Growing and Pruning Selective Ensemble Regression over Drifting Data Streams

Sheng Chen

Next Generation Wireless

School of Electronics and Computer Science

University of Southampton

Southampton SO17 1BJ

United Kingdom

E-mail: sqc@ecs.soton.ac.uk

Joint work with Mr Tong Liu, School of Automation, Chongqing University

Plenary talk at 2019 International Conference on Intelligent Computing, Nanchang, China, August 3-6, 2019

How AI is saved

- Brief AI history
  - Birth of AI and initial hype (1950s - 1970s)
  - AI winter (1980s - 1990s)
  - Endeavour: We were not calling ourselves AI, but '**intelligent computing**'
  - Reborn to new hype, this time is real

- Who save AI
  - It is often mistaken AI as belonging to **Computer** Science
  - **Electronic and digital** revolution, **mobile communication** revolution lead to connected digital world, providing solid foundation for AI to recover

- My AI equation

$$\text{AI} = e_d \cdot C^2$$

  - e_d is **E**lectronic **D**igital infrastructure; C is **C**ommunication, C is **C**omputing

Motivations

- Most real-world systems and data are nonlinear and **nonstationary**
- **Standard machine learning** of collecting training data, identifying a model, and hoping it generalizes well **does not work**
- Key to success is to **update** learner's **structure** and model **parameters** online as new data are available
- Modeling over drifting data stream: well known **stability** and **plasticity** dilemma or tradeoff
  - Stability: ability to **retain acquired knowledge** for maintaining diversity
  - Plasticity: ability to **forget** part or all **past knowledge** in order to capture new knowledge as fast as possible

Existing Techniques

- Resource allocating network: **RAN** adds RBF nodes with arriving data based on their significance
  - Growing model size, end with very large model and high prediction complexity
- Online sequential extreme learning machine (**OS-ELM**) and ensemble OS-ELM (**EOS-ELM**):
  - Fixed large model size, only update parameters, no structure adaptation
- **Fast tunable RBF**: **Chen, Gong, Hong, Chen, "A fast adaptive tunable RBF network for nonstationary systems,"** ***IEEE Trans. Cybernetics*****, 46(12), 2683-2692, 2016**
  - Replace an insignificant RBF node with new data and update parameters online (Fixed small model size, update both structure and parameters)
  - Better prediction accuracy and lower online computational complexity than RAN, OS-ELM, EOS-ELM, and other existing methods

Proposed GAP-SER

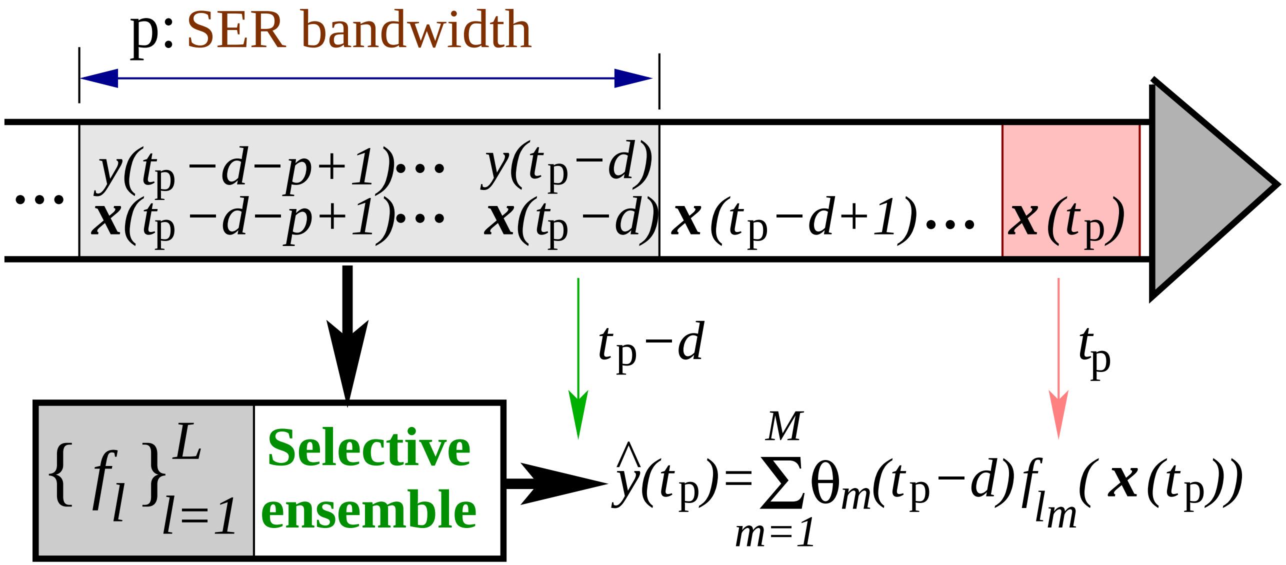
- Nonlinear and nonstationary data space partitioned with moving **window** as **local states**, each fitted with **local linear model**, yielding local model set $\{f_l\}_{l=1}^{L}$

- Growing and pruning selective ensemble regression:
  - Online prediction model is constructed as **selective ensemble** from $\{f_l\}_{l=1}^{L}$
  - **Growing** strategy: newly emerging process state is automatically identified and fitted with a local linear model
  - **Pruning** strategy: remove unwanted out of date local linear models

- Excellent stability (diversity) and plasticity properties
  - Superior online prediction accuracy and low computational complexity

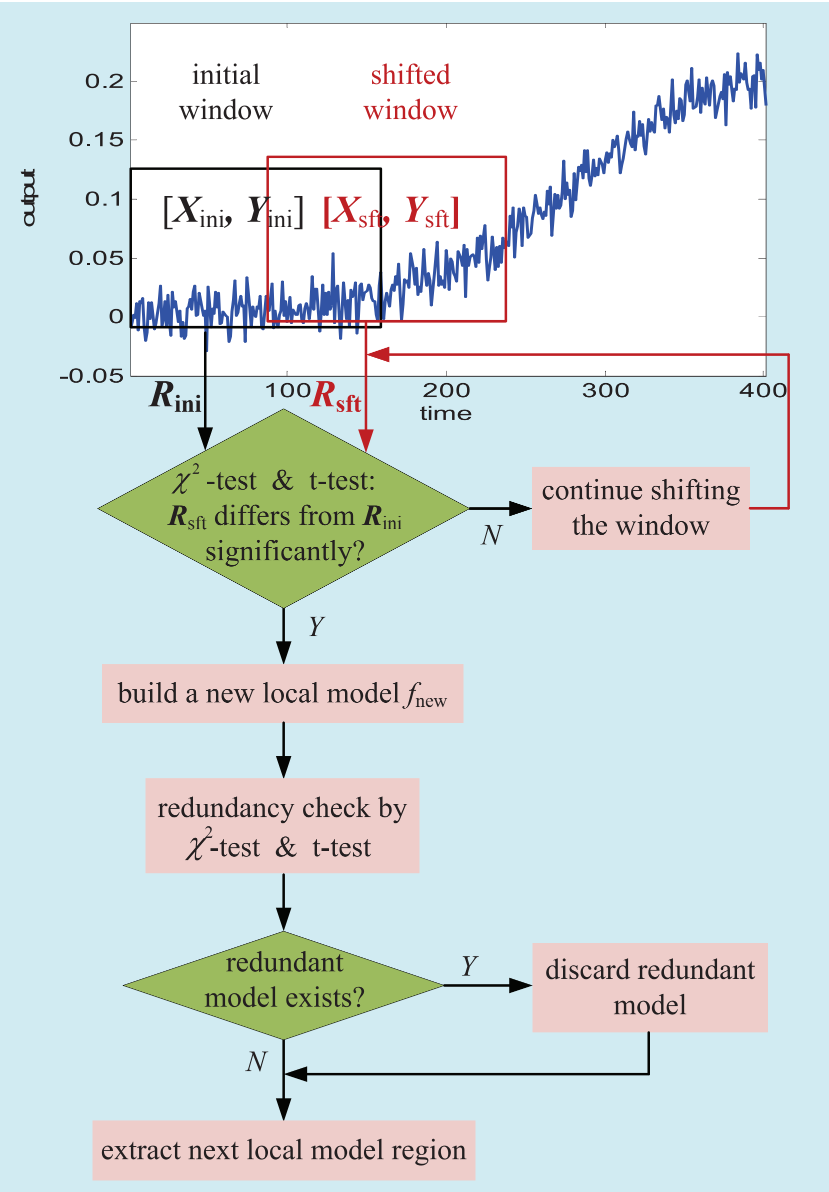
- **Liu, Chen, Liang, Harris, "Growing and pruning selective ensemble regression for nonlinear and nonstationary systems," submitted for publication**

Selective Ensemble Construction

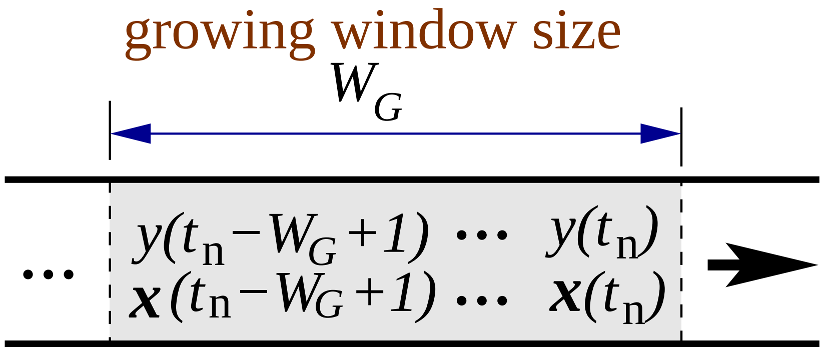


- At sample t_{p}, **SER** constructs **prediction** $\widehat{y}(t_{\mathrm{p}})$ of $y(t_{\mathrm{p}})$ by selecting M best local linear models $\{f_{l_m}\}_{m=1}^{M}$ from local model set $\{f_l\}_{l=1}^{L}$ using
  - p available samples $\big\{\boldsymbol{x}(t_{\mathrm{p}}-d-i), y(t_{\mathrm{p}}-d-i)\big\}_{i=0}^{p-1}$, where input $\boldsymbol{x}(t) \in \mathbb{R}^{m_o}$, output $y(t) \in \mathbb{R}$, $y(t_{\mathrm{p}}-d)$ is **newest** output sample available, and $d \geq 1$
- Standard **probability metric** is used for SER predictor construction

Growing Local Model Set



- At **next** sample $t_n = t_p - d + 1$, when **observation** $y(t_n)$ is available, move growing **window** one sample ahead



- **Grow** local linear model set when new process state is identified

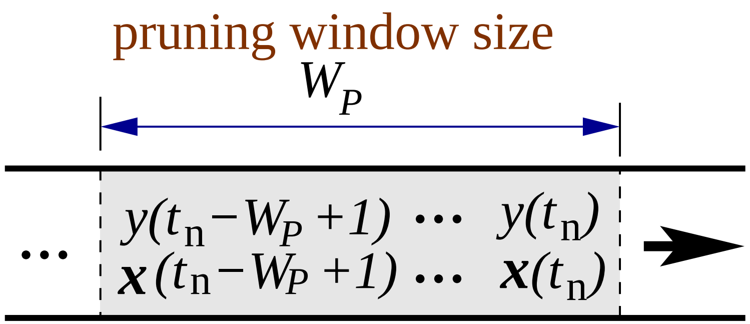
Shao, Tian, Wang, Deng, Chen, "Online soft sensor design using local partial least squares models with adaptive process state partition," *Chemometrics and Intelligent Laboratory Systems*, 144, 108-121, 2015

Pruning Local Model Set

- Over long online operation, local model set $\{f_l\}_{l=1}^{L}$ may grow to be very **large**, dramatically increasing **online** computational **complexity** of SER prediction

- Highly desired to remove out-of-date unwanted local models
  - Any local linear model in model set represents some past system **knowledge** actually occurred
  - Remove a local model not selected by current SER always runs **risk** it may be needed in future
  - How to remove 'unwanted' local model online without sacrificing diversity and accuracy of SER ?

- Our **pruning** strategy:
  - Over W_P consecutive samples, **only** if a local model has never been selected by SER predictor, it is removed from local model set



Algorithmic Parameters

- **Growing** window size W_G:
  - Small W_G leads to large number of local models, increasing online operating time but having better adaptation capability, large W_G has opposite efforts

- **Pruning** window size W_P: conveniently set to $W_P = W_G$
  - If a model is not needed consistently for current W_P prediction samples, probability it being selected in near future prediction samples is small
  - Remove 'oldest' or current 'worst'-performance local model may not be right
  - To maintain sufficient diversity, pruning will not take place if a minimum size $L_{\min}$ of local model set is reached

- **Selective ensemble regression** bandwidth p:
  - Trade off online complexity and performance – large p imposes high complexity but better robustness against noise, and small p has opposite efforts

Experimental Results

- Performance metrics
  - Test mean square error, **MSE**

$$\text{MSE}(t) = \frac{1}{t}\sum_{i=1}^{t}\big(y(i) - \widehat{y}(i)\big)^2$$

  - Online computational complexity measured by average computational time per sample, **ACTpS**

- Benchmark algorithms for comparison with our **GAP-SER**
  - **OS-ELM**: initial trained RBF model, online weight adaptation (fixed model size)
  - **EOS-ELM**: initial trained ensemble of RBF models, online weight adaptation (fixed model size)
  - **RAN**: initial zero RBF node, online growing model
  - **RANini**: initial trained RBF model, online growing model
  - Fast **tunable RBF**: initial trained RBF model, online replace worst node (fixed model size)

Lorenz Time Series

- **Lorzen** chaotic time series

$$\left\{ \begin{array}{lcl} \frac{\mathrm{d}\, x(t)}{\mathrm{d}\, t} & = & a(y(t) - x(t)) \\ \frac{\mathrm{d}\, y(t)}{\mathrm{d}\, t} & = & cx(t) - x(t)z(t) - y(t) \\ \frac{\mathrm{d}\, z(t)}{\mathrm{d}\, t} & = & x(t)y(t) - bz(t) \end{array} \right.$$

  with **time-varying** parameters:

$$a = 10, \quad b = \frac{4 + 3(1 + \sin(0.1t))}{3}, \quad c = 25 + 3\left(1 + \cos\left(2^{0.001t}\right)\right)$$

- 60-steps ahead **prediction** of $y(t)$ with

$$\boldsymbol{x}(t) = \left[y(t-60)\ y(t-66)\ y(t-72)\ y(t-78)\right]^{\mathrm{T}}$$

- **Mean** and **standard deviation** (STD) of test **MSE** and **ACTpS** over 100 realizations

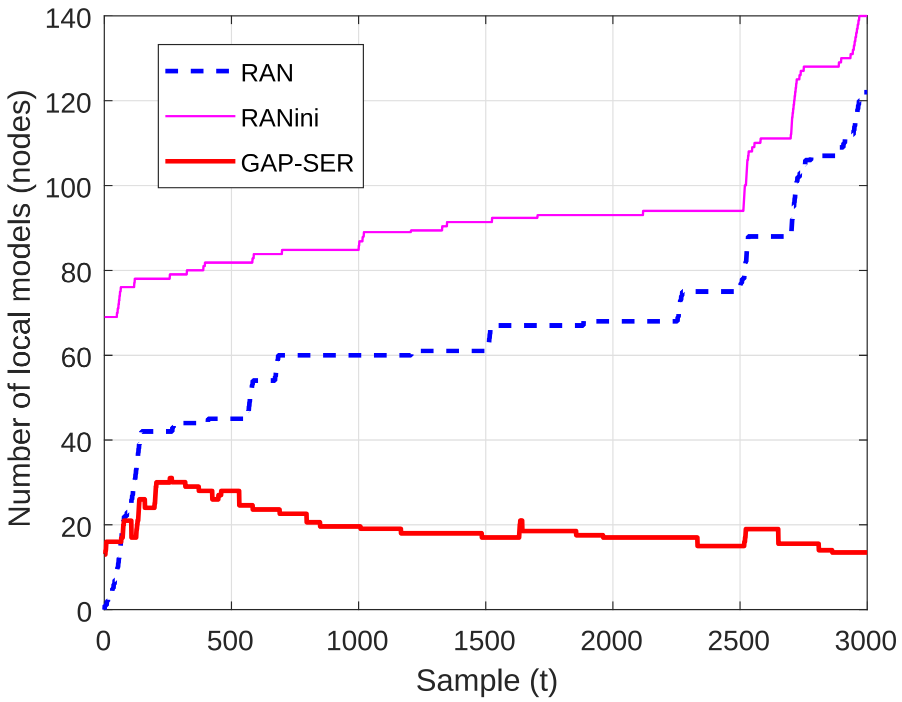
Lorenz Series Results

- First 1000 samples for initial training, last 3000 samples for online testing
- 100 random realizations are employed
- Comparison of online prediction and modeling performance (average±STD) for OS-ELM, EOS-ELM, RAN, RANini, fast tunable RBF, and proposed GAP-SER

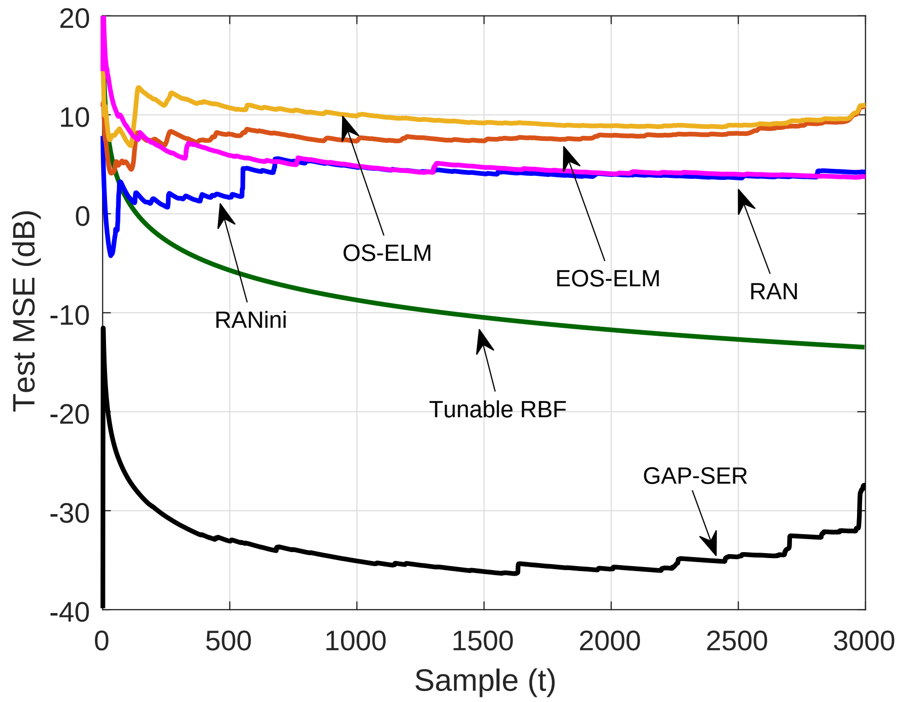
Method	MSE (dB)	ACTpS (ms)	Local models/RBF Nodes		Average ensemble size
			Initial	Final	
OS-ELM	10.87±0.01	6.21±0.31	500	500	-
	10.86±0.01	37.14±0.22	1000	1000	-
EOS-ELM	11.01±0.01	58.12±1.45	5×500	5×500	5
RAN	3.83±0.02	0.66±0.01	0	122±0	-
RANini	4.21±0.04	1.02±0.02	69±0	139.97±0.17	-
Tunable RBF	**-13.48±0.56**	**0.17±0.01**	**10**	**10**	-
GAP-SER	**-27.42±0.63**	**0.24±0.01**	**13±0**	**13.47±0.50**	**9.88±0.07**

(a) Learning curves of average numbers of local linear models (GAP-SER)/RBF nodes (RAN, RANini)

(b) Learning curves of average test MSEs of OS-ELM, EOS-ELM, RAN, RANini, fast tunable RBF, and GAP-SER

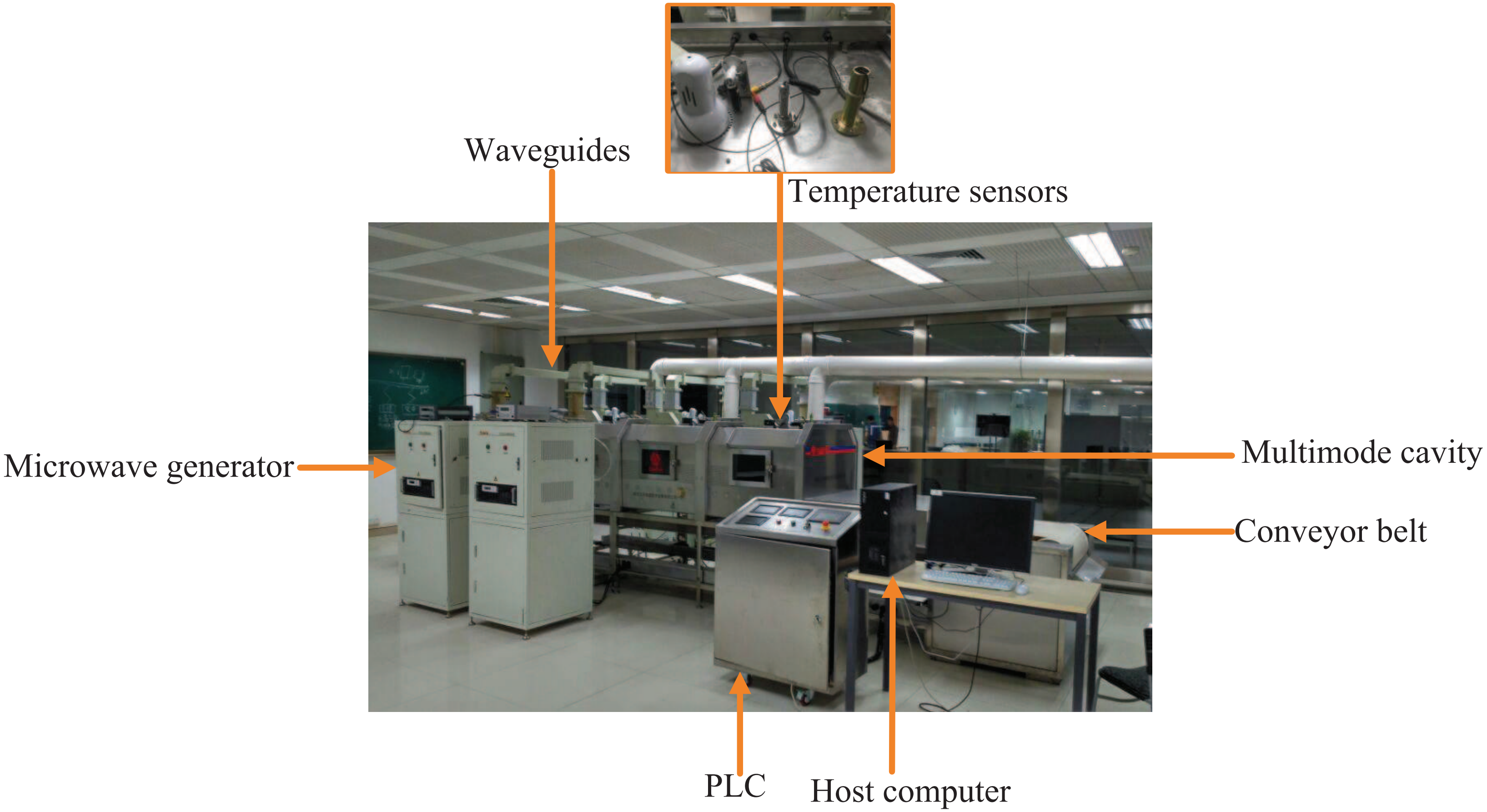


(a)



(b)

Industrial Microwave Heating System



- **Control** inputs $\boldsymbol{u}(t) = \left[u_{p_1}(t)\ u_{p_2}(t)\ u_{p_3}(t)\ u_{p_4}(t)\ u_{p_5}(t)\ v(t)\right]^{\mathrm{T}}$
  - $u_{p_i}(t)$, $1 \le i \le 5$: **microwave powers** for five microwave generators
  - $v(t)$: **conveyor speed** to cavity

Microwave Heating System

- Three **fiber optical sensors**, FOS1, FOS2 and FOS3, record multiple-points of **temperature**, $y_{s_j}(t)$, $1 \le j \le 3$

- Accurate **predictions** of $y_{s_j}(t)$ are crucial to detect **thermal runaway** in advance

- Task is to construct online **adaptive** predictors of $y_{s_j}(t)$, $1 \le j \le 3$:

$$\widehat{y}_{s_j}(t) = f_{\text{nl}-\text{ns},j}(\boldsymbol{x}_j(t); t), \;\; 1 \le j \le 3,$$

with input vector $\boldsymbol{x}_j(t) = \left[y_{s_j}(t-1) \;\; \boldsymbol{u}^{\text{T}}(t-1) \right]^{\text{T}}$

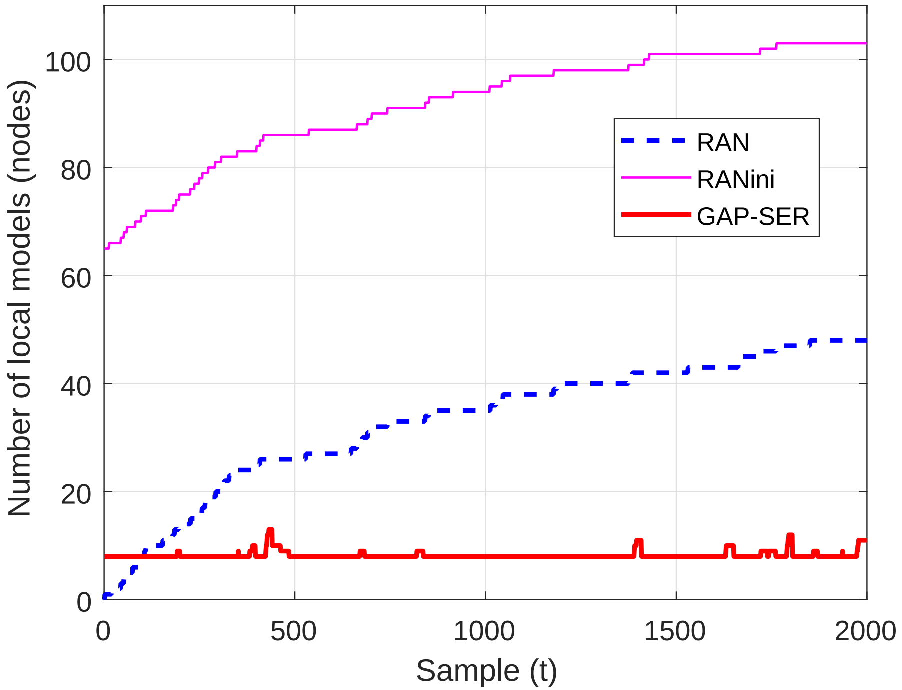
- For each FOS's dataset, first 1000 samples for initial **training** and and last 2000 samples for online **prediction**

On-line Temperture Prediction Results

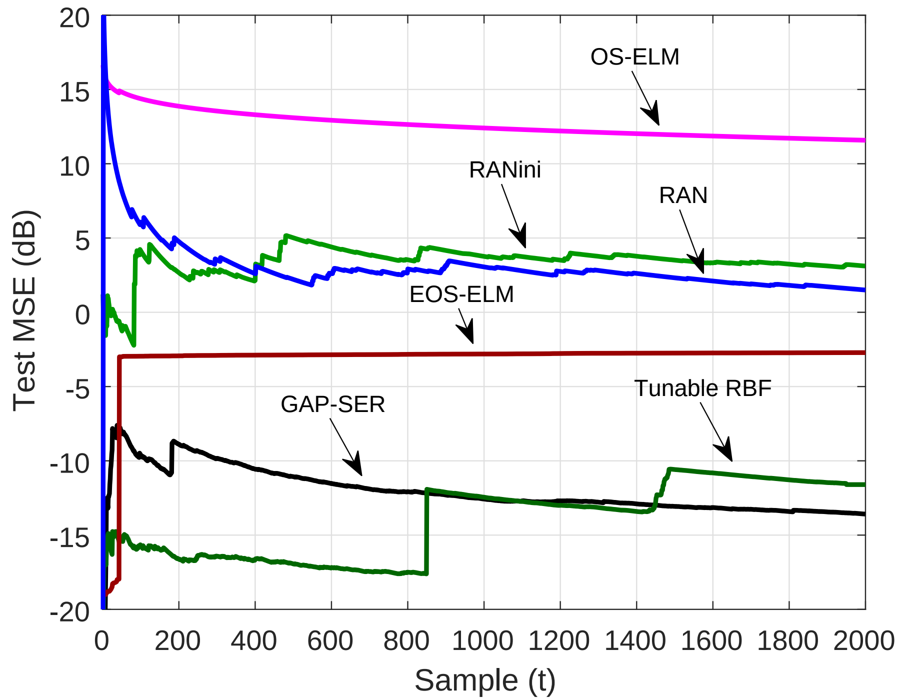
Dataset	Method	MSE (dB)	ACTpS (ms)	Local models/RBF Nodes		Average ensemble size
				Initial	Final	
FOS1	OS-ELM	18.4488	0.43	100	100	-
	EOS-ELM	6.9665	2.38	5×100	5×100	5
	RAN	1.4990	0.56	0	48	-
	RANini	3.113	1.91	65	103	-
	Tunable RBF	**-11.6108**	**0.35**	**10**	**10**	-
	GAP-SER	**-13.7951**	**0.20**	**8**	**11**	**4.15**
FOS2	OS-ELM	12.6390	0.44	100	100	-
	EOS-ELM	7.7460	2.50	5×100	5×100	5
	RAN	4.9071	0.24	0	25	-
	RANini	7.0560	0.98	45	67	-
	Tunable RBF	**-13.5971**	**0.38**	**10**	**10**	-
	GAP-SER	**-14.0198**	**0.27**	**18**	**18**	**8.94**
FOS3	OS-ELM	13.5877	0.43	100	100	-
	EOS-ELM	8.1725	2.89	5×100	5×100	5
	RAN	3.5136	0.40	0	35	-
	RANini	3.1695	1.02	48	71	-
	Tunable RBF	**-13.1200**	**0.34**	**10**	**10**	-
	GAP-SER	**-13.4187**	**0.22**	**18**	**18**	**4**

- Online temperature prediction of **FOS1**

(a) Learning curves of numbers of local linear models (GAP-SER)/RBF nodes (RAN, RANini)
(b) Learning curves of test MSEs of OS-ELM, EOS-ELM, RAN, RANini, fast tunable RBF, and GAP-SER



(a)

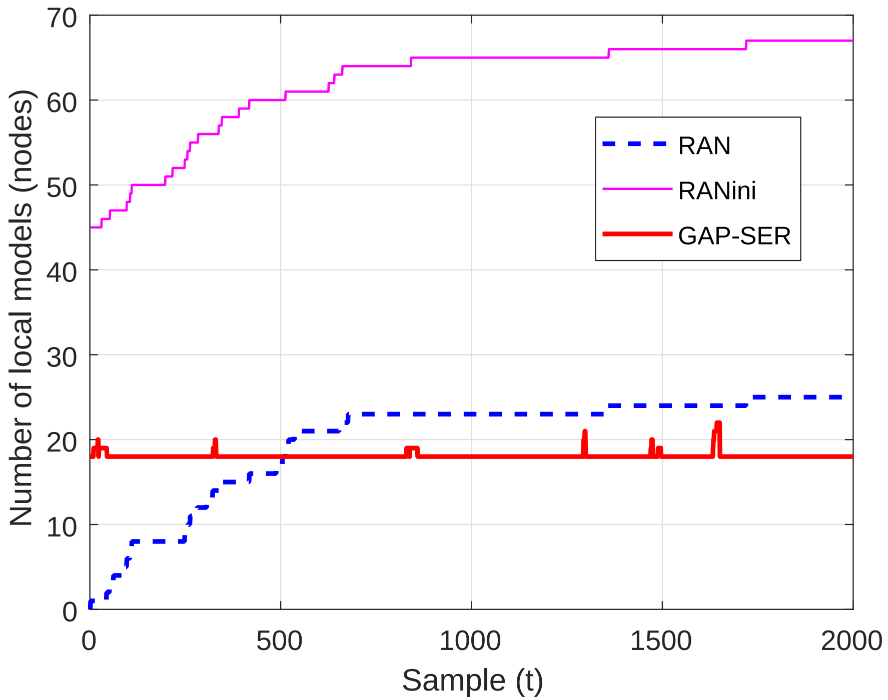


(b)

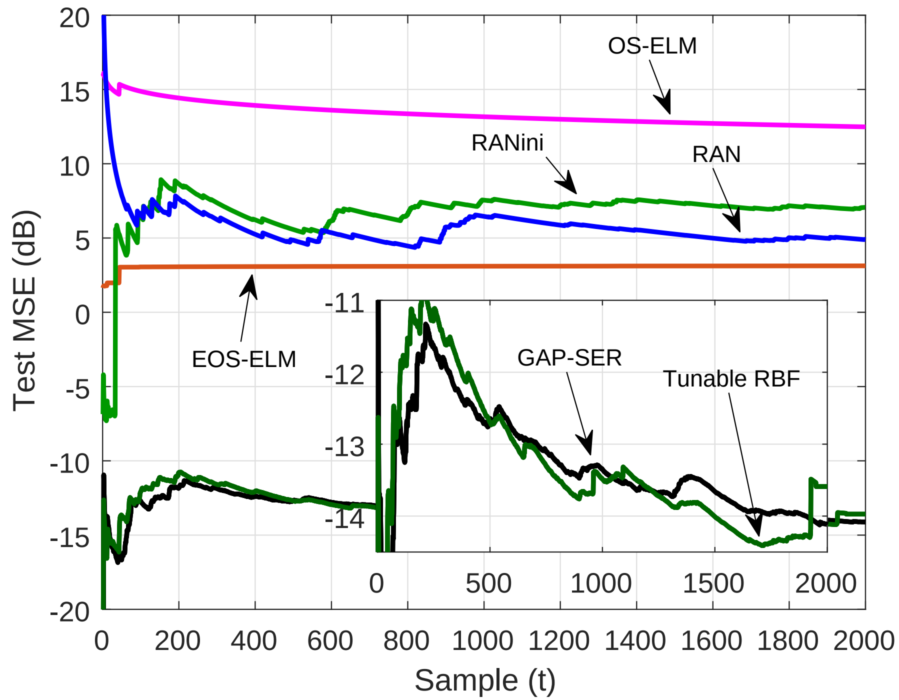
- Online temperature prediction of **FOS2**

(a) Learning curves of numbers of local linear models (GAP-SER)/RBF nodes (RAN, RANini)

(b) Learning curves of test MSEs of OS-ELM, EOS-ELM, RAN, RANini, fast tunable RBF, and GAP-SER



(a)

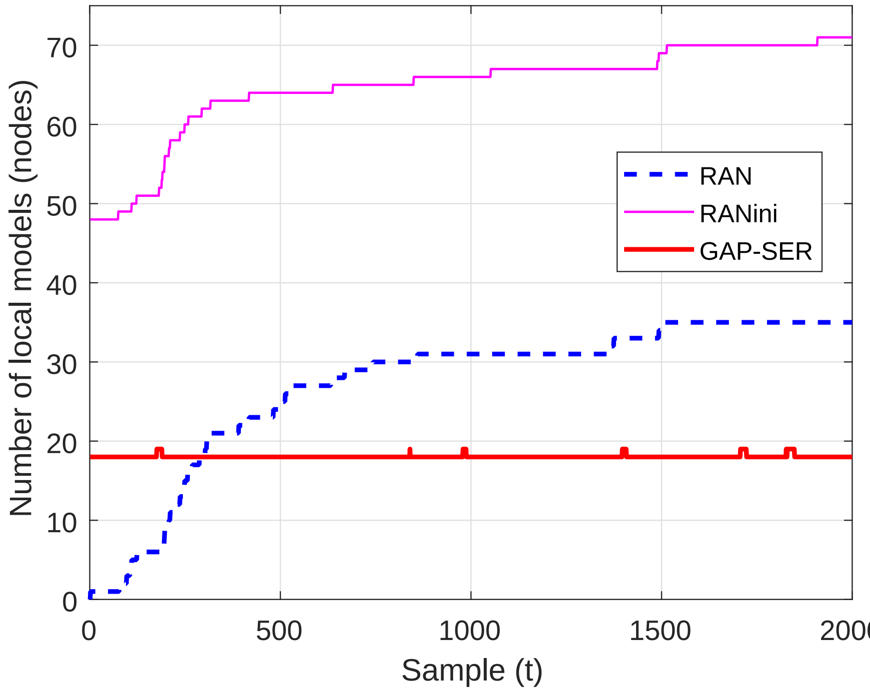


(b)

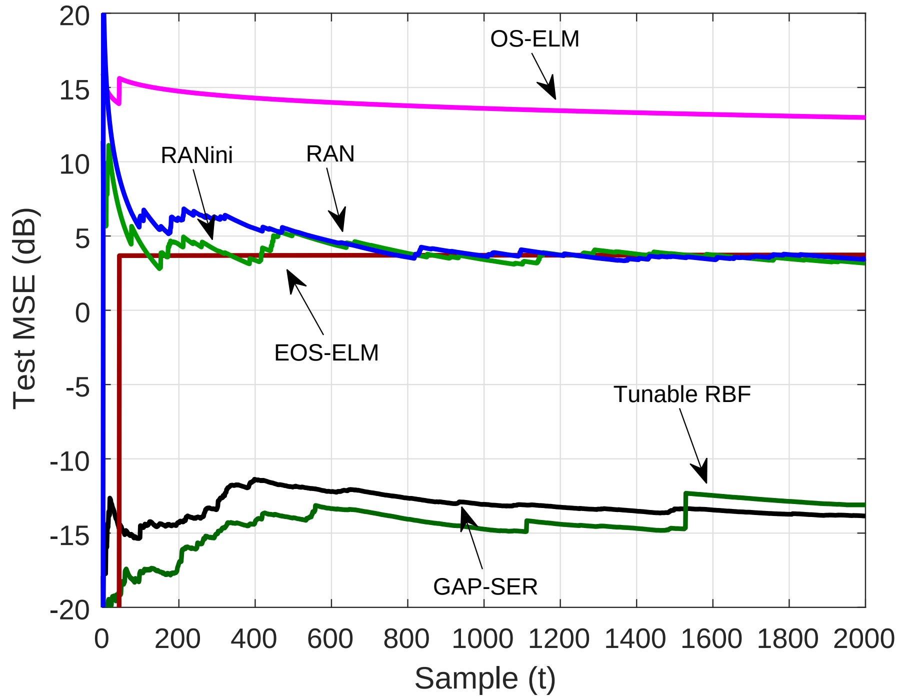
- Online temperature prediction of **FOS3**

(a) Learning curves of numbers of local linear models (GAP-SER)/RBF nodes (RAN, RANini)

(b) Learning curves of test MSEs of OS-ELM, EOS-ELM, RAN, RANini, fast tunable RBF, and GAP-SER



(a)



(b)

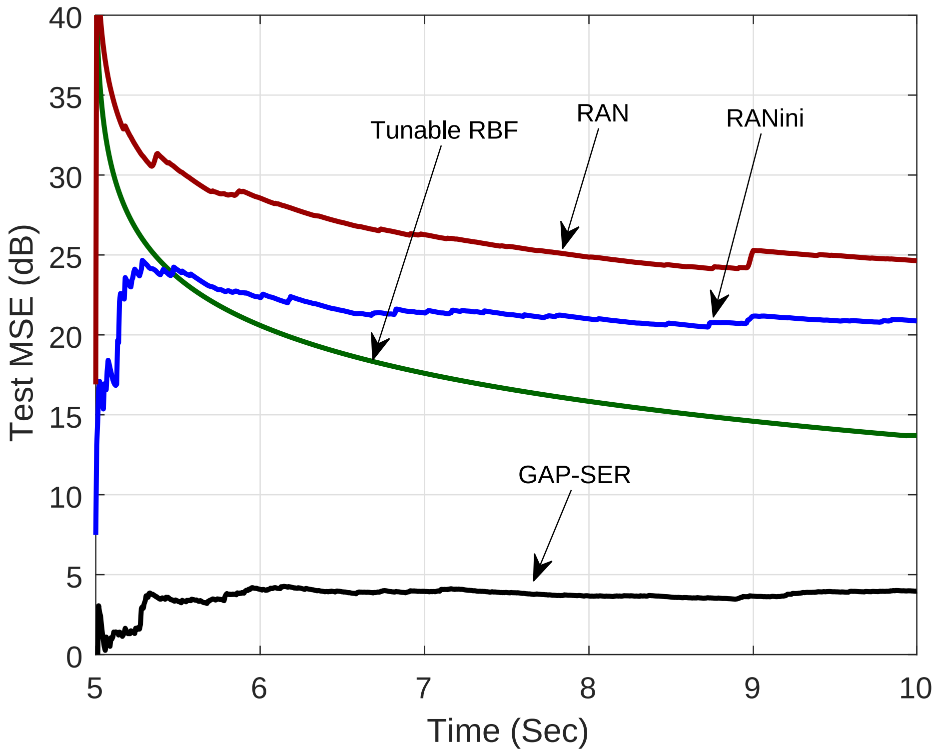
EEG Data Modeling

- **EEG** time series publicly available from the University of Bonn.
  - At sampling rate 173.61 Hz, **dataset** $D_N = \{\boldsymbol{x}(t), y(t)\}_{t=1}^{N}$ with 10 seconds length (1730 samples), where $\boldsymbol{x}(t) = [y(t-1)\ y(t-2)\ y(t-3)\ y(t-4)]^{\mathrm{T}}$
  - First 5 seconds (865 data pairs) used for initial training, and rest 5 seconds (865 data pairs) used for testing
- Comparison of **online prediction** and modeling performance for RAN, RANini, fast tunable RBF, and proposed GAP-SER

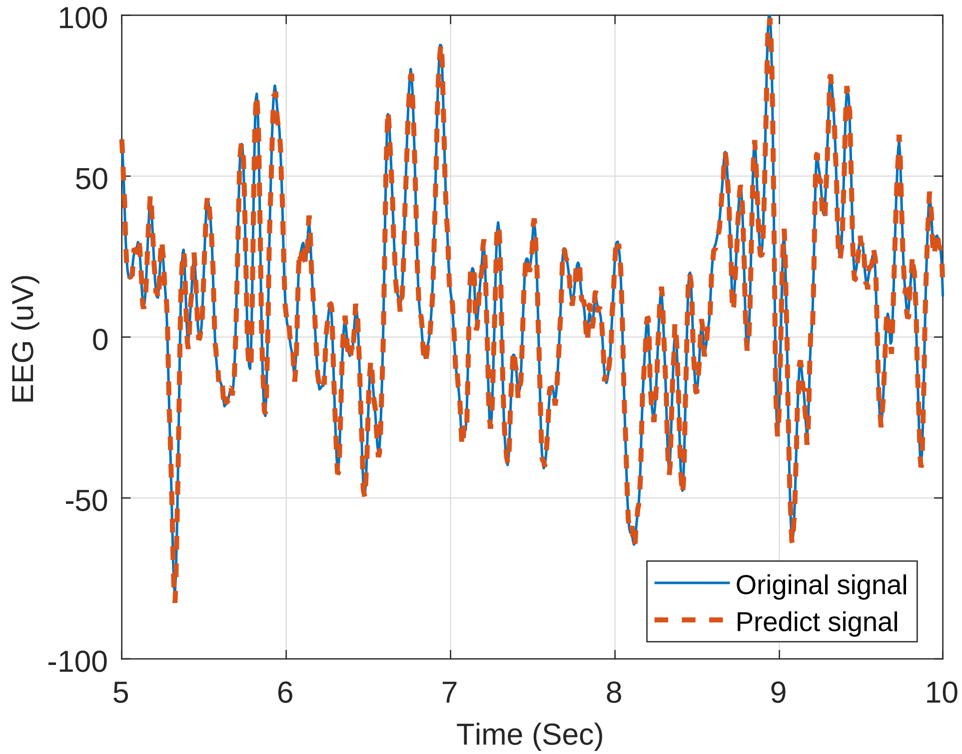
Method	MSE (dB)	ACTpS (ms)	Local models/RBF Nodes		Average ensemble size
			Initial	Final	
RAN	24.6432	1.37	0	106	-
RANini	20.8746	1.81	103	107	-
Tunable RBF	**13.6452**	**0.41**	**10**	**10**	-
GAP-SER	**3.9840**	**0.42**	**2**	**3**	**1.97**

(a) Learning curves of test MSEs of RAN, RANini, fast tunable RBF, and GAP-SER

(b) Comparison of recovered signal by GAP-SER and original EEG observations



(a)



(b)

Conclusions

- For **nonlinear** and fast **time-varying** systems and data, **traditional** machine learning relying on training does not work
  - Adapt model weights is inadequate
  - Crucial to adjust model structure as well as adapt model weights
  - Adaptation must be **fast**, **accurate**, imposing **low online** computational time

- Our growing and pruning selective ensemble regression offers **state-of-the-art** online modeling/prediction of highly nonlinear and nonstationary data
  - **Growing** strategy automatically identifies newly emerging local linear models
  - **Pruning** strategy reliably removes unwanted out-of-date local models
  - **Selective ensemble** regression adaptively constructs accurate online predictor

- This GAP-SER balances **stability** and **plasticity** well, maintains sufficient **diversity**, provides highly **accurate** adaptive modeling while imposing very low **online** computational complexity