175	0.099	0.670	3.34

SD: standard deviation; κ: excess kurtosis; A_t: daily algo return; A_t^2: daily squared algo return; $|A_t|$: daily absolute algo return; acf: autocorrelation; $\hat{\rho}_A$: sample sign correlation; df.est: estimated d.f. from $\hat{\rho}_A$

A. Data-Driven Sharpe Ratio Estimates

We use Algorithm 3 to produce Table IV for calculating annualized SR estimates. We first compute the sample sign correlation of A_t, and the DDVE using (4). Moreover, using the asymptotic variance of the data-driven volatility estimator, fuzzy α-cuts of the estimates of the daily algo volatility and annualized SR estimates are provided in Table IV. AMZN has

Algorithm 6 Dynamic rolling SR fuzzy forecasts

Require: Algo return $A_t, t = 1, \cdots, n$ from Algorithm 2, risk free rate r_f, rolling window size k

1: **for** $i = 1, \ldots, n - k + 1$ **do**
2: $\hat{\sigma}_i \leftarrow \hat{\sigma}_i^{A,DD}, RMSE_i$ {Compute DD-EWMA algo volatility forecasts and the corresponding RMSEs based on k observations $A_i, \ldots, A_{i+k-1}$ using Algorithm 4}
3: $\hat{\sigma}_i \leftarrow \hat{\sigma}_i^{A,NN}$ {Instead of 2, Compute neuro algo volatility forecasts based on k observations $A_i, \ldots, A_{i+k-1}$ using Algorithm 5}
4: $Daily.SR_i \leftarrow (mean(A_i) - r_f/252)/\hat{\sigma}_i$ {Compute daily SR forecast for each rolling window of size k using data-driven volatility forecasts}
5: **end for**
6: $Daily.SR \leftarrow mean(Daily.SR_i)$ {Compute average daily SR forecast from $n - k + 1$ rolling windows of size k}
7: $RMSE \leftarrow mean(RMSE_i)$ {Compute average RMSE from $n - k + 1$ rolling windows of size k (only for DD-EWMA algo volatility forecasts)}
8: α-cuts of Daily.SR $\leftarrow$ $(LL^{Daily.SR}, UL^{Daily.SR}) = mean(Daily.SR_i) \pm cv_\alpha * sd(Daily.SR_i)$ {Compute α-cuts of daily SR forecast}
9: $Annualized.SR \leftarrow \sqrt{252} * Daily.SR$ {Compute annualized SR forecast}
10: α-cuts of Annualized.SR $\leftarrow (\sqrt{252} * LL^{Daily.SR}, \sqrt{252} * UL^{Daily.SR})$ {Compute α-cuts of annualized SR forecast}
11: **return** Daily.SR, α-cuts of Daily.SR, RMSE, Annualized.SR, α-cuts of Annualized.SR

the highest annualized SR estimate and GOOG has the lowest one.

TABLE IV
α-CUTS OF ANNUALIZED SR ESTIMATES USING RISK AS DDVE

Assets	DDVE	0.05-cut of DDVE	ASR	0.05-cut of ASR
AAP	0.0200	(0.0186, 0.0214)	0.3085	(0.2888, 0.3310)
AAPL	0.0155	(0.0147, 0.0164)	0.7636	(0.7229, 0.8091)
MSFT	0.0145	(0.0136, 0.0154)	0.2300	(0.2170, 0.2446)
GOOG	0.0151	(0.0142, 0.0161)	-0.6390	(-0.6019, -0.6809)
IBM	0.0129	(0.0121, 0.0137)	0.7984	(0.7537, 0.8489)
AMZN	0.0181	(0.0169, 0.0192)	1.1273	(1.0615, 1.2019)

DDVE: data-driven volatility estimate; ASR: Annualized Sharpe ratio

Similar to Algorithm 3, sample standard deviation (s_n), mean absolute deviation (MAD) ($\hat{\rho}_A s_n$), and VaR ($\text{VaR}_{0.05}$) based on t distribution of A_t are used to estimate the daily volatility and annualized SR. Results are summarized in Table V, VI and VII, respectively. It can be seen that, for all stocks with t d.f. less than 4, fuzzy α-cut estimates of annualized SR using DDVE is narrower than that using sample standard deviation.

TABLE V

α-CUTS OF ANNUALIZED SR ESTIMATES USING RISK AS SD

Assets	SD	0.05-cut CI of SD	ASR	0.05-cut of ASR
AAP	0.0200	(0.0174, 0.0227)	0.3083	(0.2725, 0.3549)
AAPL	0.0155	(0.0145, 0.0166)	0.7631	(0.7150, 0.8182)
MSFT	0.0145	(0.0133, 0.0157)	0.2298	(0.2123, 0.2504)
GOOG	0.0151	(0.0134, 0.0169)	-0.6386	(-0.5734, -0.7206)
IBM	0.0129	(0.0118, 0.0140)	0.7973	(0.7357, 0.8702)
AMZN	0.0181	(0.0164, 0.0197)	1.1266	(1.0320, 1.2403)

SD: standard deviation; ASR: Annualized Sharpe ratio

TABLE VI

α-CUTS OF ANNUALIZED SR ESTIMATES USING RISK AS MAD

Assets	MAD	0.05-cut of MAD	ASR	0.05-cut of ASR
AAP	0.0127	(0.0110, 0.0144)	0.4854	(0.4291, 0.5588)
AAPL	0.0110	(0.0102, 0.0117)	1.0812	(1.0130, 1.1593)
MSFT	0.0099	(0.0091, 0.0107)	0.3365	(0.3109, 0.3666)
GOOG	0.0102	(0.0090, 0.0113)	-0.9489	(-0.8520, -1.0707)
IBM	0.0089	(0.0081, 0.0096)	1.1620	(1.0722, 1.2682)
AMZN	0.0121	(0.0110, 0.0132)	1.6810	(1.5399, 1.8507)

MAD: mean absolute deviation; ASR: Annualized Sharpe ratio

B. *Dynamic Data-Driven Rolling Sharpe Ratio Forecasts*

A rolling window approach is applied to forecast the daily and annualized SR using Algorithm 6. The selected data covers 1200 days, with 201 overlapping rolling windows. Each window of size 1000 is used to calculate a one-day-ahead algo volatility forecast and the corresponding RMSE using Algorithm 4. For each stock, the daily SR forecast is calculated by the average of the 201 daily SR forecasts from each of the 201 rolling windows. The annualized SR forecast is calculated using (7). Results are summarized in Table VIII. For each of the stocks, based on 201 RMSEs of the one-day-ahead volatility forecasts, the average RMSE is also reported in the second column of Table VIII. User time in seconds of each stock is reported in Table VIII using R function proc.time. The code is running in R version 3.6.1 with macOS Mojave, Version 10.14.5 (18F132). Usually point forecasts of SR (red line in Figure 4) are reported in the literature. In this paper, more realistic fuzzy α-cuts of the forecasts (interval forecasts) are reported in Table VIII, and plotted in Figure 4 for AMZN (blue lines and purple lines) are given. The reason is that there is strong evidence that the volatility of rolling daily SR forecasts is time varying (shown by black line in Figure 4).

TABLE VII

α-CUTS OF ANNUALIZED SR ESTIMATES USING RISK AS $\text{VaR}_{0.05}$

Assets	$\text{VaR}_{0.05}$	0.05-cut of $\text{VaR}_{0.05}$	ASR	0.05-cut of ASR
AAP	0.0267	(0.0248, 0.0285)	0.2316	(0.2165, 0.2488)
AAPL	0.0226	(0.0212, 0.0239)	0.5260	(0.4970, 0.5586)
MSFT	0.0208	(0.0196, 0.0221)	0.1599	(0.1508, 0.1703)
GOOG	0.0223	(0.0209, 0.0236)	-0.4341	(-0.4094, -0.4619)
IBM	0.0182	(0.0170, 0.0193)	0.5669	(0.5339, 0.6043)
AMZN	0.0245	(0.0229, 0.0261)	0.8316	(0.7804, 0.8899)

$\text{VaR}_{0.05}$: Value at Risk at $\alpha = 0.05$; ASR: Annualized Sharpe ratio

TABLE VIII

ROLLING SR FUZZY FORECASTS USING DD-EWMA VOLATILITY MODELS

Assets	RMSE	DSR	ASR	Time	0.05-cut of ASR	0.01-cut of ASR
AAP	0.020	0.021	0.338	0.910	(0.221, 0.456)	(0.184, 0.493)
AAPL	0.019	0.055	0.869	0.873	(0.366, 1.371)	(0.208, 1.529)
MSFT	0.016	0.009	0.138	0.850	(-0.213, 0.488)	(-0.323, 0.599)
GOOG	0.018	-0.049	-0.777	1.076	(-1.193, -0.362)	(-1.323, -0.232)
IBM	0.016	0.059	0.940	0.839	(0.455, 1.424)	(0.303, 1.576)
AMZN	0.021	0.098	1.558	0.823	(0.819, 2.297)	(0.586, 2.530)

RMSE: average root mean square error for one-step-ahead volatility forecast;
DSR: average daily SR with one-step-ahead volatility forecast;
ASR: average annualized SR with one-step-ahead volatility forecast; Time: user time in sec

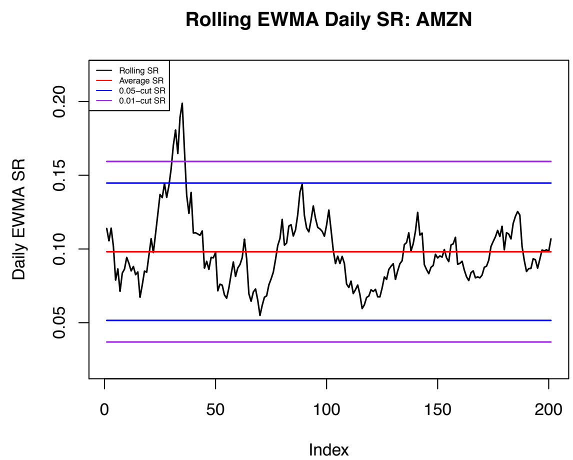

Fig. 4. Rolling daily SR forecasts and average SR forecast using DD-EWMA volatility forecasts

We also apply a rolling window approach to calculate the daily and annualized SR forecasts using neural network. The selected data covers 1200 days, and we compose 201 overlapping rolling windows, each of window size 1000 to calculate a one-day-ahead algo volatility forecast using Algorithm 5. Using Algorithm 6, the daily SR forecast is calculated by the average of the 201 rolling daily neuro SR forecasts. The annualized SR is computed using (7). Results are summarized in Table IX. Rolling daily neuro SR forecasts and its average of AMZN is plotted in Figure 5. The user time of computing the daily SR forecast and the annualized SR forecast for each stock index is listed in Table IX. It follows from Table VIII and Table IX, for each stock, the user time using DD-EWMA volatility forecasts is faster than that using data-driven neuro volatility forecasts. Fuzzy α-cuts (interval forecasts) are provided in Table IX, and plotted in Figure 5 for AMZN (blue lines and purple lines). There is strong evidence from Figure 5 that the volatility of rolling neuro daily SR forecasts (black line) is time varying.

V. SUMMARY AND DISCUSSION

A good trading strategy should have a good algo return on investment with low algo volatility. In this paper, two strategies for algorithmic trading, Bollinger bands and SMA crossover strategy, are studied based on smoothed estimates

TABLE IX
ROLLING SR FUZZY FORECASTS USING DATA-DRIVEN NEURO VOLATILITY MODELS

Assets	DSR	ASR	Time	0.05-cut of ASR	0.01-cut of ASR
AAP	0.021	0.330	28.705	(0.230, 0.430)	(0.199, 0.461)
AAPL	0.057	0.908	222.648	(0.333, 1.482)	(0.152, 1.663)
MSFT	0.008	0.128	148.617	(-0.189, 0.444)	(-0.289, 0.544)
GOOG	-0.052	-0.820	201.050	(-1.278, -0.361)	(-1.422, -0.217)
IBM	0.061	0.974	216.805	(0.301, 1.648)	(0.089, 1.860)
AMZN	0.089	1.411	106.815	(0.801, 2.021)	(0.609, 2.213)

DSR: average daily SR with one-step-ahead volatility forecast;
ASR: average annualized SR with one-step-ahead volatility forecast;
Time: user time in sec

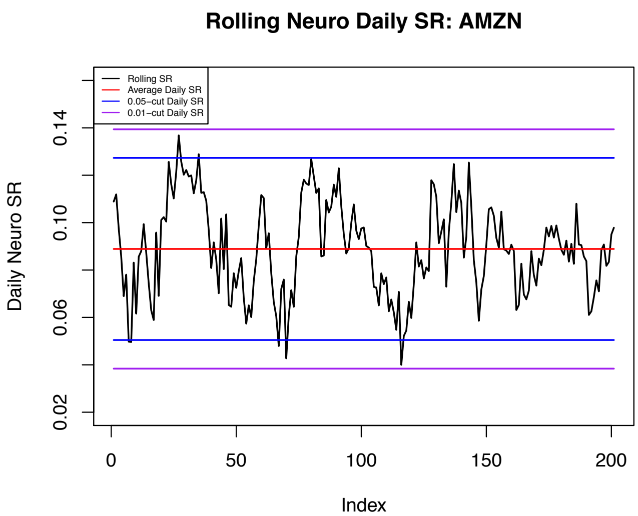

Fig. 5. Rolling daily SR forecasts and average SR forecasts using neuro volatility forecasts

and forecasts. The goal is to introduce novel data-driven volatility models to study fuzzy Bollinger bands, fuzzy SR estimates, and fuzzy SR forecasts. The main reason for using the fuzzy approach is to provide α-cuts (interval estimates or forecasts) for the volatility, Bollinger bands, and SR. For the SMA crossover strategy, fuzzy estimates of the annualized SR are computed based on various risk measures, including recently proposed DDVEs. Moreover, this study makes a strong effort to forecast SR using rolling DD-EWMA fuzzy volatility and rolling neuro fuzzy volatility models. Their performances using real financial data demonstrate that fuzzy SR forecasting models are realistic for practical purposes. Especially, the computation time using DD-EWMA volatility forecasts is faster than that using data-driven neuro volatility forecasts. For the future research, we will use dynamic models as in [13] and extend pairs trading using Kalman filtering algorithms in [6].

ACKNOWLEDGEMENT

The first and last authors acknowledge the partial funding from the Faculty of Science Inter-disciplinary research collaboration initiation grant and the Discovery grant from Natural Sciences and Engineering Research Council (NSERC). The second author acknowledges the partial funding from the Faculty of Science Start-up Fund.

REFERENCES

[1] A. Arratia, Computational Finance: An Introductory Course with R, Series ISSN 2352-3255. Atlantis Press, 2004

[2] Á. Cartea, S. Jaimungal. and J. Penalva, Algorithmic and High-Frequency Trading (Mathematics, Finance and Risk). Cambridge University Press, 2015.

[3] E. P. Chan, Algorithmic Trading: Winning Strategies and Their Rationale. New York: Wiley, 2013.

[4] R. Hyndman, G. Athanasopoulos, C. Bergmeir, et al., forecast: Forecasting functions for time series and linear models. R package version 8.12, http://pkg.robjhyndman.com/forecast, 2020.

[5] H. Konno and H. Yamazaki, "Mean-Absolute Deviation Portfolio Optimization Model and Its Applications to Tokyo Stock Market", Management Science, vol. 37 (5), pp. 519–531, 1991

[6] K. Longmore, Kalman Filter Example: Pairs Trading in R. Robot Wealth. https://robotwealth.com/kalman-filter-pairs-trading-r/, September, 2019.

[7] S. Muzzioli and B. De BAets, "Fuzzy Approaches to Option Price Modeling", IEEE Transactions on Fuzzy Systems, vol. 25 (2), pp. 392–401, 2017.

[8] A. Thavaneswaran, S. S Appadoo and A. Paseka, A, "Weighted possibilistic moments of fuzzy numbers with applications to GARCH modeling and option pricing", Math. and Comp. Modelling, vol. 49, pp. 352–368, 2009.

[9] A. Thavaneswaran, S. S Appadoo and J. Frank, Binary option pricing using fuzzy numbers. "Applied Mathematics Letters", vol. 26, pp. 65–72, 2013.

[10] A. Thavaneswaran, S. S Appadoo and J. Frank, "Generalized Value at Risk Forecasting", Communications in Statistics Theory and Methods, pp. 1–8, 2019.

[11] A. Thavaneswaran, Ruppa K. Thulasiram, J. Frank, Z. Zhu and M. Singh, "Fuzzy Option Pricing Using a Novel Data-Driven Feed Forward Neural Network Volatility Model", IEEE International Conference on Fuzzy Systems, New Orleans, LA, USA, pp. 491-496, June, 2019.

[12] A. Thavaneswaran, Ruppa K. Thulasiram, J. Frank, Z. Zhu, M.E. Hoque and N. Ravishanker, "Fuzzy Value-at-Risk Forecasts Using a Novel Data-Driven Neuro Volatility Predictive Model", IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), Milwaukee, USA, pp. 221-226, July, 2019.

[13] M. E. Thompson, "Dynamic data science and official statistics", The Canadian Journal of Statistics, vol. 46 (1), pp. 10-23, 2018.