Time Series Forecasting With a Learning Algorithm: An Approximate Dynamic Programming Approach

Ricardo Collado[1] Germán Creamer[1]

[1]School of Business
Stevens Institute of Technology
Hoboken, New Jersey

Time Series Drivers:

- Historical data:
  - Incorporated in classical time series forecast methods
  - Works best when the underlying model is fix
- Exogenous processes:
  - Not included in historical observations
  - Difficult to incorporate via classical methods
  - Could indicate changes in the underlying model

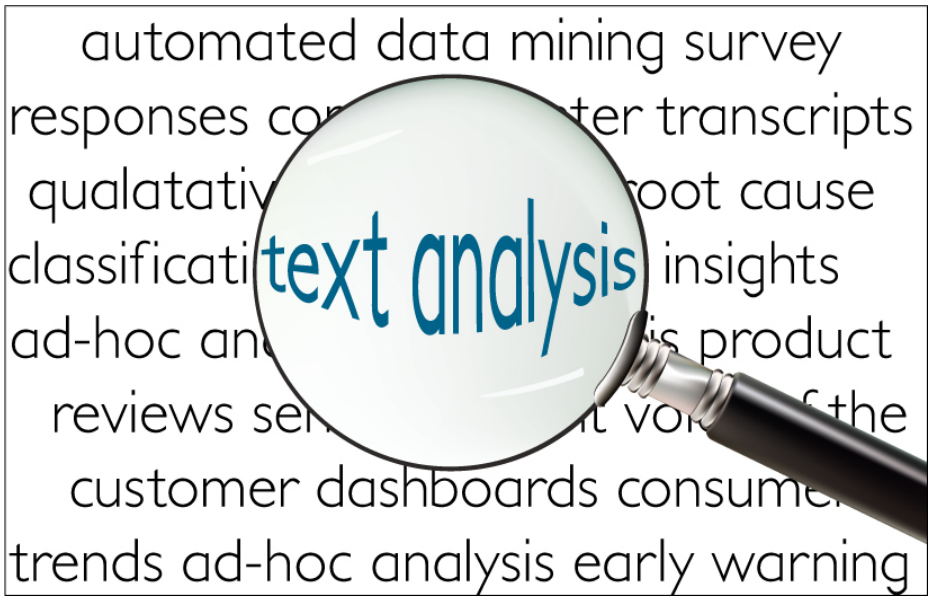
Techniques to handle changes due to external forces:

- Jump Diffusion Models
- Regime Switching Methods
- System of Weighted Experts
- Others ...

These methods do not directly integrate alternative data sources available to us

Alternative data sources:

- **Text & News Analysis**
- Social Networks Data
- Sentiment Analysis



Alternative data sources:

- Text & News Analysis
- **Social Networks Data**
- Sentiment Analysis

Alternative data sources:

- Text & News Analysis
- Social Networks Data
- **Sentiment Analysis**

We study time series forecast methods that are:

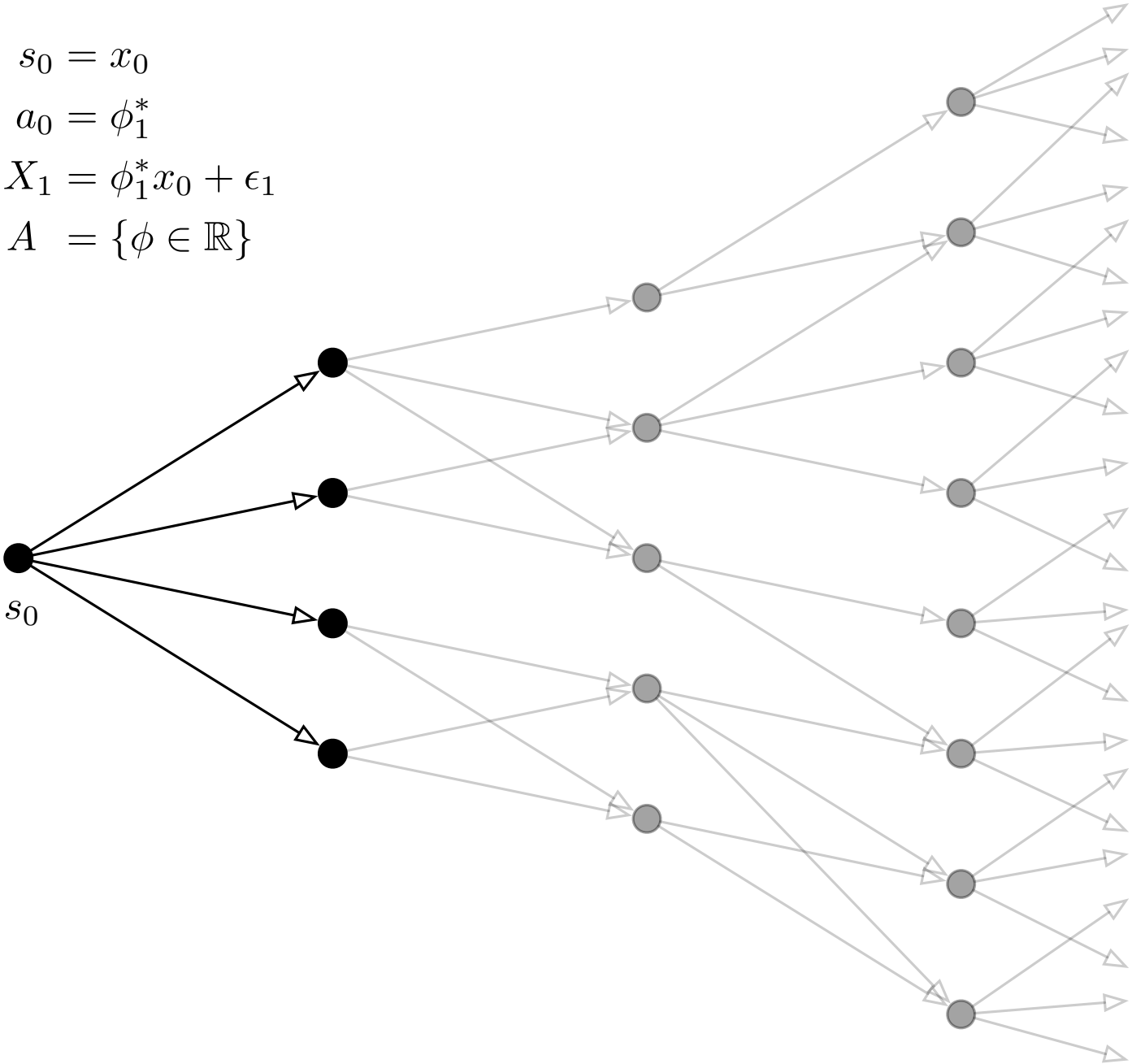
- Dynamic
- Context-Based
- Capable of Integrating Social, Text, and Sentiment Data

In this presentation we develop:

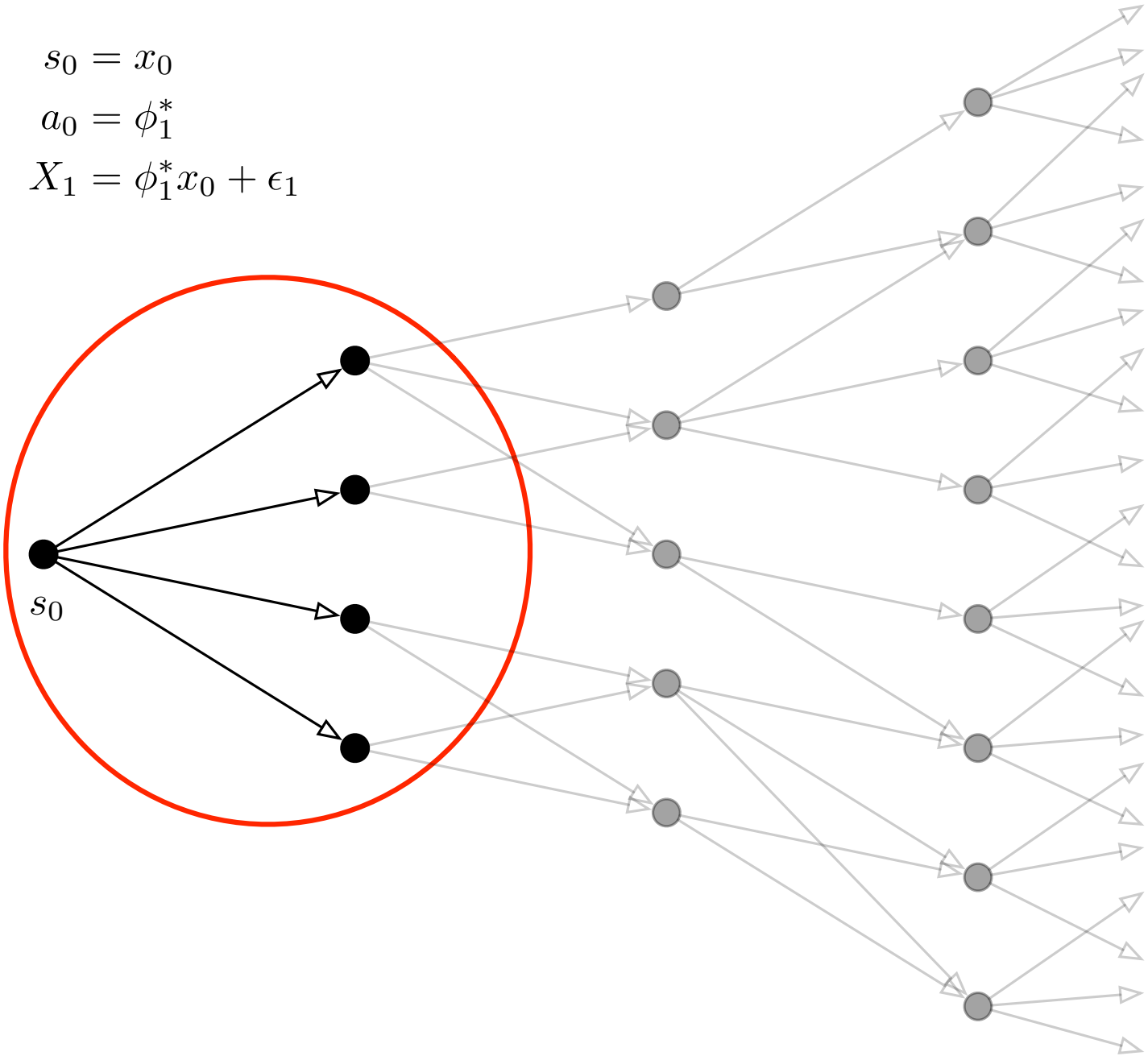
- Stochastic dynamic programming model for time series forecast
  - Rely on an "external forecast" for future values
  - External forecast allows to incorporate alternative data sources

Time Series Fitting Process

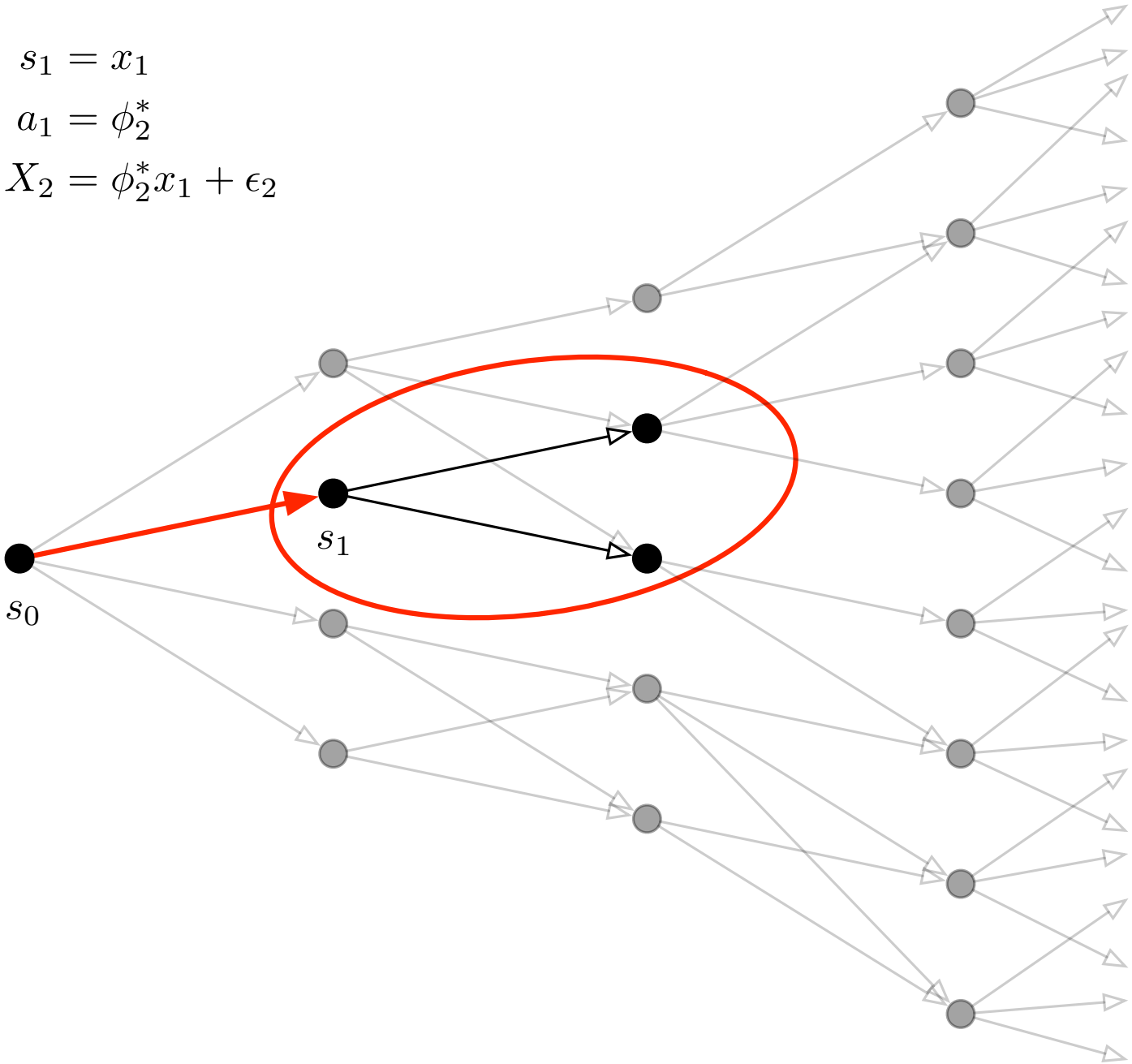
$$s_0 = x_0$$
$$a_0 = \phi_1^*$$
$$X_1 = \phi_1^* x_0 + \epsilon_1$$
$$A = \{\phi \in \mathbb{R}\}$$



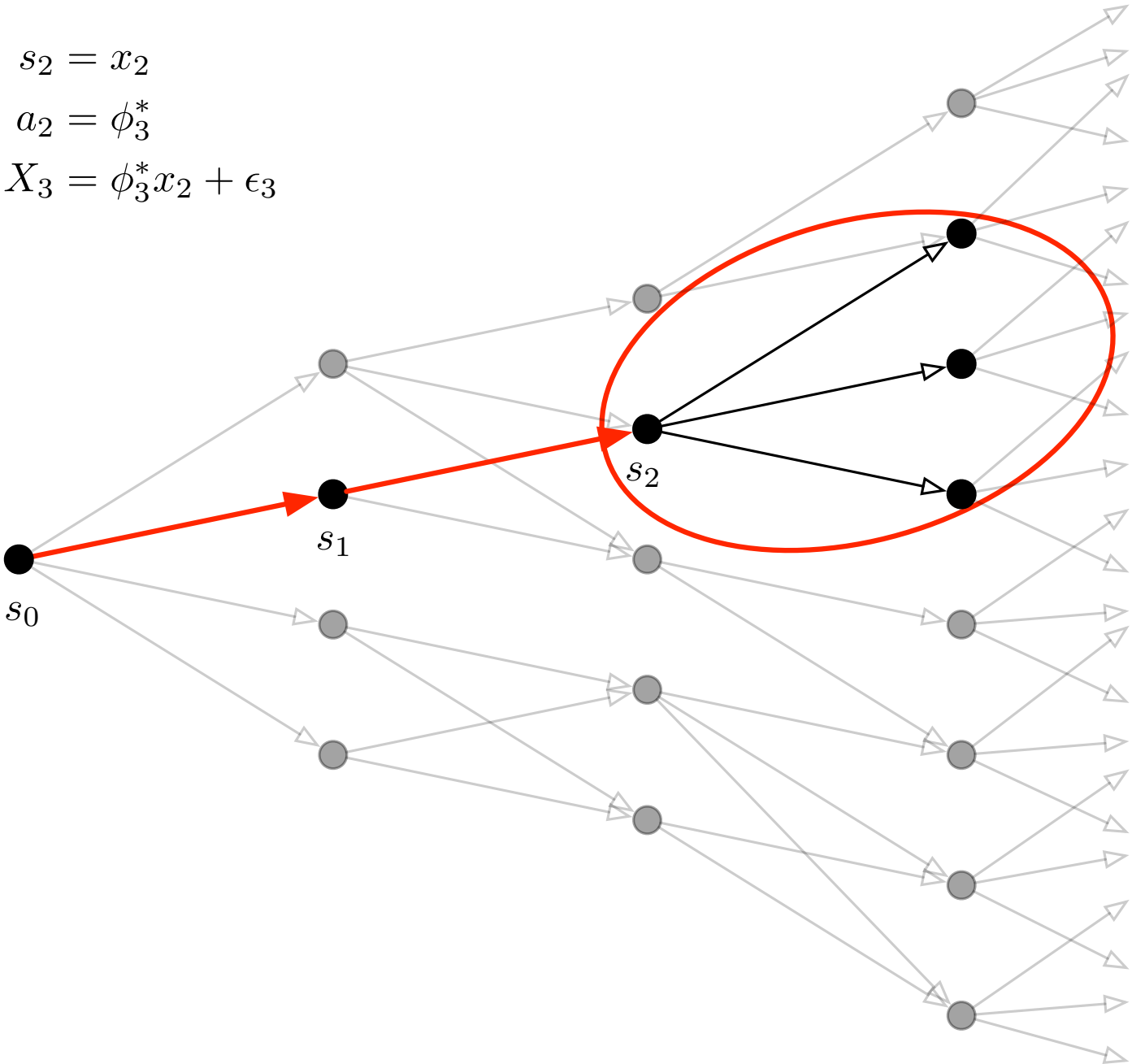
$$s_0 = x_0$$
$$a_0 = \phi_1^*$$
$$X_1 = \phi_1^* x_0 + \epsilon_1$$



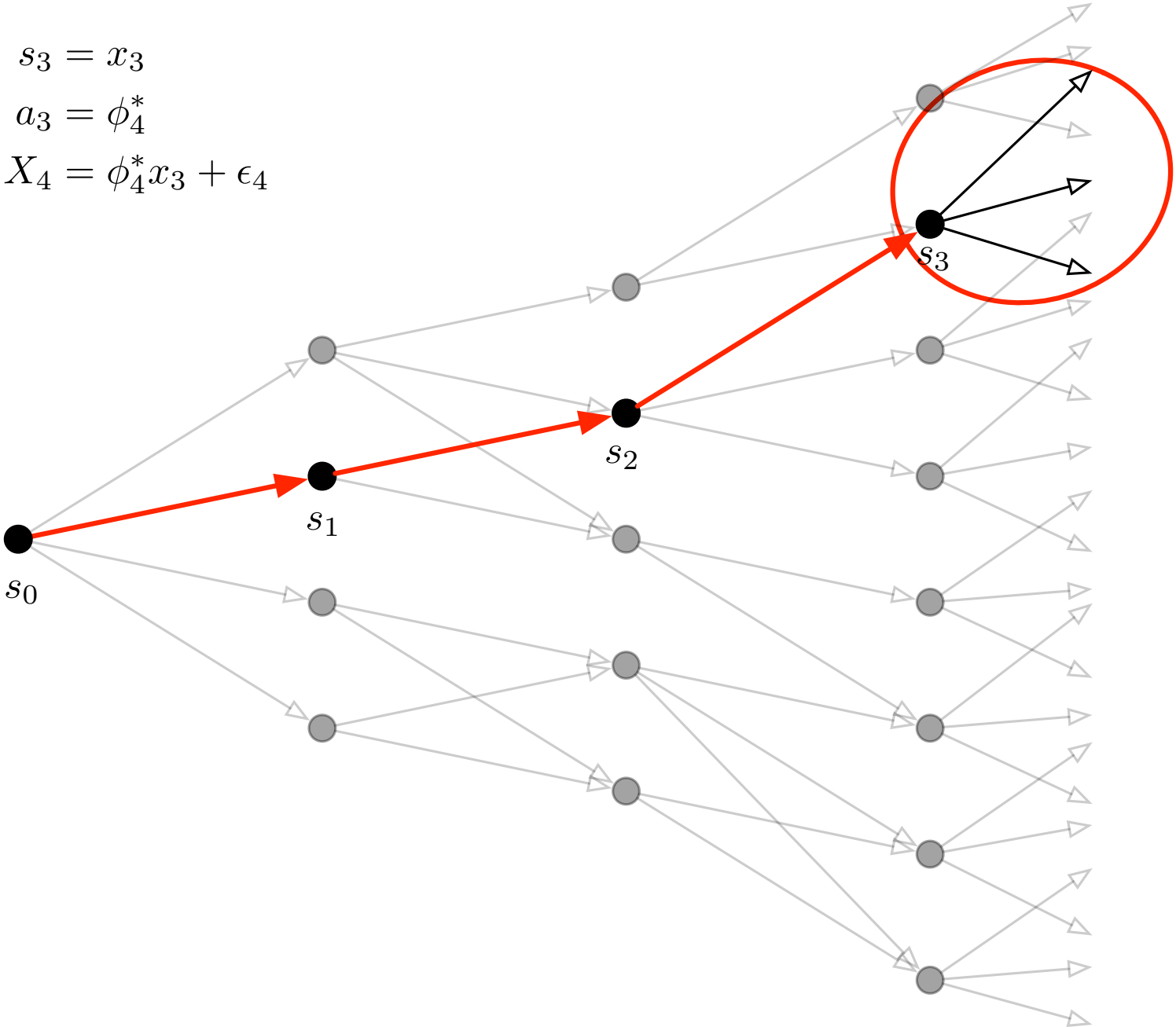
$$s_1 = x_1$$
$$a_1 = \phi_2^*$$
$$X_2 = \phi_2^* x_1 + \epsilon_2$$



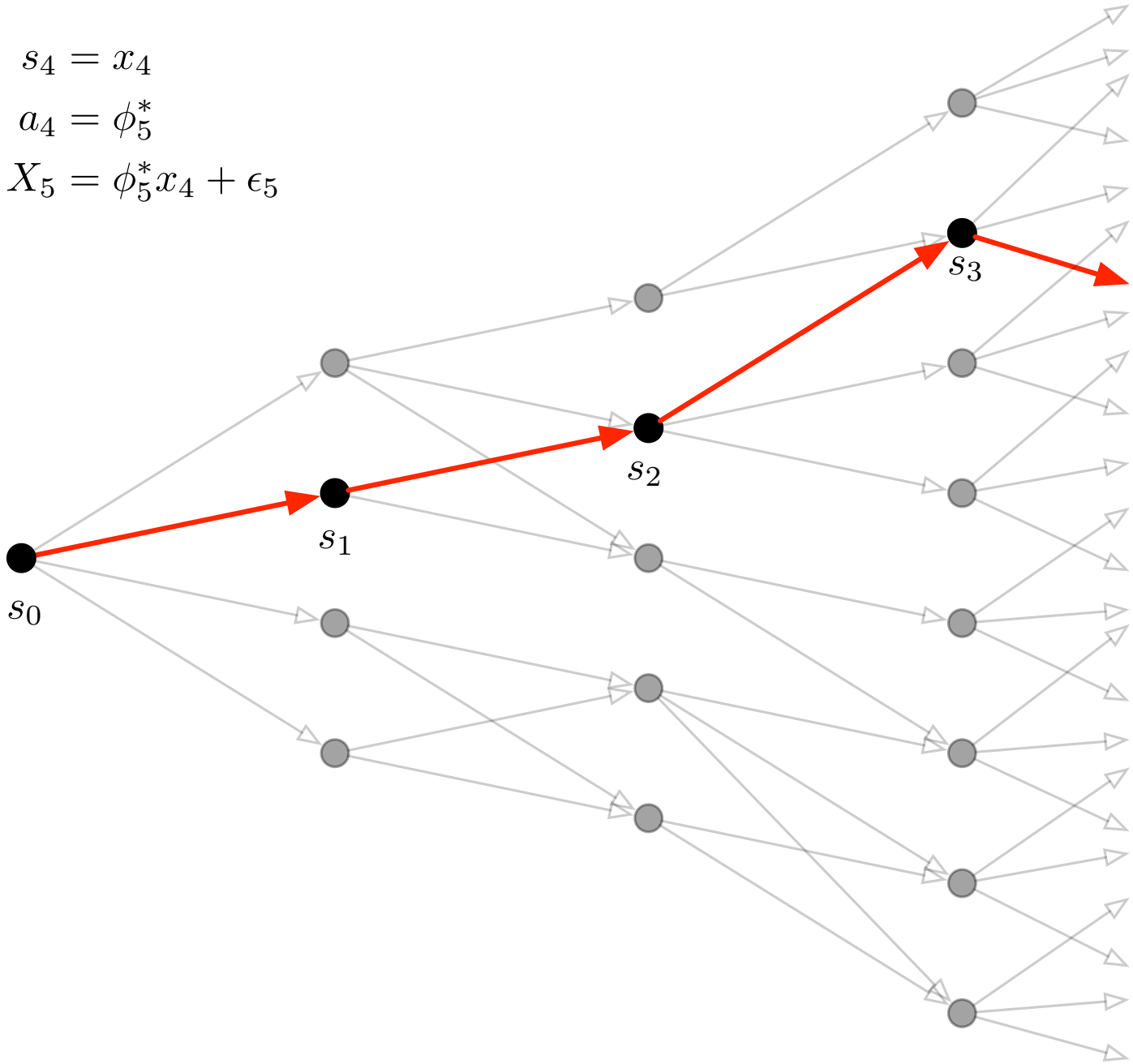
$$s_2 = x_2$$
$$a_2 = \phi_3^*$$
$$X_3 = \phi_3^* x_2 + \epsilon_3$$



$$s_3 = x_3$$
$$a_3 = \phi_4^*$$
$$X_4 = \phi_4^* x_3 + \epsilon_4$$



$$s_4 = x_4$$
$$a_4 = \phi_5^*$$
$$X_5 = \phi_5^* x_4 + \epsilon_5$$



Main Problem:

$$\min_{\pi \in \Pi} \mathbb{E}\left[\sum_{t=1}^{T} c_t(s_t, a_t)\right],$$

where $a_t = \pi_t(x_1, \ldots, x_t)$ is an admissible fitting policy.

- The time series model is parametrized by $\Theta \subseteq \mathbb{R}^d$
- $A(s) = \Theta$ for all states s
- $c_t(s, a)$ is the result of a goodness of fit test for the observations $s = (x_0, \ldots, x_t)$ and model selection $a = \theta$
- Solution via Bellman's optimality equations

Conditions to guarantee optimality:

- The set of actions $A(s)$ is compact
- The cost functions $c_t(s, \cdot)$ are lower semicontinuous
- For every measurable selection $a_t(\cdot) \in A_t(\cdot)$, the functions $s \mapsto c_t(s, a_t(s))$ and $c_T(\cdot)$ are elements of $\mathcal{L}_1(\mathcal{S}, \mathcal{B}_\mathcal{S}, P_0)$
- The DP stochastic kernel function $Q_t(s, \cdot)$ is continuous

Value Functions: Value functions $v_t : \mathcal{S} \to \mathbb{R}$, $t = 1, \ldots, T$, given recursively by:

$$v_T(s) = c_T(s)$$
$$v_t(s) = \min_{a \in A_t(s)} \{c_t(s, a) + \mathbb{E}\left[v_{t+1} \,|\, s, a\right]\},$$

for all $s \in \mathcal{S}$ and $t = T - 1, \ldots, 0$.

Bellman's Optimality Equations:

Then an optimal Markov policy $\pi^* = \{\pi_0^*, \ldots, \pi_{T-1}^*\}$ exists and satisfies the equations:

$$\pi_t^*(s) \in \arg\min_{a \in A_t(s)} \{c_t(s, a) + \mathbb{E}\left[v_{t+1} \,|\, s, a\right]\}, \; s \in \mathcal{S}, \; t = T - 1, \ldots, 0.$$

Conversely, any measurable solution of these is an optimal Markov policy π^*.

Notice that:

- Our choice of model does not affect future observations and cost.
- So, $\mathbb{E}[v \mid s, a] = \mathbb{E}[v \mid s, a']$, for any $(s, a), (s, a') \in \text{graph}(\mathcal{A})$.
- Therefore we can rewrite the optimal policy as:

$$\pi_t^*(s) \in \underset{a \in A_t(s)}{\arg\min}\{c_t(s, a)\}, \ s \in \mathcal{S}, \ t = T - 1, \ldots, 0.$$

Notice that:

- Our choice of model does not affect future observations and cost.
- So, $\mathbb{E}[v \mid s, a] = \mathbb{E}[v \mid s, a']$, for any $(s, a), (s, a') \in \text{graph}(\mathcal{A})$.
- Therefore we can rewrite the optimal policy as:

$$\pi_t^*(s) \in \underset{a \in A_t(s)}{\arg\min}\{c_t(s, a)\},\ s \in \mathcal{S},\ t = T - 1, \ldots, 0.$$

- **The optimal policy π^* is purely myopic**

Q: How to break with the myopic policy?

A: Introduce a new Markov model

New Markov Model:

- Given stochastic process $\{X_t \mid t = 0, \ldots, T\}$, s.t. $X_0 = \{\phi_0\}$
- Time series model parameterized by $\Theta \subseteq \mathbb{R}^d$
- State space:

$$\mathcal{S}_t = \left\{ (x_t, h_{t-1}, \theta_{t-1}) \;\middle|\; \begin{array}{l} x_t \text{ observation from } X_t, \\ h_t = x_0, \ldots, x_{t-1} \text{ sample sequence }, \\ \theta_{t-1} = (\phi_1, \ldots, \phi_p) \in \Theta \end{array} \right\}$$

- Action space: $A(s) = \Theta$ for all states $s \in \mathcal{S}_t$

Cost function:

$$c_t(s, \theta_t) = \gamma(s_t, \theta_t) + r\,\delta(s_t, \theta_{t-1}, \theta_t)$$

- γ: Goodness of fit test
- δ: Penalty on changes from previous model selection
- $r \geq 0$: Scaling factor used to balance fit and penalty

Example:

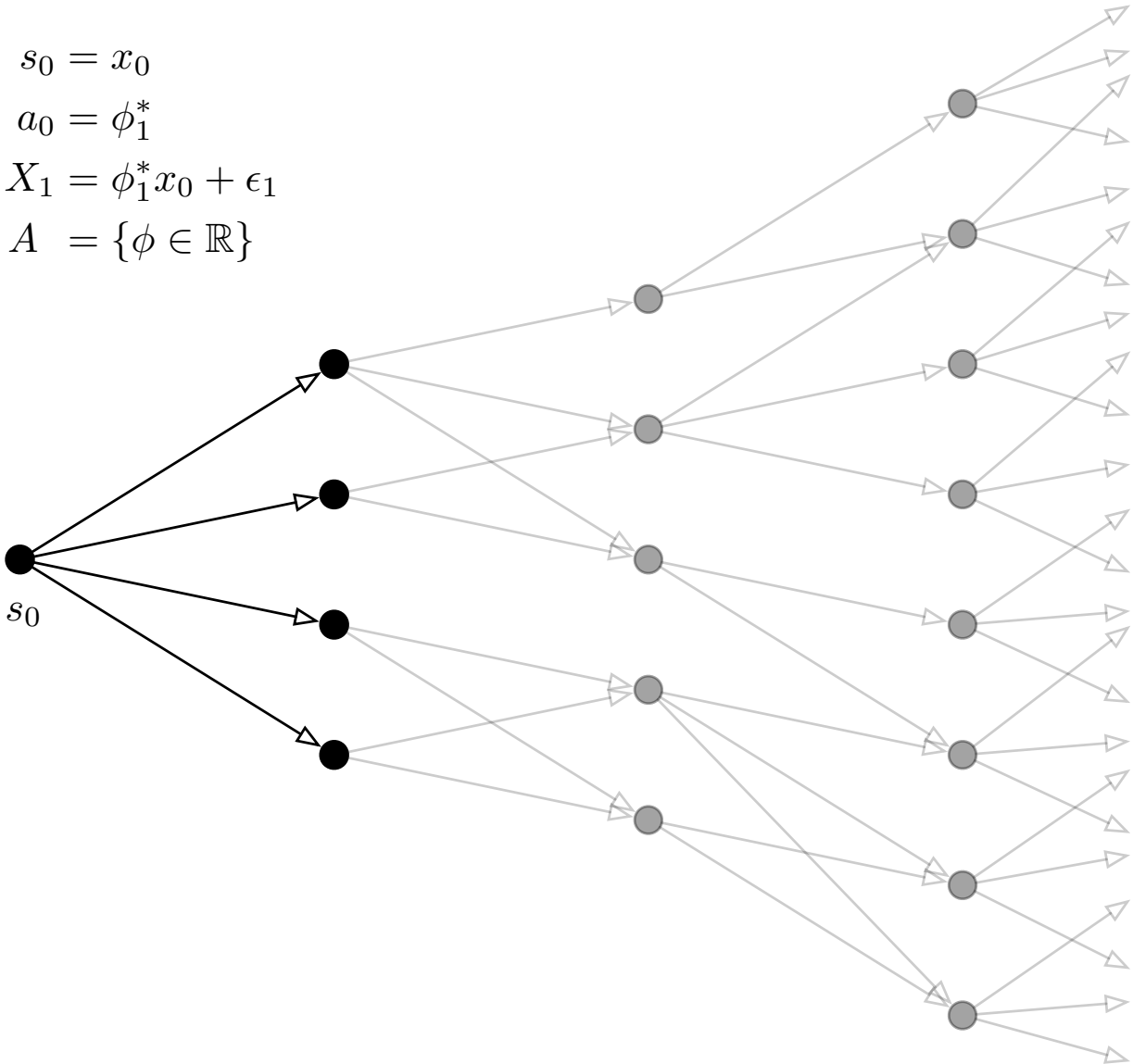
$$\chi^2\left(\theta_t \mid h_{t-1}, x_t\right) + r\left(1 - \exp\left\{-\lambda \left|\mathbb{E}\left[\mathrm{P}_{\theta_t} \mid h_{t-1}, x_t\right] - \mathbb{E}\left[\mathrm{P}_{\theta_{t-1}} \mid h_{t-1}, x_t\right]\right|\right\}\right),$$

where $r, \lambda \geq 0$.

- $\gamma(s_t, \theta_t) := \chi^2\left(\theta_t \mid h_{t-1}, x_t\right)$
- $\delta(s_t, \theta_{t-1}, \theta_t) := 1 - \exp\left\{-\lambda \left|\mathbb{E}\left[\mathrm{P}_{\theta_t} \mid h_{t-1}, x_t\right] - \mathbb{E}\left[\mathrm{P}_{\theta_{t-1}} \mid h_{t-1}, x_t\right]\right|\right\}$

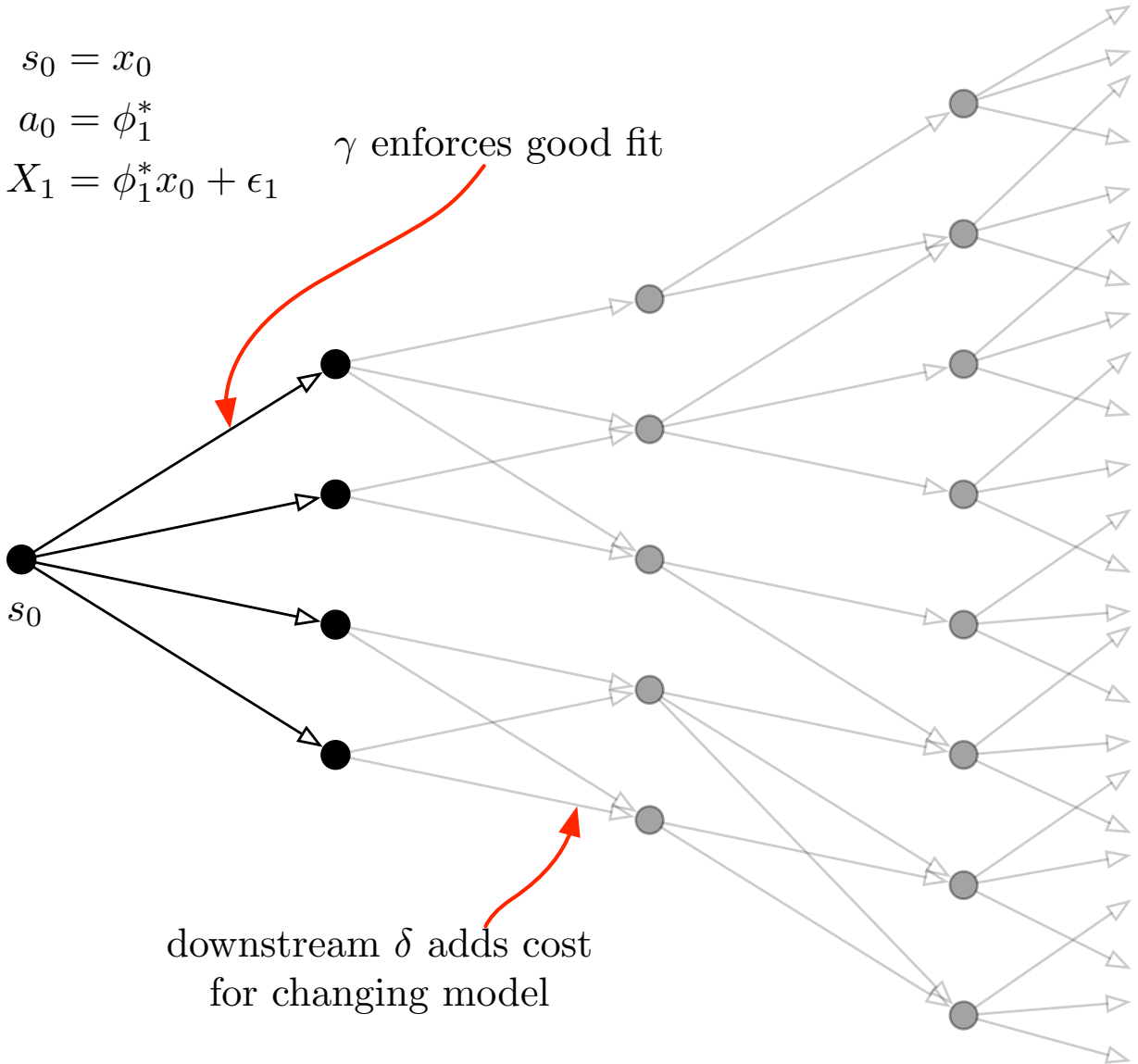
Dynamic Optimization Model:

$$
\begin{aligned}
s_0 &= x_0 \\
a_0 &= \phi_1^* \\
X_1 &= \phi_1^* x_0 + \epsilon_1 \\
A &= \{\phi \in \mathbb{R}\}
\end{aligned}
$$

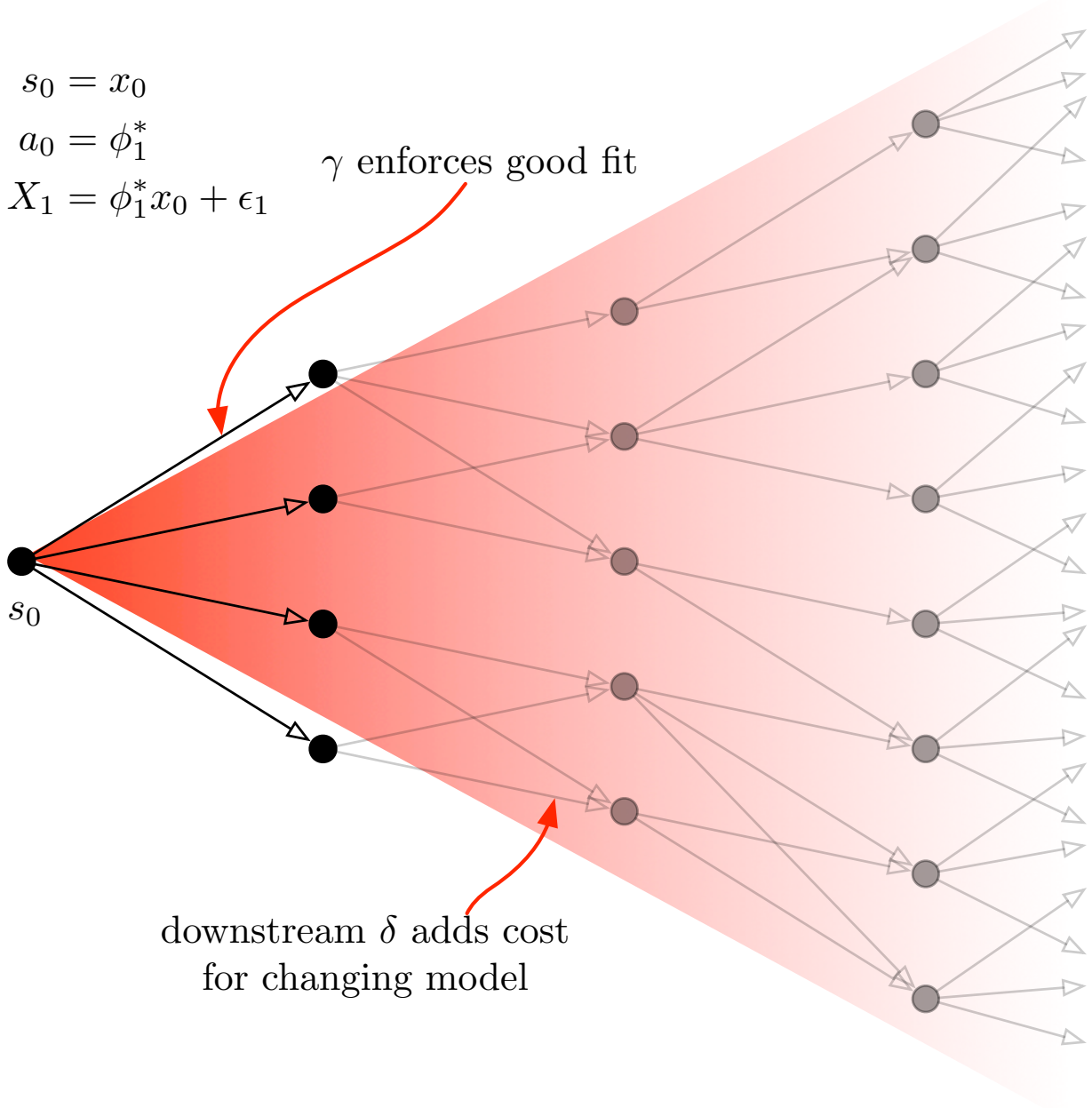


Dynamic Optimization Model:

$$s_0 = x_0$$
$$a_0 = \phi_1^*$$
$$X_1 = \phi_1^* x_0 + \epsilon_1$$

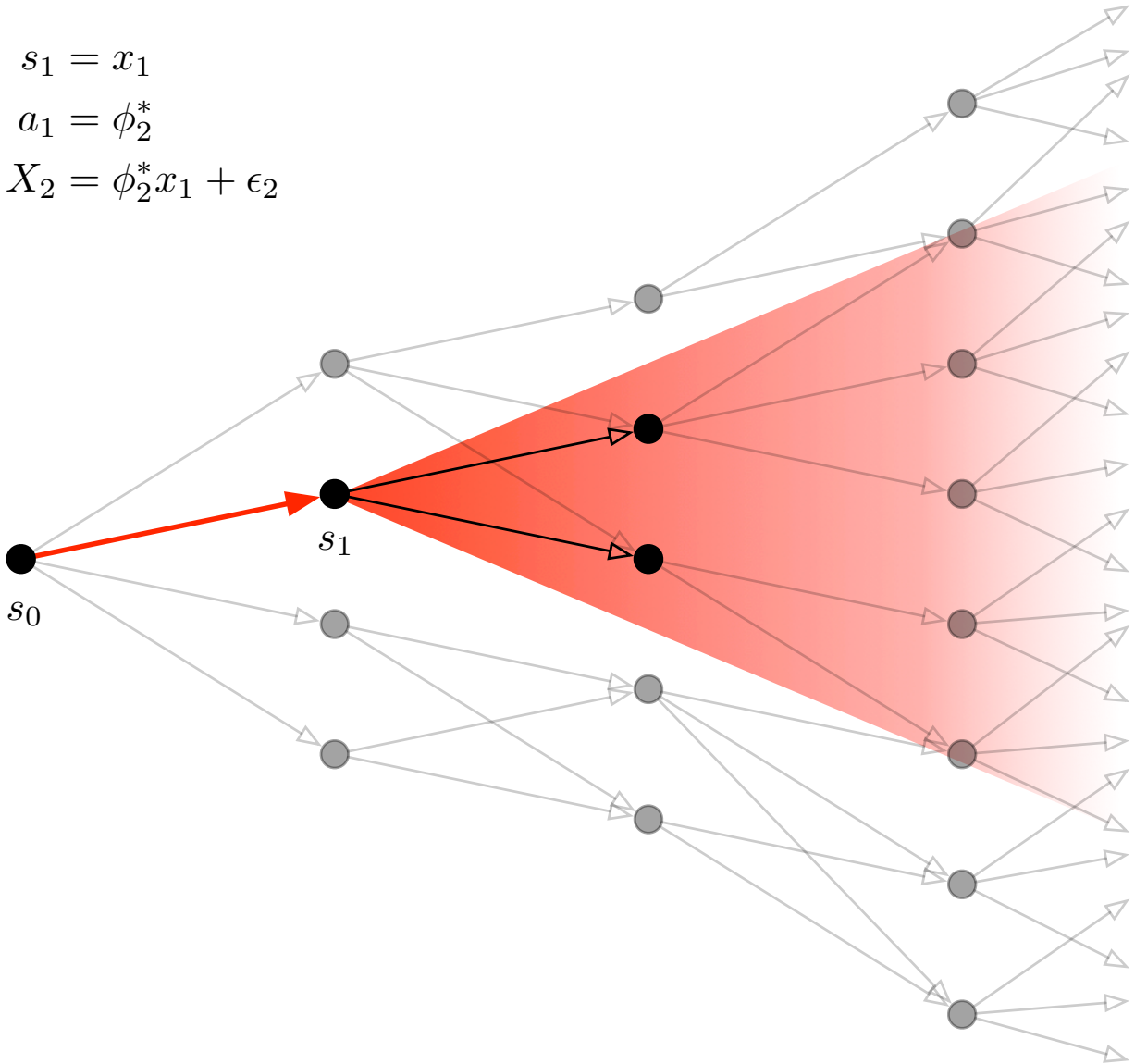


Dynamic Optimization Model:



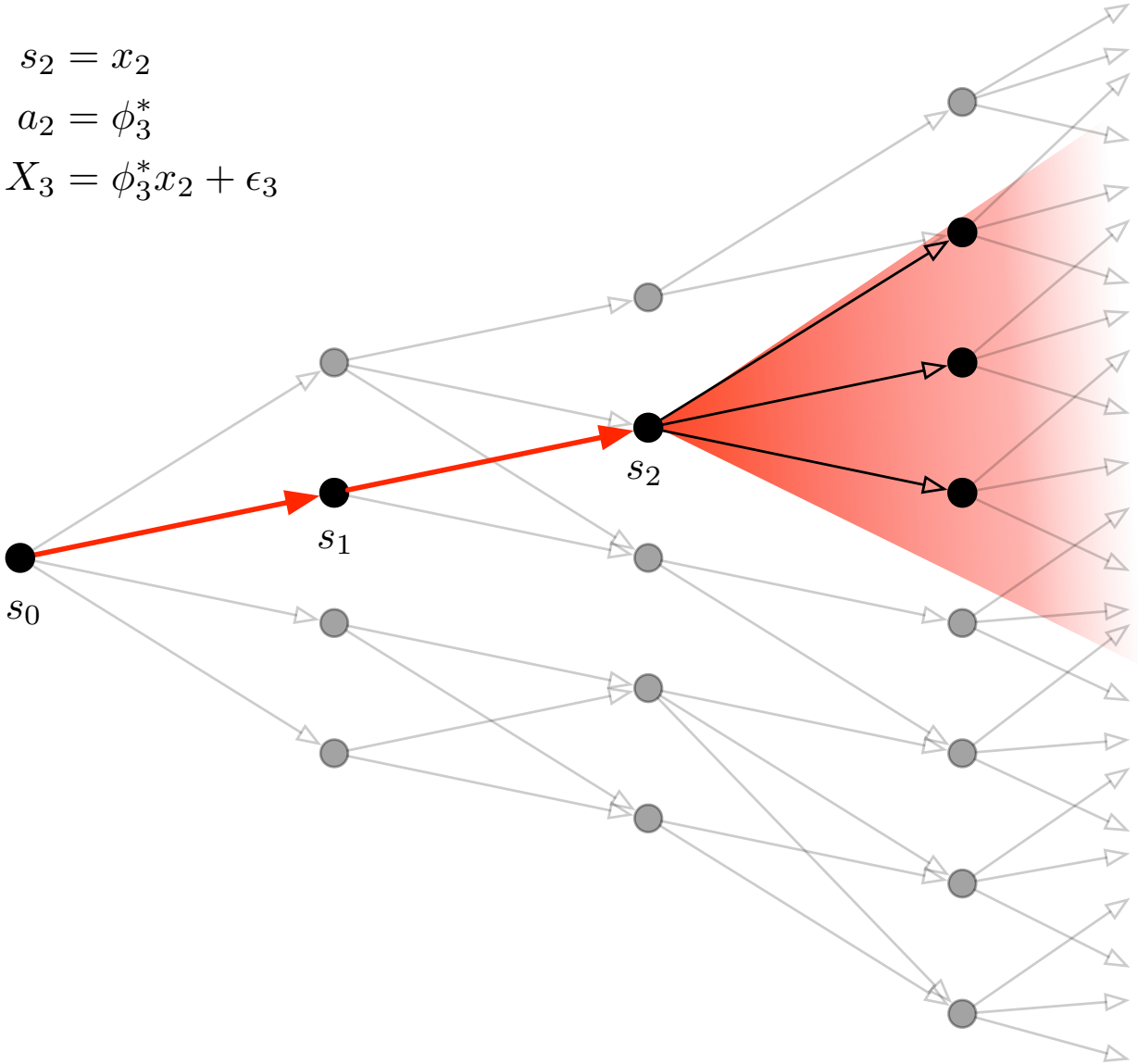
Dynamic Optimization Model:

$$s_1 = x_1$$
$$a_1 = \phi_2^*$$
$$X_2 = \phi_2^* x_1 + \epsilon_2$$



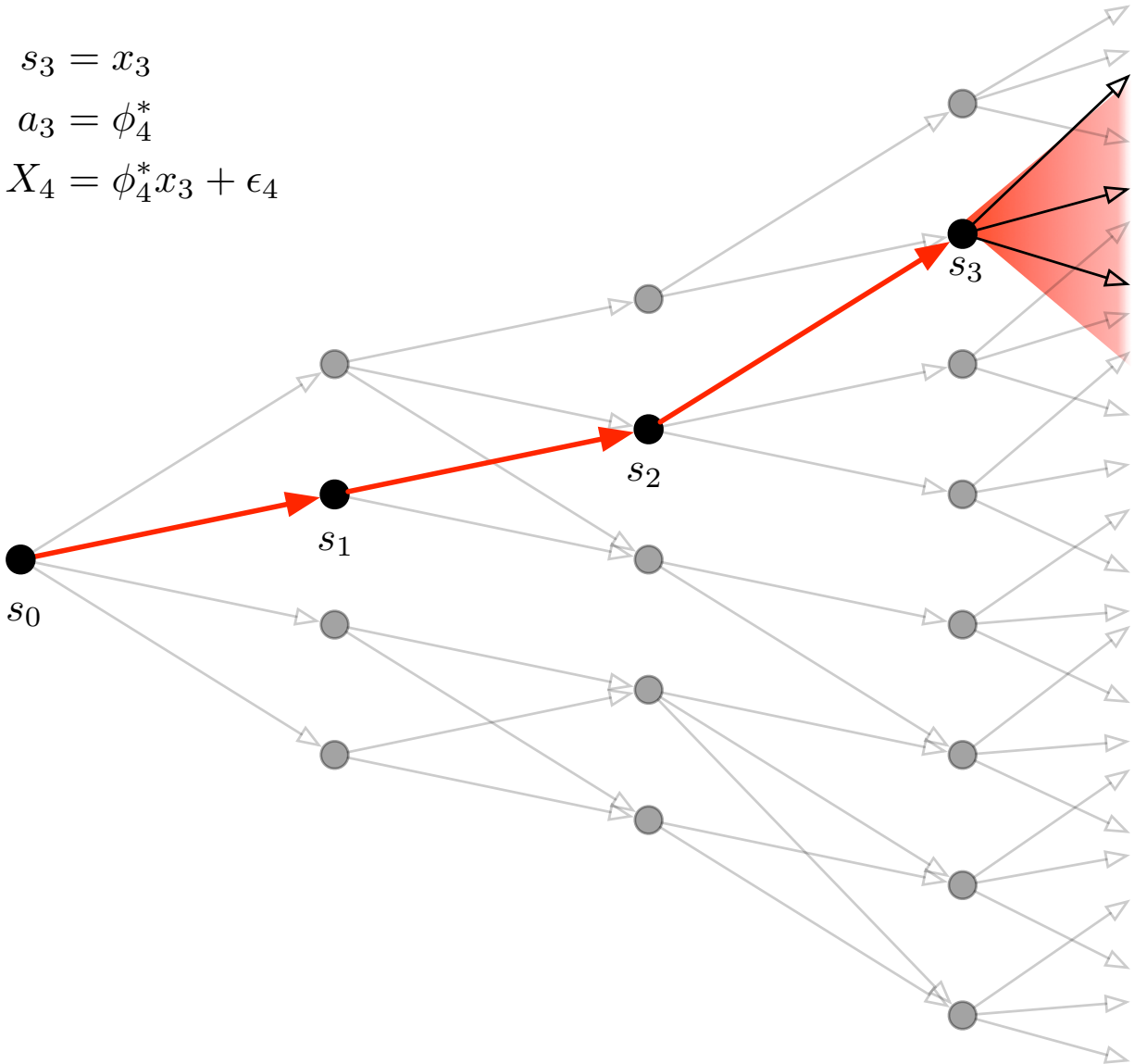
Dynamic Optimization Model:

$$s_2 = x_2$$
$$a_2 = \phi_3^*$$
$$X_3 = \phi_3^* x_2 + \epsilon_3$$



Dynamic Optimization Model:

$$s_3 = x_3$$
$$a_3 = \phi_4^*$$
$$X_4 = \phi_4^* x_3 + \epsilon_4$$



Introducing Exogenous Information Via Lookahead Methods

Lookahead Methods:

- At time t we have access to forecast random variables:

$$\widehat{X}^t_{t+1}, \ldots, \widehat{X}^t_{t+h}$$

- We assume these are discrete approximations to $X_{t+1}, \ldots, X_{t+h}$
- Each forecast induces a discrete probability measure on $\mathcal{S}$

We expect the forecasts to be finite random variables with few atoms

This has the effect of simplifying calculations, but at the expense of rough approximations.

Basic lookahead method:

Step 0 Initialization:

Step 0a Initialize $\overline{v}_t(s)$ for all time periods t and all $s \in \mathcal{S}$.

Step 0b Choose an initial state s_0^1.

Step 1 For $t = 0, \ldots, T$ do:

Step 1a Update the state variable: observe a value x_t of the stochastic process, let s_t be obtained from x_t and model selection at $t-1$, and get forecasts $\widehat{X}_{t+1}, \ldots, \widehat{X}_{t+h}$.

Step 1b Solve

$$\hat{v}_t = \min_{a_t \in A_t(s_t)} \left\{ c_t(s_t, a_t) + \mathbb{E}\left[\bar{v}_{t+1} \mid s_t, a_t \right] \right\},$$

by solving a stochastic optimization problem. Let a_t be the obtained optimal solution to the minimization problem.

Step 1c Update the value function approximation $\bar{v}_t$:

$$\bar{v}_t(s) = \begin{cases} \hat{v}_t, & \text{if } s = s_t, \\ \bar{v}_t(s), & \text{otherwise.} \end{cases}$$

Step 2 Return the value functions $\left\{ \bar{v}_t \mid t = 0, 1, \ldots, T \right\}$.

Basic Monte Carlo Algorithm:

Step 0 Initialization:

Step 0a Initialize $\overline{v}_t^0(s)$ for all time periods t and all $s \in \mathcal{S}$.

Step 0b Choose an initial state s_0^1.

Step 0c Solve the problem the previous method and denote the resulting value function approximations $\bar{v}_t^0$, $t = 0 \ldots, T$.

Step 0d Set $n = 1$.

Step 1 For $t = 0, \ldots, T$ do:

Step 1a Update the state variable: Observe a value x_t^n of the stochastic process, let s_t^n be obtained from x_n^t and model selection at $t-1$, and get forecasts $\widehat{X}_{t+1}^n, \ldots, \widehat{X}_{t+h}^n$.

Step 1b Solve

$$\hat{v}_t^n = \min_{a_t \in A_t(s_t^n)} \left\{ c_t(s_t^n, a_t) + \mathbb{E}\left[\bar{v}_{t+1} \mid s_t^n, a_t \right] \right\},$$

as before. Let a_t^n be the obtained optimal solution to the minimization problem.

Step 1c Update the value function approximation $\bar{v}_t^{n-1}$:

$$\bar{v}_t^n(s) = \begin{cases} (1-\alpha_{n-1})\bar{v}^{n-1}(s) + \alpha_{n-1}\hat{v}_t^n, & \text{if } s = s_t^n, \\ \bar{v}_t^{n-1}(s), & \text{otherwise.} \end{cases}$$

Step 2 Let $n = n+1$. If $n < N$, go to **Step 1**.

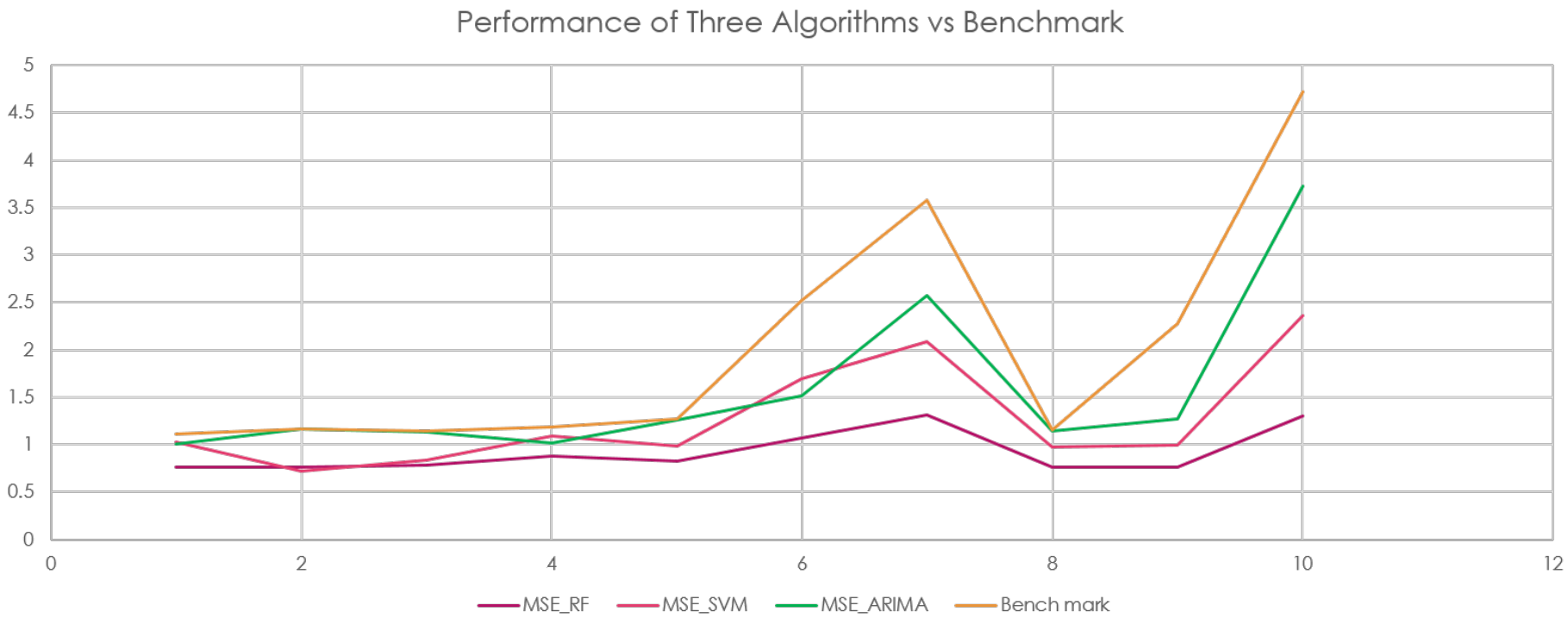
Step 3 Return the value functions $\left\{ \bar{v}_t^N \;\middle|\; t = 0, 1, \ldots, T \right\}$.

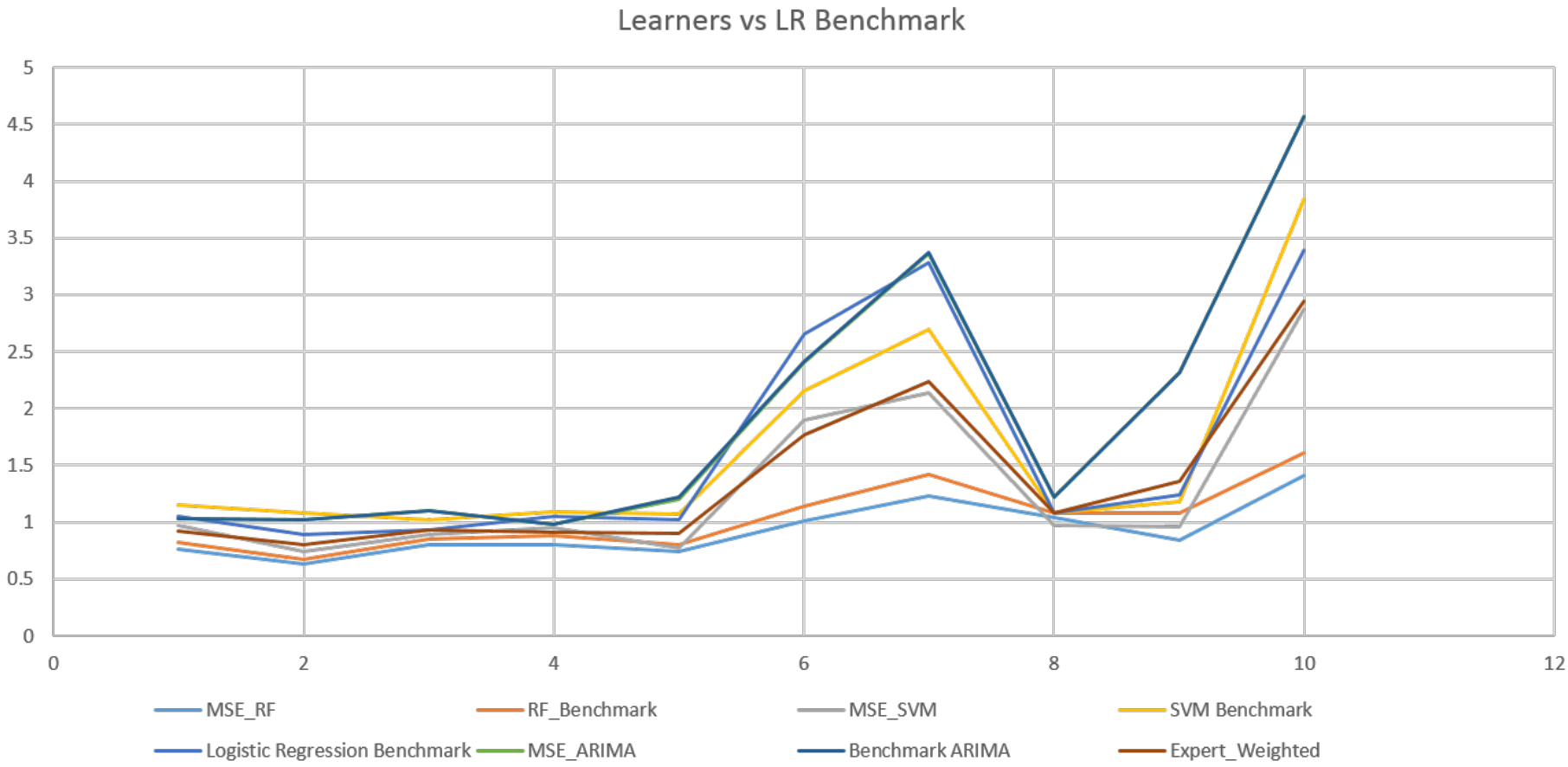
Numerical Results I

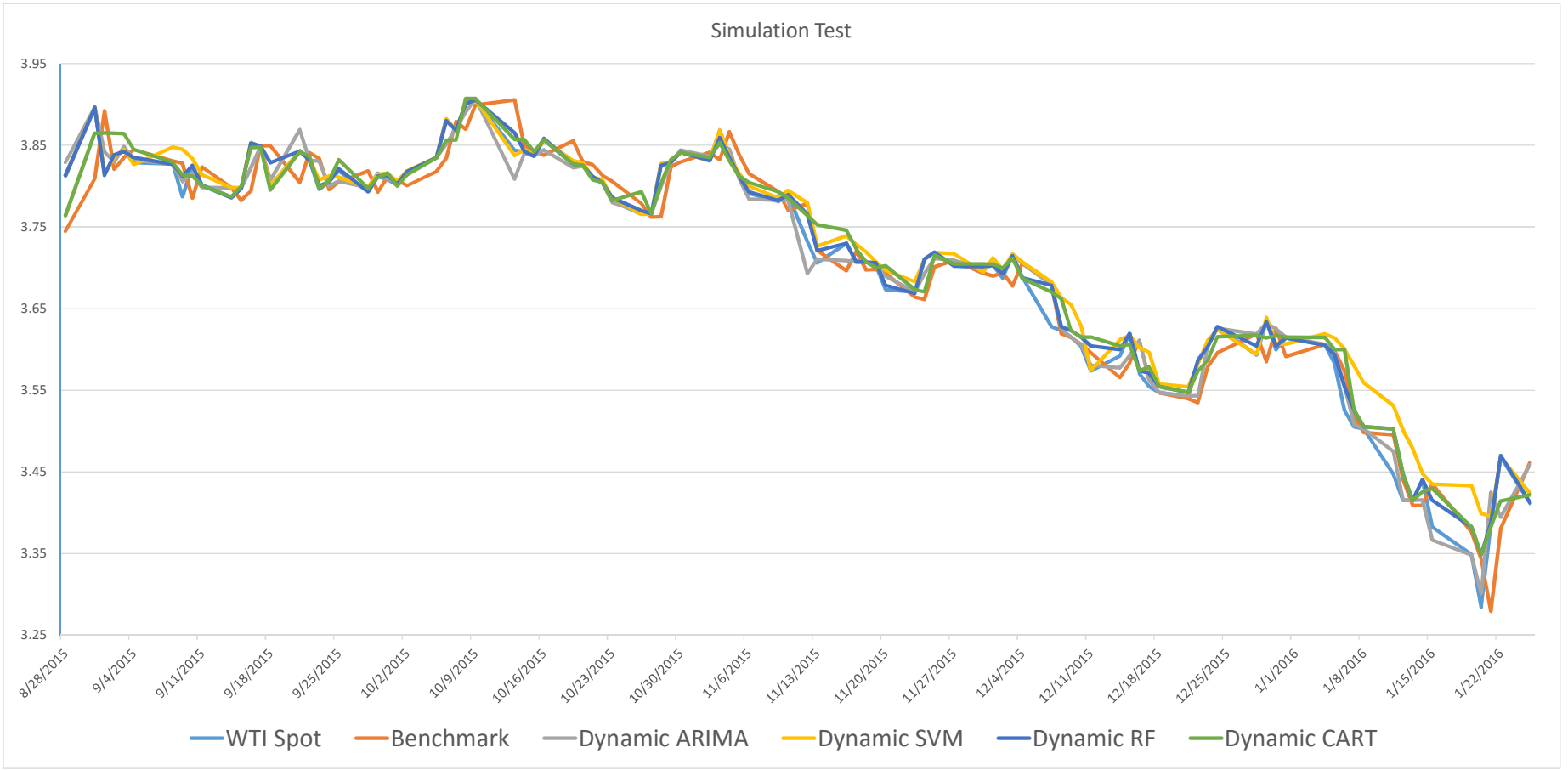
Estimation Techniques I:

- Initial test: Natural Gas Futures Contract 4 - times series (RNGC4), 1999 – 2013.
- Test approximate dynamic programming method (MSE_) over 1 to 10 days forecast window, comparing the calibration of machine learning algorithms used for regression and classification:
  - Support vector machine (SVM)
  - Random forests (RF)
  - ARIMA
  - Logistic regression (LR) – Benchmark
- 2nd test: Crude Oil Spot (WTI), Aug. 2015 – Jan. 2016.
- We made two main tests by using approximations to "futre" WTI Spot on different time windows:
  - Actual values + random white noise with increasing variance
  - Actual values + white noise + increasing bias

Energy Time Series


Performance of Three Algorithms vs Benchmark
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
2
4
6
8
10
12
MSE_RF
MSE_SVM
MSE_ARIMA
Bench mark


Learners vs LR Benchmark
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
2
4
6
8
10
12
MSE_RF
RF_Benchmark
MSE_SVM
SVM Benchmark
Logistic Regression Benchmark
MSE_ARIMA
Benchmark ARIMA
Expert_Weighted


Simulation Test
3.95
3.85
3.75
3.65
3.55
3.45
3.35
3.25
8/28/2015
9/4/2015
9/11/2015
9/18/2015
9/25/2015
10/2/2015
10/9/2015
10/16/2015
10/23/2015
10/30/2015
11/6/2015
11/13/2015
11/20/2015
11/27/2015
12/4/2015
12/11/2015
12/18/2015
12/25/2015
1/1/2016
1/8/2016
1/15/2016
1/22/2016
WTI Spot
Benchmark
Dynamic ARIMA
Dynamic SVM
Dynamic RF
Dynamic CART

	Model RMSE					
	Noise Variance					
Model	0.001	0.01	0.05	0.1	0.5	1
ARIMA	0.0003227	0.000389	0.001943	0.004214	0.008405	0.007514
SVM	0.000776849	0.000818	0.002475	0.008127	0.099774	0.16513
RF	0.000175392	0.00023	0.001843	0.003496	0.007223	0.008425
CART	0.000483638	0.000504	0.002376	0.006527	0.023588	0.024053

Benchmark	0.001116097

Unbiased Random Noise External Forecast Test

ARIMA Baseline	490.6124
RF Baseline	502.5576
CART Baseline	967.2576

	0.001	0.01	0.03	0.05	0.1	0.5	1	2
ARIMA	384.0882	380.7184		379.9081	373.224	380.9279	386.4873	473.5217
RF	28.16006	30.8183	36.77366	45.00305	56.97973	195.9457	282.2603	389.0374
CART	6.680402	29.53345	46.15347	64.68325	147.4693	497.2315	725.8215	863.787

ARIMA Baseline	627.327
RF Baseline	804.8075
CART Baseline	1588.515

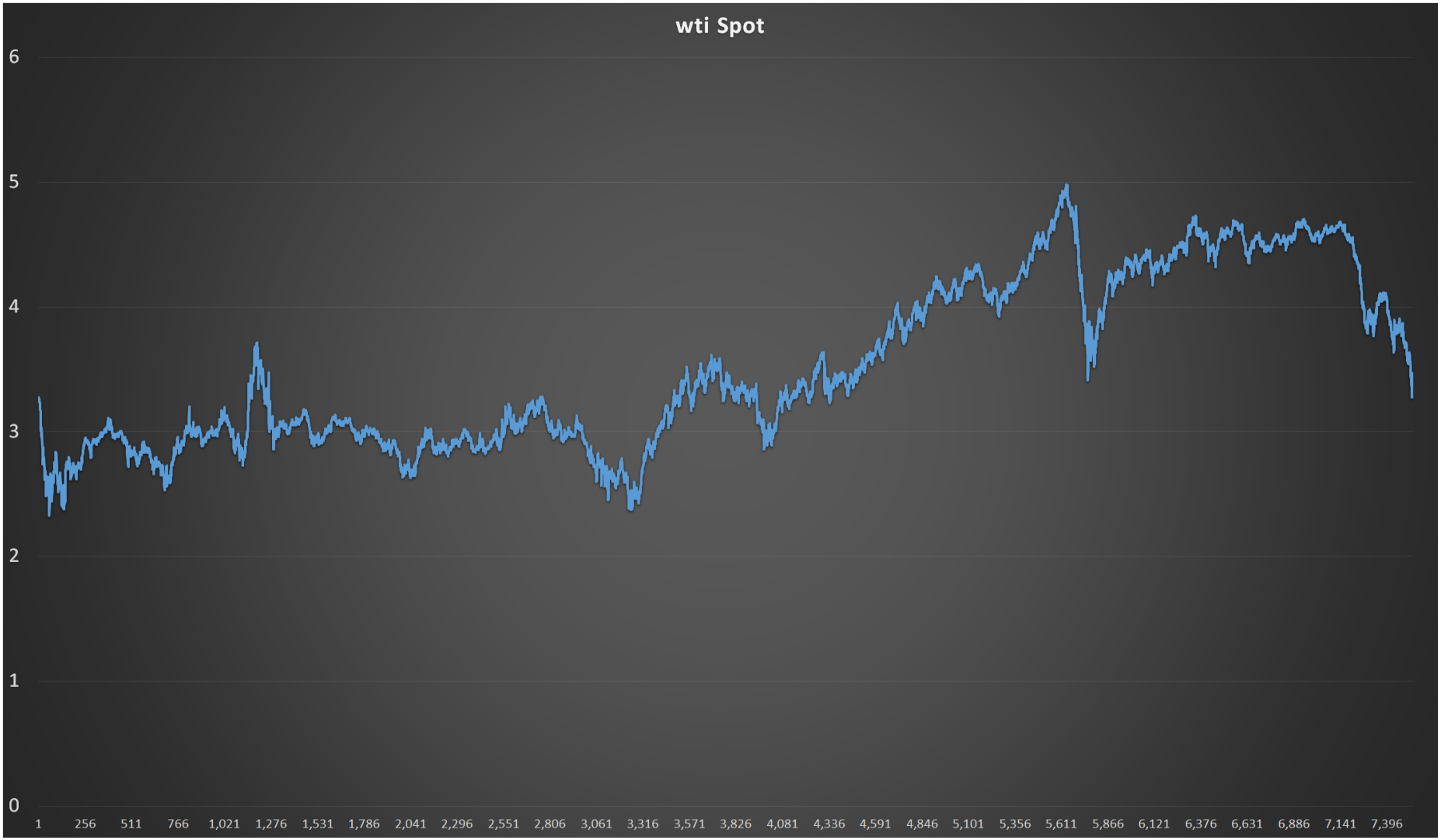
	0.001	0.01	0.03	0.05	0.1	0.5	1	2
ARIMA	586.0408	586.1985		586.4885	587.7807	590.063	624.0278	705.4027
RF	71.76113	87.90271	117.1728	146.3925	199.0221	424.4266	528.3886	618.359
CART	12.98331	65.70607	135.9031	240.7952	369.7038	913.2118	1188.266	1422.569

Biased Random Noise External Simulation: 10 and 50 Days Windows

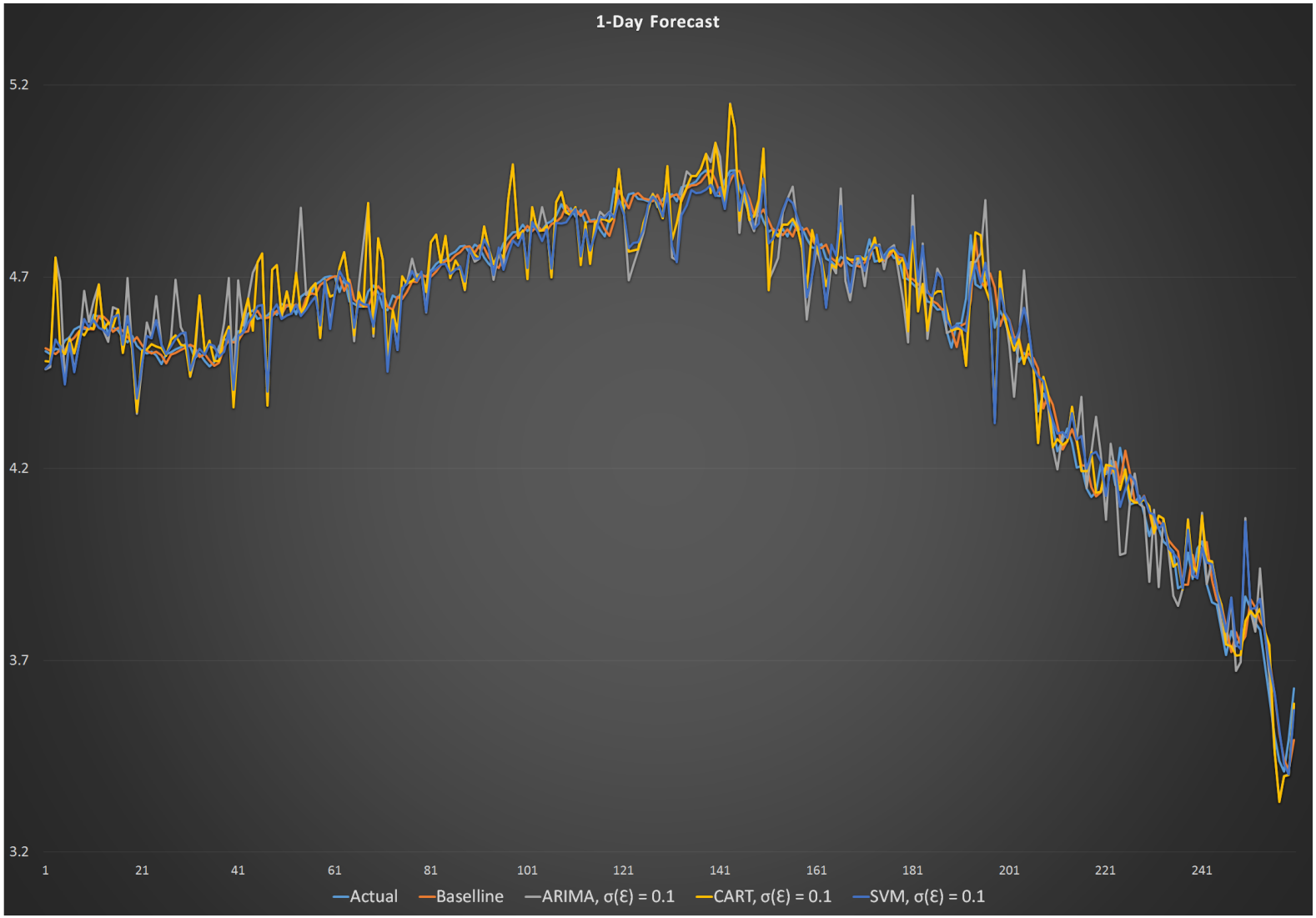
Numerical Results 2

Estimation Techniques II:

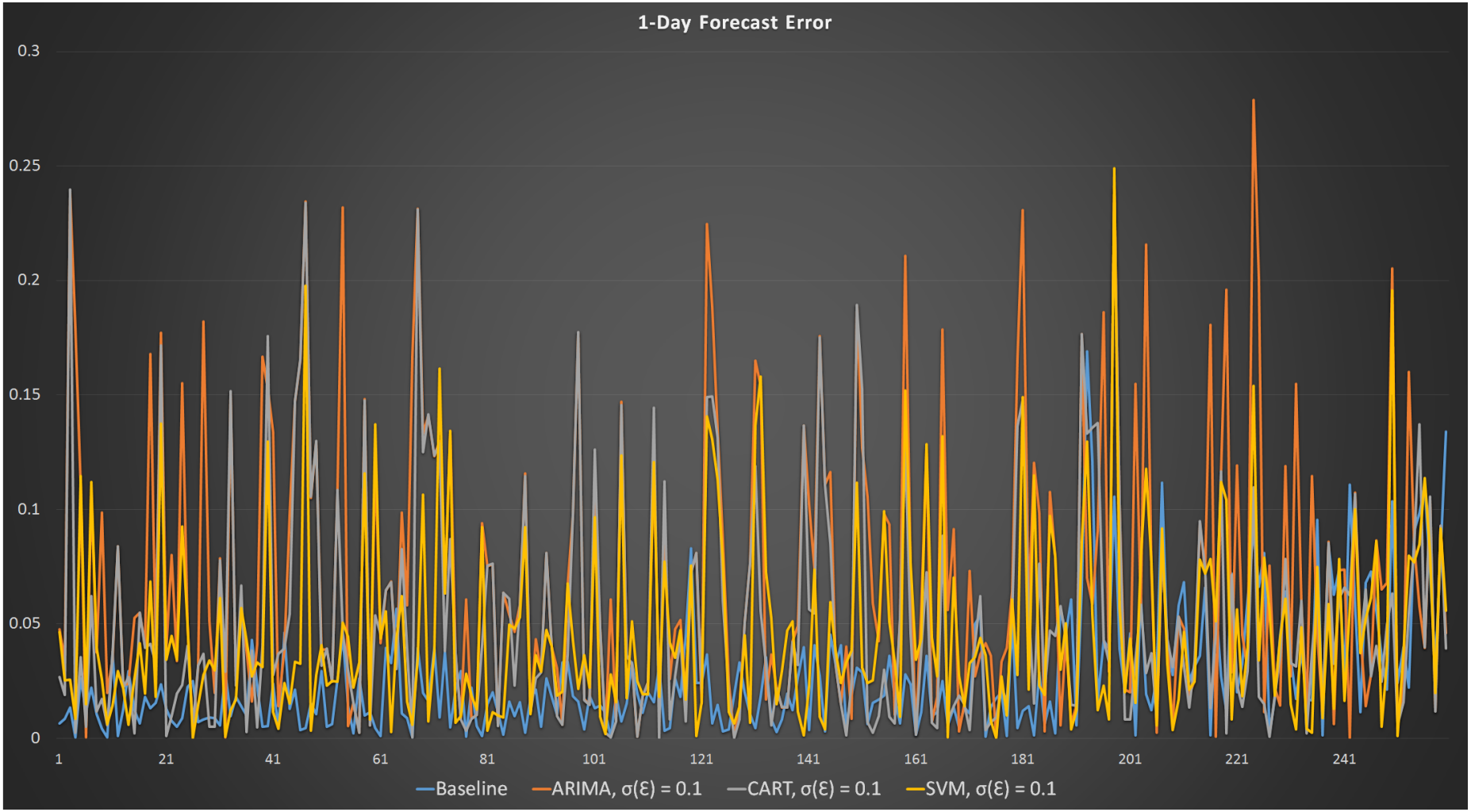
- 3rd test: Crude Oil Spot (WTI). Test approximate dynamic programming method over 1, 5, and 5 days forecast window, using different ML dynamic techniques with actual + white noise as "future" data:
  - Support vector machine (SVM)
  - Classification and regression trees (CART)
  - ARIMA
  - Logistic regression (LR) – Benchmark
- 4th test:SVM futures WTI and SVM sparse "future" data on WTI spot.


wti Spot
6
5
4
3
2
1
0
1
256
511
766
1,021
1,276
1,531
1,786
2,041
2,296
2,551
2,806
3,061
3,316
3,571
3,826
4,081
4,336
4,591
4,846
5,101
5,356
5,611
5,866
6,121
6,376
6,631
6,886
7,141
7,396

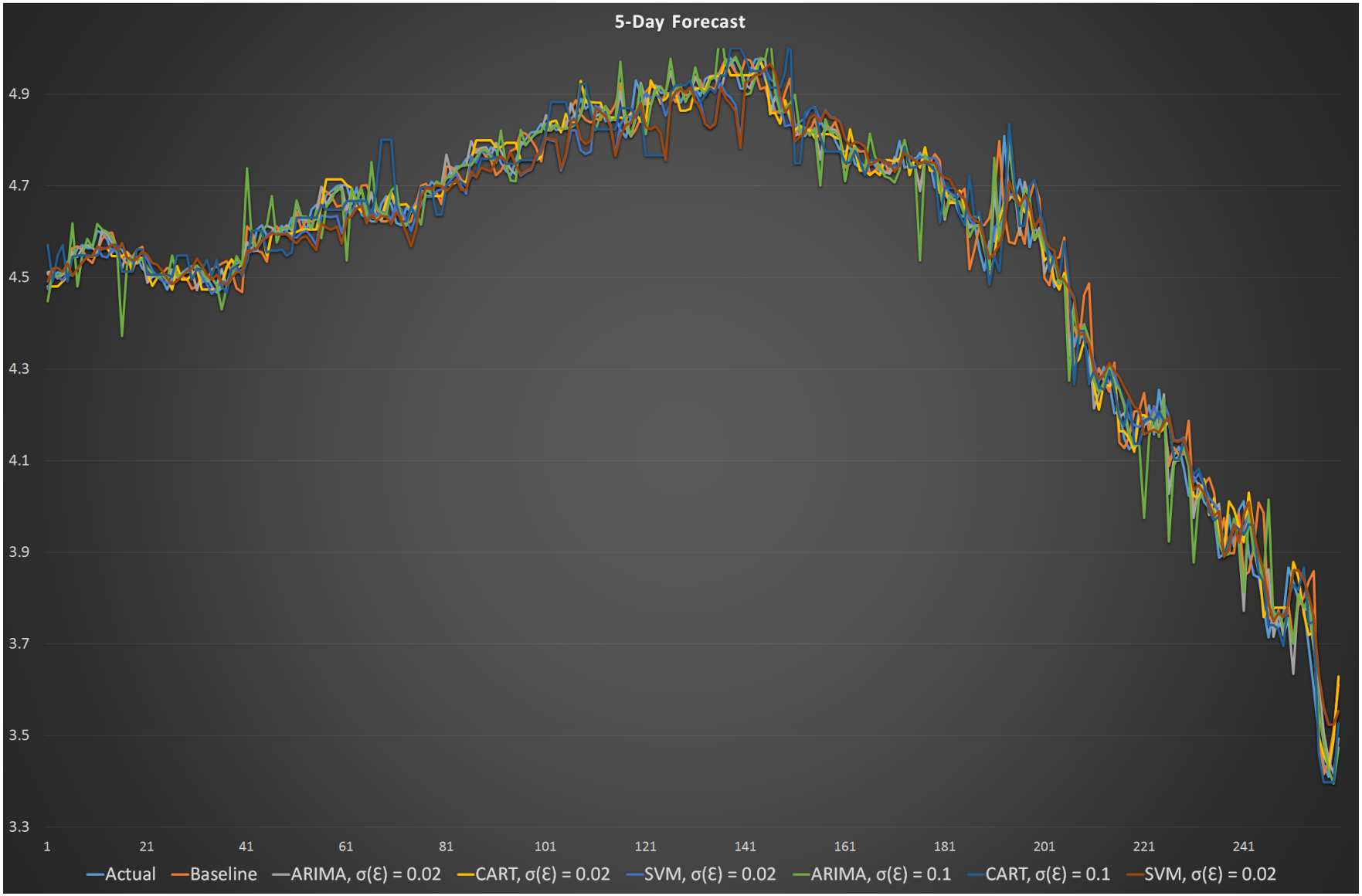
WTI Spot



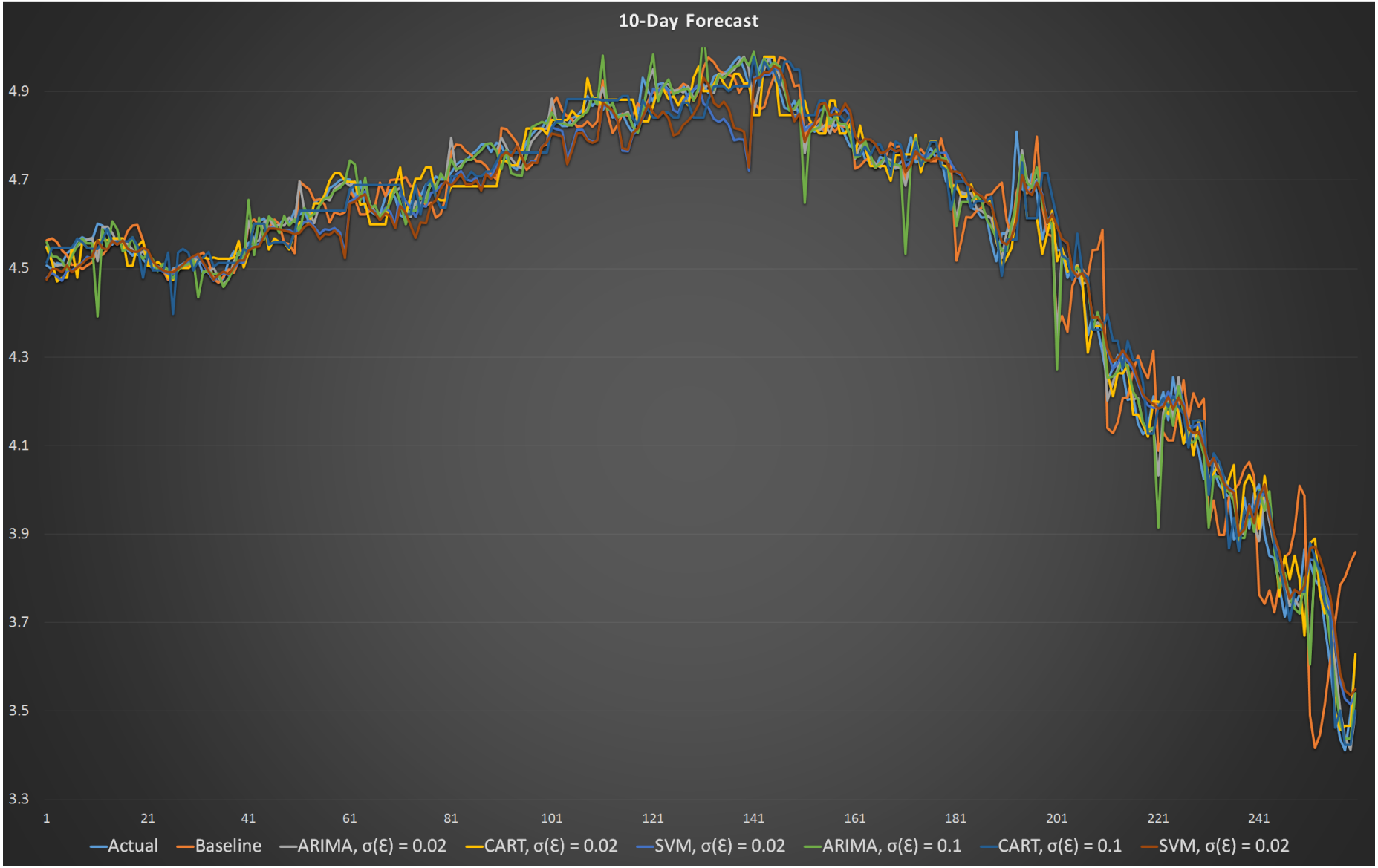
1-Day Forecast



1-Day Forecast Error



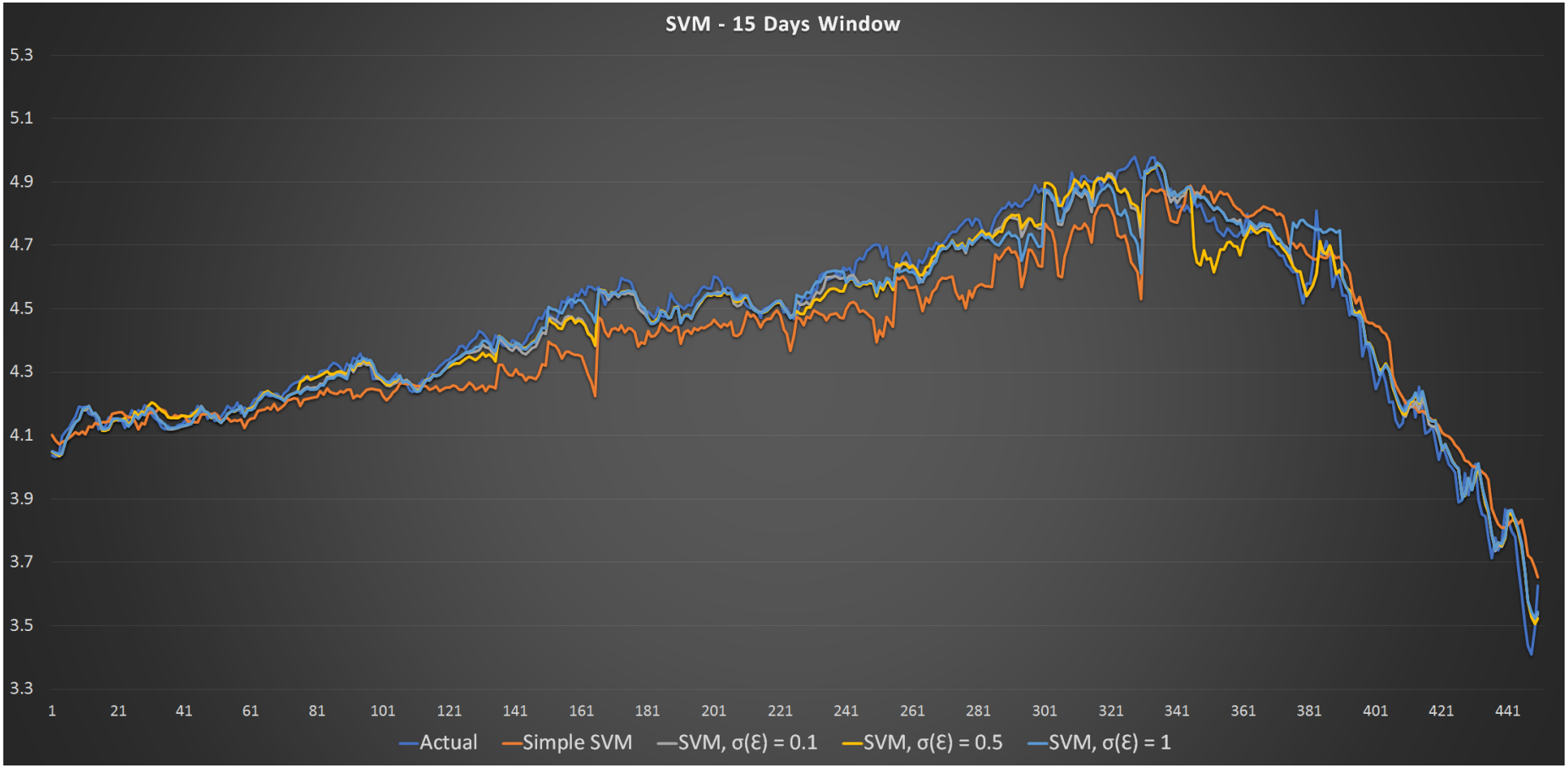
5-Day Forecast



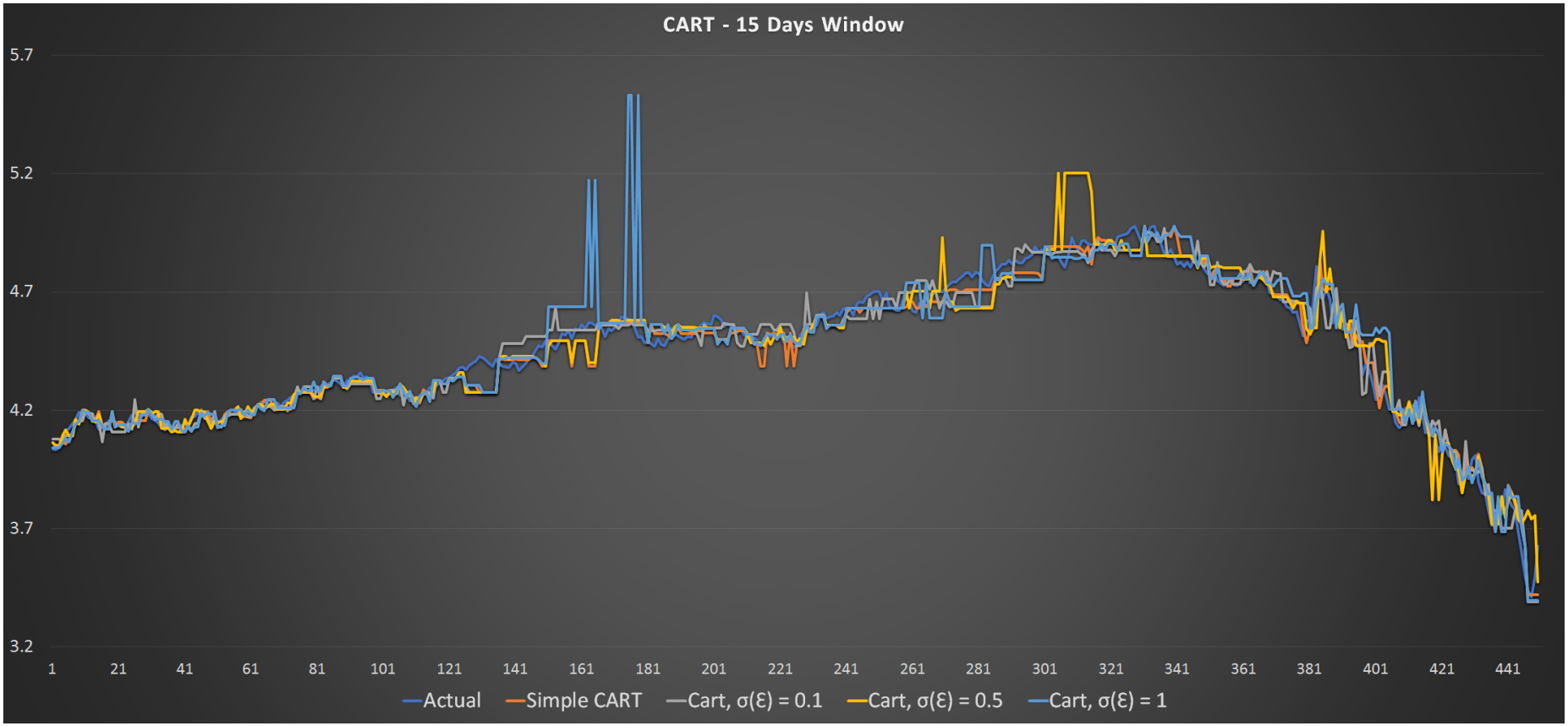
10-Day Forecast

Numerical Results 2.2

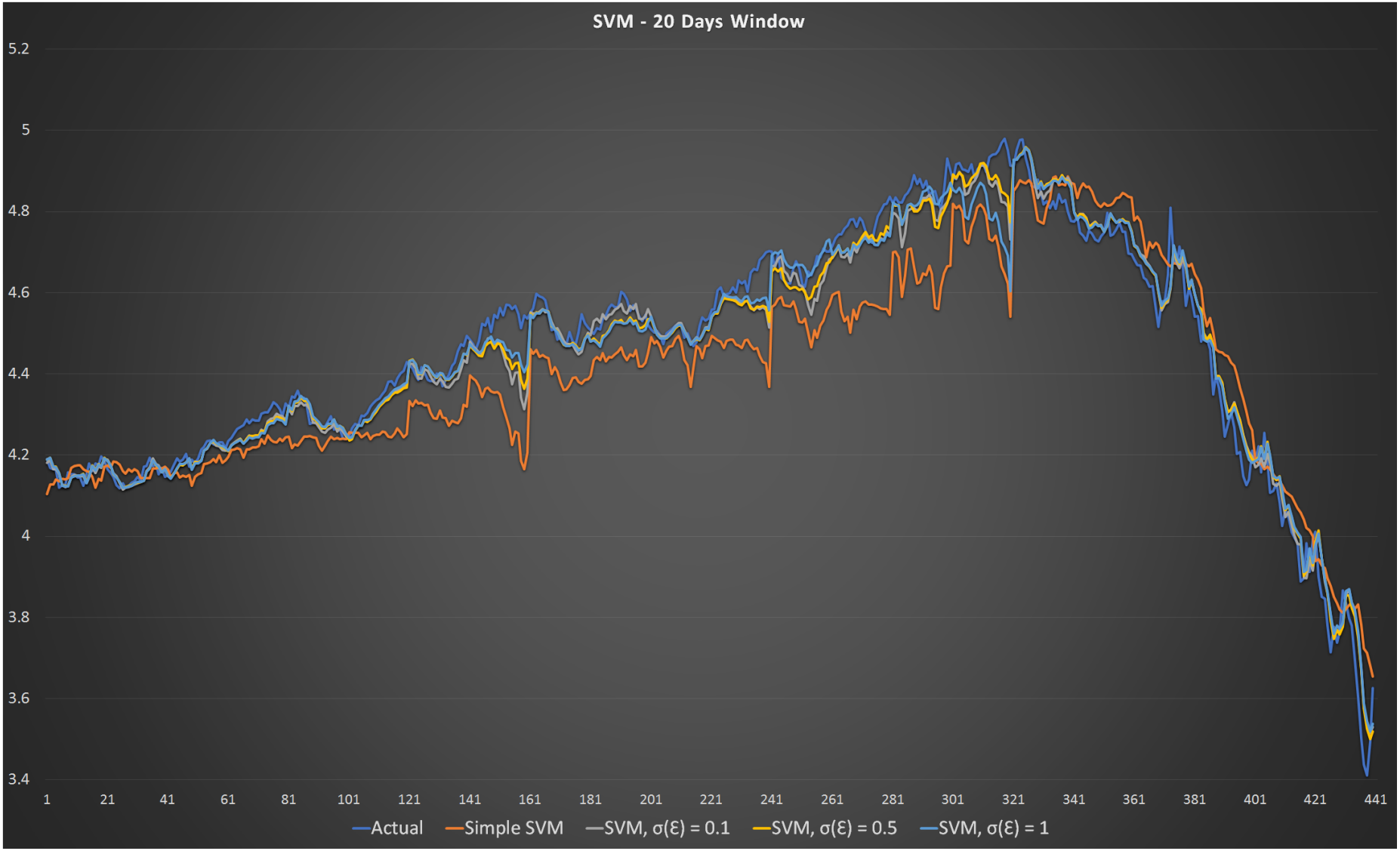
SVM – CART Analysis



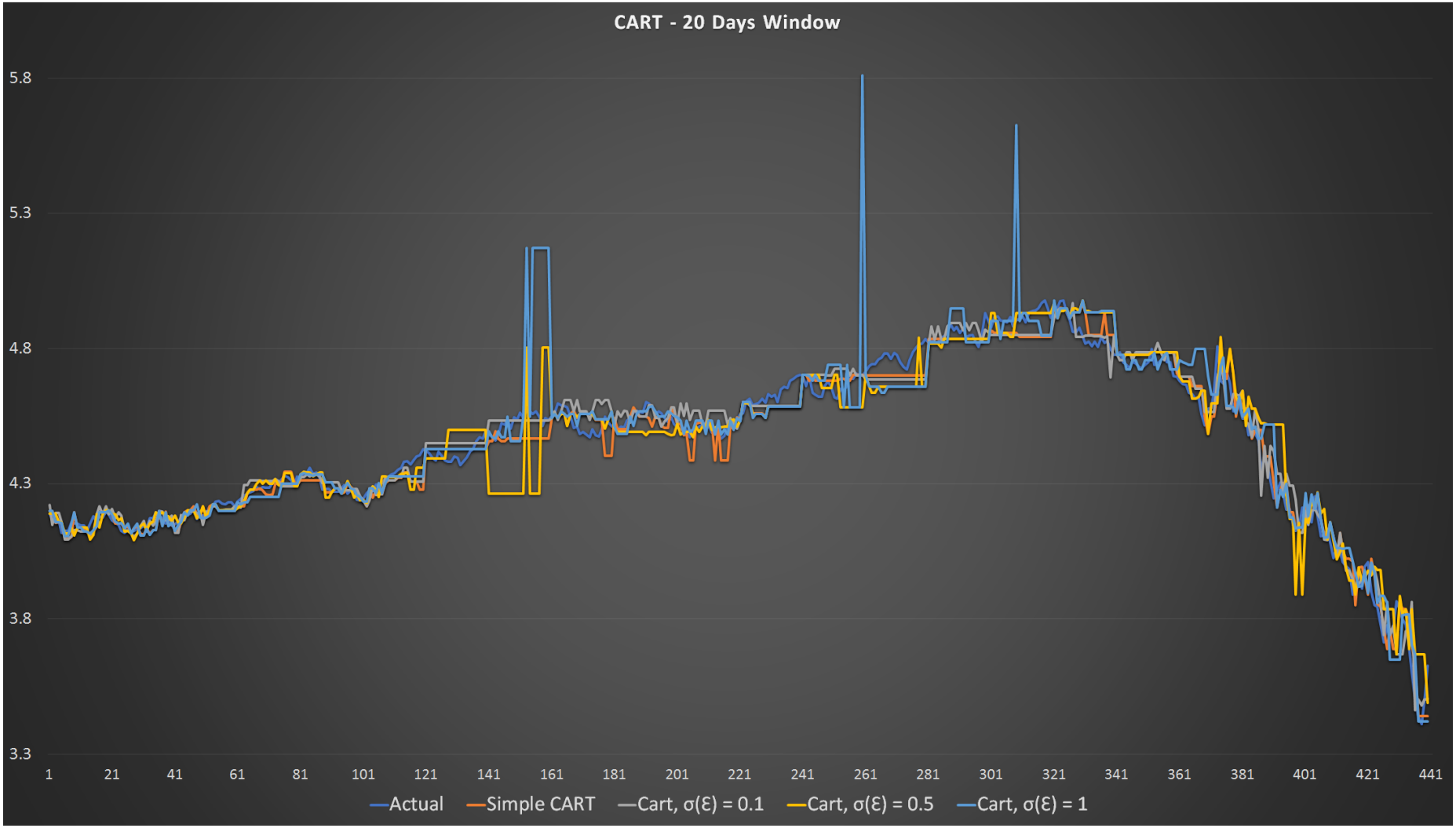
SVM 15 Days Window



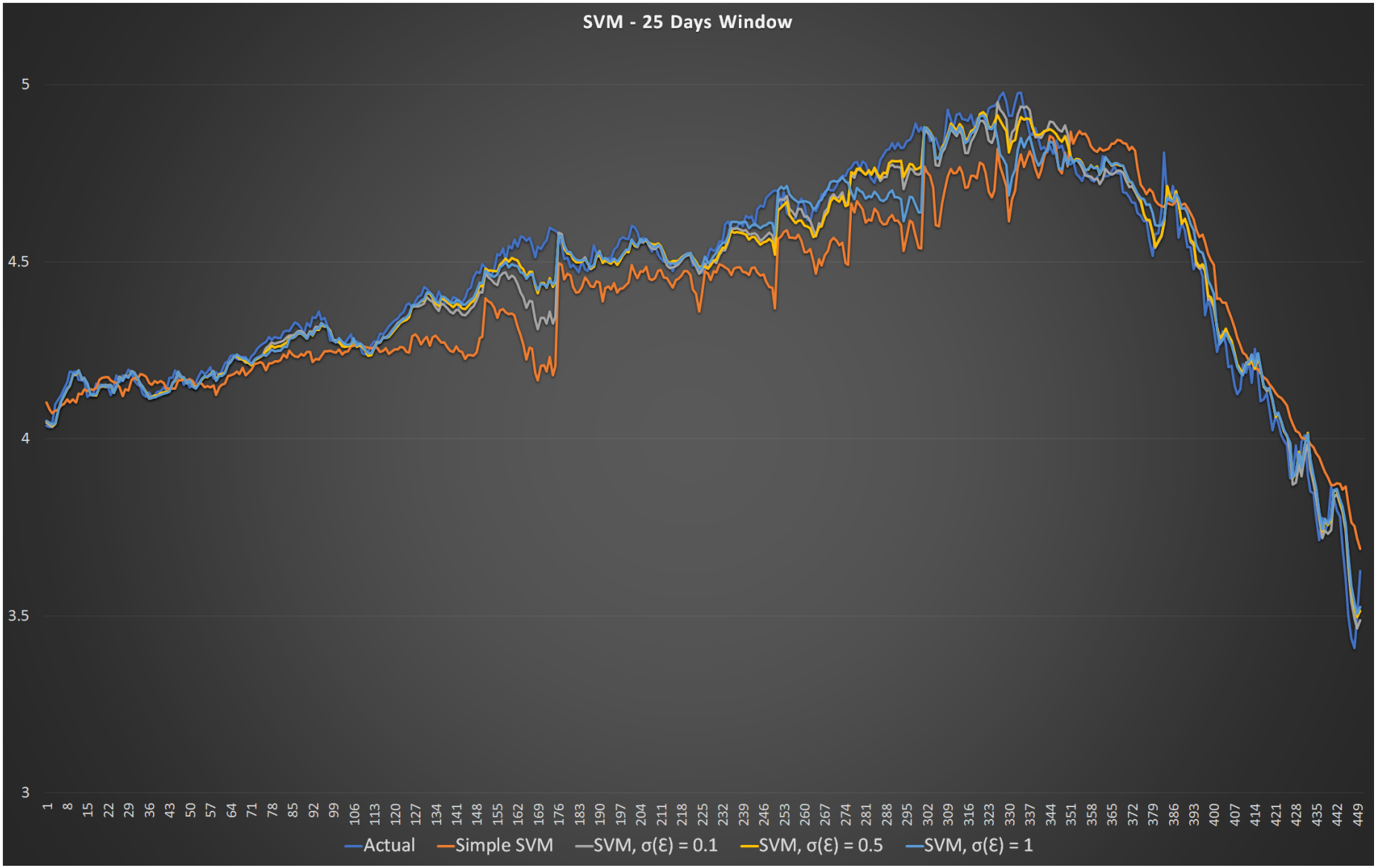
CART 15 Days Window



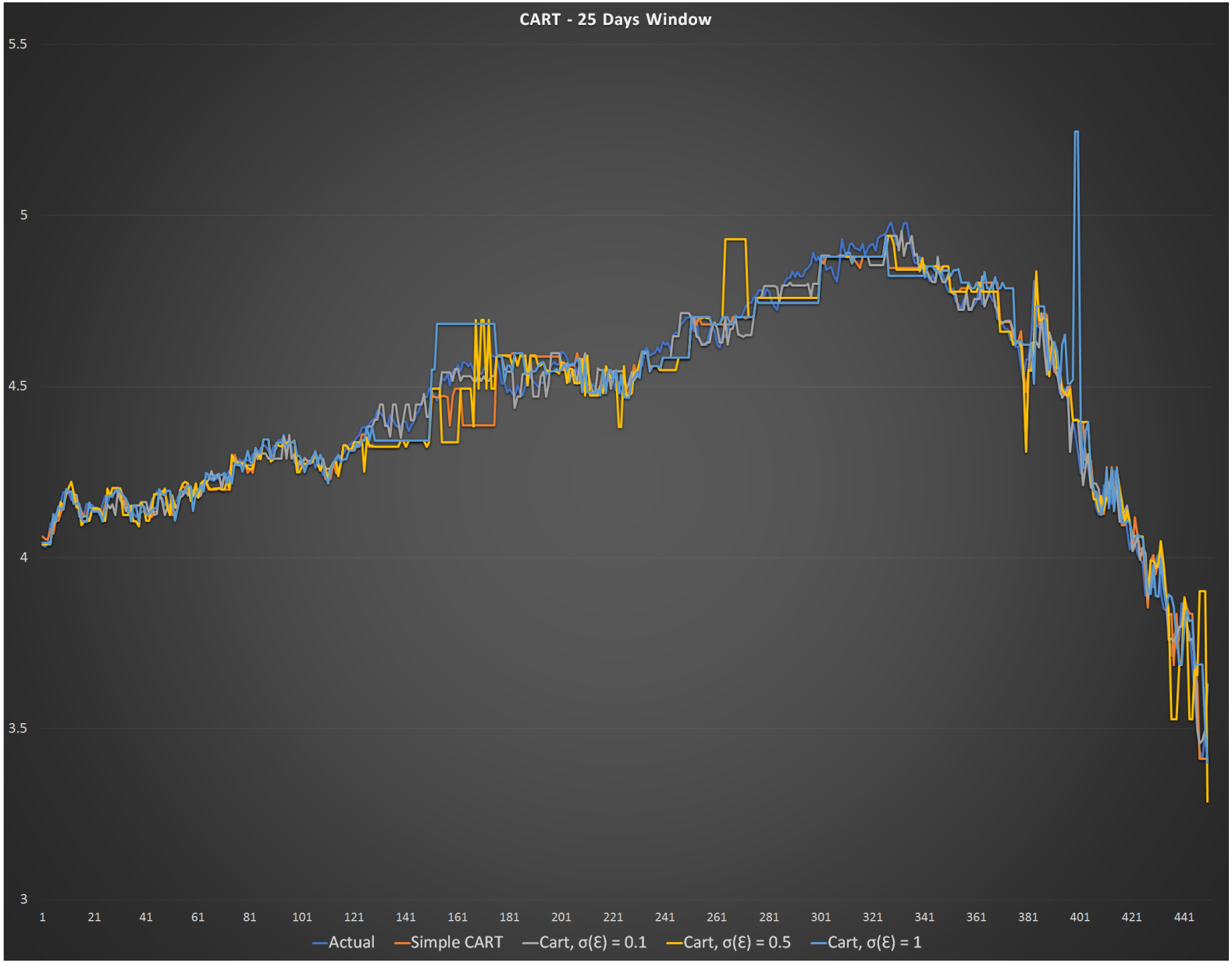
SVM 20 Days Window



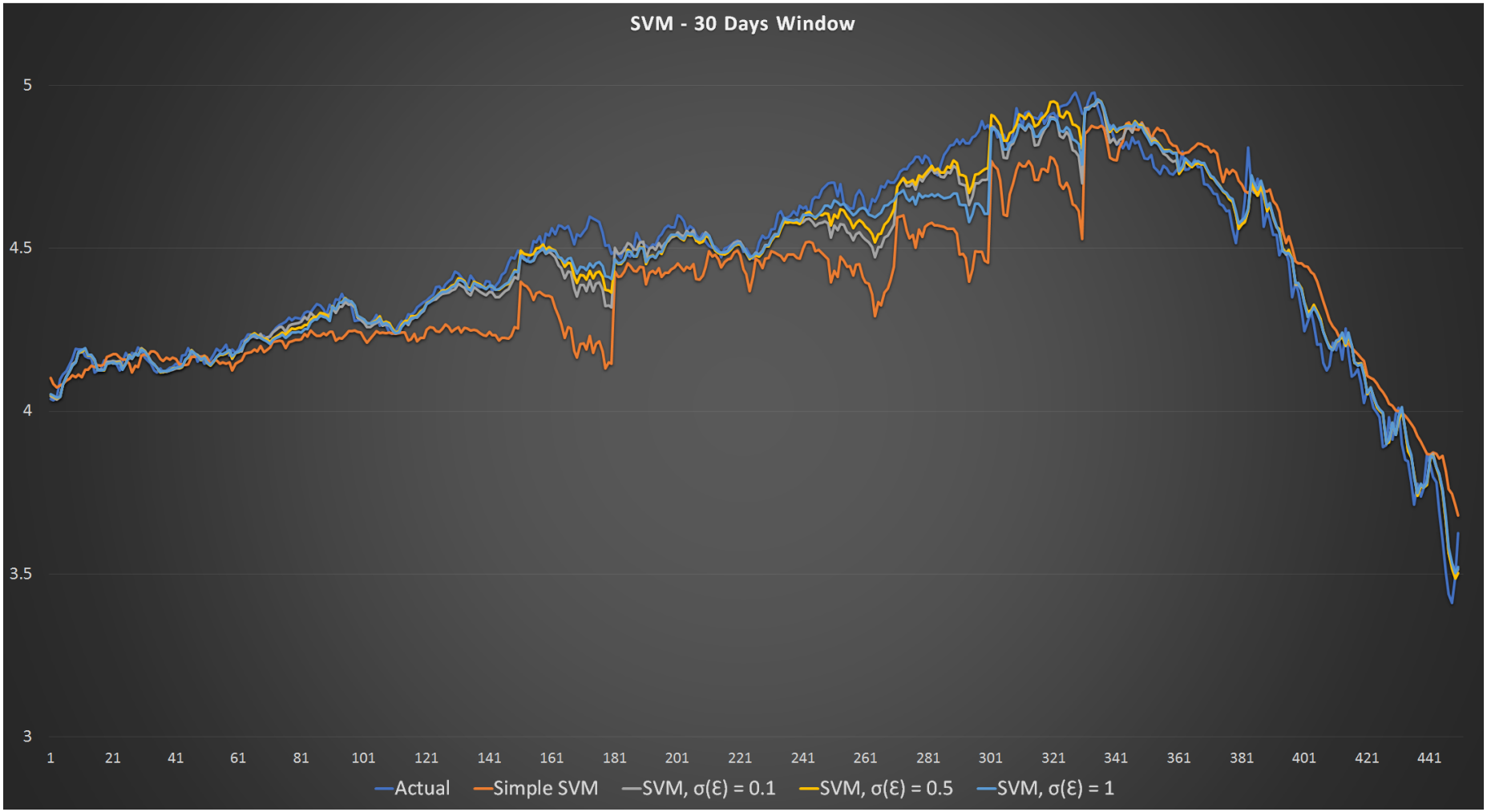
CART 20 Days Window



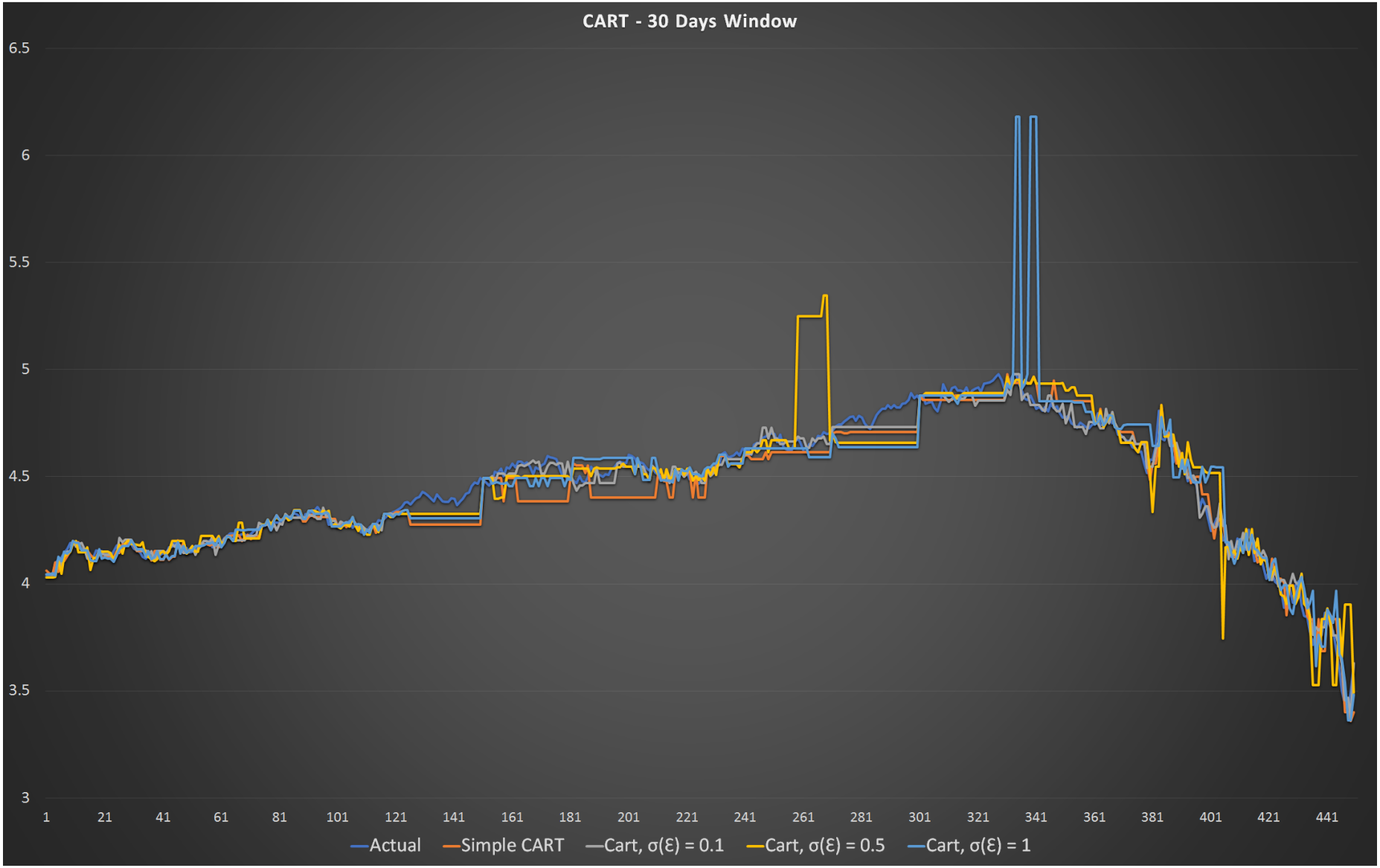
SVM 25 Days Window



CART 25 Days Window



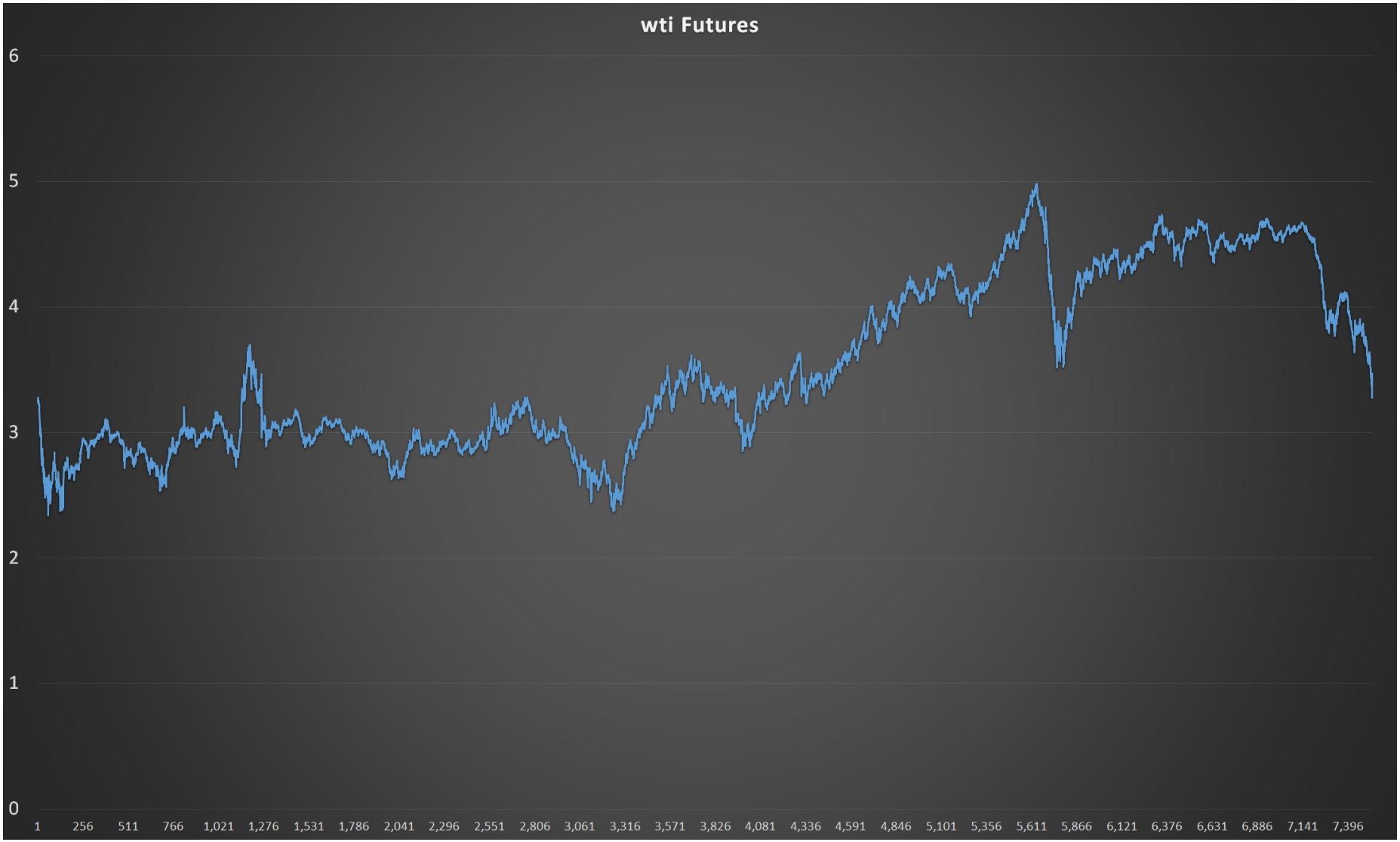
SVM 30 Days Window



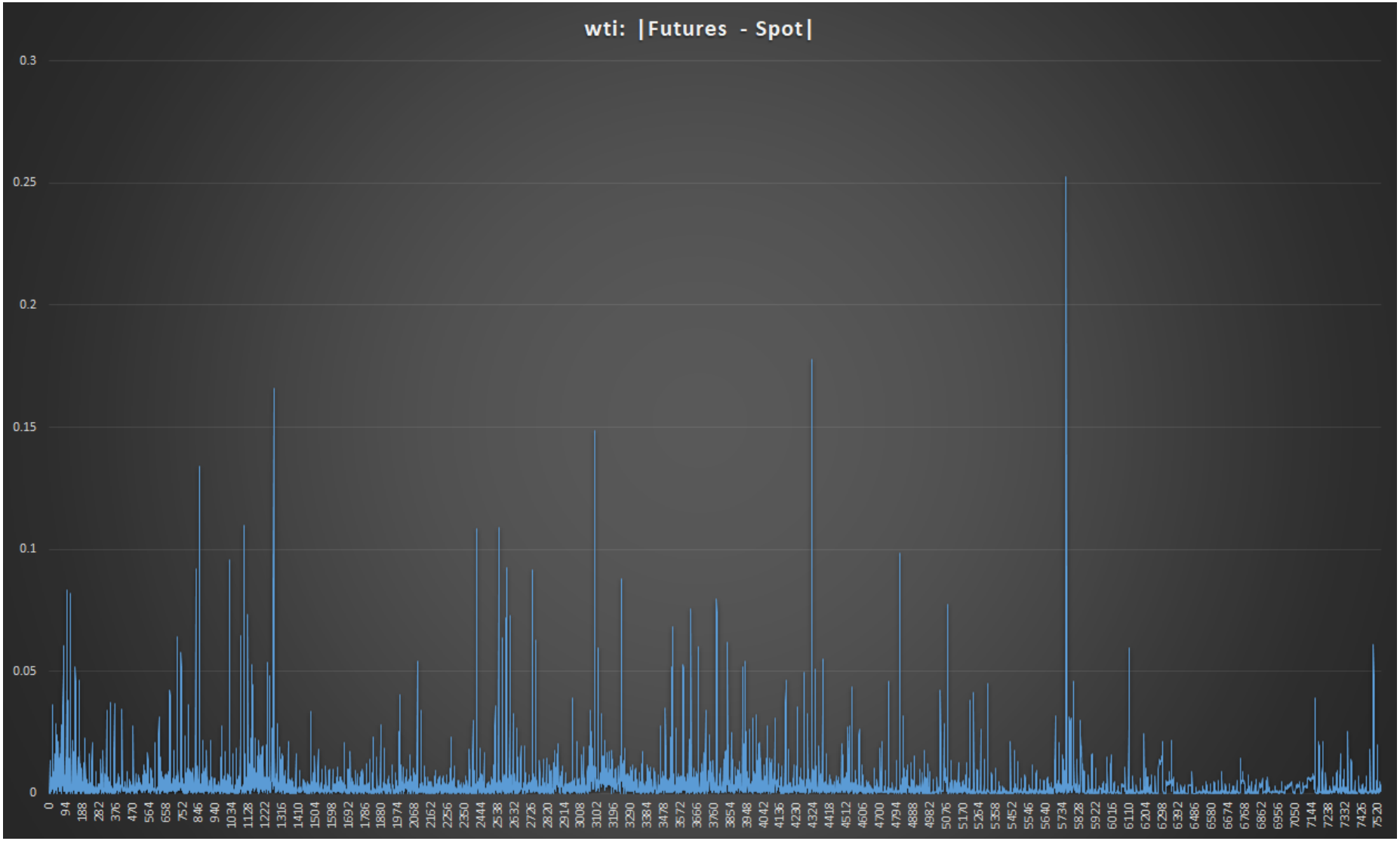
CART 30 Days Window

Numerical Results 2.3

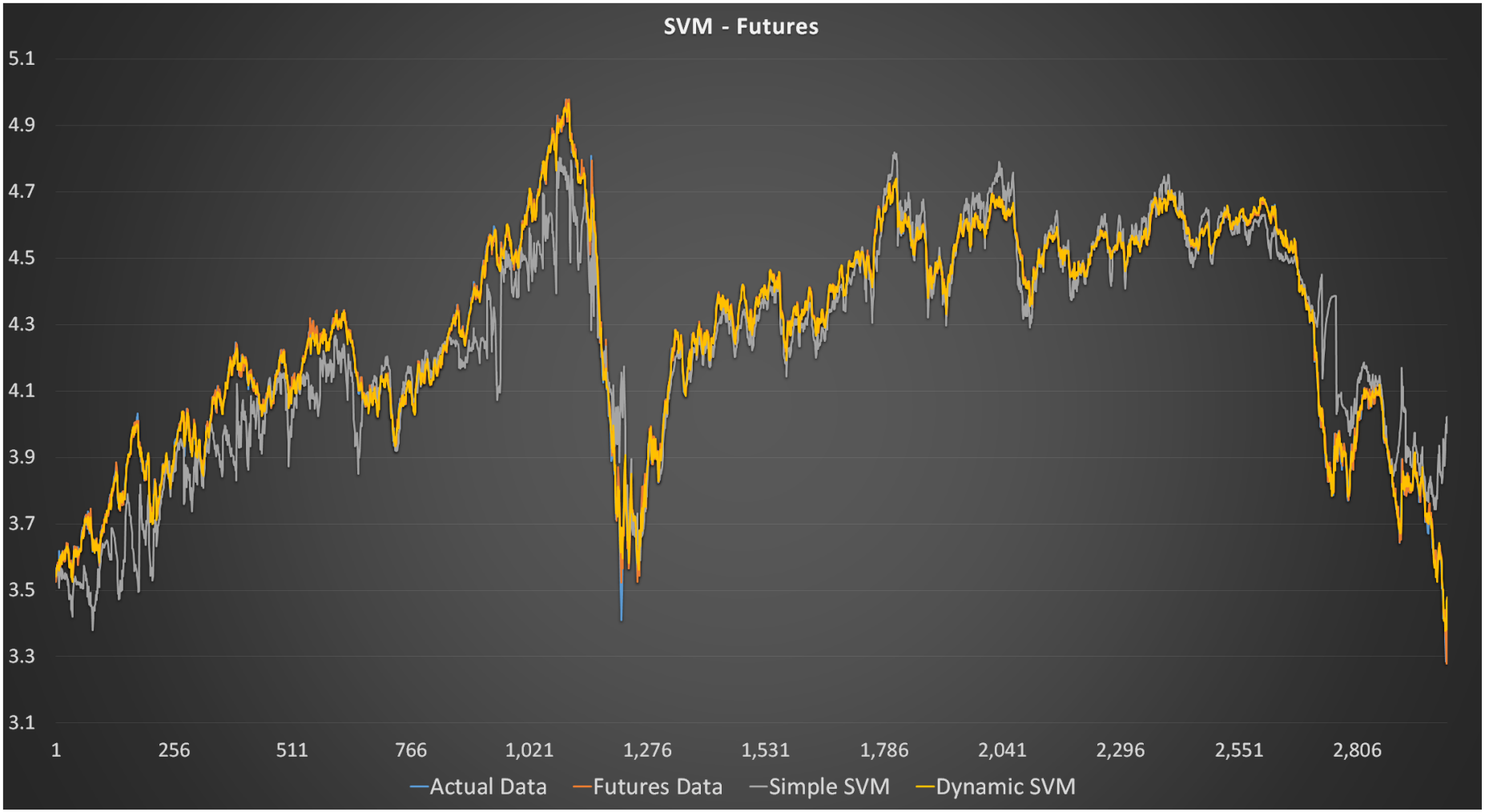
WTI Futures – Sparse SVM Analysis


wti Futures
6
5
4
3
2
1
0
1
256
511
766
1,021
1,276
1,531
1,786
2,041
2,296
2,551
2,806
3,061
3,316
3,571
3,826
4,081
4,336
4,591
4,846
5,101
5,356
5,611
5,866
6,121
6,376
6,631
6,886
7,141
7,396

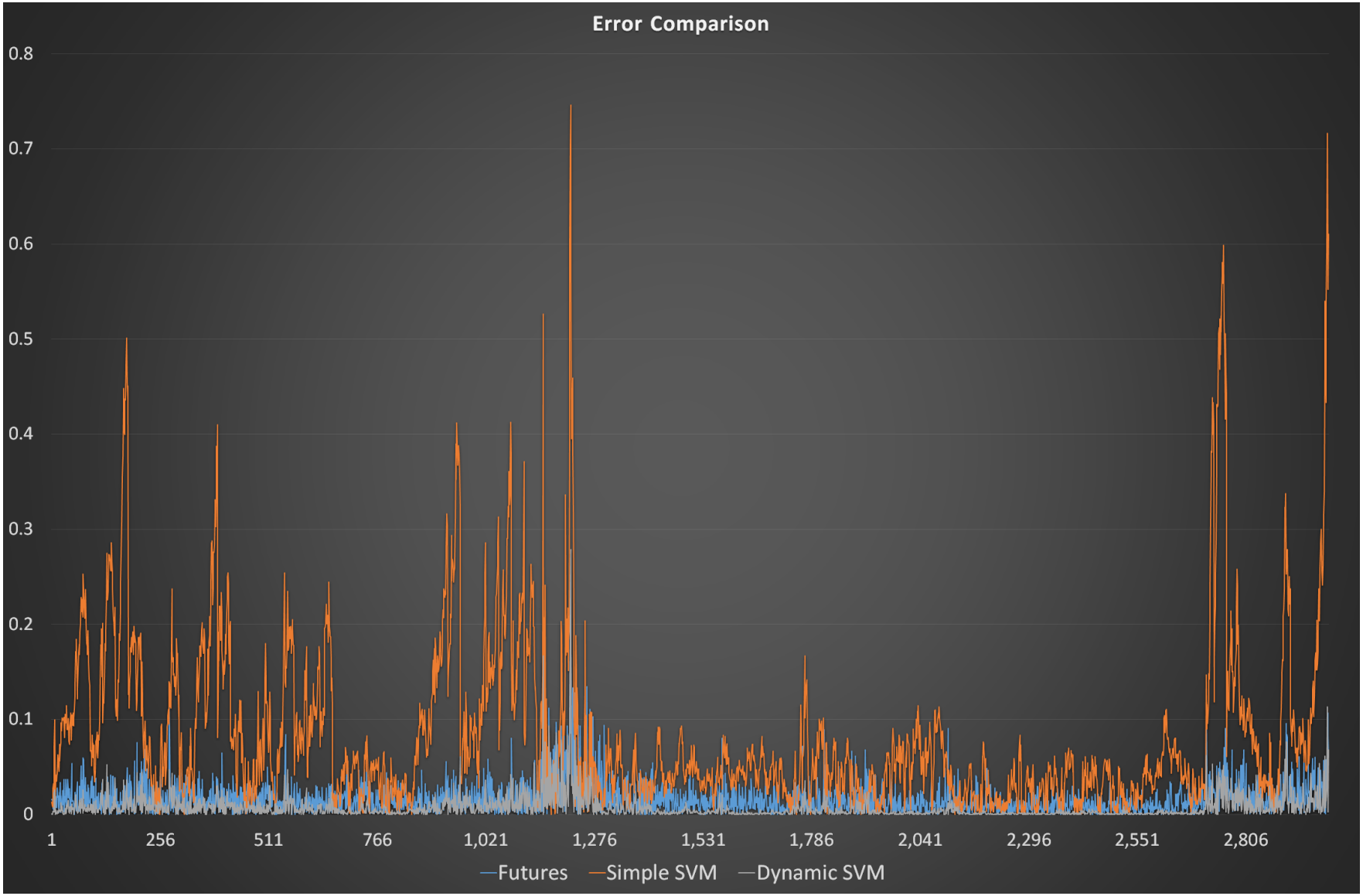
WTI Futures



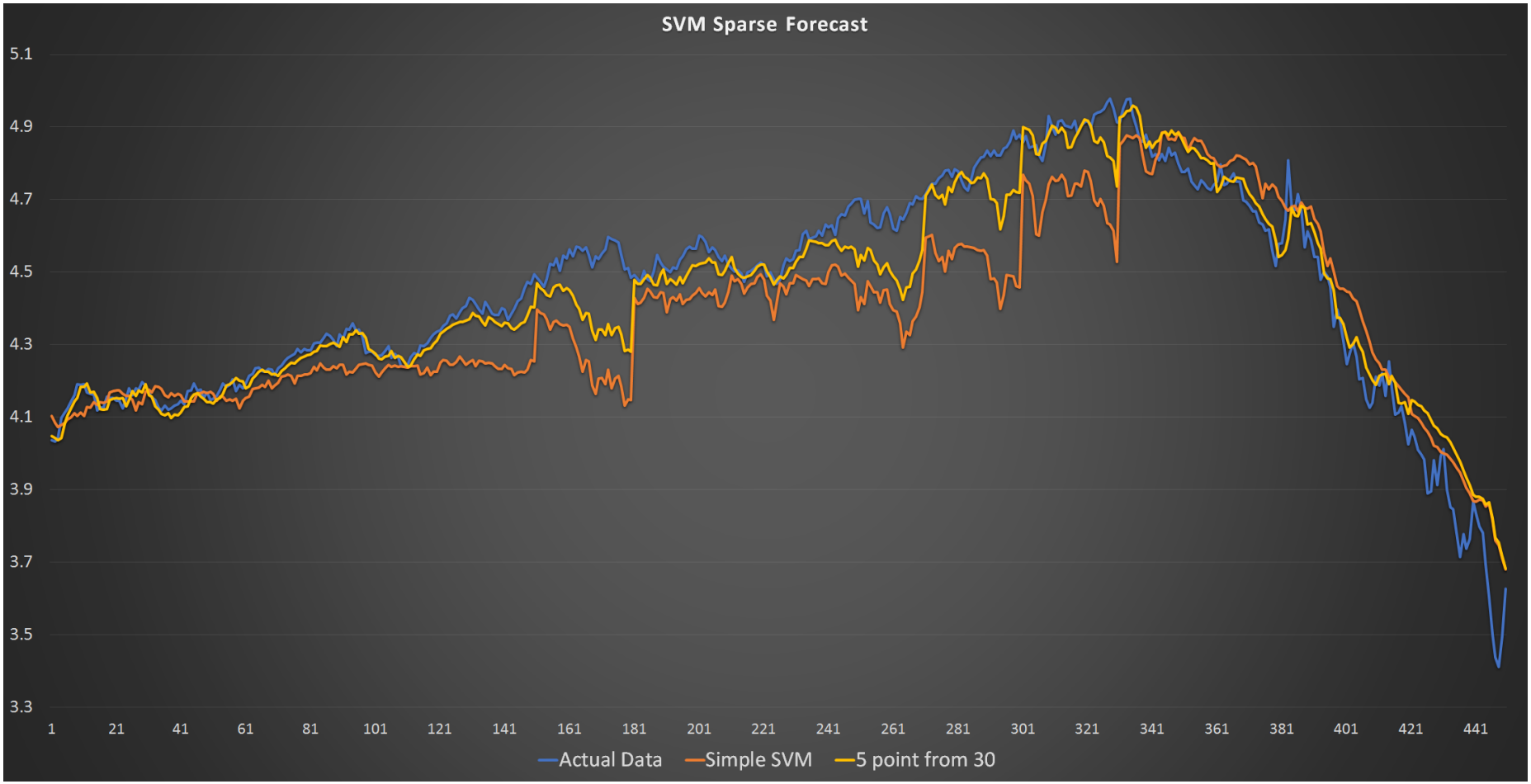
WTI: |Futures − Spot|


SVM - Futures
5.1
4.9
4.7
4.5
4.3
4.1
3.9
3.7
3.5
3.3
3.1
1
256
511
766
1,021
1,276
1,531
1,786
2,041
2,296
2,551
2,806
Actual Data
Futures Data
Simple SVM
Dynamic SVM

SVM Futures



SVM Futures Error Comparison

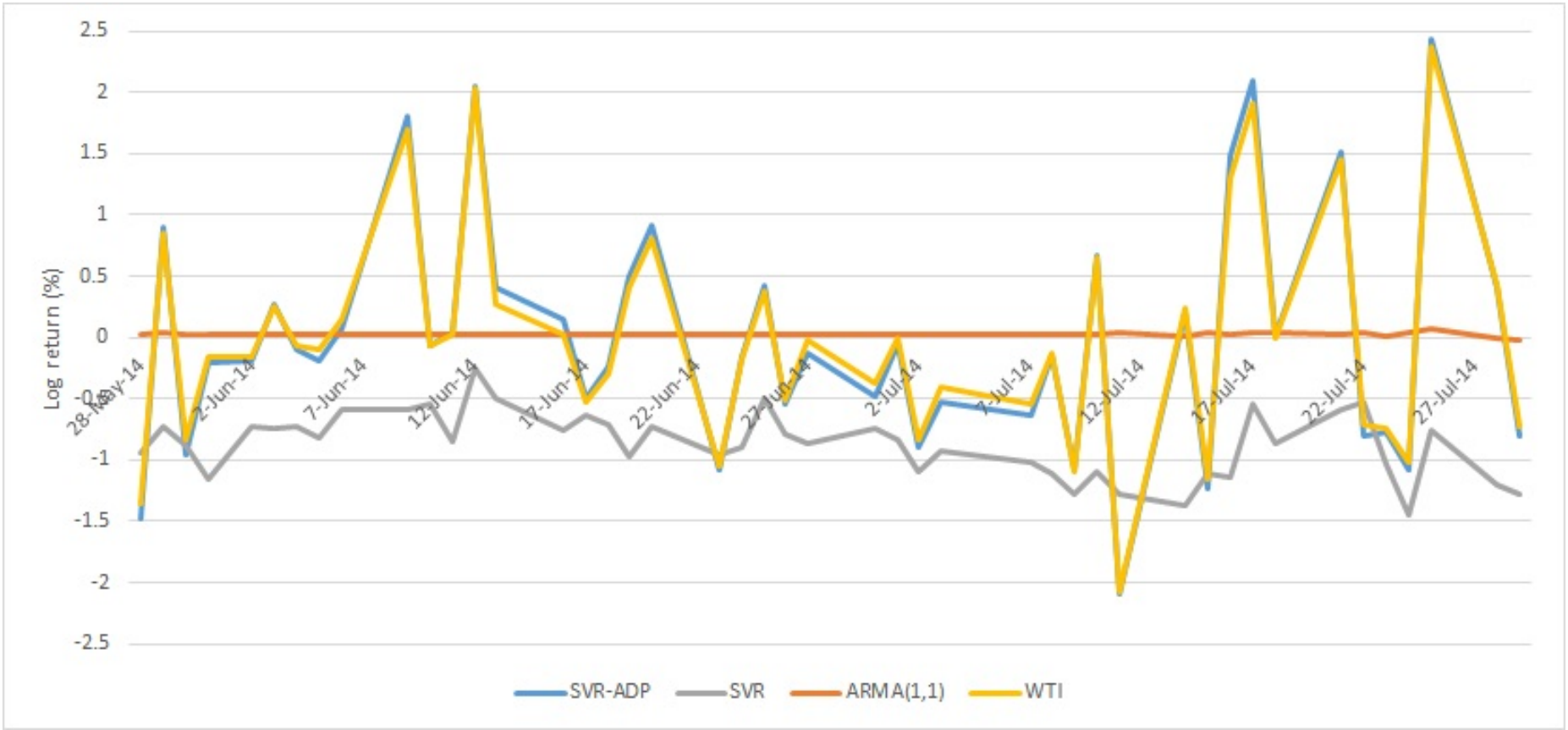


SVM Sparse Forecast

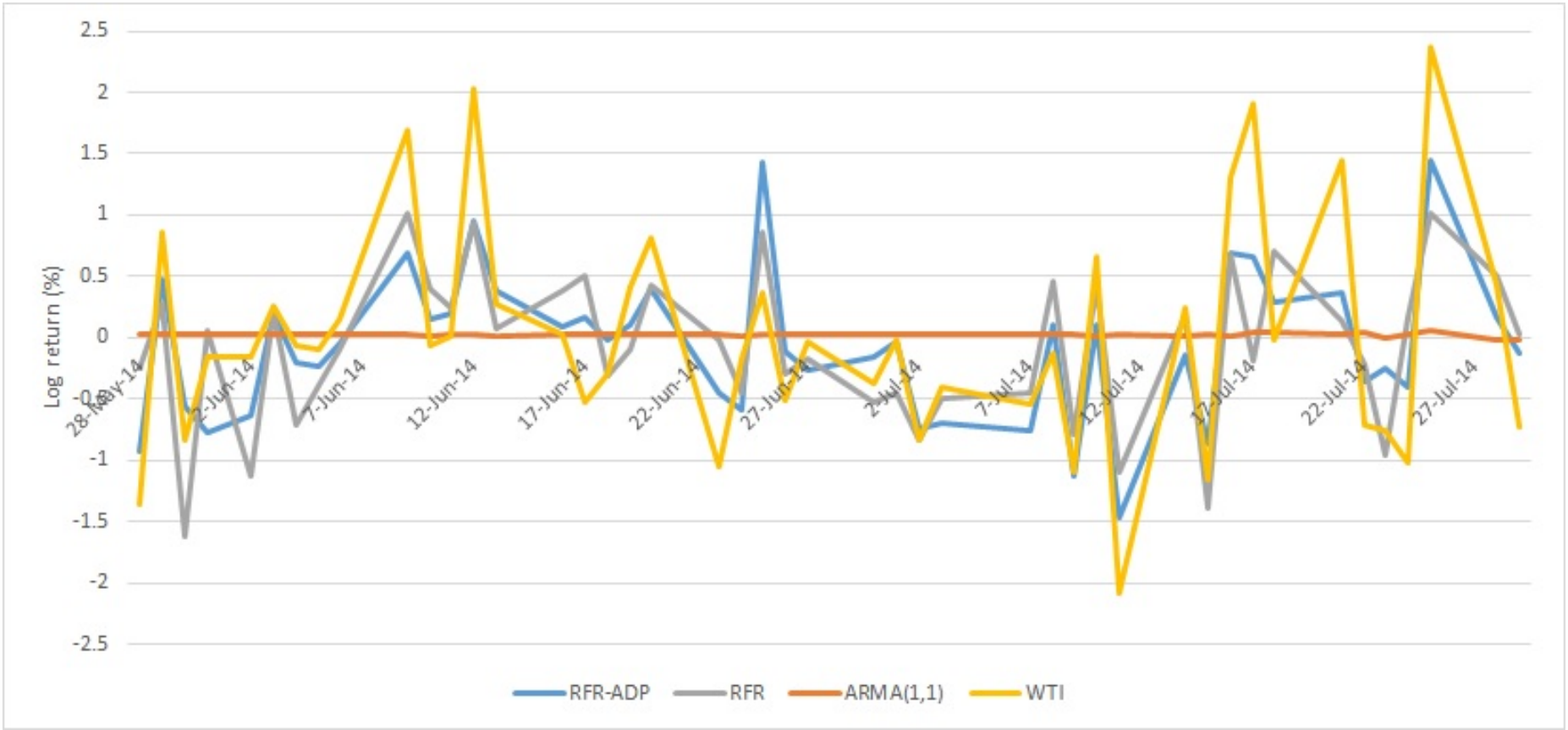
Numerical Results 3

Estimation Techniques III:

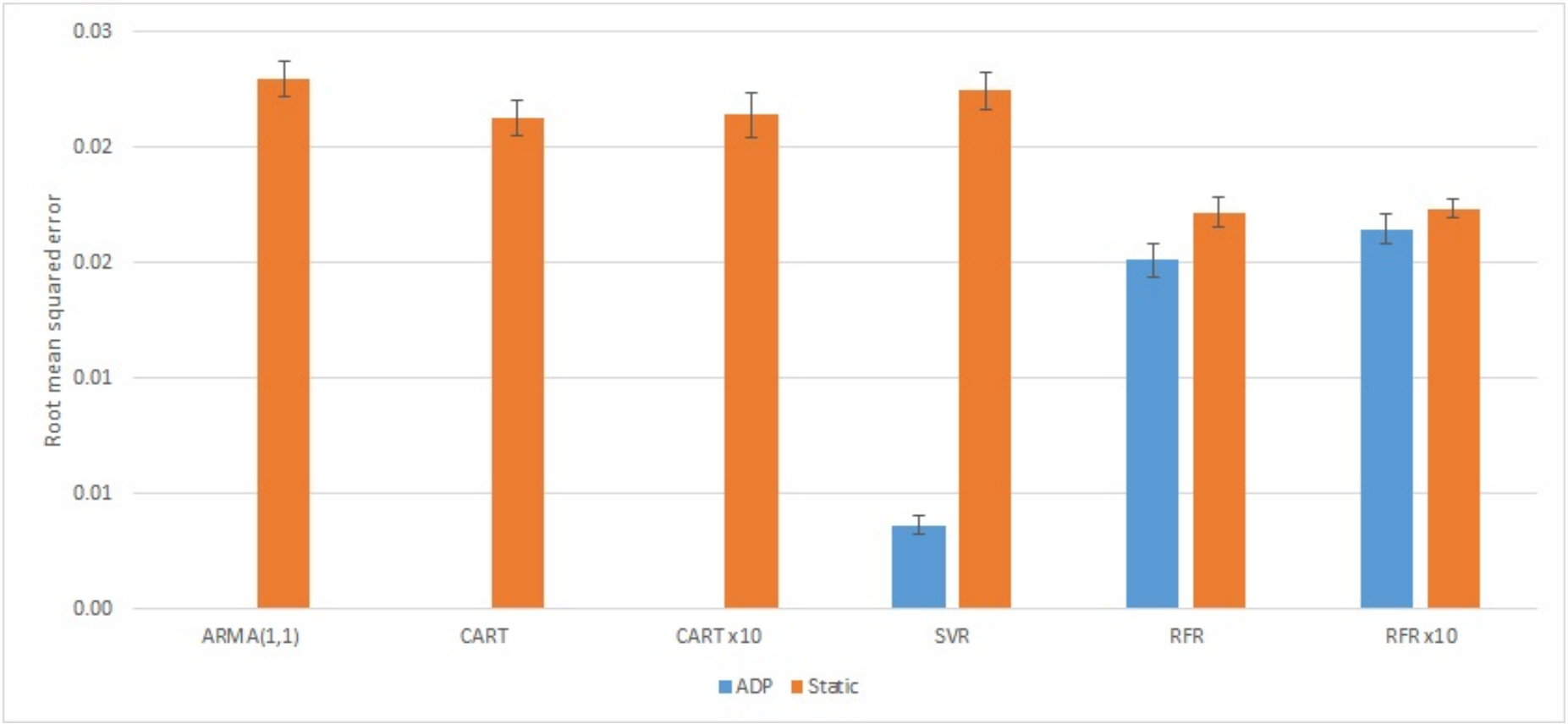
- Test our approximate programming dynamic method comparing the calibration of two machine learning algorithms used for regression and classification: support vector machine (SVM) and random forests (RF).
- For regression analysis:
  - Support vector regression (SVR): minimizes a loss function using only the most relevant values.
  - Random forests for regression (RFR): explores a large search space based on random selection of its features and samples.
- External forecast function for daily data of WTI log returns: 1 month WTI futures price at the end of every month.
- Training & test dataset: 15 lags of the dependent variable and technical indicators:
  - Price indicators: Simple moving averages with 10 and 20 days.
  - Momentum indicators: Relative strength index with 10 and 20 days, and the moving average convergence divergence.
- The test sample includes the forecast function.



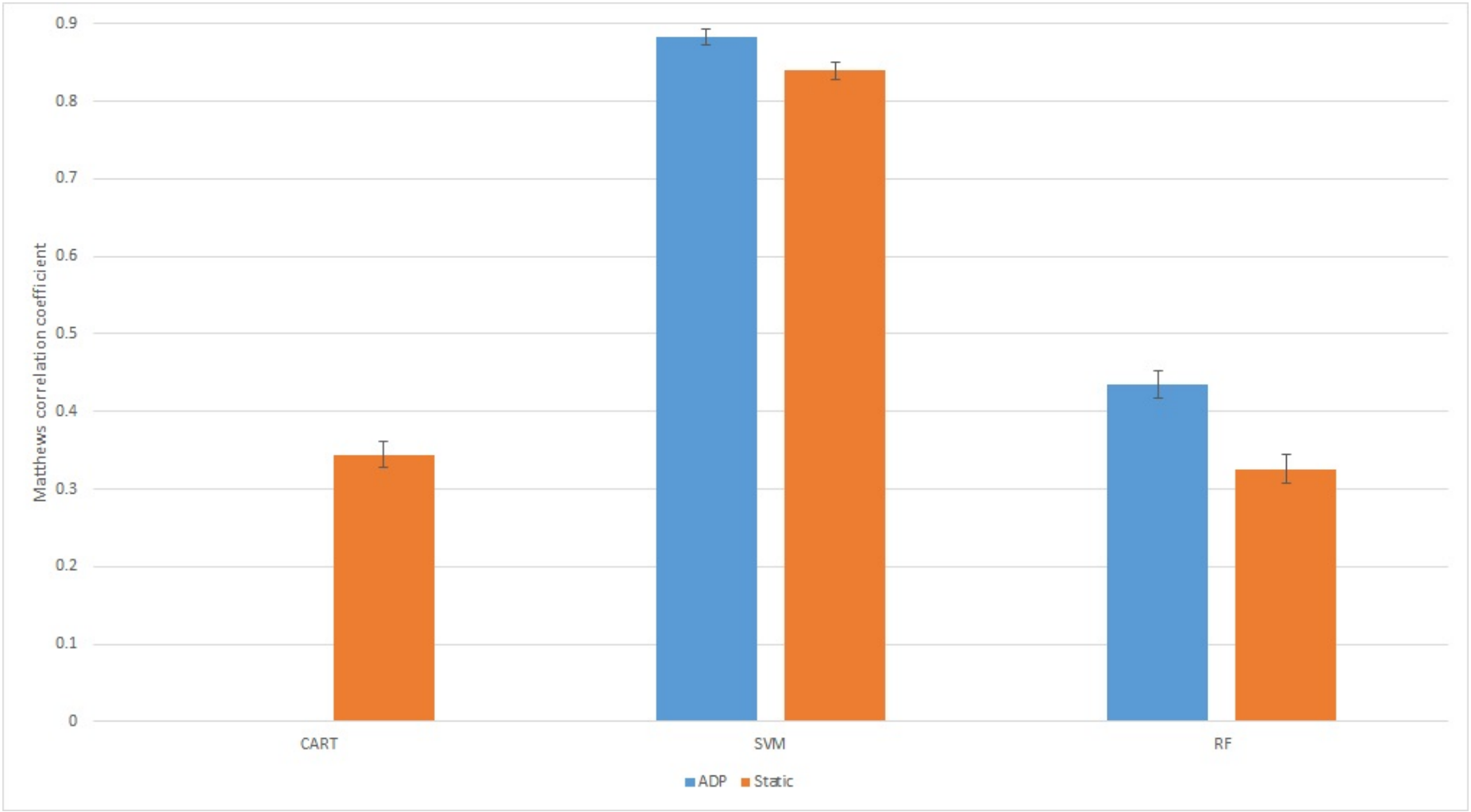
WTI log return and forecasts by ARMA(1,1), support vector regression (SVR), and the approximate dynamic programming SVR (SVR-ADP) for June and July 2014.



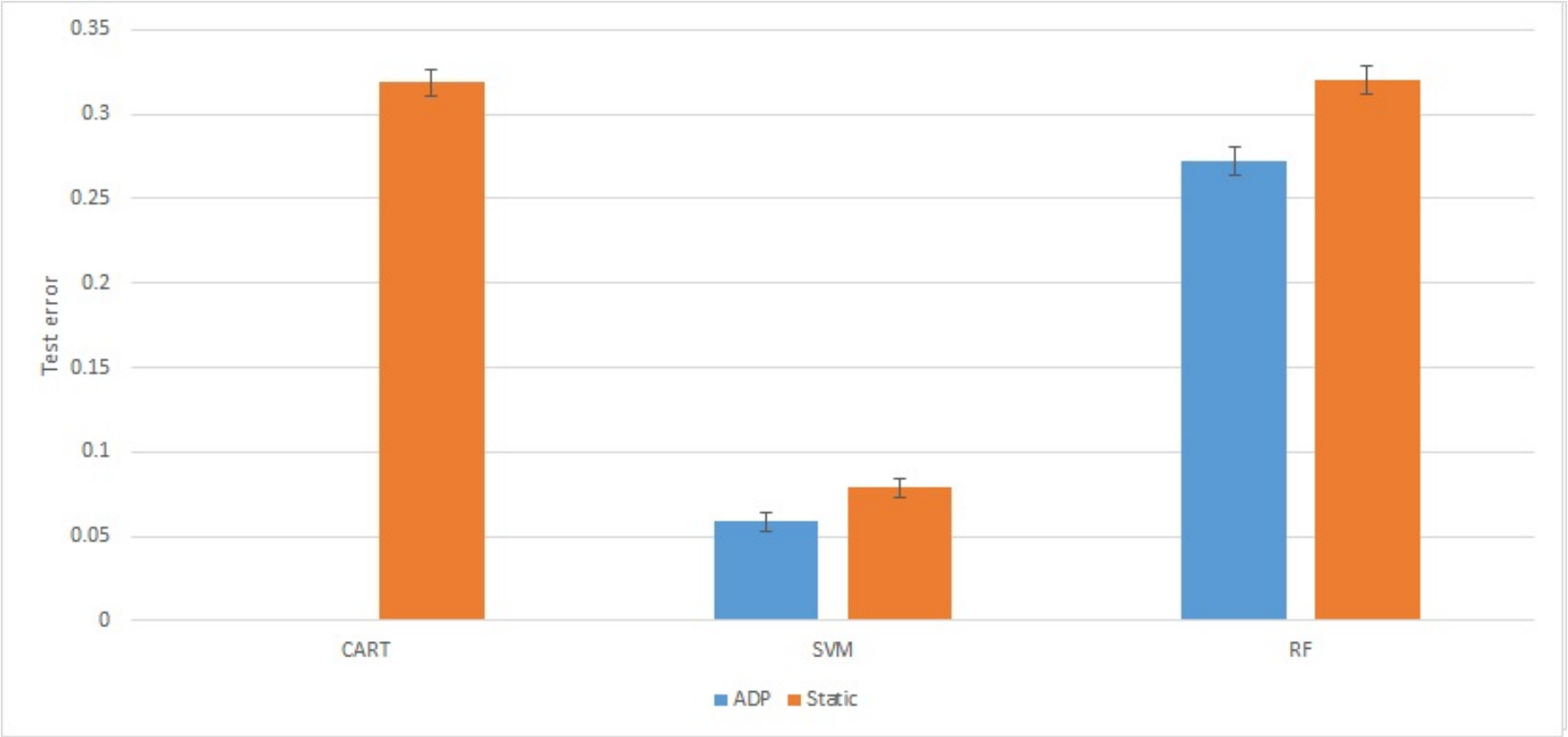
WTI log return and forecasts by ARMA(1,1), random forest for regression (RFR), and the approximate dynamic programming RFR (RFR-ADP) for June and July 2014.



Root mean squared error (RMSE) of ADP and static (benchmark) methods. SVR, RFR, and x10 stand for support vector regression, random forest for regression, and 10 times 40 folds cross-validation respectively.
The error bars represent standard error. RMSE mean differences of SVR-ADP and RFR-ADP in relation to their static versions and ARMA(1,1) and CART are statistically significant with 99% confidence level .



Matthews correlation coefficient of ADP and static (benchmark) methods. SVM and RF stand for support vector machine and random forest respectively. The error bars represen standard error. MCC mean differences of SVM-ADP and RF-ADP in relation to their static versions and CART are statistically significant with 95% conf.level.



Test error of ADP and static (benchmark) methods. SVM and RF stand for support vector machine and random forest respectively. The error bars represent standard error. Test error mean differences of SVM-ADP and RF-ADP in relation to their static versions and CART are statistically significant with 95% confidence level.