LETS Forecast: Learning Embedology for Time Series Forecasting

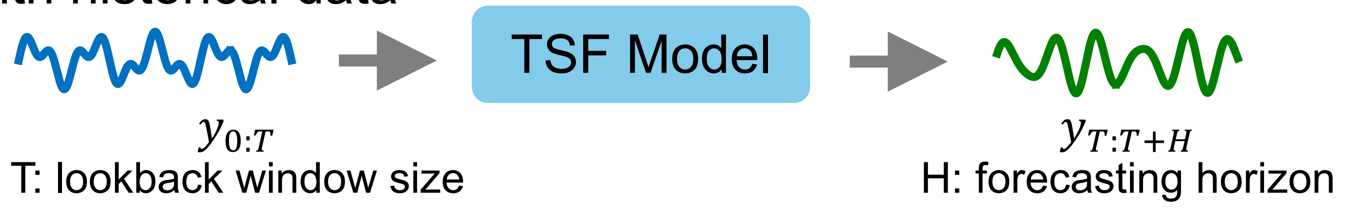
Abrar Majeedi, Viswanatha Reddy Gajjala, Srinath Namburi, Nada Elkordi, Yin Li

University of Wisconsin-Madison

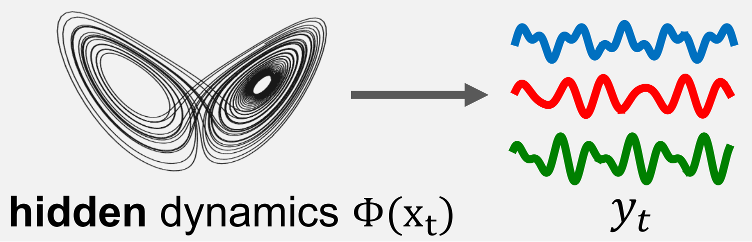
ICML International Conference On Machine Learning

Motivation

Time Series Forecasting (TSF): Predicting future values with historical data



Real world time series from hidden dynamical systems



$x_{t+1} = \Phi(x_t)$

$y_t = h(x_t) + \epsilon$

h: observation function

ϵ: measurement noise

Deep learning method have dominated TSF, yet they often ignore underlying dynamics.

Dynamical System Modeling + Deep Learning = More Accurate Time Series Forecasting?

Key Idea

Modeling hidden dynamics from just the observed data is challenging, but *Takens' Theorem* offers a solution.

Takens' Theorem: For a generic smooth dynamical system (Φ: $M \to M$), the delay embedding:

$$(h(x), h(\Phi(x)), \dots, h(\Phi^{m-1}(x))),$$

is an embedding of the manifold M into $\mathbb{R}^m$ when $m > 2d$, where d is the dimension of M, and $h: (M \to \mathbb{R})$ is a smooth observation function.

TL;DR: With observed 1D time series y, its time delayed version has similar topology as the underlying state x, and thus, can be used to "reconstruct" the dynamics.

Key Challenges:

- Takens' theorem assumes *noise-free measurements*, yet real-world data are almost always noisy.
- How can we design *learning-based models* to leverage Takens' theorem for time series forecasting?

Work was partially supported by National Science Foundation under Grant No. CNS 2333491, and by the Army Research Lab under contract number W911NF-2020221.

Prior Work: Empirical Dynamical Modeling (EDM)

Empirical Dynamical Modeling (EDM): a computational model that builds on Takens' theorem for time series forecasting

Input: univariate time series $y_{0:T}$ Output: predictions of $y_{T+\Delta t}$ $\forall \Delta t \in [1, H]$ (assuming $H \ll T$)

1. Time-delay the input $y_{0:T}$ by $2d+1$ steps as $\hat{y}_{0:T}$
2. Find $2d+2$ nearest neighbors $\{y_{N_i}\}$ for $\hat{y}_T$ using a kernel function k in the time delayed space
3. Interpolation using nearest neighbors (i.e., Nadaraya-Watson estimator), following

$$y^{\text{Pred}}_{T+\Delta t} = \frac{1}{\sum_i^{2d+2} w_i} \sum_{i=1}^{2d+2} w_i \cdot y_{N_i+\Delta t},$$ where $w_i = k(\hat{y}_T, \hat{y}_{N_i})$ with k the kernel in Step 2

- A separate model for each time series
- Designed for univariate time series
- Limited forecasting horizon
- Sensitive to input noise

Our Work: DeepEDM

DeepEDM = deep learning + dynamical system modeling (EDM)

Input: Each channel $y_{0:T}$ of a time series
Output: predictions of $y_{T+\Delta t}$ $\forall \Delta t \in [1, H]$
(H may be larger than T)

① Initial prediction: $y^{p}_{T:T+H} = f(y_{1:T})$

② Time delay and encoding:

$\hat{y}_{1:T+H} = \text{TimeDelay}([y_{1:T}, y^{p}_{T:T+H}])$

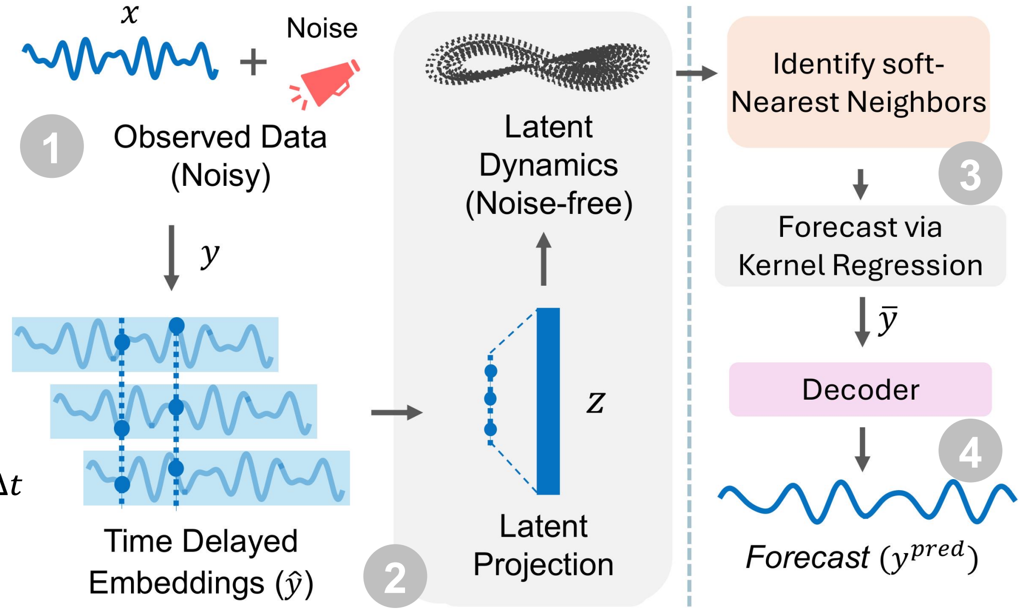
$z_{1:T+H} = \text{Encoder}(\hat{y}_{1:T+H})$

③ Kernel regression:

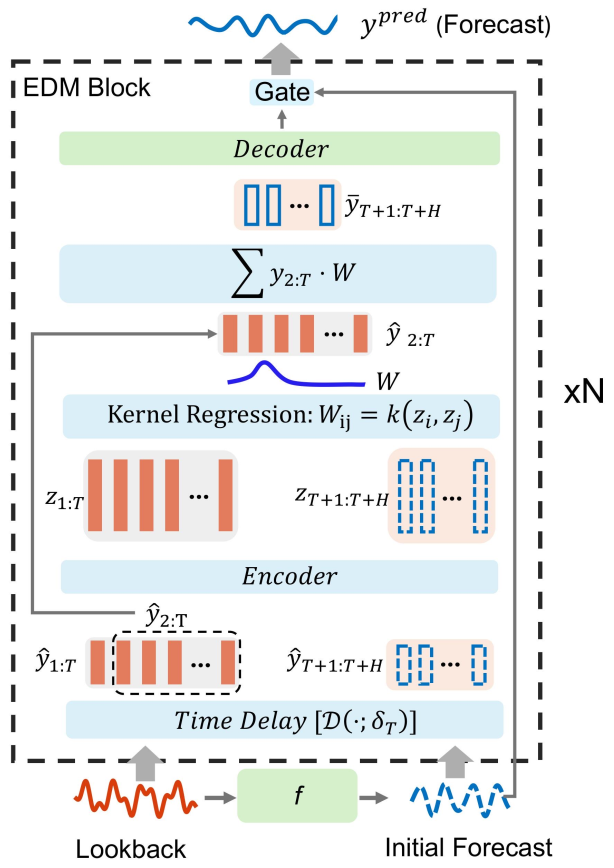
$$\bar{y}_{t'+\Delta t} = \frac{1}{\sum_{t=1}^{T} k(z_t, z_{t'})} \sum_{t=1}^{T} k(z_t, z_{t'}) \cdot \hat{y}_{t+\Delta t}$$

④ Prediction decoding:

$y^{\text{Pred}}_{T+1:T+H} = \text{Decoder}(\bar{y}_{T+1:T+H})$



Architecture (EDM blocks)



Key Design

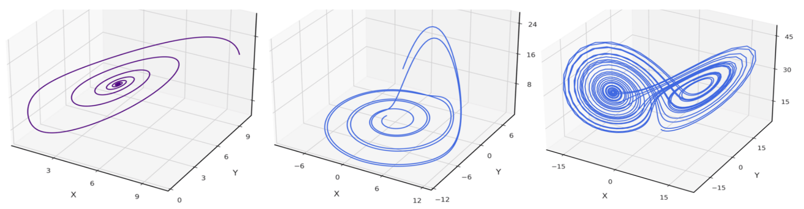
- A single model shared across time series
- Channel-wise modeling for multivariate series
- Cascading from initial prediction ① to allow long-term forecasting
- Time delay ② + kernel regression ③ = EDM
- Neighbor-based interpolation -> learnable kernel regression ③ for noise robustness
- Encoding-decoding architecture ②, ④ for improved performance

Relationship to Transformers:

DeepEDM = Transformer (with input patching and a variant of self-attention)

Experiments and Results

Synthetic Datasets: Chaotic and non-Chaotic Lorenz, Rossler Systems, with varied levels of Gaussian noise

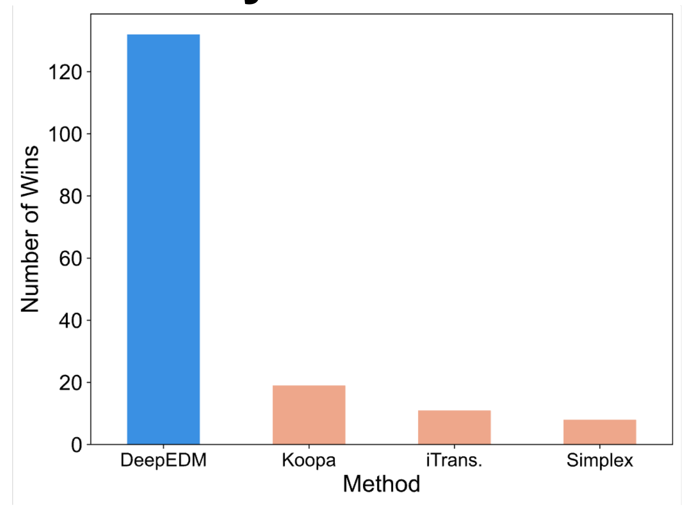


Real-world Benchmarks: Stocks, Health, Traffic, Electricity, M4, etc. (8 in total)

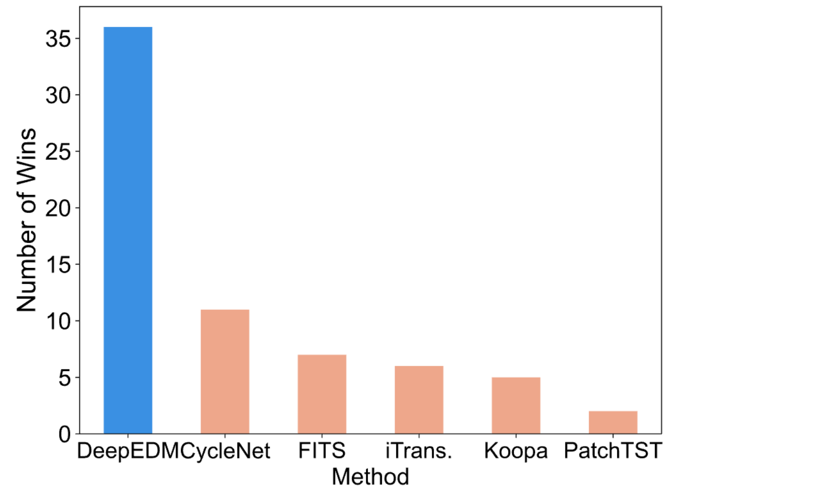
Baselines: Koopa, iTransformer, FITS, PatchTST, Dlinear, etc. (11 in total)

Metrics: Prediction error (MSE, MAE)

Results on Synthetic Datasets

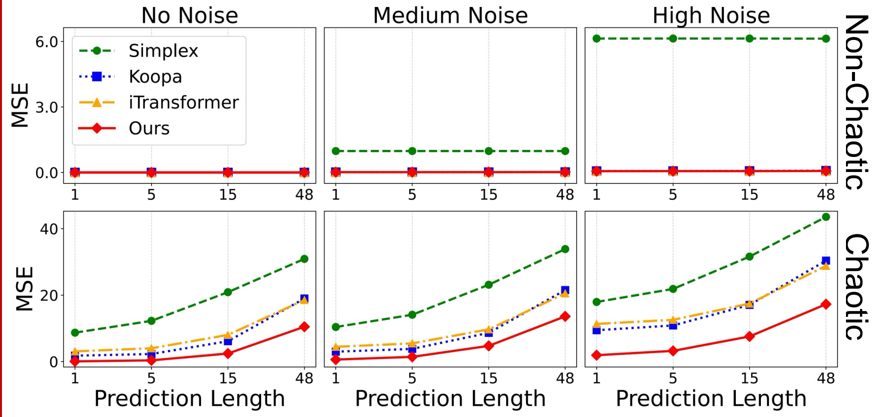


Results on Real-world Benchmarks



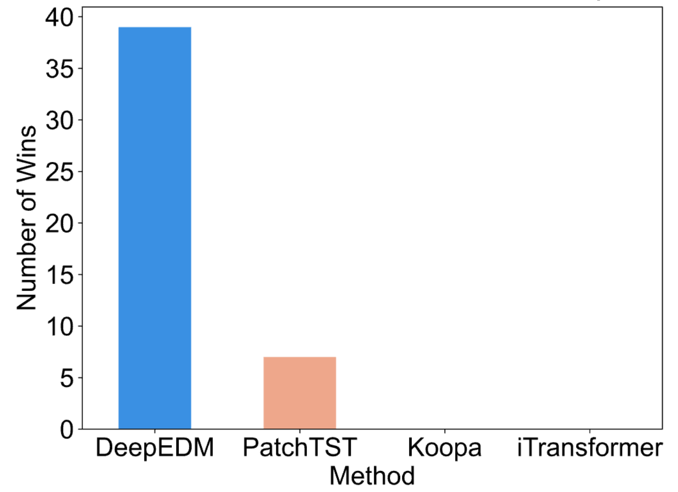
Robustness to noise

(Synthetic data with varied noise levels)



Generalization to Unseen Series

(real-world data with unseen series)



State-of-the-art performance on synesthetic and real-world benchmarks, across many settings, including *fixed lookback window, lookback window searching, varied forecasting horizons, input noise, and unseen series*

Conclusion

- ✓ A novel framework (DeepEDM) that integrates deep learning and dynamical system modeling
- ✓ DeepEDM builds on Takens' theorem, overcomes key EDM limitations, and sheds light on the success of prior Transformer models
- ✓ State-of-the-art performance on real-world and synthetic benchmarks, with robust to noise and generalization to unseen time series