Prediction after a Horizon of Predictability: Non-predictable Points and Partial Multistep Prediction for Chaotic Time Series

Authors: Vasilii A. Gromov & Philip S. Baranov
Presented by: Marwan Bouabidi

Introduction

- Chaotic Time Series: what are they?
  - Representing evolution of chaotic systems with respect to time.
  - Examples: stocks, temperature, heart beat, etc.
- Lyapunov instability: $\boldsymbol{\varepsilon(t) = \varepsilon(0)e^{\lambda t}, \lambda>0}$
- Horizon of predictability: $\boldsymbol{T \sim 1/\lambda \ln(\varepsilon_{max}/\varepsilon_0)}$
- How to predict the evolution of these unstable TS? A prediction beyond **T** possible?

The research:

- Main characteristics:
  - Deals with a chaotic time series.
  - Multistep prediction: **h steps** into the future after the last known observation **t. But Only based on the t first observations!**
  - Partial prediction: not all points are predicted!
  - Non-predictable points: some predictions are rejected!
  - Assumption: the TS evolves around a strange attractor.

Data

- A chaotic TS divided into 2 parts:
  - Y_1: observations up until a moment **t** in time, the **training set**
  - Y_2: real data to predict, from **t** to **t+h**, the **testing set**
  - **$Y = Y_1$ U Y_2**
- **Y1** must be normalized before use.

Mathematical model: generating the training set

- Patterns:
    - Int vector of steps, size **L-1**: $\alpha=(k_1, k_2, \ldots, k_{L-1})$
    - K_max: limit of the steps
    - Example for L=4: **(2, 3, 4)**
- Motifs:
    - Vectors of observations, size **L**: $C_\alpha=(m^\alpha_1, m^\alpha_2, \ldots, m^\alpha_L)$
    - Many possible motifs from each pattern.
    - Examples (L=4): $(y_0, y_2, y_5, y_9), \ldots, (y_{t-9}, y_{t-7}, y_{t-4}, y_t)$
    - Could be clustered before use (**Wishart clustering**, for every pattern)

Mathematical model: optimization problem

- Minimize:
    - Number of non-predictable points
    - Error for the predictable ones

$$I_1 = \sum_{t+h \in Y_2} \left(1 - \zeta\left(\widehat{S}_{t+h}^{(p)}\right)\right),$$

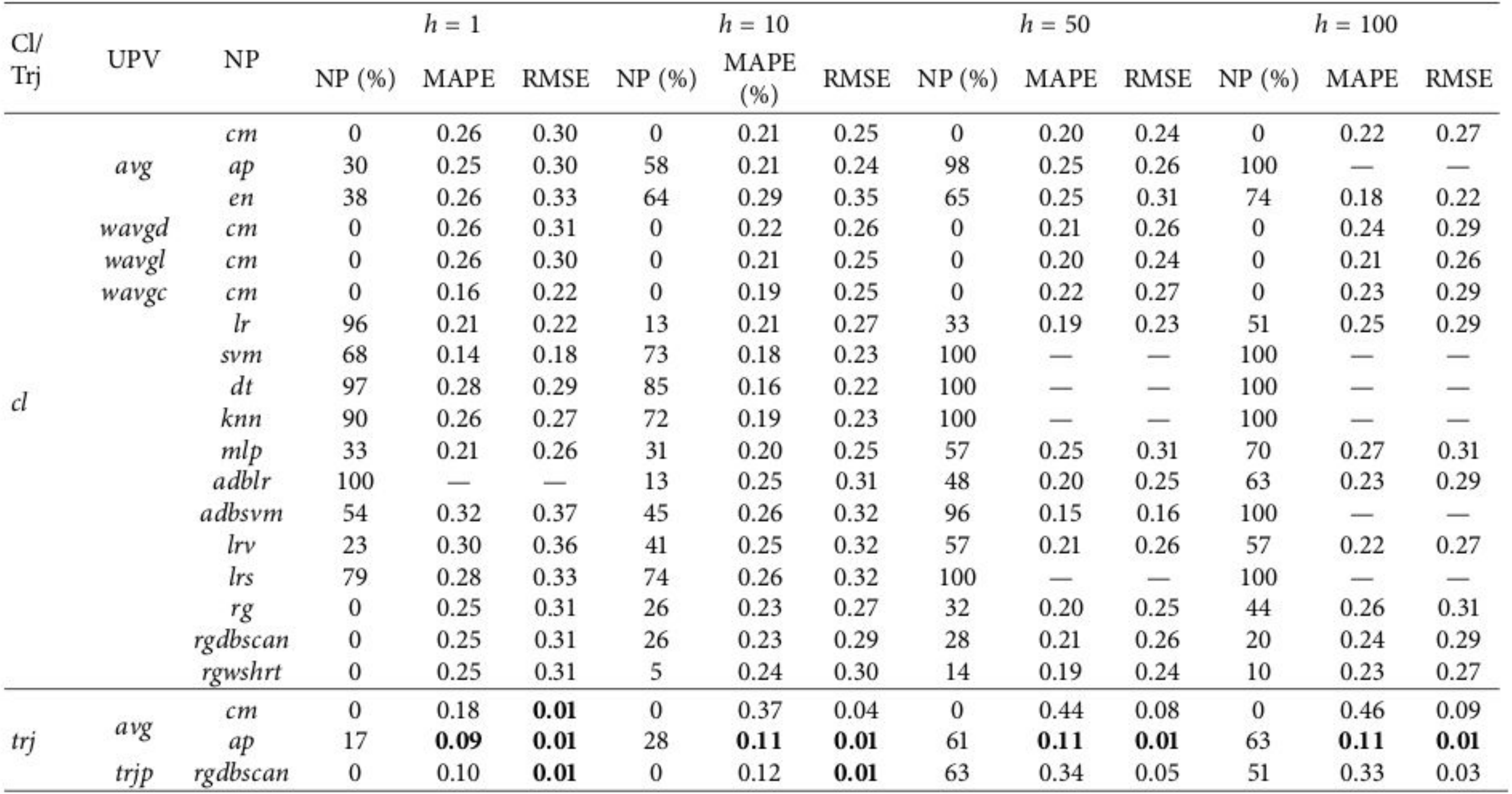
$$I_2 = \frac{1}{\left|Y_2\right|} \sum_{t+h \in Y_2} \zeta\left(\widehat{S}_{t+h}^{(p)}\right) \left\| g\left(\widehat{S}_{t+h}^{(p)}\right) - y_{t+h} \right\|.$$

Mathematical model: prediction algorithm

- Set **ε** to 0.01 and normalize the observations $\mathbf{Y_1}$.
- Generate all patterns based on the values of **L** and $\mathbf{K_{max}}$: α= $(k_1, k_2, ..., k_{L-1})$
- For every pattern α, generate all possible motifs:
- For every point **t<"t+i"<=t+h**:
    - Go through all patterns, and their respective motifs C_α^{trunc}(truncated from its last value), and try to find **similar enough sequences** of data to **C**.
    - $C=(y_{t+i-k1-k2-...-kL-1}, \dots , y_{t+i-kL-1})$: to compare with all C_α
    - If found, $d(C, C_\alpha^{trunc})<\varepsilon$, its last element is a possible prediction value for t+i
- Calculate a **prediction** for **t+i**, **<u>if predictable</u>**, based on the collected values.

Mathematical model: prediction algorithm

- How to calculate a **Unified Prediction value (UPV)** for t+i based on possible values?
  1. **Current position (cp)**: average, weighted average, clustering…
  2. **Prediction Trajectory (trj):** consider all possible predictions of points **t+1** up to **t+i. UPV is the average of the last points of certain chosen trajectory.**
- Ways to identify non-predictable points?
  1. Set of possible prediction values **empty**
  2. Prediction error > a certain threshold **ε**
  3. **Clustering** the set of possible predictions (equally sized, or none at all)



Cl/ Trj	UPV	NP	*h* = 1			*h* = 10			*h* = 50			*h* = 100		
			NP (%)	MAPE	RMSE	NP (%)	MAPE (%)	RMSE	NP (%)	MAPE	RMSE	NP (%)	MAPE	RMSE
cl	*avg*	*cm*	0	0.26	0.30	0	0.21	0.25	0	0.20	0.24	0	0.22	0.27
		ap	30	0.25	0.30	58	0.21	0.24	98	0.25	0.26	100	—	—
		en	38	0.26	0.33	64	0.29	0.35	65	0.25	0.31	74	0.18	0.22
	wavgd	*cm*	0	0.26	0.31	0	0.22	0.26	0	0.21	0.26	0	0.24	0.29
	wavgl	*cm*	0	0.26	0.30	0	0.21	0.25	0	0.20	0.24	0	0.21	0.26
	wavgc	*cm*	0	0.16	0.22	0	0.19	0.25	0	0.22	0.27	0	0.23	0.29
		lr	96	0.21	0.22	13	0.21	0.27	33	0.19	0.23	51	0.25	0.29
		svm	68	0.14	0.18	73	0.18	0.23	100	—	—	100	—	—
		dt	97	0.28	0.29	85	0.16	0.22	100	—	—	100	—	—
		knn	90	0.26	0.27	72	0.19	0.23	100	—	—	100	—	—
		mlp	33	0.21	0.26	31	0.20	0.25	57	0.25	0.31	70	0.27	0.31
		adblr	100	—	—	13	0.25	0.31	48	0.20	0.25	63	0.23	0.29
		adbsvm	54	0.32	0.37	45	0.26	0.32	96	0.15	0.16	100	—	—
		lrv	23	0.30	0.36	41	0.25	0.32	57	0.21	0.26	57	0.22	0.27
		lrs	79	0.28	0.33	74	0.26	0.32	100	—	—	100	—	—
		rg	0	0.25	0.31	26	0.23	0.27	32	0.20	0.25	44	0.26	0.31
		rgdbscan	0	0.25	0.31	26	0.23	0.29	28	0.21	0.26	20	0.24	0.29
		rgwshrt	0	0.25	0.31	5	0.24	0.30	14	0.19	0.24	10	0.23	0.27
trj	*avg*	*cm*	0	0.18	**0.01**	0	0.37	0.04	0	0.44	0.08	0	0.46	0.09
		ap	17	**0.09**	**0.01**	28	**0.11**	**0.01**	61	**0.11**	**0.01**	63	**0.11**	**0.01**
	trjp	*rgdbscan*	0	0.10	**0.01**	0	0.12	**0.01**	63	0.34	0.05	51	0.33	0.03

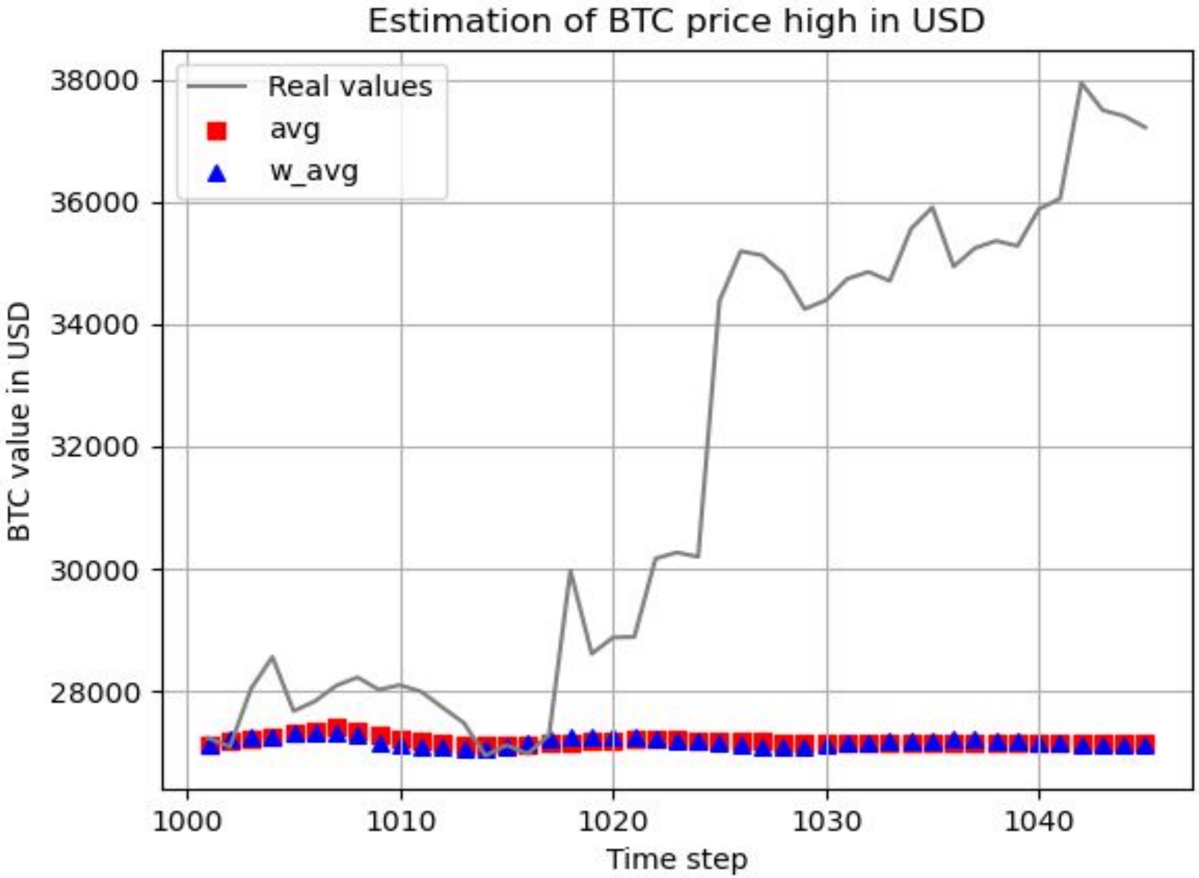
Results: Electric grid load of Germany (hourly) from 2014-12-31 to 2016-02-20

Application

- **Language**: Python
- **Dataset**: BTC daily high in USD from 01.01.2021 to 12.11.2023 (investing.com)
    - Size 1046, t=1000, h=45
- **Parameters**:
    - **ε = 0.01** or **0.001**
    - **$L=4$**, **$K_{max}=10$**, no clustering of the motifs.
    - **UPV for cp**: average and weighted average ($\varepsilon-\rho(C, C_{\square}^{trunc}))/\varepsilon$)
    - Min-max normalization: **(x-min)/(max-min)**
    - Prediction error: $|y_{pred}-y_{real}|/|y_{real}|$

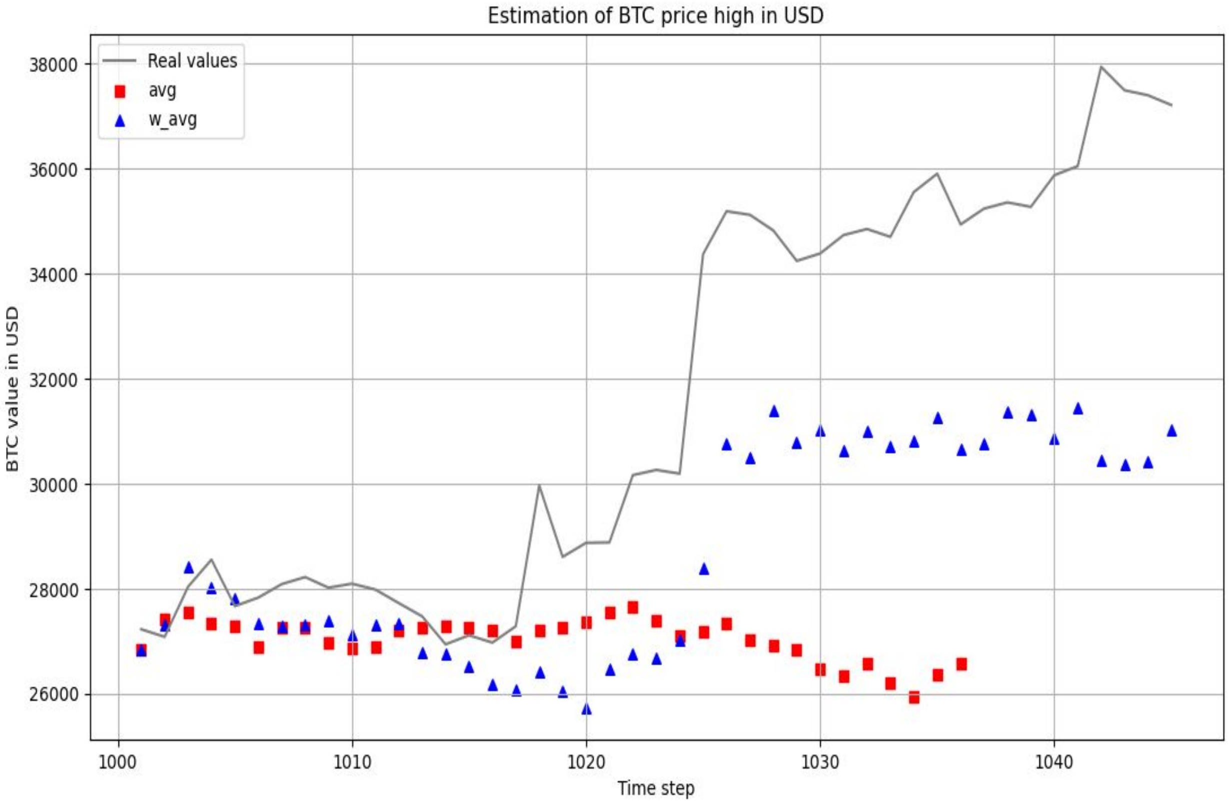
Application

- ε=0.01
- t=1000
- h=45
- NP method 1



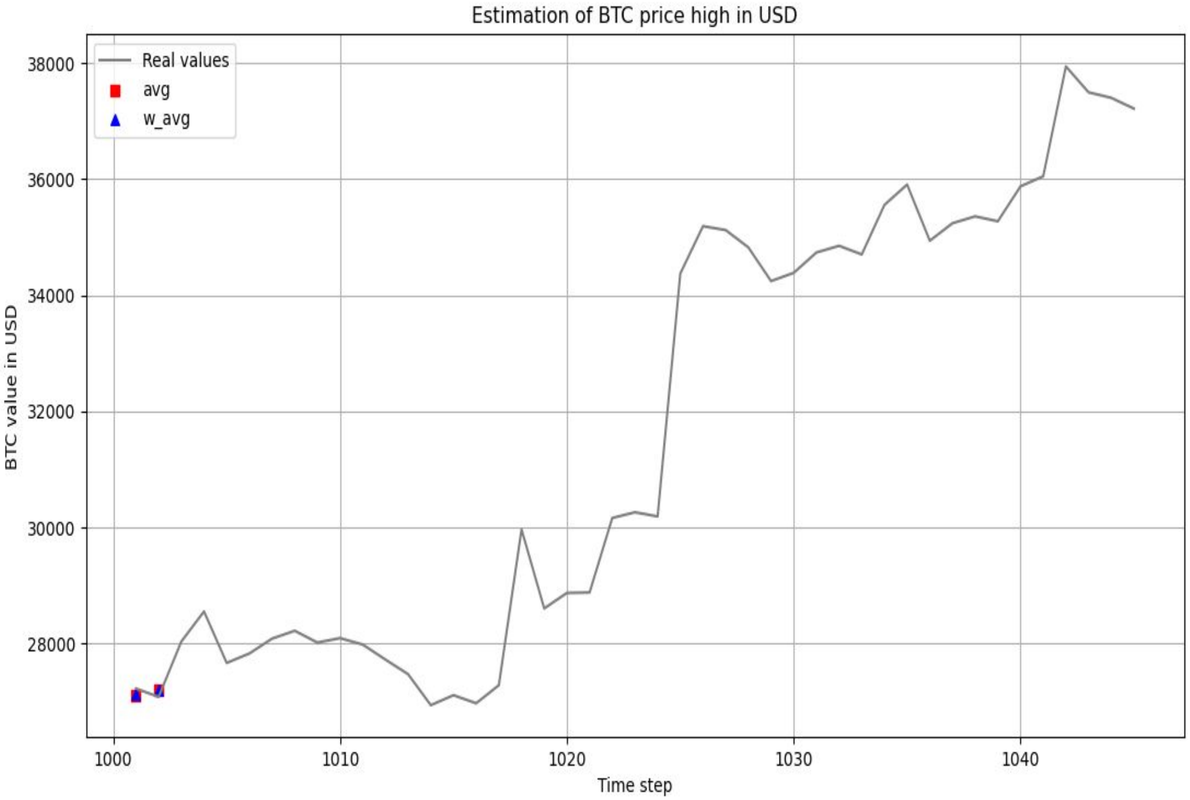
Application

- ε=0.001
- t=1000
- h=45
- NP method 1



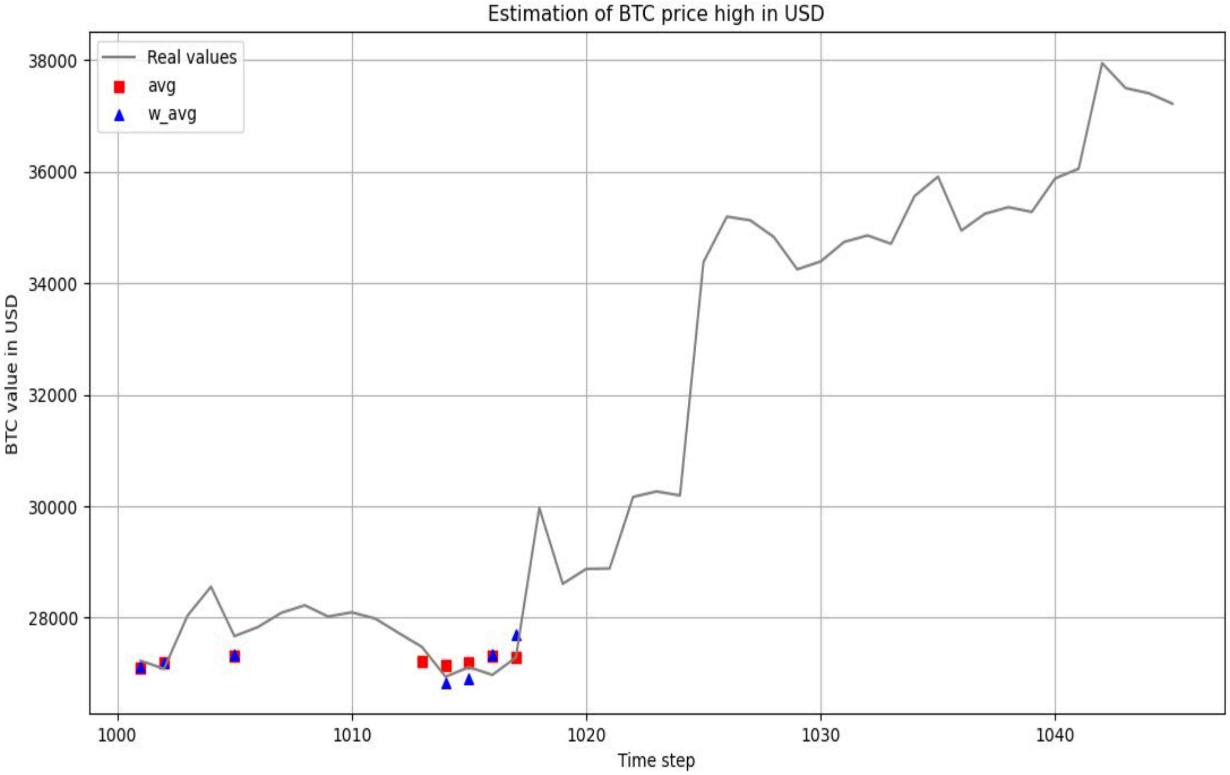
Application

- ε=0.01
- t=1000
- h=45
- NP method 2 (ap)
- ε_{max}=0.01



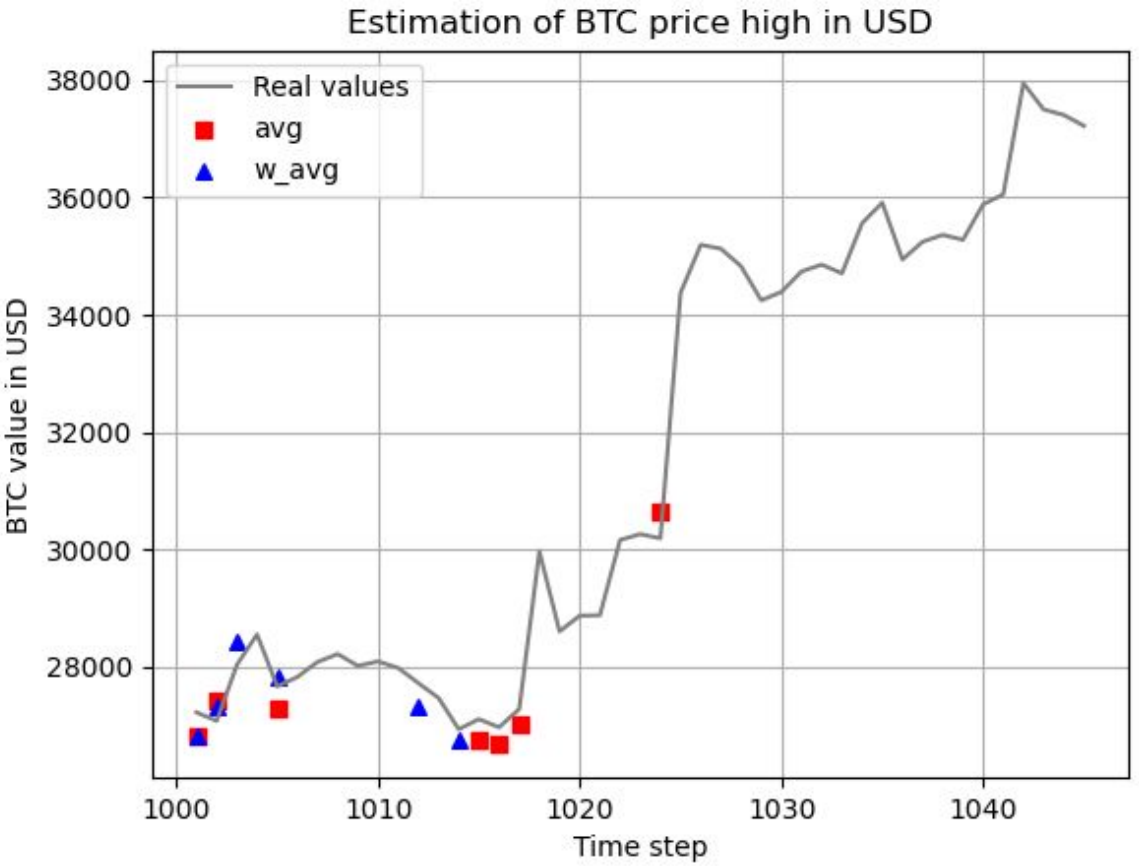
Application

- ε=0.01
- t=1000
- h=45
- NP method 2 + rounded error
- ε_{max}=0.01



Application

- ε=0.001
- t=1000
- h=45
- NP method 2 + rounded error
- ε_{max}=0.01



Conclusion

- An interesting approach to chaotic TS.
- Concept of non-predictable points:
  - Adds more quality to the prediction, but takes away from the quantity.
  - Their choice heavily influences the quality of the prediction.
- Prediction based on possible trajectories more accurate than the one based on current position (goes even beyond T).

Thank you for your attention!