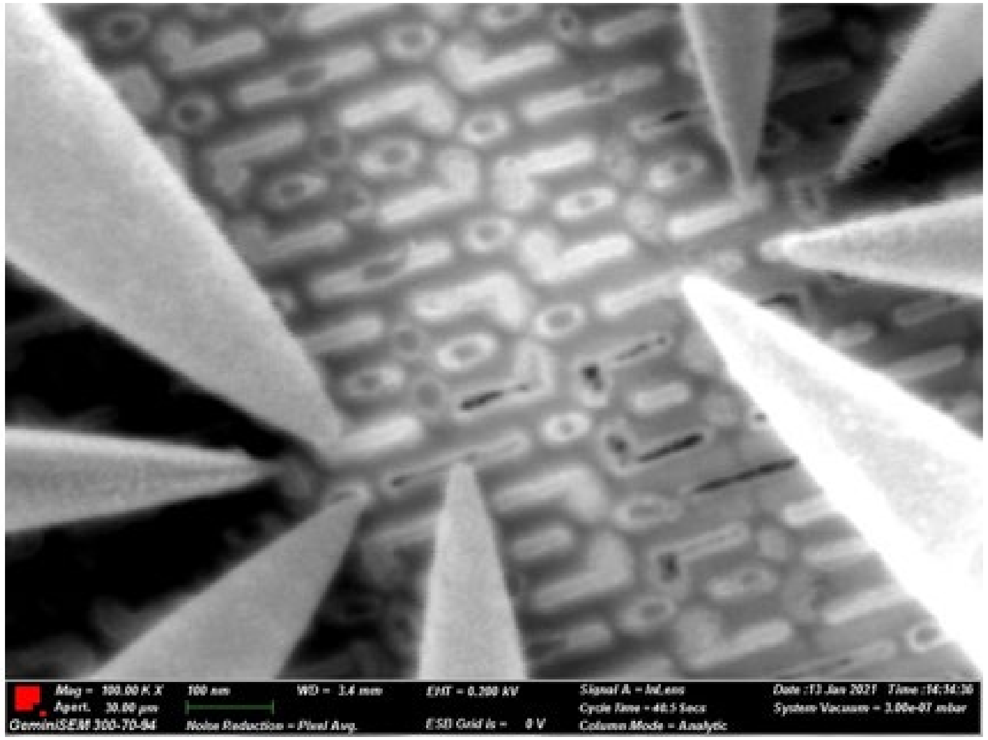
Evaluating ML Models for Modelling Piezo Dynamics

29.7.24

David Kleindiek

Introduction

- Nanoprobing
- Semiconductors Failure Analysis



- Driven by piezo motors
- Sub-nanometre accuracy

Devices

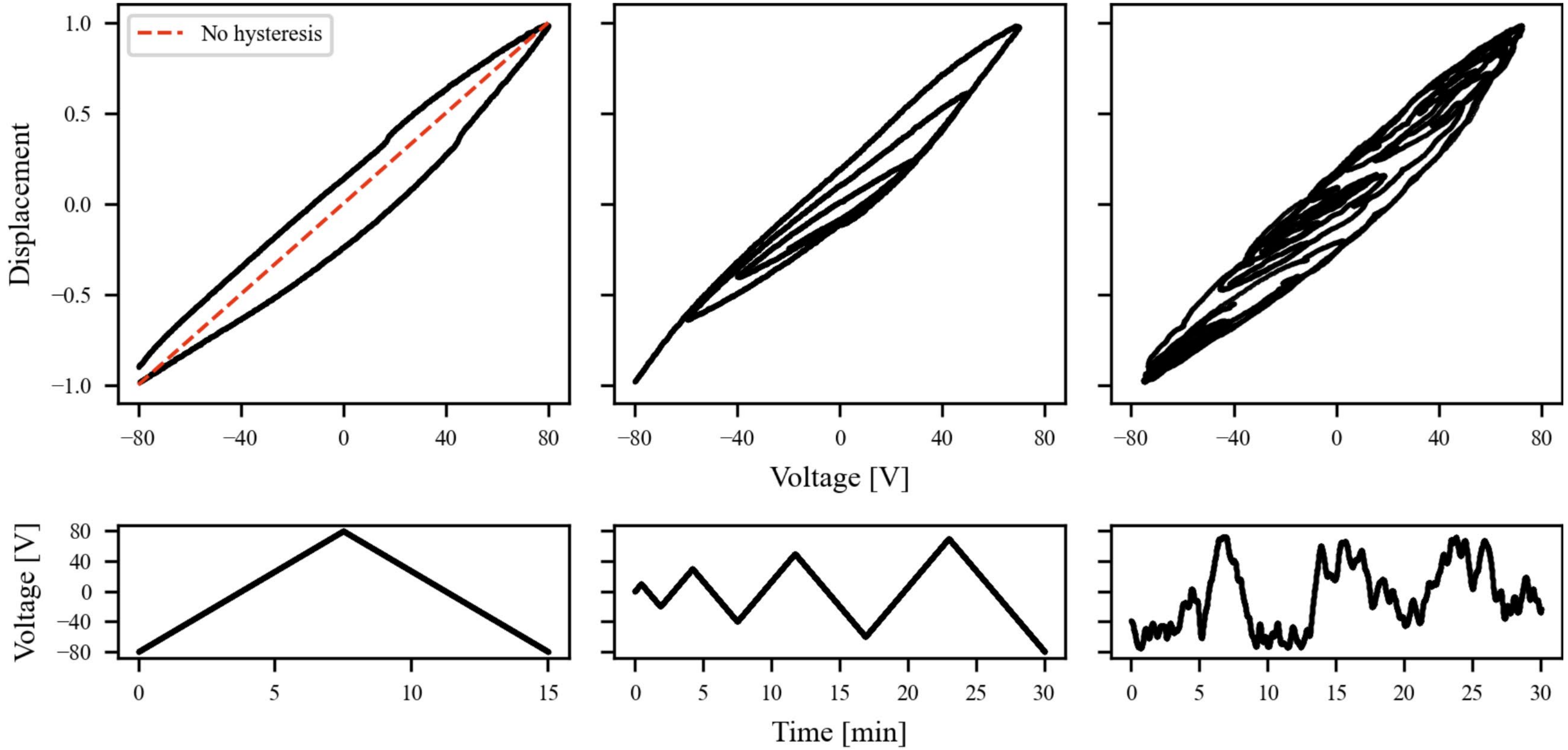
MM3E

- Three axis
- Input voltage: -80V to 80V
- Max displacement: a, b=20µm, c=3µm
- Encoder resolution: a, b=250nm c=20nm

⇨ C Axis used to capture data

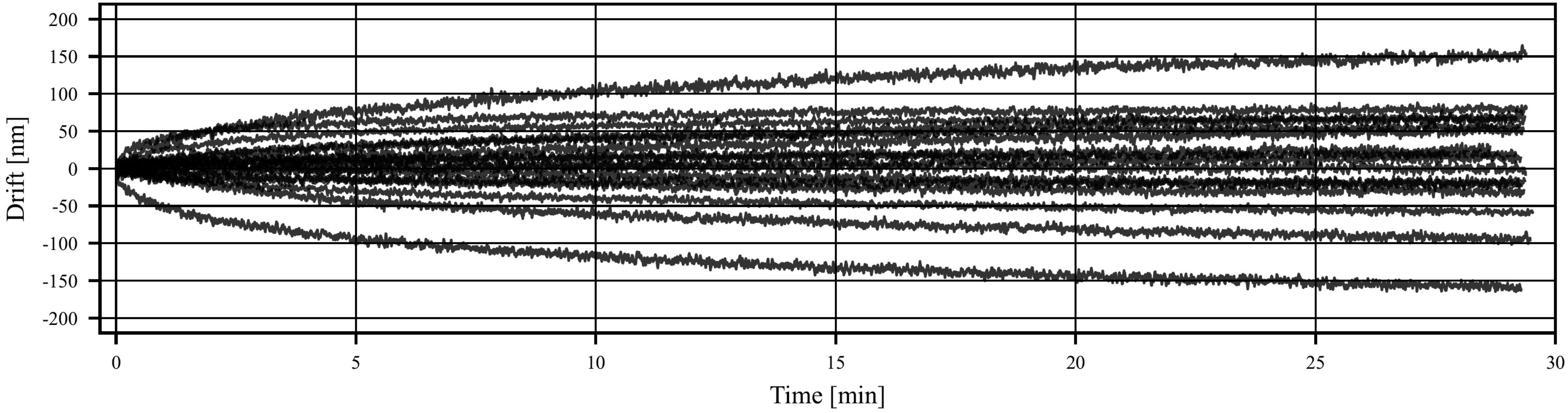
Challenges: Hysteresis

- Non-linear relation between input voltage and displacement
- Dependent on current and past applied voltages



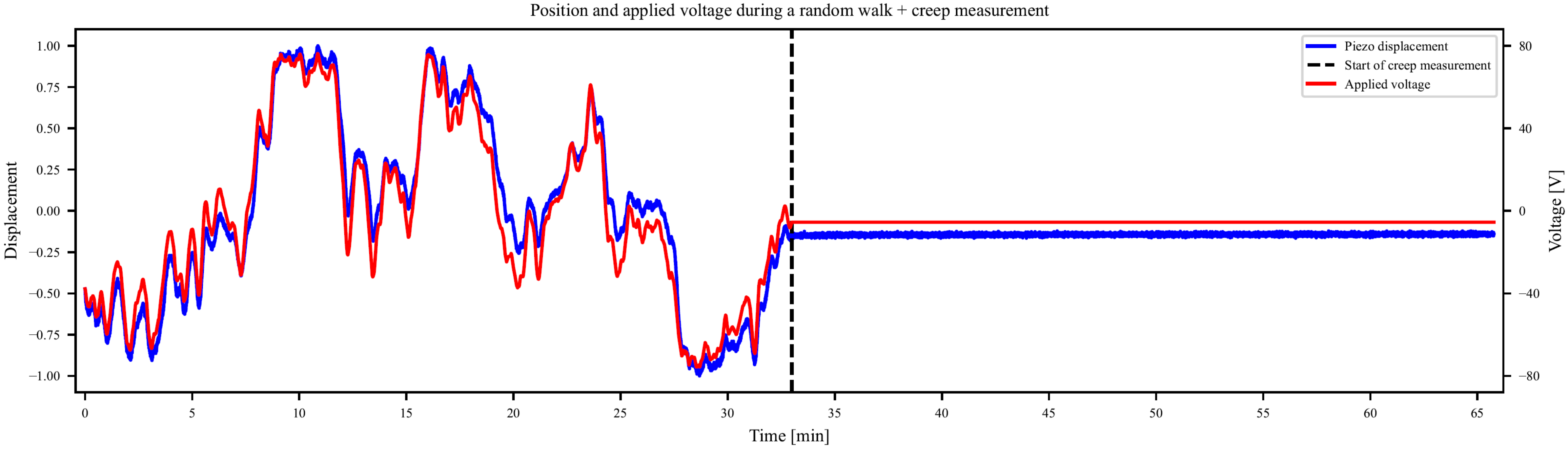
Challenges: Creep

- Slow drift of displacement after voltage step
- Up to 15% deviation from desired position
- Time dependent on previous applied voltages



Data collection

- Modified random walk
- >32 hours of movement data
- >7 hours of creep recording



Feature enginering

- Dependency on past applied voltages
- Step size

➔Rolling exponential sums
➔Lagged position
➔Voltage differentiating

- Lasso regression for feature selection

Models

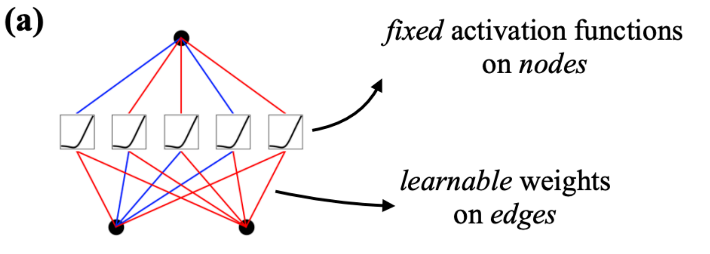
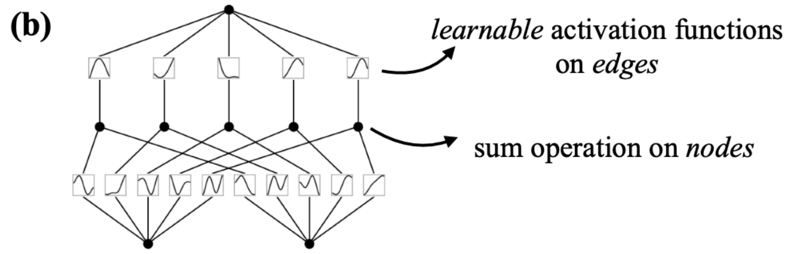
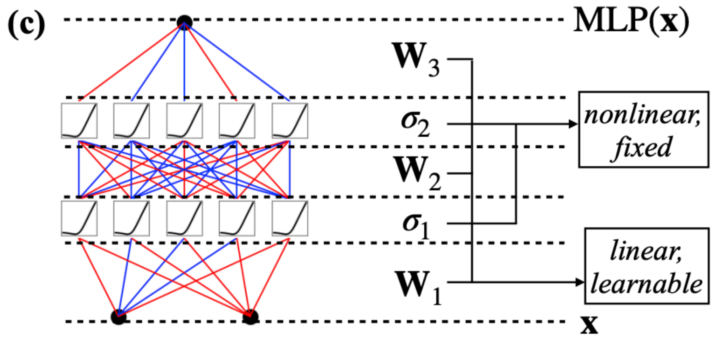
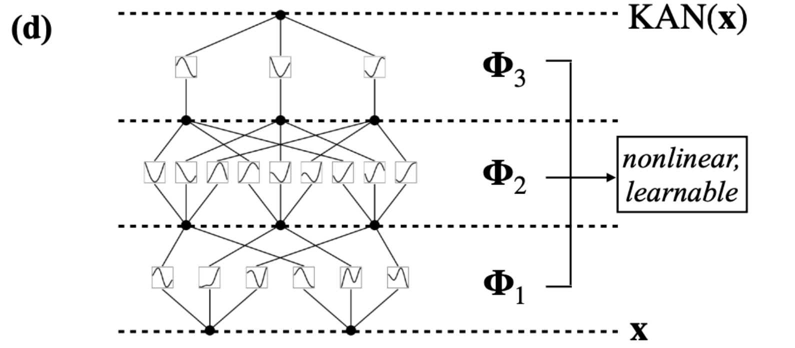
- Linear Regression
- Vanilla Neural Network
- Long short-term memory
- Kolmogorov-Arnold Networks

Hyperparameter Optimization Framework:

OPTUNA

https://github.com/optuna/optuna

KAN: Kolmogorov-Arnold Networks

Model	Multi-Layer Perceptron (MLP)	Kolmogorov-Arnold Network (KAN)
Theorem	**Universal Approximation Theorem**	**Kolmogorov-Arnold Representation Theorem**
Formula (Shallow)	$f(\mathbf{x}) \approx \sum_{i=1}^{N(\epsilon)} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i)$	$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right)$
Model (Shallow)		
Formula (Deep)	$\mathrm{MLP}(\mathbf{x}) = (\mathbf{W}_3 \circ \sigma_2 \circ \mathbf{W}_2 \circ \sigma_1 \circ \mathbf{W}_1)(\mathbf{x})$	$\mathrm{KAN}(\mathbf{x}) = (\mathbf{\Phi}_3 \circ \mathbf{\Phi}_2 \circ \mathbf{\Phi}_1)(\mathbf{x})$
Model (Deep)		

- Faster scaling
- Interpretability

https://github.com/KindXiaoming/pykan

KAN: Training

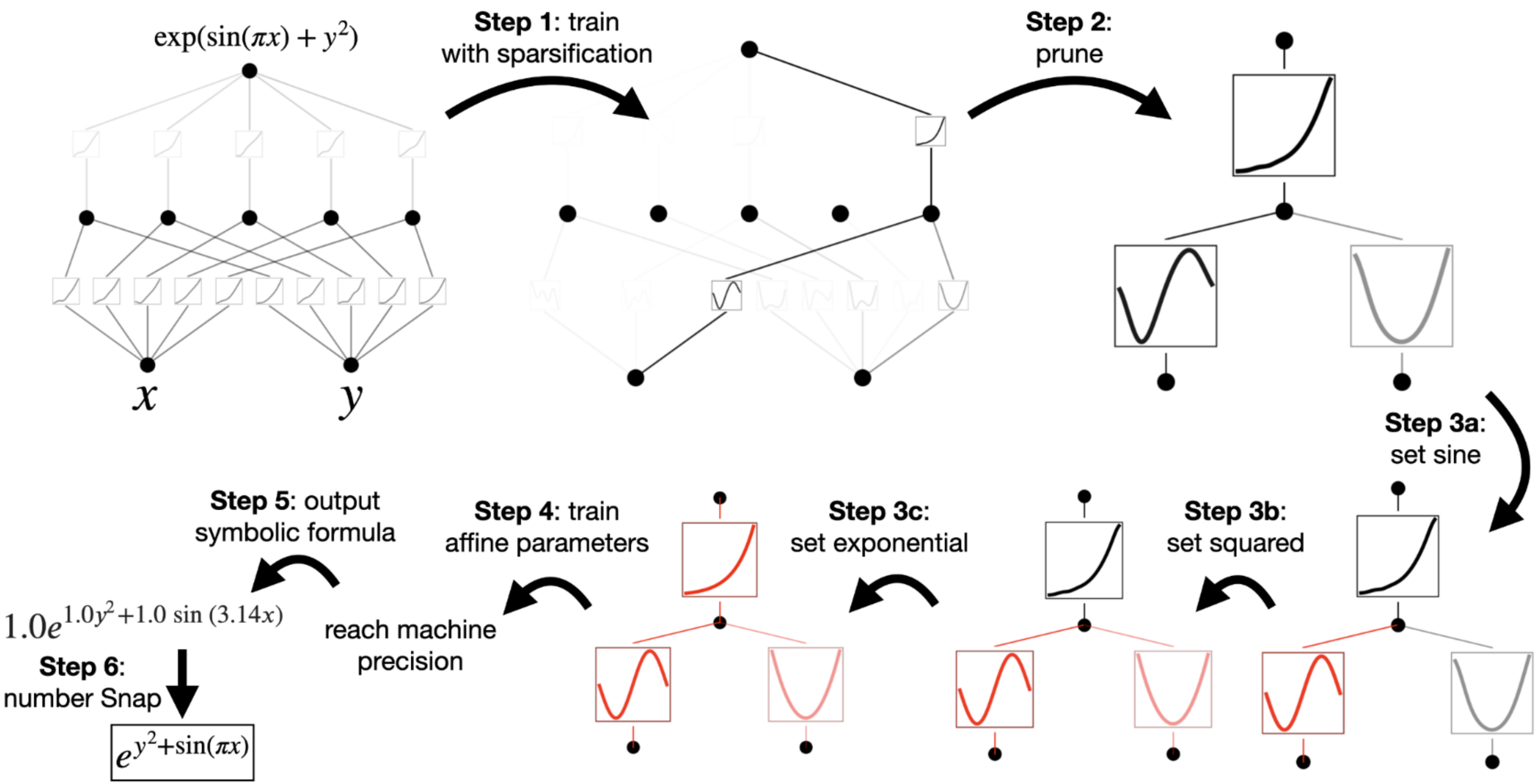
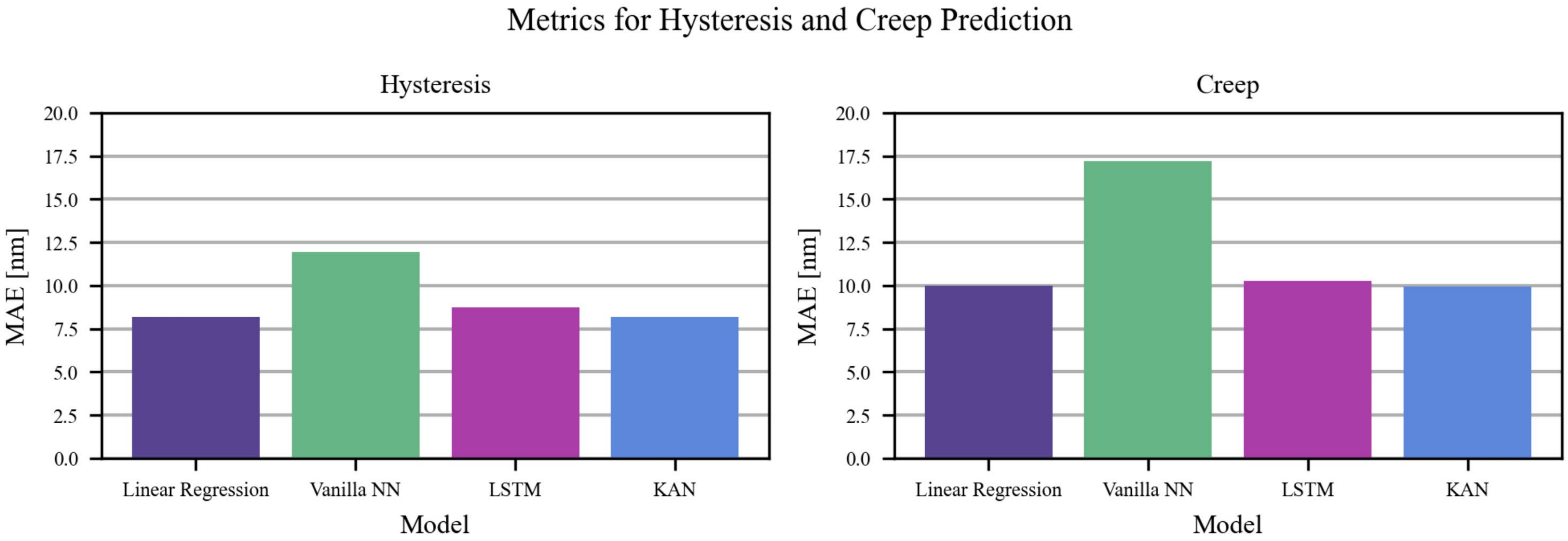


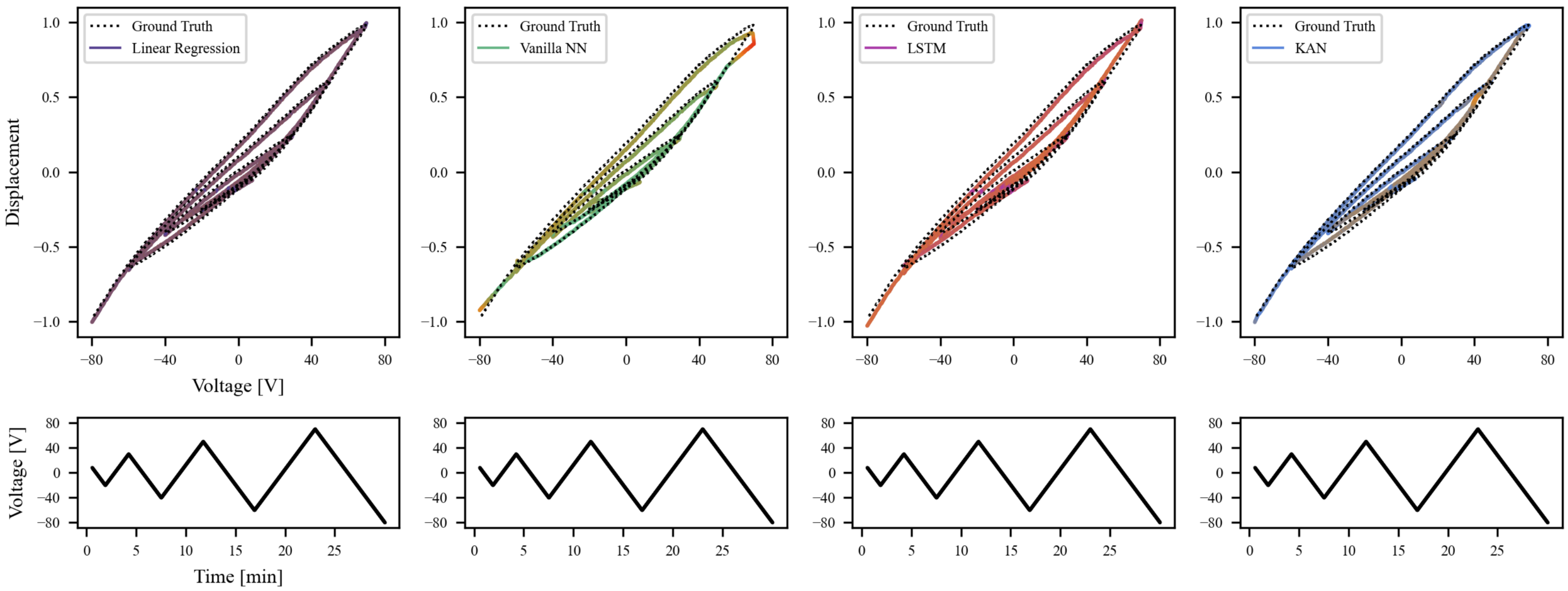
Figure 2.4: An example of how to do symbolic regression with KAN.

https://arxiv.org/abs/2404.19756

Metrics

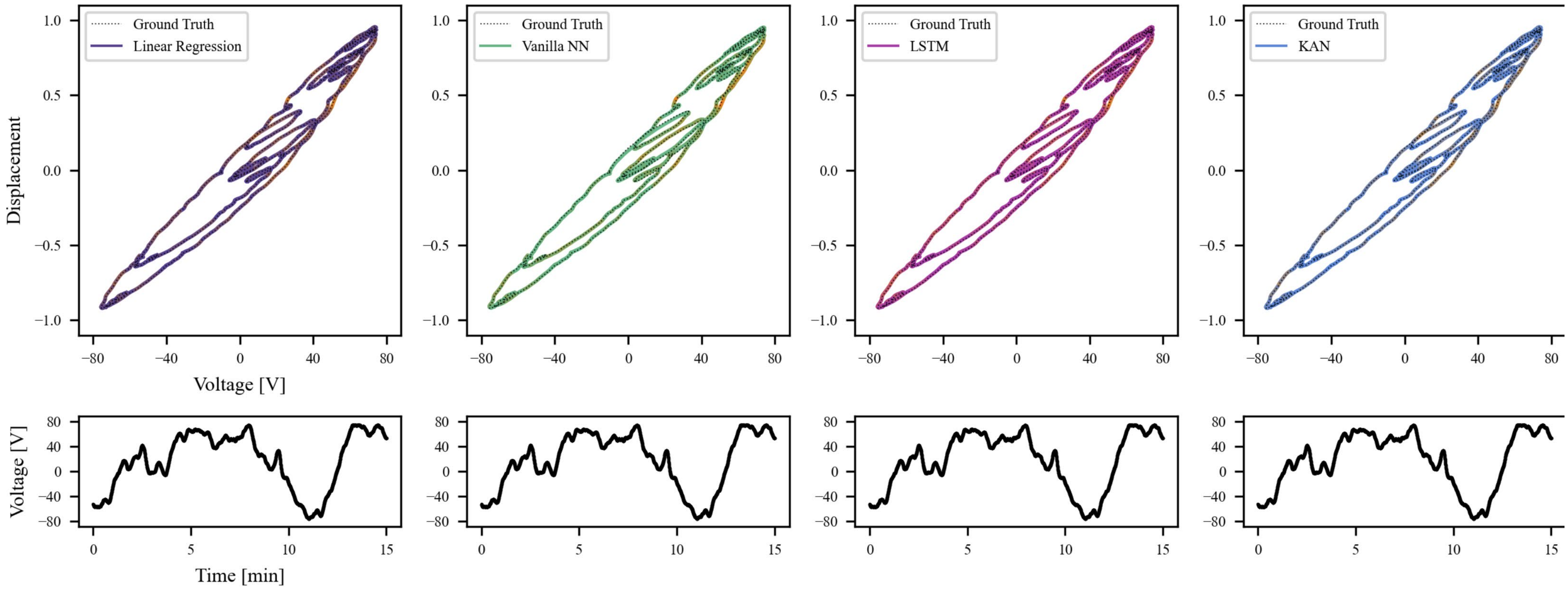

Metrics for Hysteresis and Creep Prediction
Hysteresis
Creep
MAE [nm]
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Linear Regression
Vanilla NN
LSTM
KAN
Model

Evaluation: Hysteresis



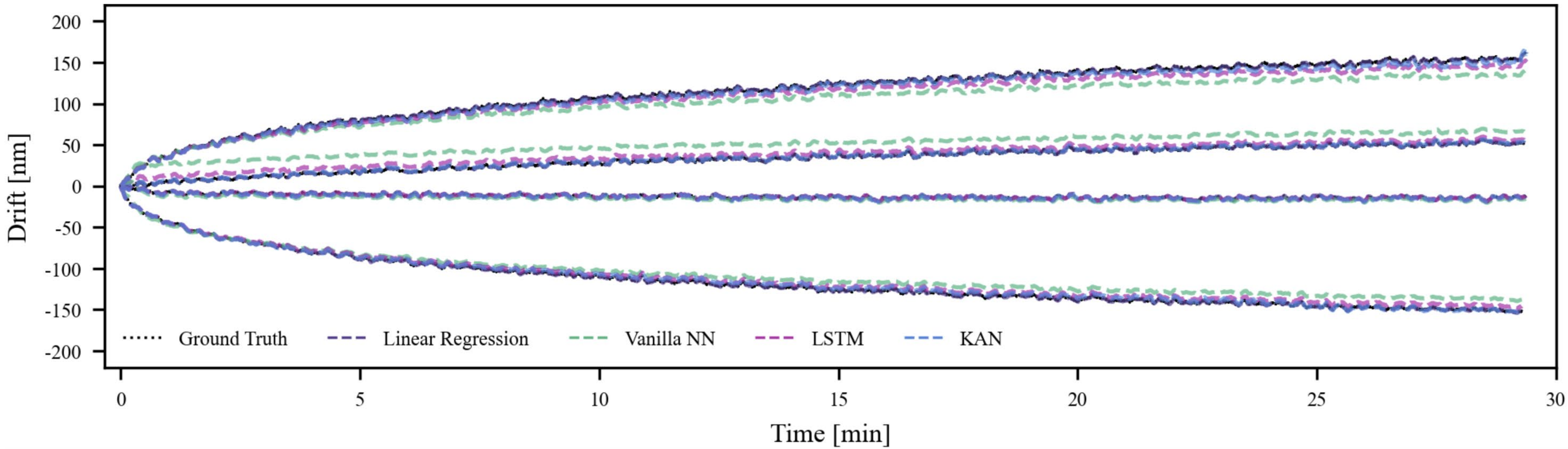
- Turning points pose a problem for neural network

Evaluation: Hysteresis



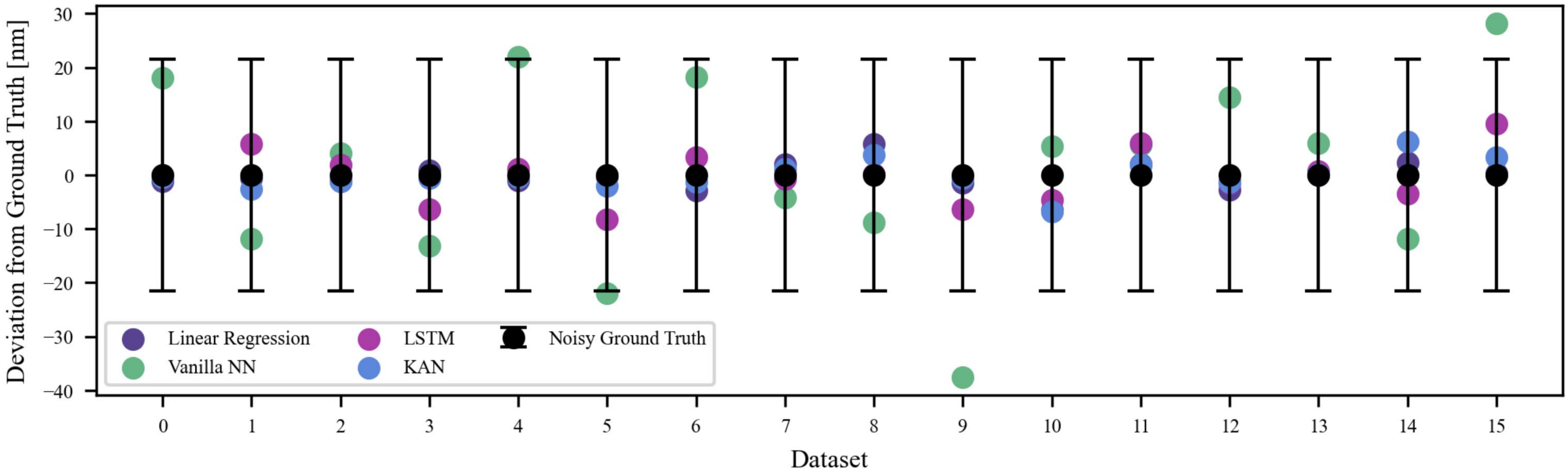
Evaluation: Creep

- Vanilla NN and LSTM deficate slightly
- KAN near perfect



Evaluation: Creep

- Endposition + encoder noise characteristic



Outlook

- Create an inverse model to compensate the effects
- Extend model to predict voltage for goal position
- Integrate models into the controller of the manipulators