MS&E 448 Presentation Final

H. Rezaei, R. Perez, H. Khan, Q. Chen

Description of Technical Analysis Strategy

Identify regularities in the time series of prices by extracting nonlinear patterns from noisy data.

Use a class of smoothing estimators to extract nonlinear relations by "averaging out" the noise.

1) Smoothing Estimators and Kernel Regression
2) Definitions of Technical Patterns
3) The Identification Algorithm
4) Automating Technical Analysis

1) Smoothing Estimators and Kernel Regression

Assume the prices take the following format:

$$P_t = m(X_t) + \epsilon_t, \qquad t = 1, \ldots, T,$$

where $m(X_t)$ is an arbitrary fixed but unknown nonlinear function of a state variable X_t and ε_t is white noise

Smoothing:

- Estimate the nonlinear relationship
- Replicate the way human recognition extracts regularities from noisy data

Smoothing estimator:

$$\hat{m}(x) \equiv \frac{1}{T} \sum_{t=1}^{T} \omega_t(x) P_t,$$

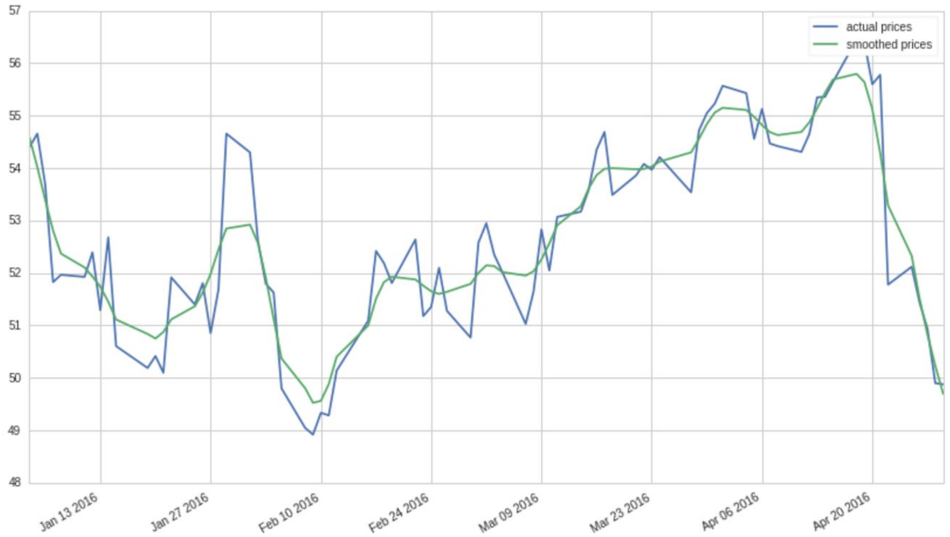
where ω_t is the weighting factor

1) Smoothing Estimators and Kernel Regression

Kernel Regression Estimator:

$$\hat{m}_h(x) = \frac{1}{T}\sum_{t=1}^{T}\omega_{t,h}(x)Y_t = \frac{\sum_{t=1}^{T}\mathrm{K}_h(x-X_t)Y_t}{\sum_{t=1}^{T}\mathrm{K}_h(x-X_t)}.$$

Microsoft (MSFT) smoothing from January 1st 2016 through May 1st 2016



2) Definitions of Technical Patterns

Head and Shoulders (HS)

Inverse Head and Shoulders (IHS)

Broadening Top (BTOP)

Broadening Bottom (BBOT)

Triangle Top (TTOP)

Triangle Bottom (TBOT)

Rectangle Top (RTOP)

Rectangle Bottom (RBOT)

E_1, E_2, E_3, E_4, E_5 are a sequence of consecutive local extrema

$$\text{HS} \equiv \begin{cases} E_1 \text{ is a maximum} \\ E_3 > E_1,\ E_3 > E_5 \\ E_1 \text{ and } E_5 \text{ are within 1.5 percent of their average} \\ E_2 \text{ and } E_4 \text{ are within 1.5 percent of their average,} \end{cases}$$

$$\text{TTOP} \equiv \begin{cases} E_1 \text{ is a maximum} \\ E_1 > E_3 > E_5 \\ E_2 < E_4 \end{cases}$$

3) The Identification Algorithm

- Given a sample of prices $P_1, \ldots, P_T$, we fit kernel regressions, one for each window from t to t+l+d-1, where t varies from 1 to T-l-d+1,

- Fixes the length of the window at l + d to distinguish signal from noise in this case.

- l: length of the window

- d: the number of days following the completion of a pattern that must pass before the pattern is detected. The lag d ensures that we are computing our conditional returns without any "look-ahead" bias.

- Within each window, we estimate a kernel regression using the prices in that window:

$$\hat{m}_h(\tau) = \frac{\sum_{s=t}^{t+l+d-1} K_h(\tau - s) P_s}{\sum_{s=t}^{t+l+d-1} K_h(\tau - s)}, \qquad t = 1, \ldots, T - l - d + 1,$$

- Proceed to check for the presence of the various technical patterns after we have identified all of the local extrema in the window [t, t + l + d - 1]

4) Automating Technical Analysis

1. Define each technical pattern in terms of its geometric properties, for example, local extrema.

2. Construct a kernel estimator of a given time series of prices so that its extrema can be determined numerically.

3. Analyze the kernel estimator for occurrences of each technical pattern.

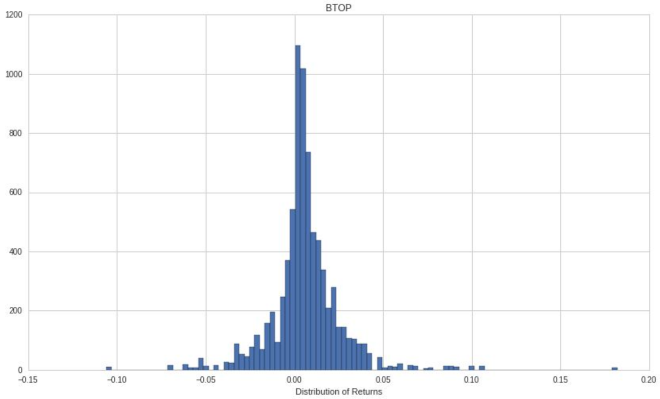
Training set

- Data from the 100 most liquid stocks from 2002-1-1 through 2010-12-31. A period of 9 years.
- Identify the patterns
- Compute returns based on waiting and holding period
- Optimize parameters:
    - Waiting period (w): number of days after recognizing the pattern before entering position
    - Holding period (l): number of days that position is hold

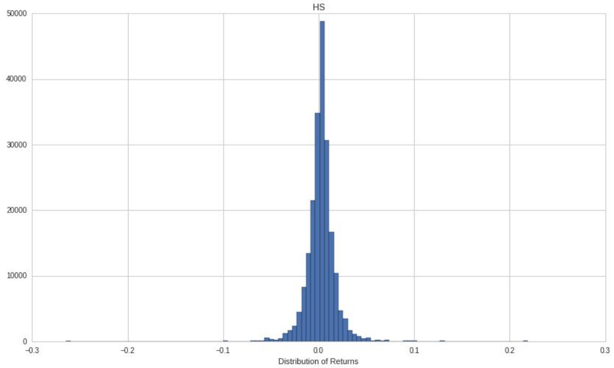
Optimized Waiting and Holding Periods

Pattern	**Waiting Period** (days)	**Holding Period** (days)
Head and Shoulders	1	1
Broadening Top	1	1
Rectangle Top	1	1
Triangle Top	2	1
Inverse Head and Shoulders	1	1
Broadening Bottom	1	1
Rectangle Bottom	1	1
Traingle Bottom	2	1

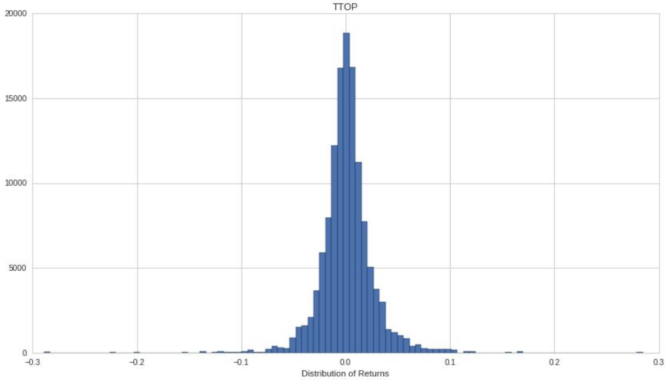
Distribution of Returns: Bearish Patterns



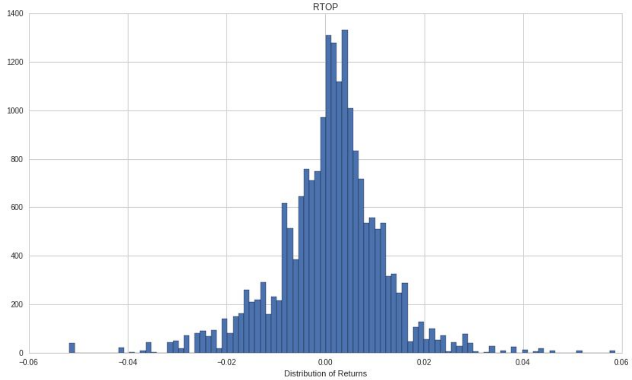
BTOP:
Mean: 0.006174
SD: 0.020361
SR: 4.6564



HS:
Mean: 0.002715
SD: 0.014904
SR: 2.8912

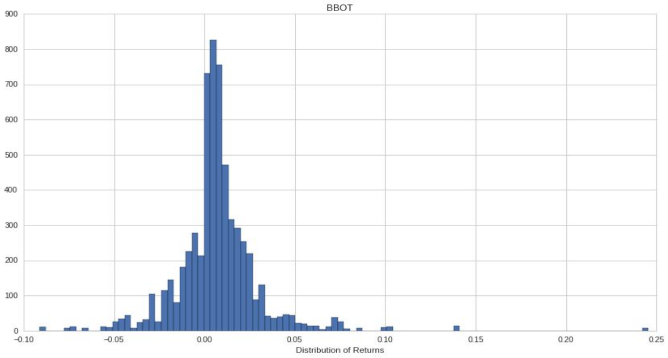


TTOP:
Mean: 0.001341
SD: 0.026575
SR: 0.8010

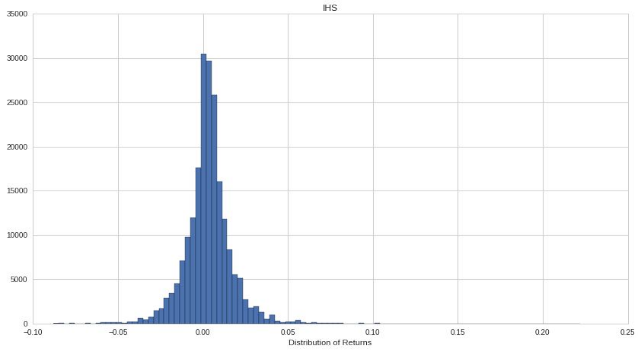


RTOP:
Mean: 0.000930
SD: 0.010806
SR: 0.45250

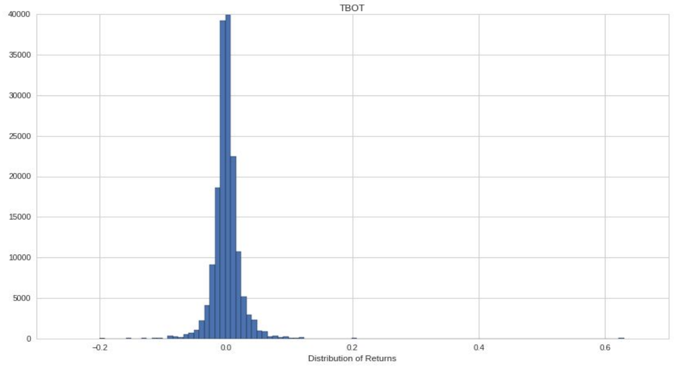
Distribution of Returns: Bullish Patterns



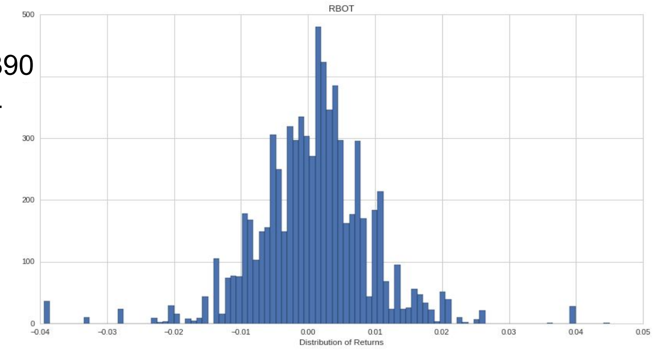
BBOT:
Mean: 0.006803
SD: 0.023195
SR: 4.6563



IHS::
Mean: 0.003169
SD: 0.013959
SR: 3.6040



TBOT:
Mean: 0.002390
SD: 0.021694
SR:1.7494



RBOT:
Mean: 0.000224
SD: 0.007862
SR: 0.4524

Best Patterns

Broadening Top (BTOP)

Inverse Head and Shoulders (IHS)

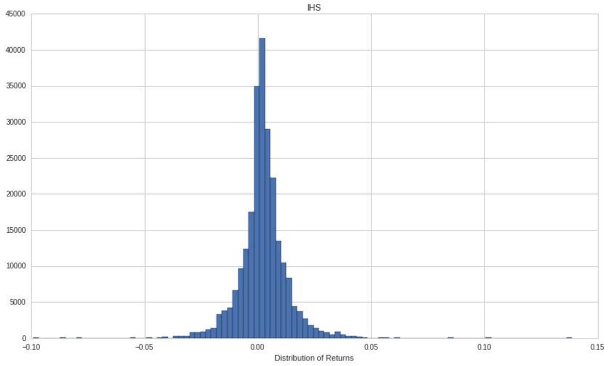
Broadening Bottom (BBOT)

Head and Shoulders (HS)

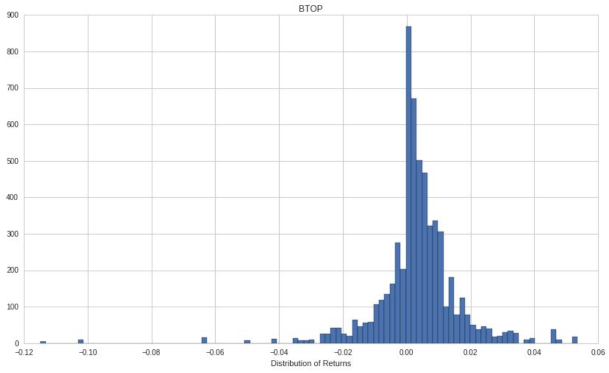
Training set

- Out of sample data from the 100 and 1000 most liquid stocks from 2011-1-1 through 2017-12-31. A period of 7 years.
- Using optimized parameters, we compute return distribution for the best 4 patterns

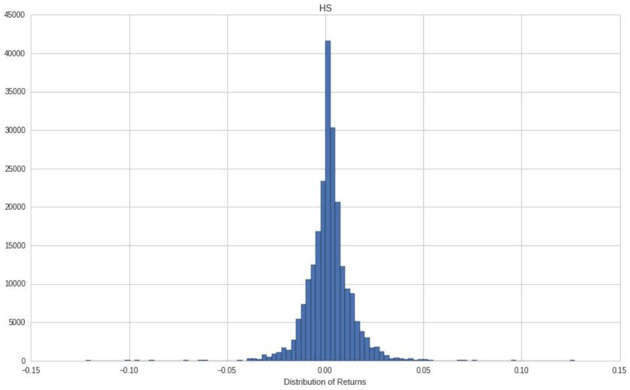
Universe of 100 Stocks



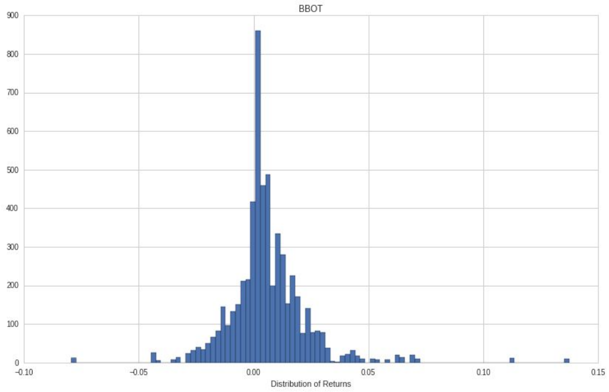
IHS:
Mean: 0.002277
SD: 0.010785
SR: 3.3525



BTOP:
Mean: 0.002919
SD: 0.013236
SR: 3.99147

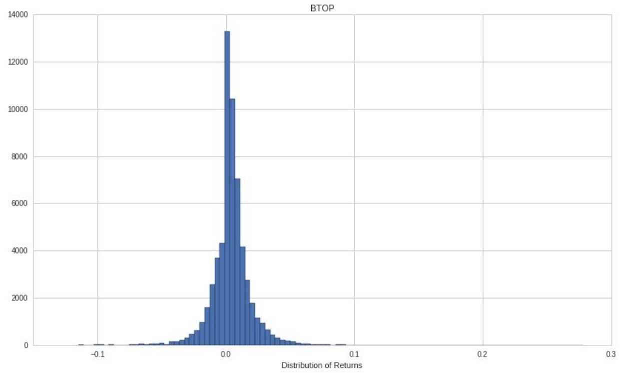


HS:
Mean: 0.001825
SD: .0.011382
SR: 2.5466

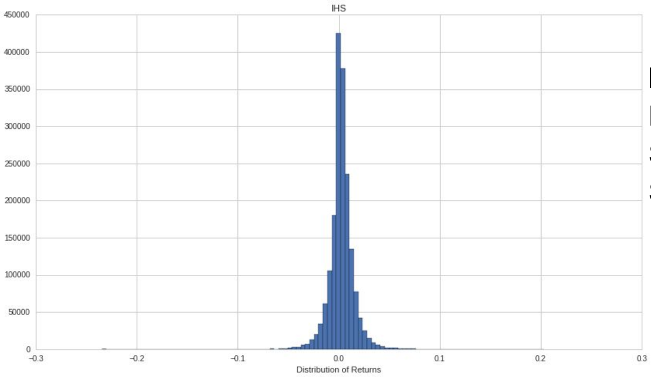


BBOT:
Mean: 0.003049
SD: 0.013959
SR: 5.5683

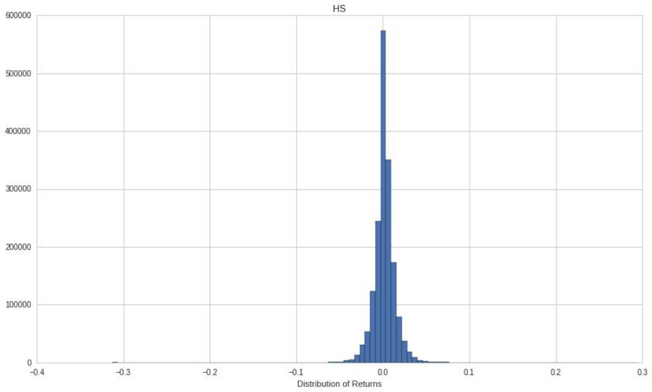
Universe of 1000 Stocks



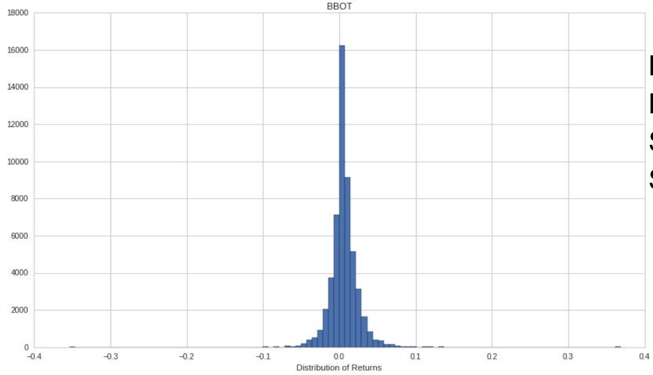
BTOP:
Mean: 0.004089
SD: 0.015237
SR: 4.2604



IHS:
Mean: 0.002583
SD: .0.012463
SR: 3.2902



HS:
Mean:0.002086
SD: .0.012589
SR: 2.6305

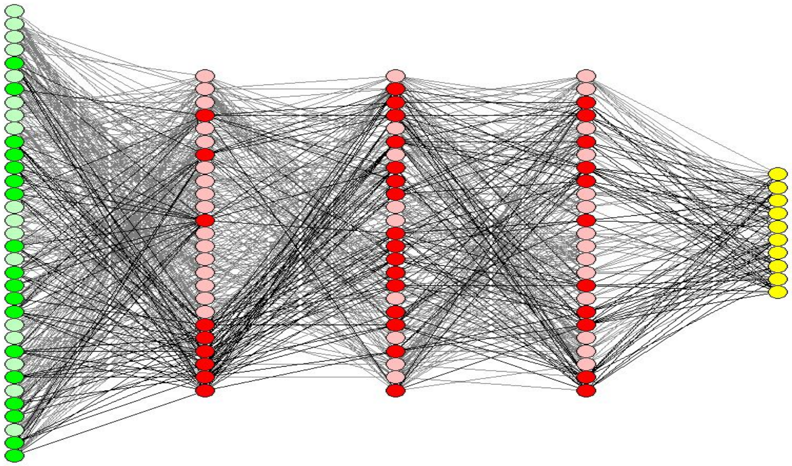


BBOT:
Mean: 0.006314
SD: 0.020119
SR: 4.9824

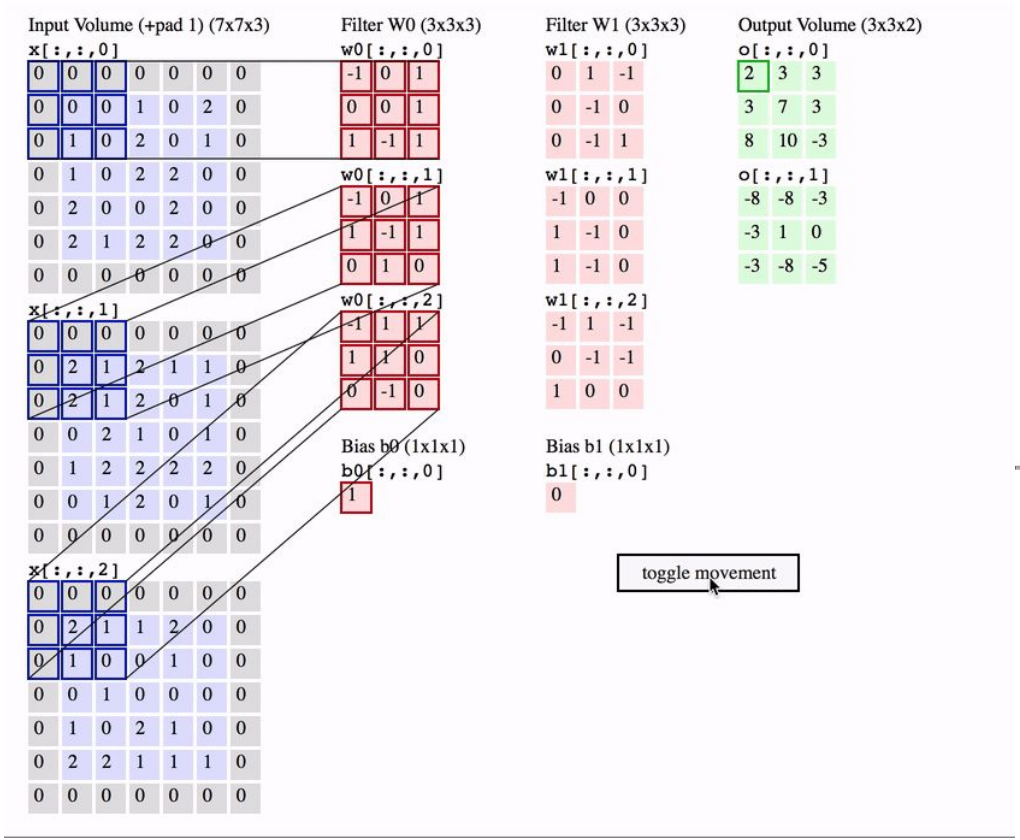
Training The Network

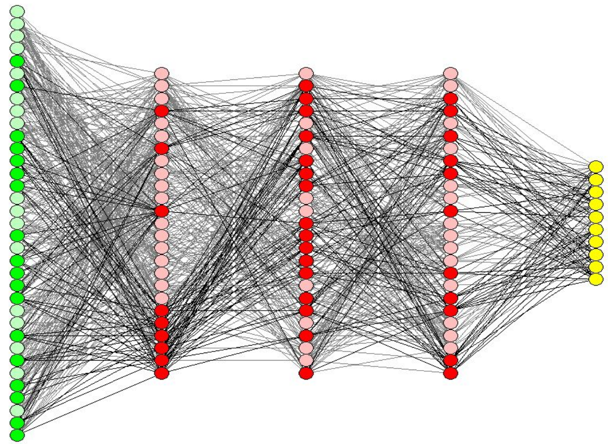
- Most stocks on the market have a degree of built in reflexivity.

- Our baseline idea is to let an extremely dense neural net extract a plethora of features from a daily priceline index of one stock.

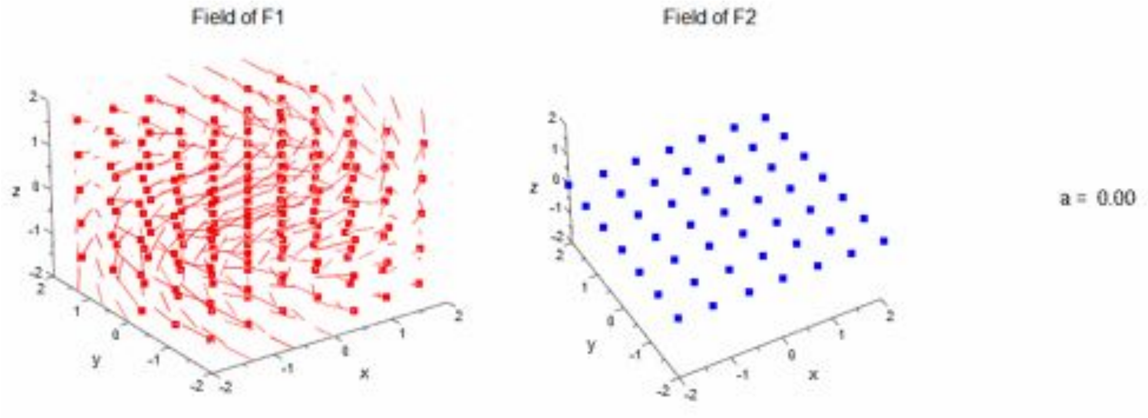


Training The Network

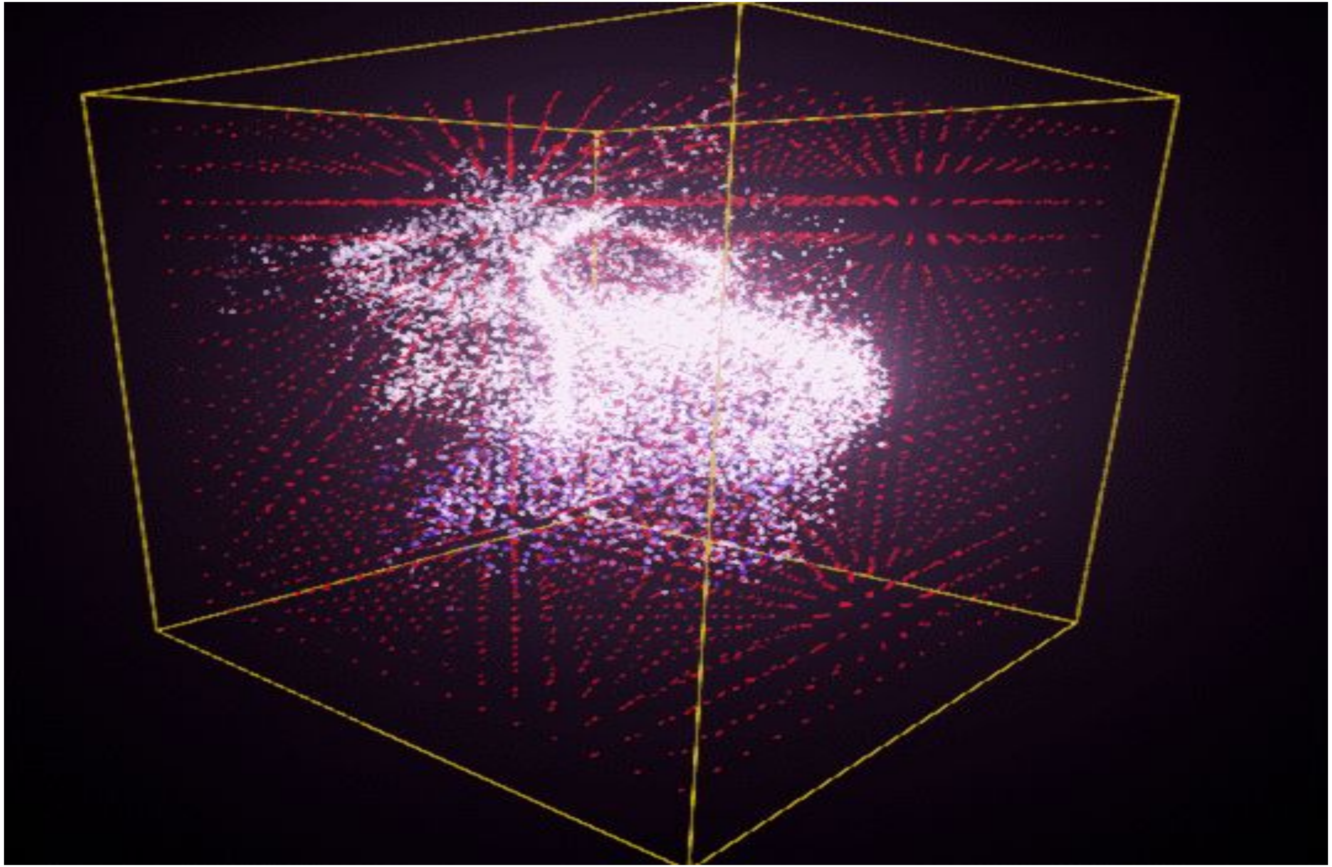




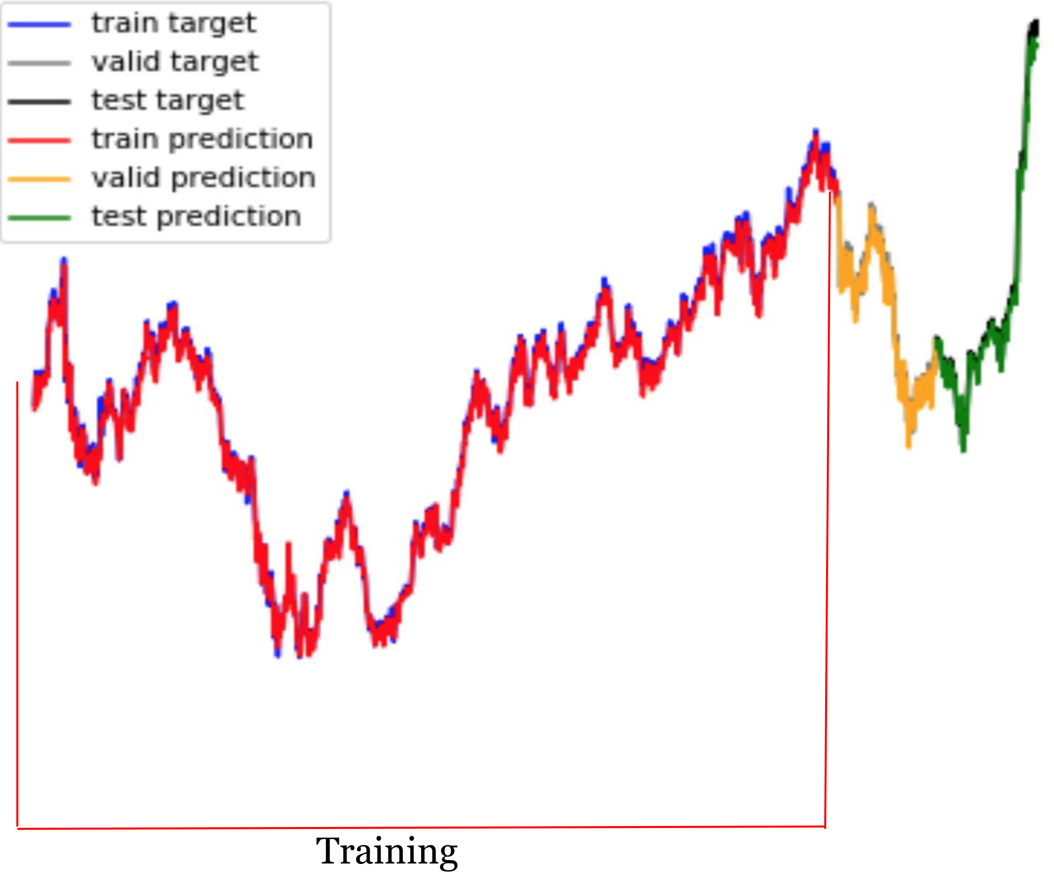
Static vs Dynamic Field



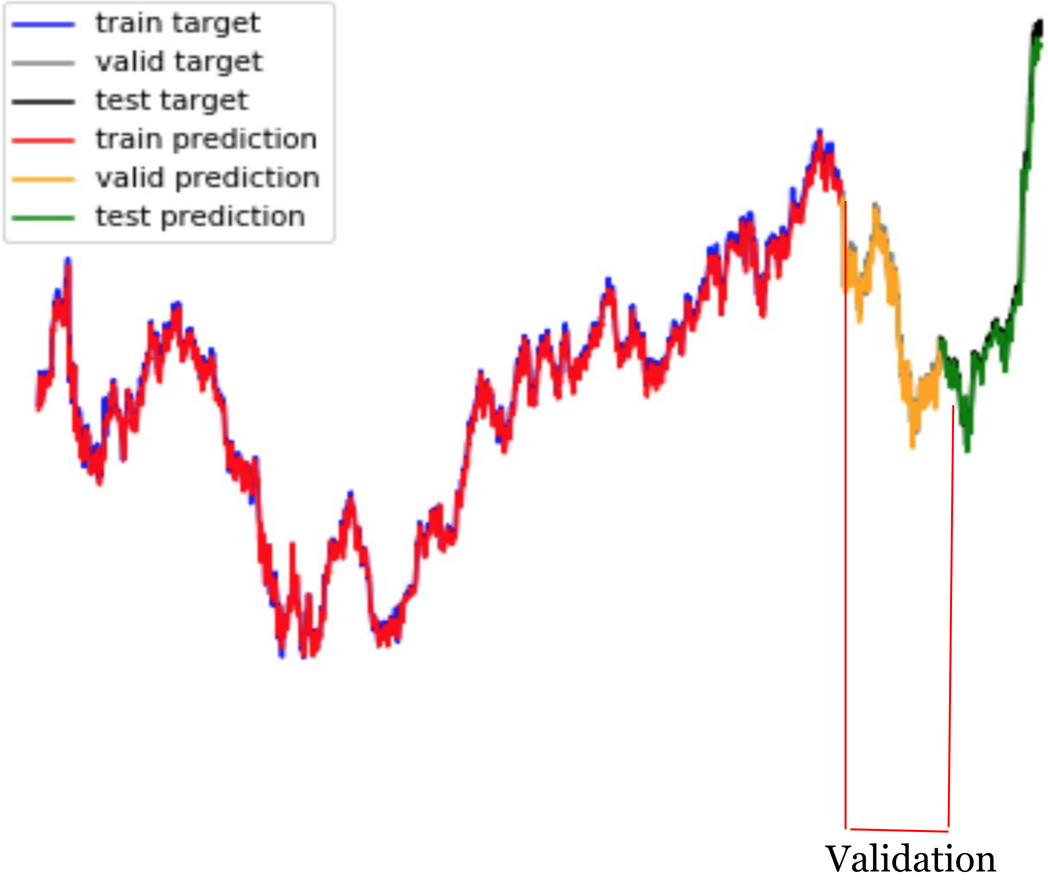
Capturing a Dynamic Field



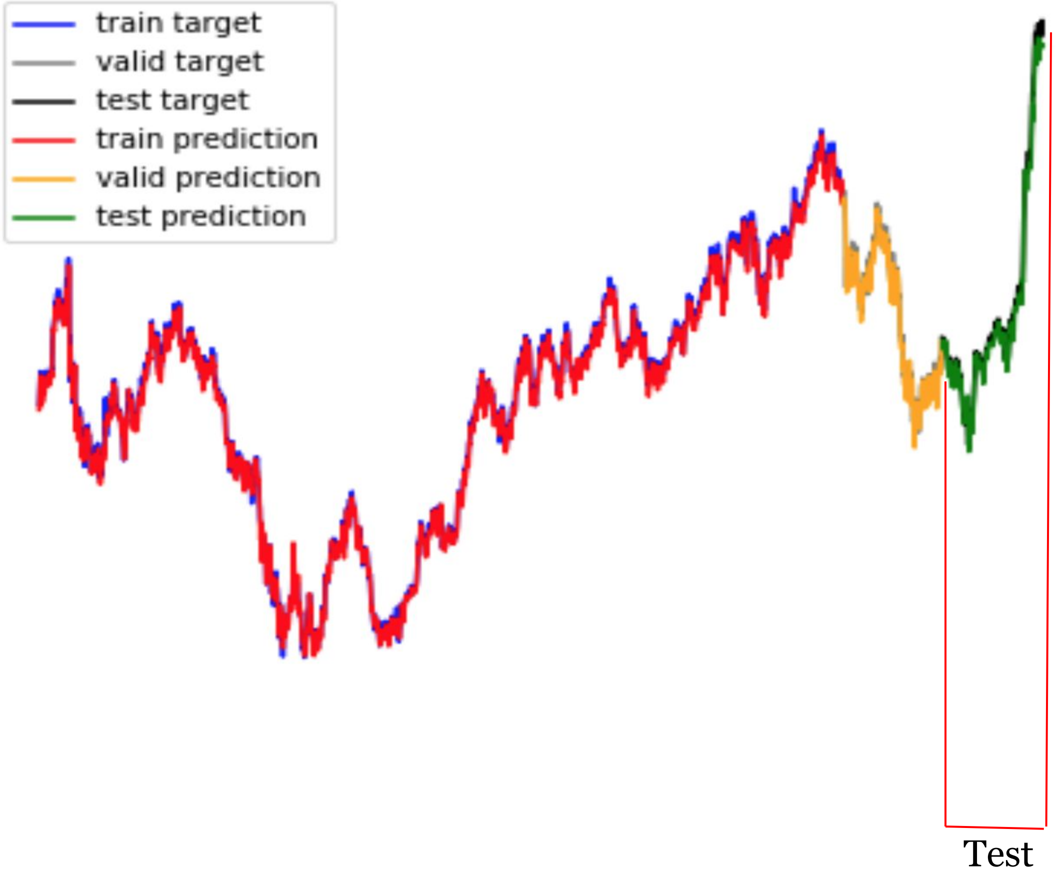
LSTM Prediction 61.00% - MOE 4.00%



LSTM Prediction 61.00% - MOE 4.00%



LSTM Prediction 61.00% - MOE 4.00%

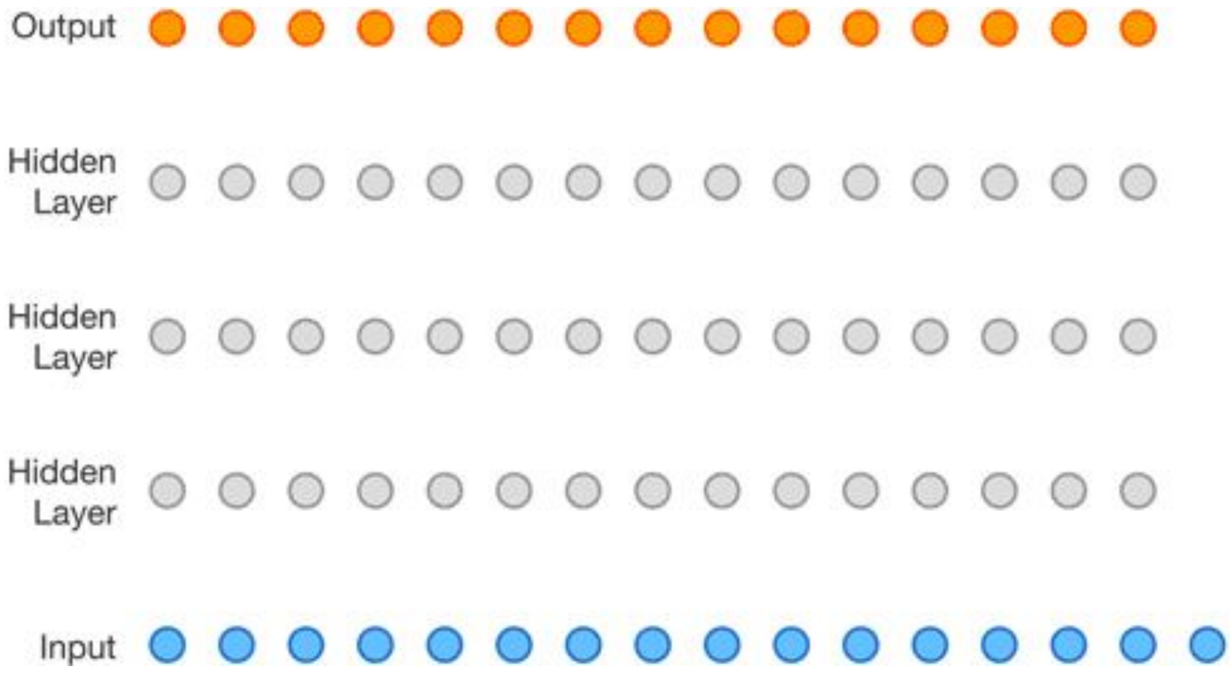


Stanford University

LSTM Prediction 61.00% - MOE 4.00%

```
0.00 epochs: MSE train/valid = 0.196804/0.297400
4.99 epochs: MSE train/valid = 0.001803/0.002345
9.97 epochs: MSE train/valid = 0.001369/0.003114
14.96 epochs: MSE train/valid = 0.000522/0.000866
19.94 epochs: MSE train/valid = 0.000516/0.001022
24.93 epochs: MSE train/valid = 0.000421/0.000620
29.91 epochs: MSE train/valid = 0.000290/0.000540
34.90 epochs: MSE train/valid = 0.000246/0.000444
39.89 epochs: MSE train/valid = 0.000335/0.000790
44.87 epochs: MSE train/valid = 0.000265/0.000452
49.86 epochs: MSE train/valid = 0.000231/0.000393
54.84 epochs: MSE train/valid = 0.000162/0.000397
59.83 epochs: MSE train/valid = 0.000198/0.000494
64.81 epochs: MSE train/valid = 0.000171/0.000334
69.80 epochs: MSE train/valid = 0.000199/0.000415
74.78 epochs: MSE train/valid = 0.000176/0.000423
79.77 epochs: MSE train/valid = 0.000145/0.000329
84.76 epochs: MSE train/valid = 0.000167/0.000352
89.74 epochs: MSE train/valid = 0.000172/0.000391
94.73 epochs: MSE train/valid = 0.000156/0.000337
99.71 epochs: MSE train/valid = 0.000147/0.000328
```

LSTM to produce future prediction



Moving Forward

1) Breakdown Signals Analysis Using Neural Network.

2) Bayesian Neural Network.