Mini-Project (ML for Time Series) - MVA 2021/2022
On the paper titled:
"**Multivariate singular spectrum analysis: A general view and new vector forecasting approach**" of Hossein Hassani & Rahim Mahmoudvand

1 Introduction and contributions

Multivariate Singular Spectrum Analysis (MSSA) is a robust technique for analyzing time series without any assumptions about the underlying system. MSSA can be used to solve many problems like extraction of seasonality components, smoothing the signal and forecasting a multivariate time series. It aims to compose the signal into three components of trend, harmonic, and noise. It then reconstructs the series, using the estimated trend and harmonic components and computes the forecasts based on the reconstructed series.

In this project, we use the main findings of [1] to forecast multivariate time series. We chose to forecast two datasets, the first is simulated data consisting of a direct component, a linear trend, a harmonic seasonality component, and an additive white noise, whereas the second is the daily price of the 14 most popular Crypto-currency Assets.

- Re-partition of Work: We worked jointly on the project, almost all steps of the implementation and writing the report was done at refined by us at the same time.
- Source of available source code:
  - We didn't use any available source code.
  - Percentage of the source code that has been reused: Almost 0% (except for the packages needed)
- Experiments: We experimented on two different data sets, toy data generated as described in the data section of the report and the Crypto data that are known to contain a lot of noise, we wanted to see the effect of highly noised data on this method. Normally (M)SSA is robust and requires no normalization of the data and the results of the analysis of the prices of the Crypto data were not as great as we expected, had we had more time to work on the project we would have inspected why, and perhaps use the daily return (or log return) because they are more stable as features.
- Improvement on the original method: Not much improvement on the original method, we would like to mention that the original paper contained lots of errors and typing mistakes that we needed more time to fix for the implementation to work correctly. Some results of the paper about choosing the optimal r were very ambiguous and hard to interpret.

2 Method

Let us consider M time-series with none-necessarily equal lengths N_i ; $Y_{N_i}^{(i)} = (y_1^{(i)}, ..., y_{N_i}^{(i)})$, $(i = 1, ..., M)$. In the following sub-sections, we depicts the stages and steps of a MSSA algorithm as it was described in the studied paper [1], the uni-variate form can be acquired then by setting M = 1 for all multivariate algorithms considered in [1].

The (M)SSA technique consists of two complementary stages: decomposition and reconstruction.

2.1 Stage I. Decomposition

2.1.1 Step 1: Embedding

Consider one-dimensional time series $Y_{N_i}^{(i)} = (y_1^{(i)}, ..., y_{N_i}^{(i)})$ with length N_i. let $2 \leq L_i \leq N_i$ be a window length and $K_i = N_i - L_i + 1$. Embedding is a mapping that transfers a time series $Y_{N_i}^{(i)}$ into the trajectory matrix $\mathrm{X}^{(i)} = [X_1^{(i)}, \cdots, X_{K_i}^{(i)}] = \begin{pmatrix} y_1^{(i)} & y_2^{(i)} & \cdots & y_{K_i}^{(i)} \\ y_2^{(i)} & y_3^{(i)} & \cdots & y_{K_i+1}^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ y_{L_i}^{(i)} & y_{L_i+1}^{(i)} & \cdots & y_{N_i}^{(i)} \end{pmatrix} \in \mathbb{R}^{L_i \times K_i}$, which is a Hankel matrix. Applying this method for a M multivariate time series with different lengths N_i with windows lengths L_i ($L_{sum} = \sum_{i=1}^{M} L_i, K_{sum} = \sum_{i=1}^{M} K_i$), provides us with a trajectory matrix:

- For vertical form (V MSSA): $X_V = \begin{bmatrix} X^{(1)} \\ \vdots \\ X^{(M)} \end{bmatrix} \in \mathbb{R}^{L_{sum} \times K}$ s.t $K = K_1 = \cdots = K_M$.

- For horizontal form (H MSSA): $\mathrm{X}_H = \left[X^{(1)} : \cdots : X^{(M)}\right] \in \mathbb{R}^{L \times K_{sum}}$ s.t $L = L_i = \cdots = L_M$.

We get:

$$X_V X_V = \begin{bmatrix} X^{(1)}X^{(1)T} & X^{(1)}X^{(2)T} & \ldots & X^{(1)}X^{(M)T} \\ X^{(2)}X^{(1)T} & X^{(2)}X^{(2)T} & \ldots & X^{(2)}X^{(M)T} \\ \vdots & \vdots & \ddots & \vdots \\ X^{(M)}X^{(1)T} & X^{(M)}X^{(2)T} & \ldots & X^{(M)}X^{(M)T} \end{bmatrix}, X_H X_H = X^{(1)}X^{(1)T} + \cdots + \mathrm{X}^{(M)}X^{(M)T}.$$

2.1.2 Step 2: Singular Value Decomposition (SVD).

The SVD of X_V can be written as $X_V = X_{V_1} + + X_{V_{L_{sum}}}$, where $X_{V_i} = \sqrt{\lambda_{V_i}} U_{V_i} V_{V_i}^T$ and $V_{V_i} = X_V^T U_{V_i} / \sqrt{\lambda_{V_i}} (X_{V_i} = 0 if \lambda_{V_i} = 0)$. Where $\lambda_{V_1} \geq \cdots \geq \lambda_{V_{L_{sum}}} \geq 0$ denote the eigenvalues and $U_{V_1}, \cdots, U_{V_{L_{sum}}}$ denote the eigenvectors of $X_V X_V^T$.

2.2 Stage II. Reconstruction

2.2.1 Step 1: Grouping.

Grouping is the procedure of splitting the set of indices $\{1, ..., L_{sum}\}$ into disjoint subsets $I_1, \cdots, I_m$ corresponding to the representation $X_V = X_{I_1} + \cdots + X_{I_m}$. The contribution of the component X_{V_I} is measured by the share of the corresponding eigenvalues: $\sum_{i \in I} \lambda_{V_i} / \sum_{i=1}^{rank(X_V)}$.

2.2.2 Step 3: Reconstructed Matrix and VMSSA recurrent forecasting algorithm (VMSSA-R).

Let $U_{V_j} = (U_j^{(1)}, \cdots, U_j^{(M)})^T$ be the j-th eigenvector of $X_V X_V^T$, where $U_j^{(i)}$ with length L_i corresponds to the series $Y_{N_i}^{(i)}$. The reconstructed matrix achieved from a chosen r eigentriples is by:

$$\hat{X}_V = \left[\hat{X}^{(1)} : \cdots : \hat{X}^{(K)}\right] = \sum_{i=1}^{r} U_{V_i} U_{V_i}^T X_V$$

- In order to convert the reconstructed matrix, $\hat{X}_{V_i}$ to a time series we use diagonal-averaging (or Hankelization) over the anti-diagonalss' $i + j = k + 1$ to convert it to a Hankel matrix. Consider matrix $\tilde{X}^{(i)} = \mathcal{H}\hat{X}^{(i)} (i = 1, ..., M)$ as the result of the Hankelization procedure of the matrix $\hat{X}^{(i)}$ obtained from the previous step, where $\mathcal{H}$ is a Hankel operator.

- Let $U_j^{(i)\nabla}$ be the vector of the first $L_i - 1$ components of the vector $U_j^{(i)}$ and $\pi_j^{(i)}$ be the last component of the vector $U_j^{(i)}$.

- For a chosen r we define the matrix $U^{\nabla M} = (U_1^{\nabla M}, \cdots, U_r^{\nabla M})$ where $U_j^{\nabla M} = \begin{bmatrix} \mathrm{U}_j^{(1)\nabla} \\ \vdots \\ \mathrm{U}_j^{(M)\nabla} \end{bmatrix}$.

- Let W be the matrix $(\pi_j^{(i)})_{1 \le i \le r, 1 \le j \le M}$.

- If the matrix $(I_M - WW^T)^{-1}$ exists and if if $r \le L_{sum} - M$, we define the **h-step ahead VMSSA forecasts** as follows:

  $$\left[\hat{y}_{j_1}^{(1)}, \cdots, \hat{y}_{j_M}^{(M)}\right]^T = \begin{cases} \left[\tilde{y}_{j_1}^{(1)}, \cdots, \tilde{y}_{j_M}^{(M)}\right]^T, & J_i = 1, \cdots, N_i \\ (I_M - WW^T)^{-1} W U^{\nabla M T} Z_h, & J_i = N_i + 1, \cdots, N_i + h \end{cases}$$

  such that $Z_h = \left[Z_h^{(1)}, \cdots, Z_h^{(M)}\right]$ and $Z_h^{(i)} = \left[\hat{y}_{N_i - L_i + h + 1}^{(i)}, \cdots, \hat{y}_{N_i + h + 1}^{(i)}.\right]$

2.2.3 Step 3: Reconstructed Matrix for HMSSA.

The reconstruction of the trajectory matrix for the HMSSA, follows the same previous rules. The only difference is that for the HMSSA, we choose the same L_i and the trajectory matrix is different. More details about this will be found in the HMSSA-R forecasting algorithm.

2.3 Recurrent forecasting algorithm

The purpose of this section is to describe the way we forecast the multivariate time series through the reconstructed matrix.

2.3.1 HMSSA recurrent forecasting algorithm: HMSSA-R

1. For a fixed L, we construct the trajectory matrix $X^{(i)}$ of the signal $Y_{N_i}^{(i)}$ as previously described.

2. Construct the block trajectory matrix: $X_H = \left[X^{(1)} : \cdots : X^{(M)}\right] \in \mathbb{R}^{L \times K_{sum}}$.

3. Let $U_{H_k} = (u_{1j}, \cdots, u_{Lj})^T$ with length L be the j-th eigenvector of $X_H X_H^T$.

4. Consider $\hat{X}_H = \sum_{i=1}^{r} U_{H_i} U_{H_i}^T X_V$ for a chosen r.

5. Consider matrix $\tilde{X}^{(i)} = \mathcal{H}\hat{X}^{(i)} (i = 1, ..., M)$ as the result of the Hankelization of $\hat{X}^{(i)}$.

6. Let $U_{H_j}^{\nabla}$ be the vector of the first $L-1$ components of the vector U^{H_j} and π_{H_j} be the last component of the vector U^{H_j}.

7. We define $v^2 = \sum_{j=1}^{r} {\pi_{H_j}}^2$.

8. Denote the linear Coefficients vector $\mathcal{R}$ as follows: $\mathcal{R} = \frac{1}{1-v^2} \sum_{j=1}^{r} \pi_{H_j} U_{H_j}^{\nabla}$.

9. If $v^2 < 1$, the h-step ahead HMSSA forecasts exists and is calculated by the following formula:

$$\left[\hat{y}_{j_1}^{(1)}, \cdots, \hat{y}_{j_M}^{(M)}\right]^T = \begin{cases} \left[\tilde{y}_{j_1}^{(1)}, \cdots, \tilde{y}_{j_M}^{(M)}\right]^T, & J_i = 1, \cdots, N_i \\ \mathcal{R}^T Z_h, & J_i = N_i + 1, \cdots, N_i + h \end{cases}$$

such that $Z_h = \left[Z_h^{(1)}, \cdots, Z_h^{(M)}\right]$ and $Z_h^{(i)} = \left[\hat{y}_{N_i - L_i + h + 1}^{(i)}, \cdots, \hat{y}_{N_i + h + 1}^{(i)}.\right]$ and $i = 1, \cdots, M$

2.4 New MSSA Forecasting Procedure

2.4.1 HMSSA Vector forecasting algorithm (HMSSA-V)

Let's consider the vector $H\{_H$, v^2 and $\mathcal{R}$ defined in section 2.3.1.

1. Consider $\Pi = H\{_H \mathrm{H}\{_H{}^T + (1 - v^2)\mathcal{R}\mathcal{R}^T$.

2. Consider the operator $\mathcal{P}^{(v)} : \mathcal{L}_r \to \mathbb{R}^L$ such that for $Y \in \mathcal{L}_r, \mathcal{P}^{(v)} Y = (\Pi Y_\Delta, \mathcal{R}^T Y_\Delta)^T$

3. Define the vector $Z_j^{(i)}$ $(i = 1, \cdots, M)$ such that
$$Z_j^{(i)} = \begin{cases} \tilde{X}_j^{(i)}, & j = 1, \cdots, K_i \\ \mathcal{P}^{(v)} Z_{j-1}^{(i)}, & j = K_i + 1, \cdots, K_i + h + L - 1 \end{cases}$$

4. Construct $Z^{(i)} = \left[Z_1^{(i)}, \cdots, Z_{K_i+h+L-1}^{(i)}\right]$ and perform diagonal averaging to obtain the series $\hat{y}_1^{(i)}, \cdots, \hat{y}_{N_i+h+L-1}^{(i)}$. Notice that $\hat{y}_{N_i+1}^{(i)}, \cdots, \hat{y}_{N_i+h}^{(i)}$ provides the h-step ahead of HMSSA-V forecast.

2.4.2 VMSSA Vector forecasting algorithm (VMSSA-V)

VMSSA-V forecasting algorithm:

1. Define $\mathcal{P}^{(v)} : \mathcal{L}_r \to \mathbb{R}^{L_{sum}-M}$ such that

$$\mathcal{P}^{(v)}Y = \begin{bmatrix} \Pi^{(1)}Y_\Delta \\ \mathcal{R}^{(1)T}Y_\Delta) \\ \vdots \\ \Pi^{(M)}Y_\Delta \\ \mathcal{R}^{(M)T}Y_\Delta) \end{bmatrix}$$ where $Y_\Delta^T = (Y_\Delta^{(1)}, \cdots, Y_\Delta^{(M)}$ such that $Y_\Delta^{(i)}$ is the last $L_i - 1$ element of entities Y_i with length L_i.

2. Define the vector $Z_j^{(i)}$ $(i = 1, \cdots, M)$ such that
$$Z_j^{(i)} = \begin{cases} \tilde{X}^i, & i = 1, \cdots, K \\ \mathcal{P}^{(v)}Z_{i-1}^{(i)}, & i = K+1, \cdots, K+h+L_{max}-1 \end{cases}$$

3. Construct the matrix $Z = [Z_1 : \cdots : Z_{K+h+L_{max}-1}]$ and make its hankelization to obtain to obtain the series $\hat{y}_1^{(i)}, \cdots, \hat{y}_1^{(i)}{}_{N_i+h+L_{max}}$, for $i = 1, \cdots, M$.

4. The numbers $\hat{y}^(i)_{N_i+1}, \cdots, \hat{y}_{N_i+h}^{(i)}$ for $i = 1, \cdots, M$ form the h-step ahead *VMSSA-V* forecasts.

Remark: VMSSA-R and VMSSA-V forecasts are unique.

2.5 Optimal Values of L and r

Assuming that the M-multivariate series have the same length N and same window shape L, we can get precise values for L. As specified in the paper, it is now possible to choose the proper value of L such that $L = [\frac{N+1}{2}]$, $L = [\frac{N+1}{M+1}]$ and $L = [\frac{M}{M+1}(N+1)]$ for SSA, VMSSA and HMSSA, respectively.

The choice of the optimal r remains purely heuristically, the paper provided no theoretical values for the optimal r. The optimal r depends on the signals.

1. Choosing a small r leads to a loss of precision as parts of the signals in all series will be lost.

2. Choosing a big r includes the noise in the reconstructed series.

3. For highly interrelated series sharing several common components, it's better to choose a small r.

4. Choosing a relatively big r is necessary when the series analyzed have very little relation to each other.

3 Data

As the authors of [1] did, we tested the four versions of the algorithm of interest on two types of data sets. The first consists of a simulated signal that can be decomposed as:

$$y_t^{(S)} = A^{(S)} + B^{(S)}t + C^{(S)}sin(2\pi f^{(S)}t) + D^{(S)}\varepsilon_t$$

Where $\varepsilon_t \sim \mathcal{N}(0,1)$, and where we have simulated two signals with the following parameters (see figure 1 for the plots):

$$(A^{(S1)}, B^{(S1)}, C^{(S1)}, f^{(S1)}, D^{(S1)}) = (5.0, 0.01, 1.0, \frac{1}{5.0}, 0.1)$$

$$(A^{(S2)}, B^{(S2)}, C^{(S2)}, f^{(S2)}, D^{(S2)}) = (5.0, 0.01, 1.0, \frac{1}{7.0}, 0.1)$$

Whereas the second one is a Kaggle Crypto's dataset that we have preprocessed in order to make it computationally friendly (see figure 4 for some signal examples).

4 Results

As we couldn't find any working implementation of any of the previously described versions of MSSA, we have coded the latter as learning classes **VMSSAR**, **HMSSAR**, **VMSSAV**, and **HMSSAV** that can be trained then used to forecast new values of the fitted data sets.

Figures A.2, 5 show forecasting results for both the simulated and Crypto's data sets. One can observe that the forecasting results on the simulated data set are almost perfect, with a mean-square-error of the order of 10^{-4} for all versions of the MSSA algorithm, conversely to the Crypto's data set on which the forecasting is far from accurate. This being said, the four versions of the algorithm were able to give some hints about the direction of evolution of the assets' price.

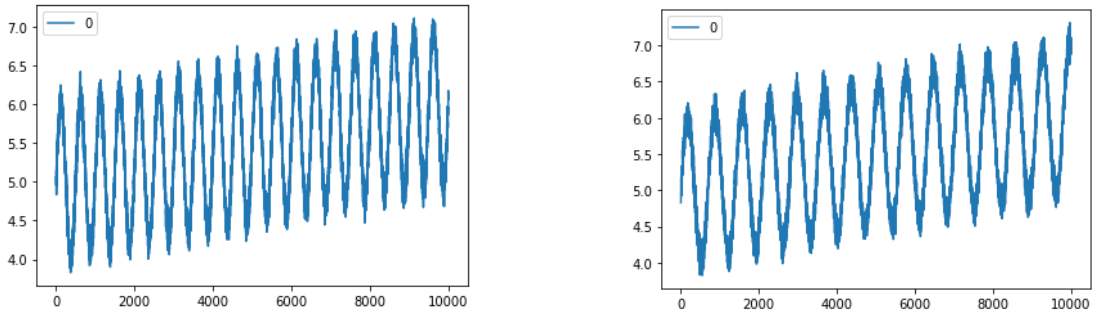
Finally, tables 1, 2 summerize forecasting test mean-square-error and R^2 of the different versions of the studied algorithm on the different used Crypto's signal. The results of the first table show that MSSA doesn't work as good as it does on real-world signals, while the negativity of all R^2 factors that can be observed in the second table proves, once again, that it is (almost) impossible beat the constant predictor when considering financial data, a fact that is widely known as the *Principle of no-arbitrage opportunity*.

References

[1] HOSSEIN HASSANI and RAHIM MAHMOUDVAND. *MULTIVARIATE SINGULAR SPECTRUM ANALYSIS: A GENERAL VIEW AND NEW VECTOR FORECASTING APPROACH*. Mar. 2013. DOI: 10.1142/s2335680413500051. URL: http://dx.doi.org/10.1142/S2335680413500051.

A Appendix

A.1 Simulated data set

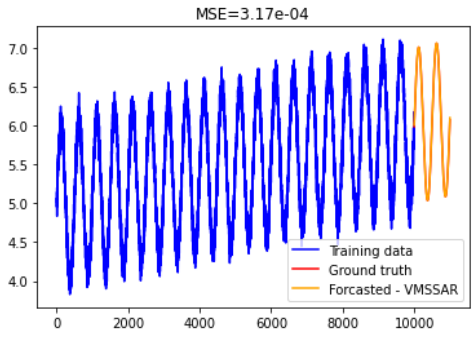


(a) First signal of the simulated data set.

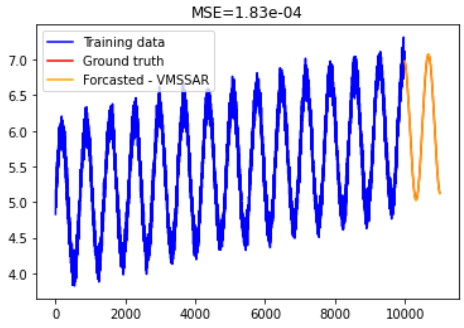
(b) Second signal of the simulated data set.

Figure 1: Simulated data set.

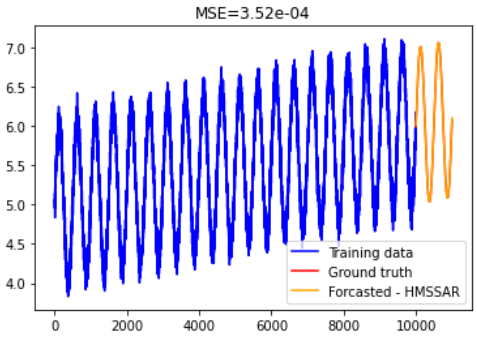
A.2 Forecasting results in the simulated data set



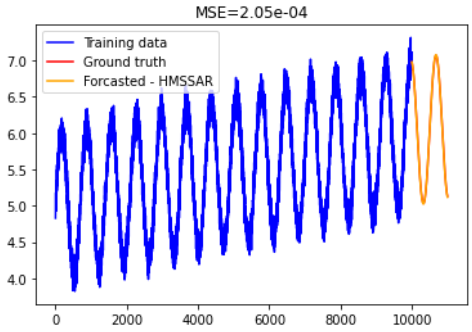
(a) VMSSA-R on the first signal.



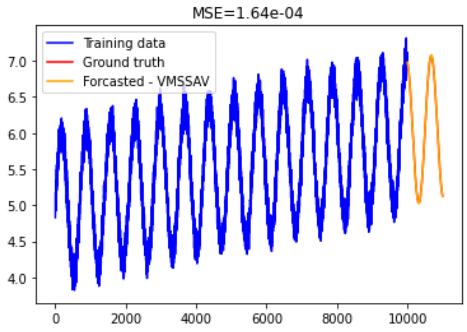
(b) VMSSA-R on the second signal.



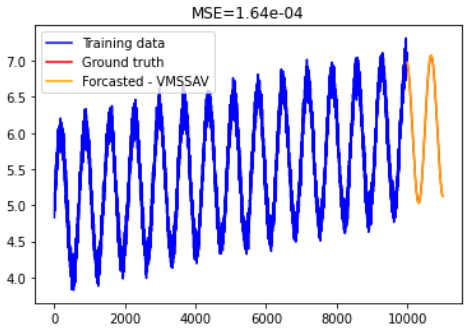
(c) HMSSA-R on the first signal.



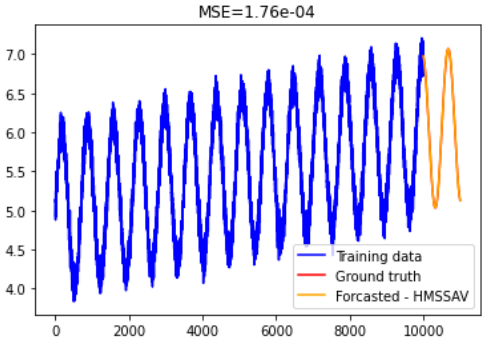
(d) HMSSA-R on the second signal.



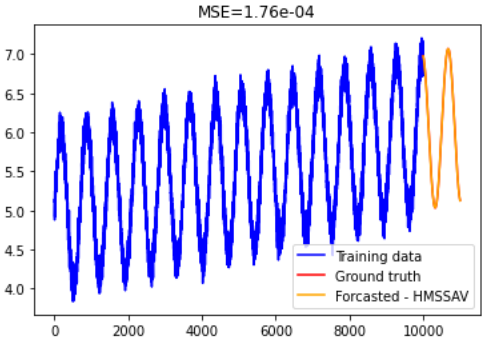
(a) VMSSA-V on the first signal.



(b) VMSSA-V on the second signal.



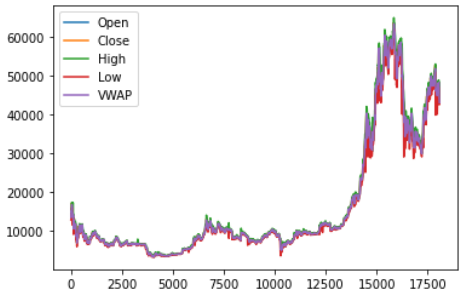
(c) HMSSA-V on the first signal.



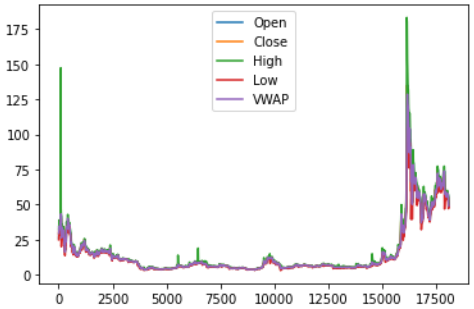
(d) HMSSA-V on the second signal.

Figure 3: MSSA the two signals of the simulated dataset.

A.3 Examples of signals from the Crypto's dataset



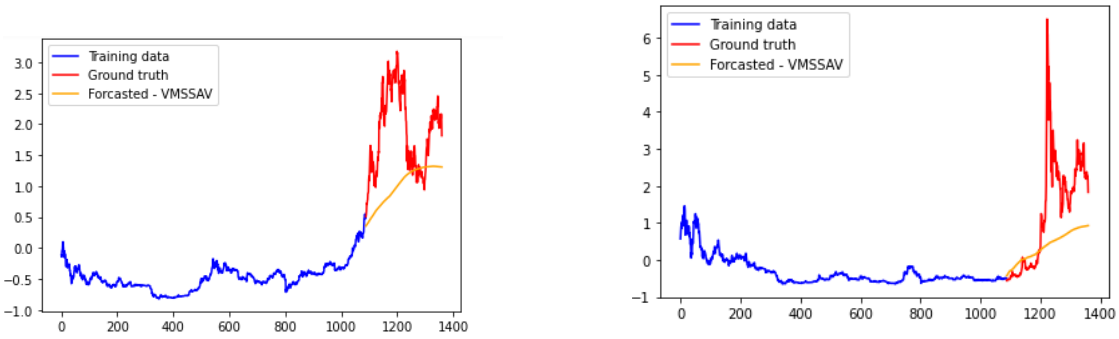
(a) Bitcoin related signals.



(b) Ethereum related signals.

Figure 4: Examples of signals from the Crypto's dataset.

A.4 Examples of forecasting results on the Crypto's dataset



(a) Bitcoin related signals.

(b) Ethereum related signals.

Figure 5: Examples of forecasting results on the Crypto's dataset.

A.5 Test MSE's of the different models

	BTC	EOS	ETH
HMSSA-R	0.23	1.08	6.55
VMSSA-R	0.23	1.29	6.50
HMSSA-V	0.14	1.19	5.72
VMSSA-V	0.26	1.12	4.60

Table 1: Forcasting test mean-square-error for the different versions of the algorithm on the different used Crypto's signal.

A.6 Test R^2's of the different models

	BTC	EOS	ETH
HMSSA-R	-1.50	-185.23	-1846.29
VMSSA-R	-1.47	-219.57	-1832.92
HMSSA-V	-0.54	-205.57	-1641.12
VMSSA-V	-1.44	-192.04	-1324.98

Table 2: Forcasting test R^2 for the different versions of the algorithm on the different used Crypto's signal.