Towards Fully Adaptive Deep Neural Networks

Professor Sheng Chen

School of Electronics and Computer Science

University of Southampton

Southampton SO17 1BJ, United Kingdom

Keynote speech at LSMS2021 & ICSEE2021

October 30 - November 1, Hangzhou, China

Joint work with Dr Tong Liu, Department of Computer Science, University of Sheffield, U.K.

Background

- Artificial neural networks have evolved from '**shallow**' one-hidden-layer architecture, such as RBF, to '**deep**' architecture
  - **Deep learning** has achieved **breakthrough** progress in many walks of life
  - Deep neural networks have been applied to modeling of industrial processes

- Deep learning's success coincides with **digital big data** era
  - With massive historical data, training of deep neural network models becomes practical
  - Enabling the exploitation of deep learning capability to capture complex underlying nonlinear dynamic behavious from data

- Many real-life processes are not only nonlinear but also highly **nonstationary**
  - During online operation, system's nonlinear dynamics can change significantly
  - Deep neural network model must **adapt fast** to such change

Motivations

- **Sampling period** of many industrial processes is **small**, and **adaptation** must be **sufficiently fast** to be completed within a sampling period
  - **Impossible to adapt structure** of deep neural network model, such as SAE, within sampling period
  - Instead, adaptation is taken place **only on weights of output regression layer**
  - **Insufficient** for tracking significant and fast changes in system
- We have proposed an adaptive **gradient radial basis function** network
  - Adapting structure of GRBF is not only optimal but also imposes **litter** online computation complexity
  - Completely feasible to complete adaptation within a sample period
  - GRBF is a **shallow** neural network
- **Combining** deep learning capability of **deep neural network**, such as SAE, with excellent adaptability of **GRBF**? $\Rightarrow$ Motivate this research

System Model

- **Nonlinear** and **nonstationary** system

$$y_t = f_{\mathrm{sys}}(\boldsymbol{x}_t; t) + \xi_t$$

  - **Output** y_t with lag n_y
  - **Input** vector $\boldsymbol{u}_t \in \mathbb{R}^m$ with lag n_y
  - **Noise** ξ_t
  - Unknown nonlinear and nonstationary system map $f_{\mathrm{sys}}(\cdot; t)$
  - System 'input' **embedding** vector with dimension $n = n_y + m \cdot n_u$

$$\boldsymbol{x}_t = [x_{1,t} \cdots x_{n,t}]^{\mathrm{T}} = \left[y_{t-1} \cdots y_{t-n_y} \; \boldsymbol{u}_{t-1} \cdots \boldsymbol{u}_{t-n_u}\right]^{\mathrm{T}}$$

- This is **one-step** ahead predictor model
  - Extension to **multi-step** ahead predictor straightforward

GRBF Network

- **Differencing output series** to reduce nonstationarity: **GRBF** input

$$\boldsymbol{x}'_t = \left[y_{t-1} - y_{t-2} \cdots y_{t-n_y} - y_{t-n_y-1} \cdots \right]^{\mathrm{T}}$$

By comparison, **RBF** input

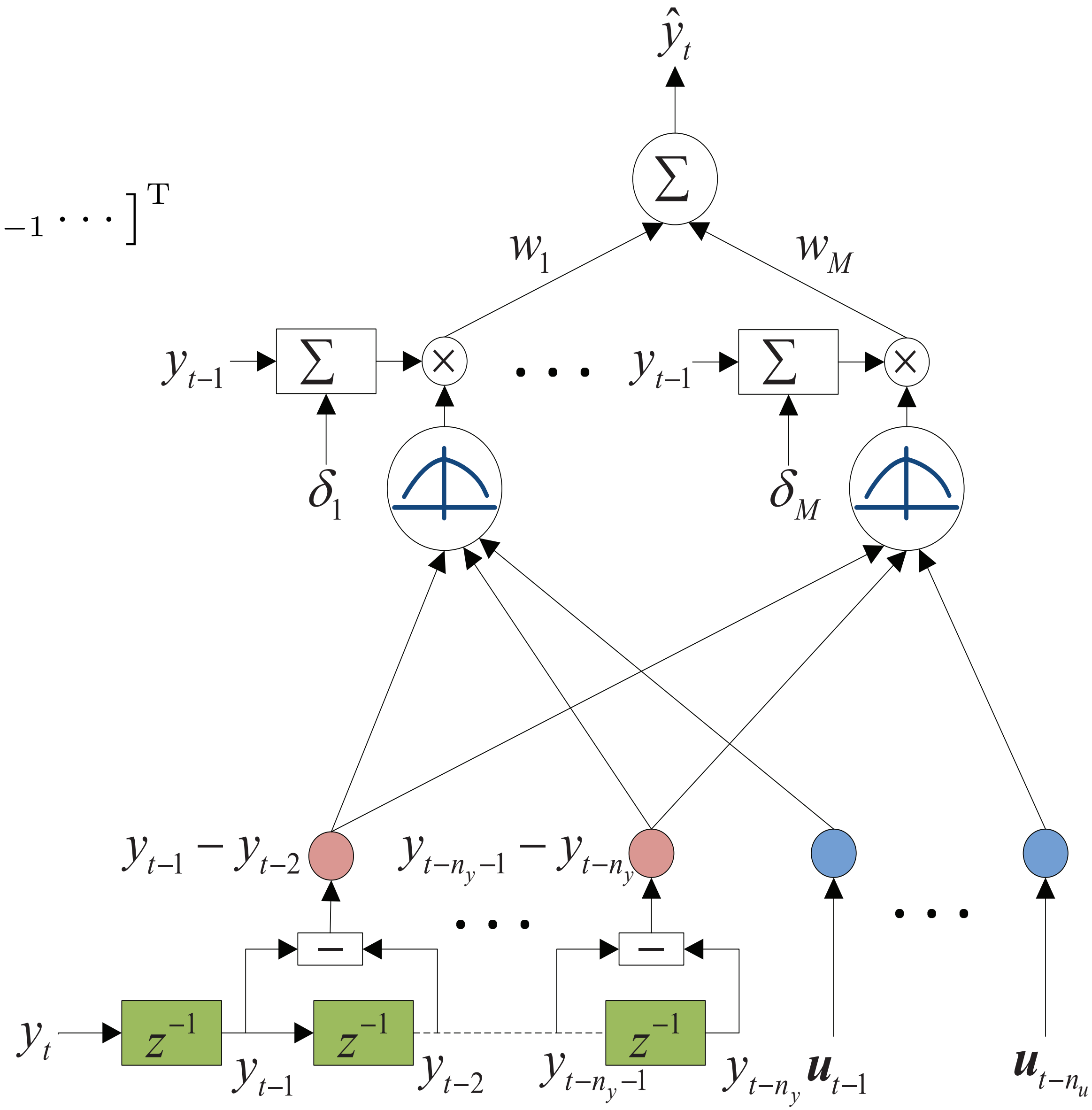
$$\boldsymbol{x}_t = \left[y_{t-1} \cdots y_{t-n_y} \cdots \right]^{\mathrm{T}}$$

- **Hidden node as local predictor** of y_t: **GRBF** node

$$\varphi_j(\boldsymbol{x}'_t) = (y_{t-1} + \delta_j) \cdot e^{-\frac{\|\boldsymbol{x}'_t - \boldsymbol{c}_j\|^2}{2\sigma^2}}$$

By comparison, **RBF** node

$$\varphi_j(\boldsymbol{x}'_t) = e^{-\frac{\|\boldsymbol{x}'_t - \boldsymbol{c}_j\|^2}{2\sigma^2}}$$



Efficient Training

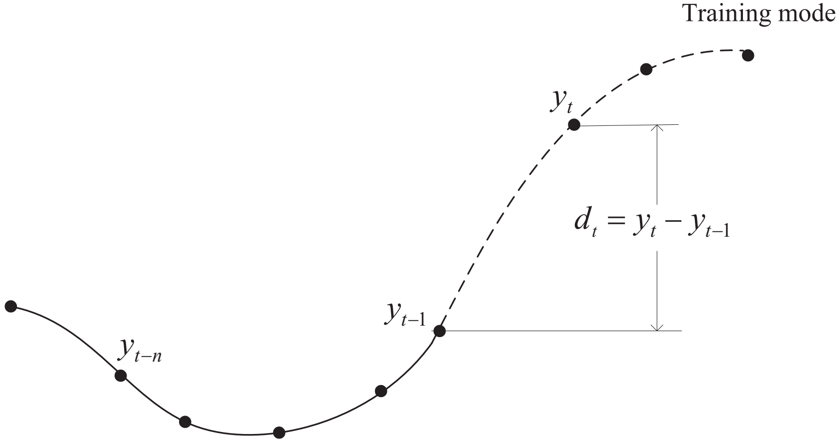
- Like RBF network, efficient training is achieved with OLS algorithm

  Training data $\left\{\boldsymbol{x}_t, d_t = y_t - y_{t-1}; y_t\right\}_{t=1}^{N} \underbrace{\Longrightarrow}_{\text{OLS selects subset model}} \left\{\boldsymbol{c}_{t_j}, \delta_{t_j}\right\}_{j=1}^{M}$

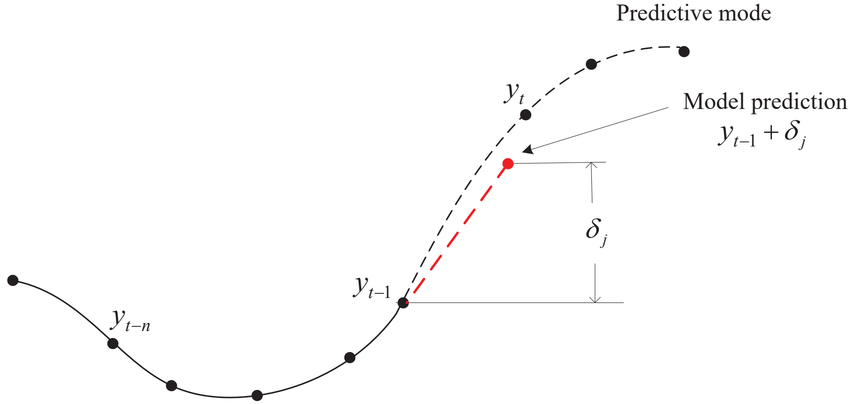
- Geometric interpretation of GRBF hidden node: each center encodes an independent system state and each node acts as a local predictor of system output y_t

(a) In training, if $\boldsymbol{x}_t$ selected as jth center, set $\delta_j = d_t \rightarrow j$th node is perfect predictor of y_t

(b) In prediction, if $\boldsymbol{x}_t$ is close to jth center $\rightarrow j$th node is very good predictor of y_t



(a)



(b)

Online Adaptation

- During online operation, system's underlying dynamics can **change** significantly
  - A model must **adapt** to changing operation environment in **real time**
  - Optimizing **structure** of neural networks, both shallow and deep ones, online is computationally **prohibitive**

- Typically, when observation/measurement of y_t becomes available, RLS is used for online adaptation of **weights** of output layer **only**
  - For highly nonstationary process, this is **insufficient**

- Online learning or adaptive modeling principle: balance 'stability' and 'plasticity'
  - Online leaner should have ability to retain acquired knowledge (**stability**)
  - At same time, has ability to forget out-of-the-date past knowledge so as to learn new one as quickly as possible (**plasticity**)

- Adaptive GRBF achieves **balanced** or optimal trade-off of stability and plasticity

Adaptive GRBF

- During online operation, when current modeling $\widehat{y}_t$ is insufficient:

$$\left(y_t - \widehat{y}_t\right)^2 / y_t^2 \geq \text{thresold}$$

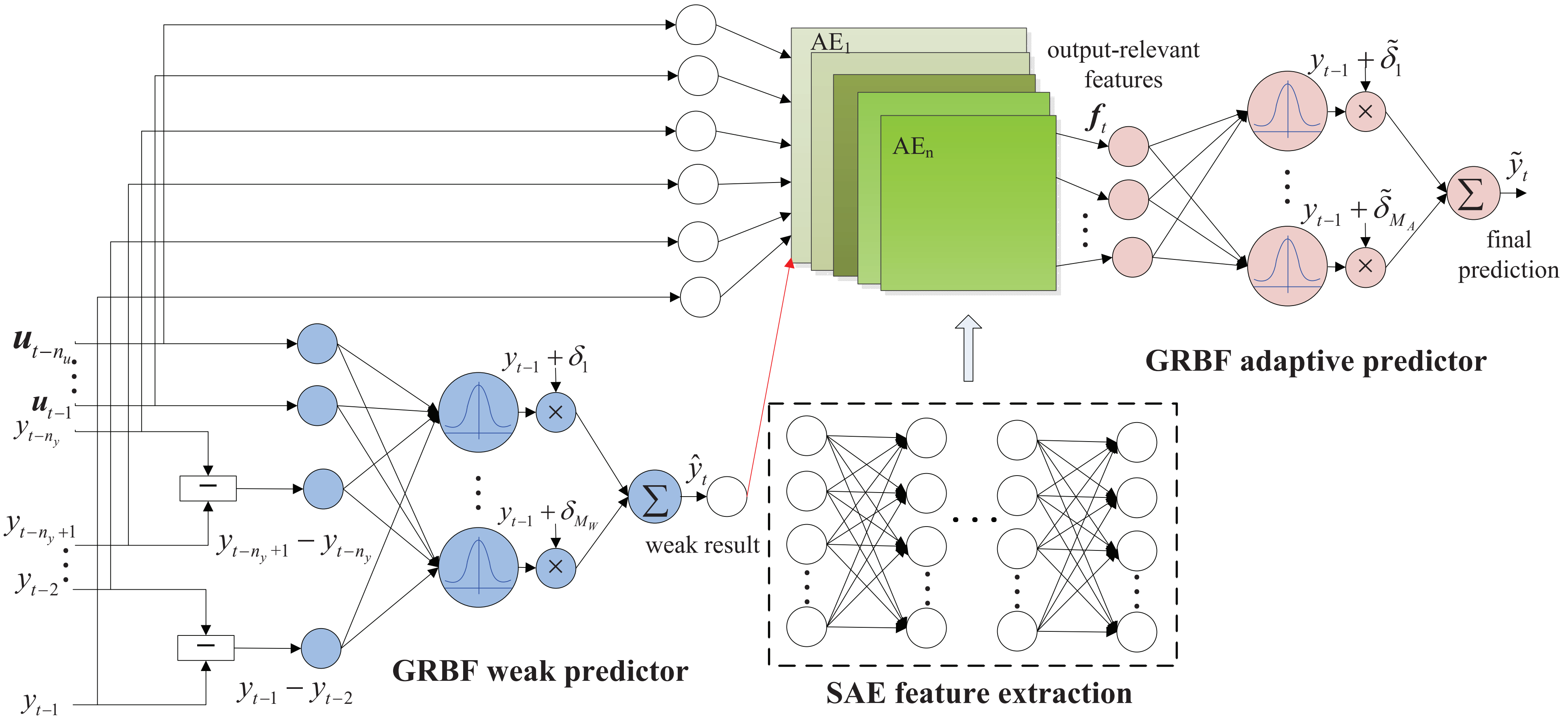
Worst node (smallest squared weighted node output) replaced with a new node:

$$\text{node center } \boldsymbol{c}_r \leftarrow \boldsymbol{x}_t' \quad \text{node scalar } \delta_r \leftarrow y_t - y_{t-1}$$

  - Most nodes do not change - nodes encode independent system states acquired from historical data - stability
  - Most out-of-date node is replaced - plasticity, to encode **newly emerging system state** and new node is **perfect** local predictor of y_t
  - Online **complexity**: regularized LS estimation of output layer weights

Liu, Chen, Liang, Du, Harris, "Fast tunable gradient RBF networks for online modeling of nonlinear and nonstationary dynamic processes," ***J. Process Control*****, 93, 53–65, 2020**

Proposed Deep Neural Network: Structure



- **GRBF weak predictor** module, provide preliminary output prediction
- **Output-enhanced stacked autoencoder** module, provide deep output-relevant features
- **GRBF adaptive predictor** module, provide final output prediction

Proposed Deep Neural Network: Rationale

- **SAE** is a **deep neural network** finding its way to **regression** application
  - Layers of stacked autoencoders extract deep features from input
  - Given information of output y_t, SAE can extract much better-quality features
- Impossible to provide y_t as input to SAE - We do next best thing, provide a perdition of y_t as input to SAE by **GRBF weak predictor**
- Instead of usual linear output regression layer on top of SAE to provide prediction of y_t, we replace it by a much stronger **GRBF adaptive predictor**
- **Training** of proposed deep neural network
  - **OLS** for GRBF weak predictor
  - **Standard optimization** procedure for SAE
  - **OLS** for GRBF adaptive predictor

Proposed Deep Neural Network: Operation

- Proposed DNN: SAE enhanced by GRBF weak predictor maps process input space onto deep **feature space**, and GRBF adaptive predictor then maps feature space onto process **output space**

- During online operation, GRBF weak predictor and SAE are **fixed** (impossible to adapt SAE online anyway)

- GRBF adaptive predictor is **adapted** online to track process's changing dynamics
  - When underlying system dynamics change significant, feature space changes accordingly
  - GRBF adaptive predictor capable of fast adapting to changing process dynamics
  - while imposing very low online computational complexity, capable of meeting **real-time** constraint of small sampling period

- Proposed deep neural network integrates **deep learning capability** of **SAE** with **excellent adaptability** of **GRBF**

Experiment Setup

- **Proposed** DNN is compared with following **benchmarks**
  - Long short-term memory (LSTM): during online operation, LSTM is **fixed**
  - Stacked autoencoder (SAE): during online operation, SAE is **fixed**
  - Adaptive SAE: during online operation, only **weights** of output regression layer is **adapted** by RLS
  - **Adaptive GRBF** (AGRBF)

- Performance measure: **test mean square error** (MSE)

- Online computational complexity: measured by **averaged computation time per sample** (ACTpS) in [ms]

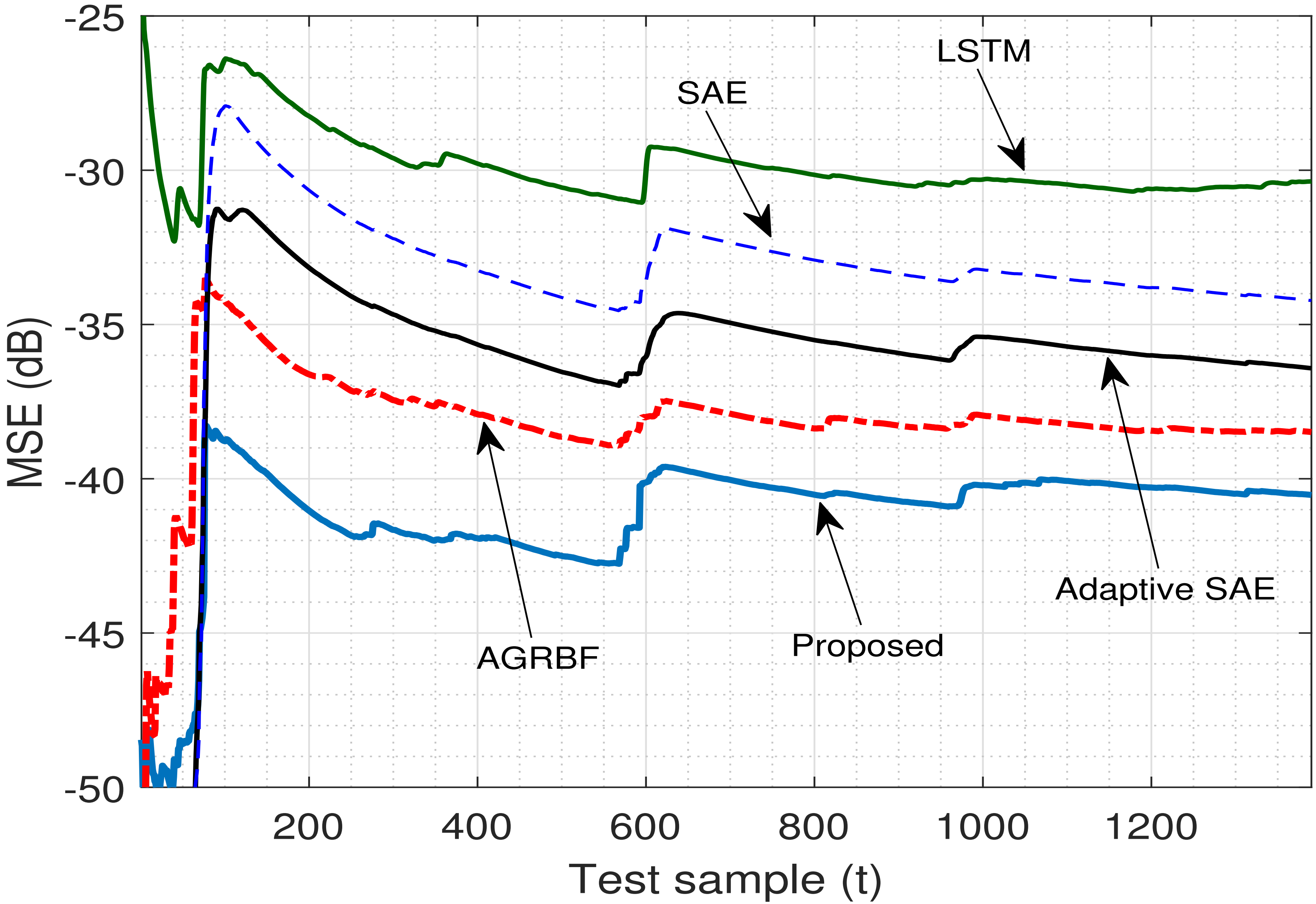
Liu, Tian, Chen, Wang, Harris, "Deep cascade gradient RBF networks with output-relevant feature extraction and adaptation for nonlinear and nonstationary processes," submitted to *IEEE Trans. Cybernetics*

Case 1: Debutanizer Column Process

Method	Initial training MSE (dB)	Online prediction/modeling MSE (dB)	Online prediction/modeling ACTpS [ms]
AGRBF	-41.5539	**-38.4860**	0.4989
LSTM	-35.1764±1.1892	-30.3595±0.7991	NA
SAE	-51.3509±0.6323	-34.2239±4.4788	NA
Adaptive SAE	-51.0977±0.6402	-36.4189±4.0396	**0.0021**
Proposed DNN	-42.4724±1.6812	**-40.5255±0.8252**	**0.2477**

- SAE, adaptive SAE, LSTM, and proposed DNN depend on **initialization**
  - Average MSE and standard deviation over 20 independent runs are given
- SAE and adaptive SAE achieve spectacular training performance but online **test MSE degrades** considerably
  - Adaptive SAE has smallest ACTpS, as it **only** adapts 4 linear weights
- Proposed DNN has **best test MSE** with ACTpS smaller than AGRBF
  - Dimension of deep feature space is much smaller than that of input space

Case 1: Test MSE learning curves

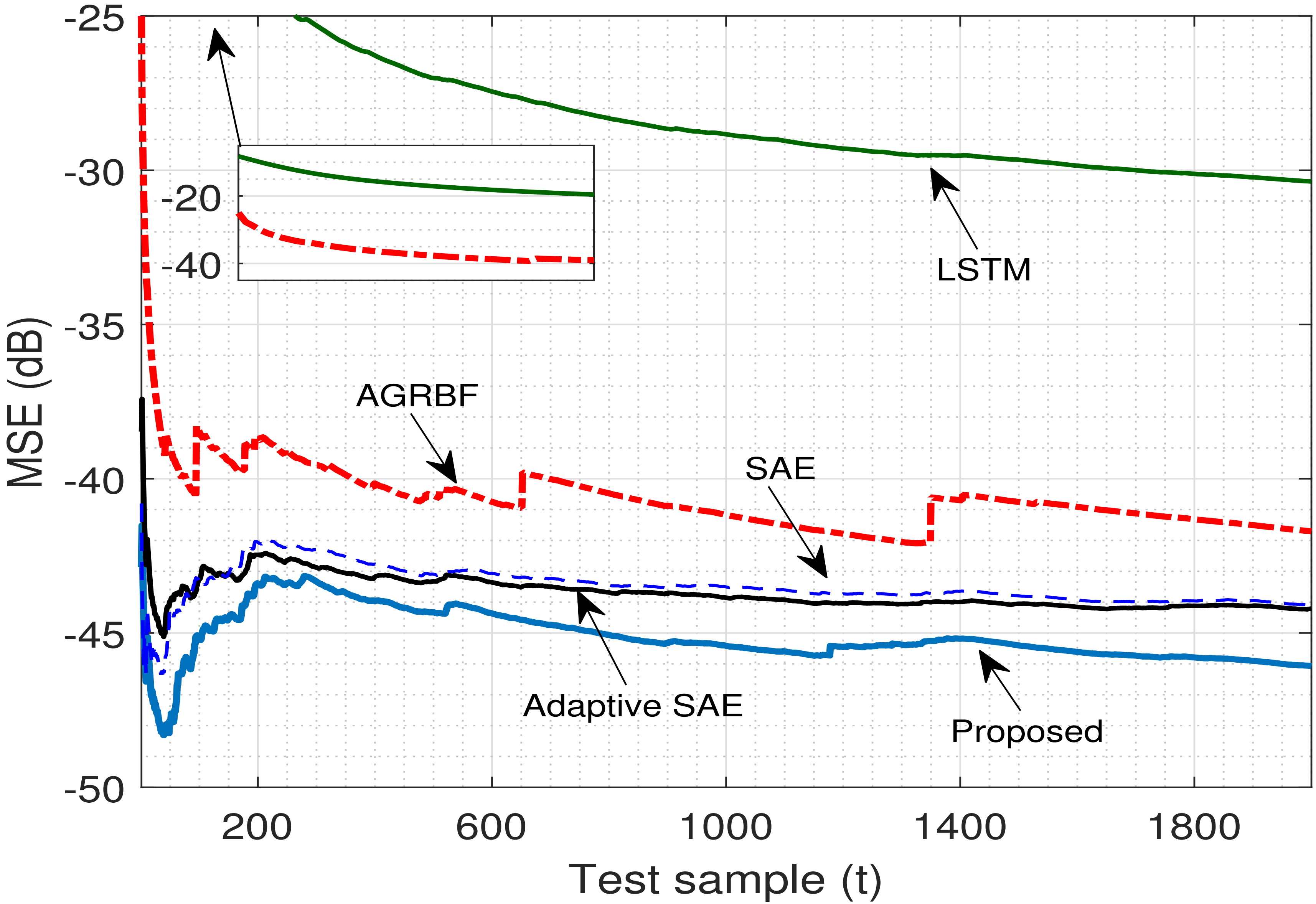


Case 2: Microwave Heating Process

Method	Initial training MSE (dB)	Online prediction/modeling MSE (dB)	Online prediction/modeling ACTpS (ms)
AGRBF	-30.1656	-41.7033	0.0372
LSTM	-30.0209±2.0842	-30.3644±1.5311	NA
SAE	-42.4144±4.5252	-44.0824±4.1411	NA
Adaptive SAE	-40.6789±9.8211	**-44.2089±4.5475**	**0.0020**
Proposed DNN	-43.9761±1.2653	**-46.0683±1.1564**	**0.0121**

- Nonstationaity of this process is not severe
- Proposed DNN has **best test MSE** with ACTpS smaller than AGRBF
  - Dimension of deep feature space is much smaller than that of input space
- Adaptive SAE has second best test MSE
  - Adaptive SAE has smallest ACTpS, as it **only** adapts 3 linear weights

Case 2: Test MSE Learning Curves

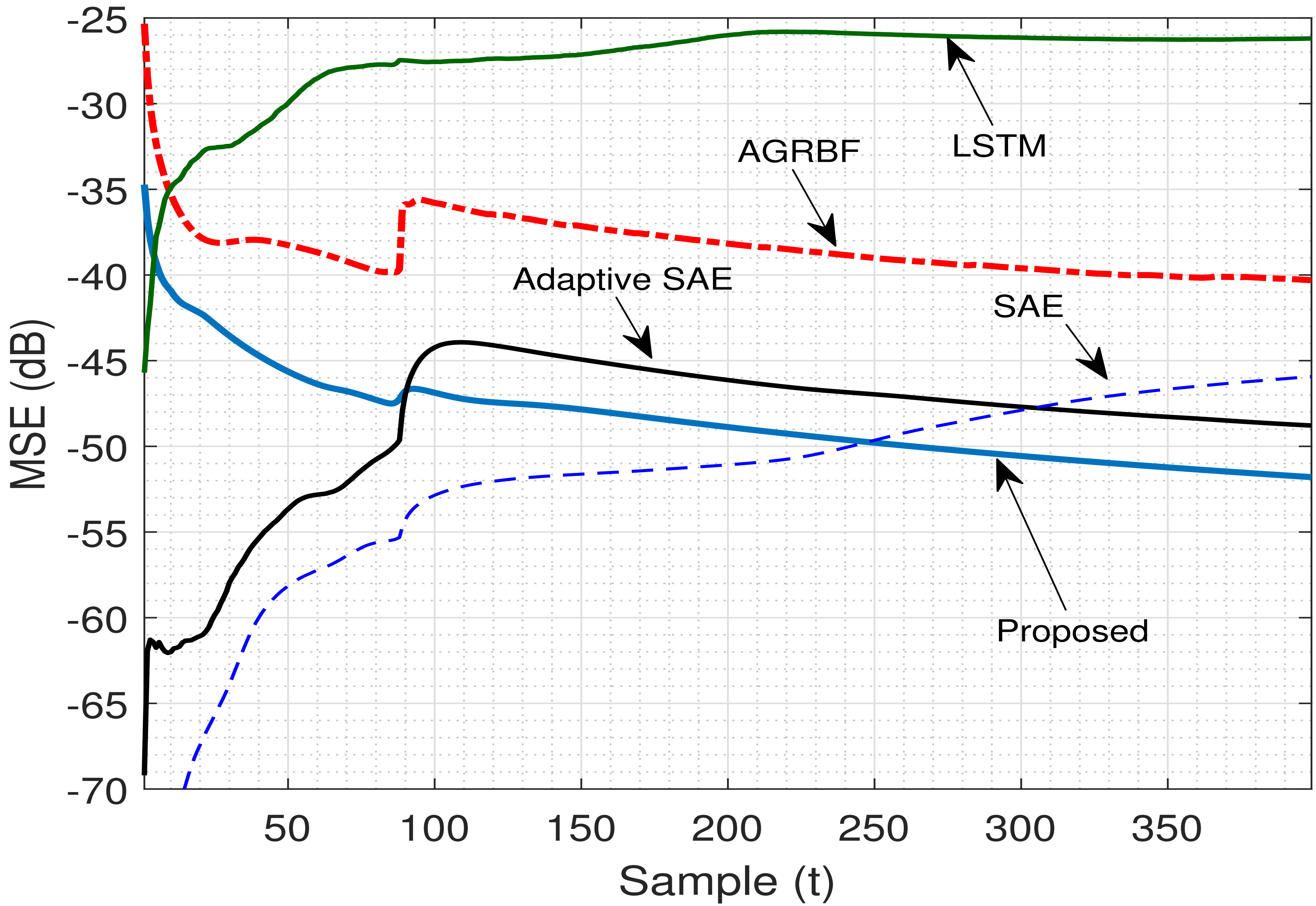


Case 3: Penicillin Fermentation Process

Method	Initial training MSE (dB)	Online prediction/modeling MSE (dB)	Online prediction/modeling ACTpS (ms)
AGRBF	-33.5760	-40.3049	0.0600
LSTM	-32.6484±4.7905	-26.1942±4.4434	NA
SAE	-89.0346±9.7129	-45.9289±6.4825	NA
Adaptive SAE	-73.4173±10.2068	**-48.7861±5.2173**	**0.0033**
Proposed DNN	-37.6025±1.1217	**-51.7963±1.9480**	**0.0498**

- SAE and adaptive SAE achieve spectacular training performance but online **test MSE degrades** considerably
- Proposed DNN has **best test MSE** with ACTpS smaller than AGRBF
  - Dimension of deep feature space is much smaller than that of input space
- Adaptive SAE has second best test MSE, and smallest ACTpS as it **only** adapts 4 linear weights

Case 3: Test MSE Learning Curves



Conclusions

- **Deep neural networks**, such as stacked autoencoder, has **deep nonlinear learning** capability, but it is **impossible to adapt** network structure online in real time

- **Shallow gradient RBF** network has **excellent adaptability**

- We have shown how to **integrate deep nonlinear learning** capability of SAE with **excellent adaptability** of adaptive GRBF

- Proposed deep neural network architecture is capable of adapting to changing underlying system dynamics in **real-time**
  - Particularly suitable for **online modeling** of **highly nonlinear and nonstationary** industrial processes