

# IPLNA - In-Parameter Linear Nonlinear neural Architectures

Comparison to larger Neural Networks

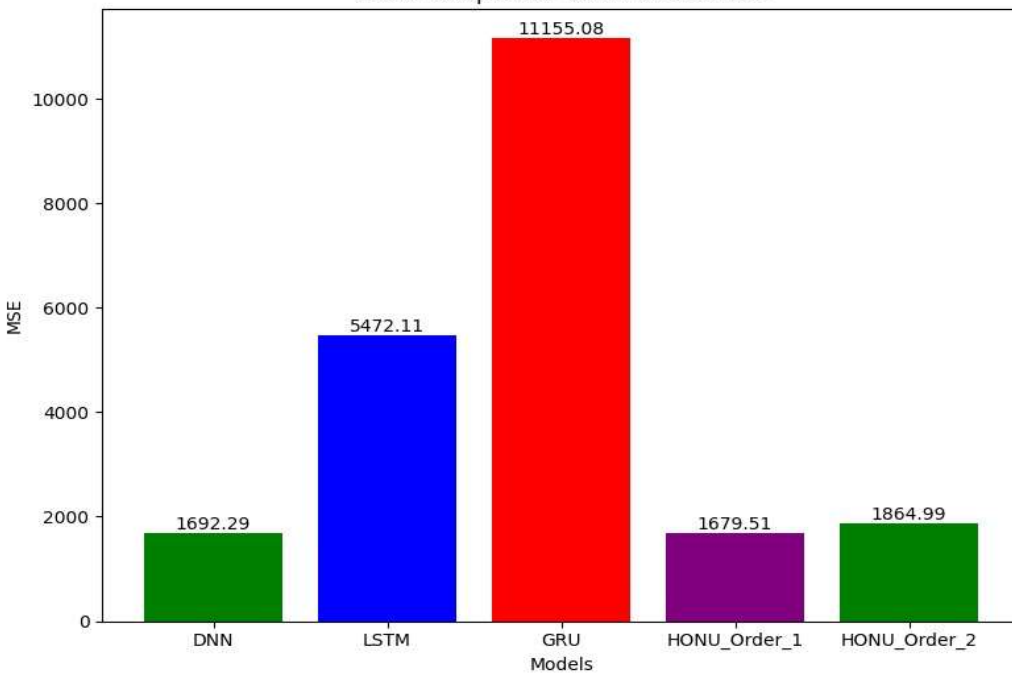
“Explainability”, “Physics-Informed”

Learning Weight Convergence,

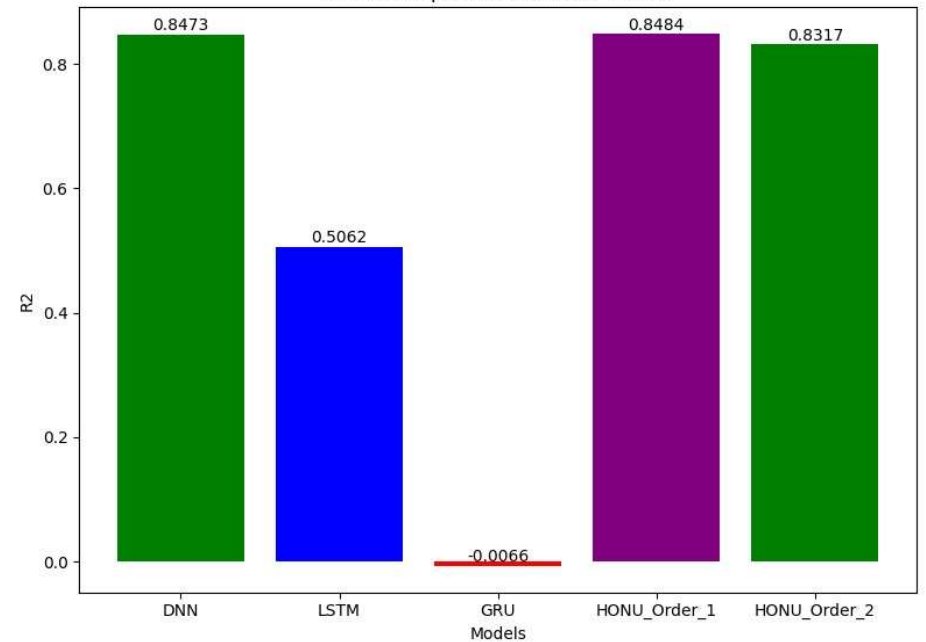
Ivo Bukovsky (FSci, USB), Moritz Sontheimer (NTUST), Vladimir Maly (VM-Engineering)

# Photovoltaic Power Prediction Model Comparison (Deep vs Shallow Neural Networks)

Model Comparison based on Test MSE



Model Comparison based on Test R2

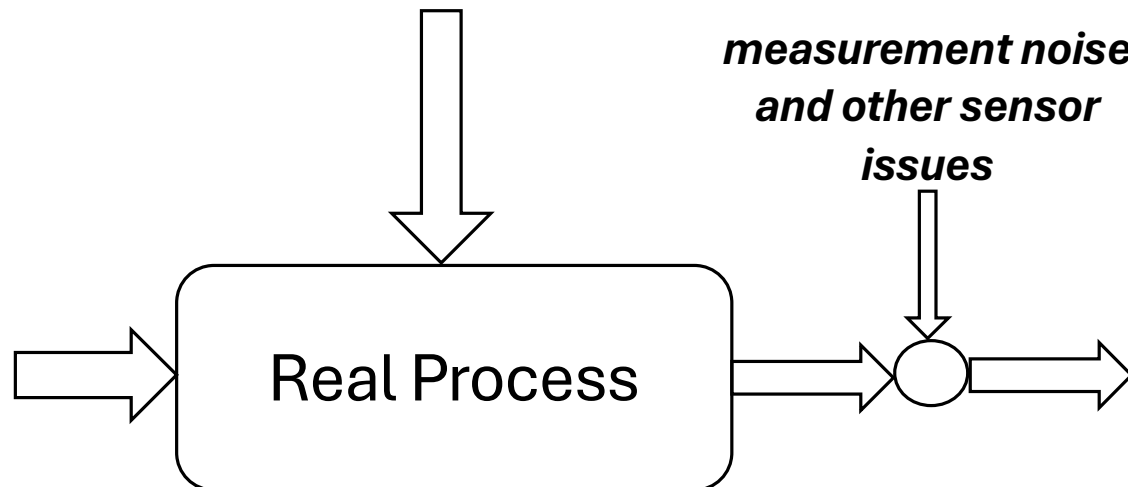




## ***Nonstationarity, Perturbations, Incompleteness of Process Observations***

***process perturbations,  
component wear and fatigue in time,  
external environment changes,  
untold changes***

***Process inputs  
that we can affect  
(control inputs,  
setups,  
configurations)***



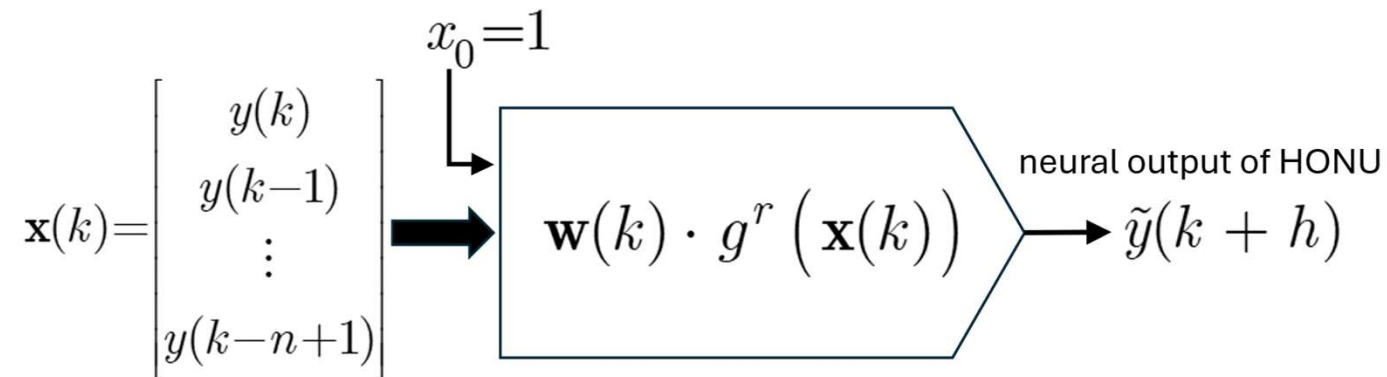
***The observable  
outputs  
(do we measure/see  
enough variables to  
predict future  
behavior?)***

Mini AI (?)

IPLNA

HONU

**HONU ... Higher-Order Neural Unit** is a standalone neuron, i.e, merely a polynomial of  $r$ -th order nonlinearity [7-9,14]



$$g^r(\mathbf{x}) = \left[ \begin{array}{c} i=0 \dots n \\ j=i \dots n \\ \kappa=j \dots n \\ \vdots \end{array} \left\{ x_i \cdot x_j \cdot x_\kappa \cdots \right\} \right]$$

$$r = \left\{ \begin{array}{l} 1 \dots \text{Linear Neural Unit (LNU)} \\ 2 \dots \text{Quadratic Neural Unit (QNU)} \\ 3 \dots \text{Cubic Neural Unit (CNU)} \\ \vdots \end{array} \right.$$

$$\text{LNU: } g^{r=1}(\mathbf{x}(k)) = \mathbf{x}(k)$$



# Some Considerations

- Larger NN (LSTM etc)
  - “have memory” so they can learn long data without changing all weights (parameters) of the model seasonally
  - But they are not analyzable (explainable, convergence/stability,...)
- Simpler (single “neuron” model, IPLNA, HONU)
  - All weights respond to current data (learn and forgets quickly temporal dependency in data, no memory for long data)
  - The data behavior is encoded into a vector of neural weights
  - We can use them if not enough data, if noisy data,...
  - We may explain system states according to learn weights
  - We can analyze/guarantee model convergence (learning stability) for any gradient learning
  - We can analyze stability of a real system from learned data
  - We can apply learning Entropy
- IPLNA have one global minima (we solve set of linear equations); however, the input data must be linearly independent (no collinearity (redundancy) in  $\mathbf{x}$ )



# IPMAI

Interpretable Prescriptive Maintenance  
using Artificial Intelligence

**Interreg**  
Austria – Czechia



Co-funded by the  
European Union



UNIVERSITY  
OF APPLIED SCIENCES  
UPPER AUSTRIA



Průmyslověcká  
fakulta  
Faculty  
of Science

Jihočeská univerzita  
v Českých Budějovicích  
University of South Bohemia  
in České Budějovice

intemac



scch {  
software  
competence  
center  
hagenberg  
}



V S P  
College of  
Polytechnics  
Jihlava

dataPartner<sup>®</sup>

