Learning in heterogeneous, high-dimensional, and distributed scenarios: statistical approaches towards hybrid and trustworthy AI.

Faïcel Chamroukhi

Seminar @ LaMSN – April 30, 2025

Scientific Challenges

- **Motivation :** Modern ML/AI must handle heterogeneous, often unlabeled, high-dimensional, and distributed data.
- **Challenges :**
  - Sparsity and interpretability, scalability
  - Privacy, uncertainty quantification, distributed computations
  - Exploiting prior knowledge (eg. structures, physics)

Scientific framework

↪ Latent variable **models** : $f(x|\boldsymbol{\theta}) = \int_{\mathcal{Z}} f(x, z|\boldsymbol{\theta}) \mathrm{d}z$

↪ **Learning**, **unsupervised representation** and **Selection** in high-dimension

↪ **A Numerical/statistical learning approch :** learning perspective focused on

- the design of latent variable models
- with learning and selection guarantees and approximation capabilities
- upon regularization enabling hybridization and training fostering trust.

Outline

Guidelines

- Design models that allow well-principled (with statistical guarantees) predictions in heterogeneous, high-dimensional and distributed situations,
- Construct efficient algorithms that operate in unsupervised way and provide interpretable solutions with computational guarantees.

Learning with high-dimensional predictors

Questioning : Prediction (non-linear regr., classification) & clustering in presence of

[1.] **High-dimensional** predictors : $\boldsymbol{X}_i \in \mathbb{R}^p$ with $p \gg n$, in a heterogeneous population and complex distributions

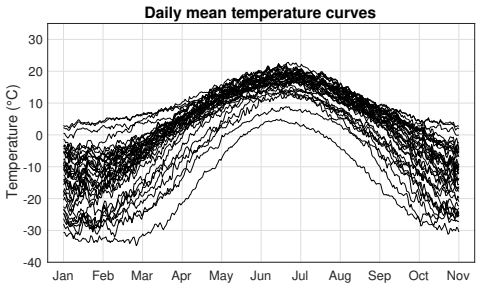
Questioning : Prediction (non-linear regr., classification) & clustering in presence of

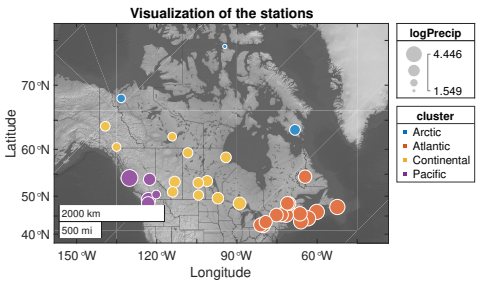
[1.] **High-dimensional** predictors : $\boldsymbol{X}_i \in \mathbb{R}^p$ with $p \gg n$, in a heterogeneous population and complex distributions

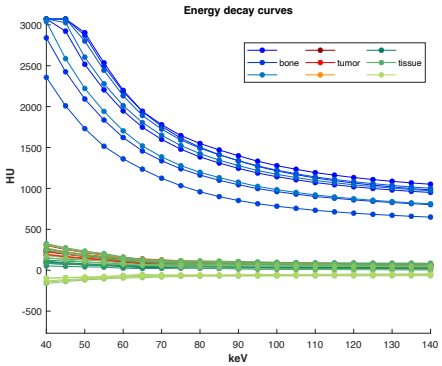
[2.] **Functional** predictors : $X_i(t)$, $t \in \mathcal{T} \subseteq \mathbb{R}$ (eg. continuously recorded variables)

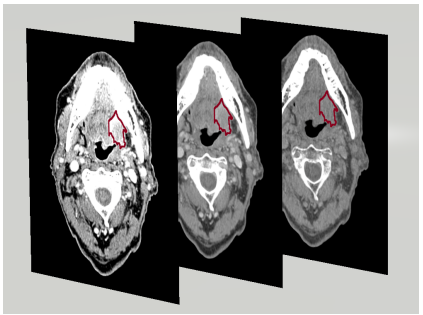
$\hookrightarrow$ Look for parsimonious and interpretable models

e.g : Relate functional predictors $\{X(t) \in \mathbb{R}; t \in \mathcal{T} \subset \mathbb{R}\}$ to a scalar response $Y \in \mathcal{Y} \subset \mathbb{R}$







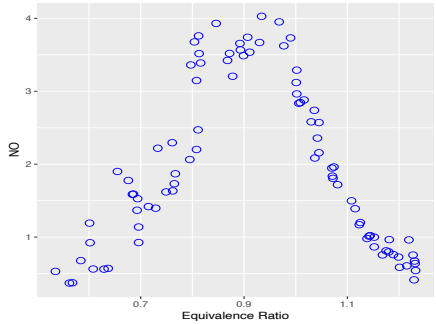


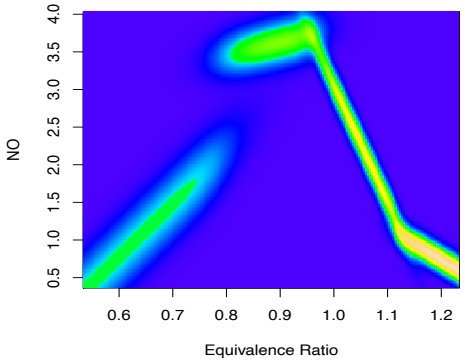
Functional regression (e.g. Roche'23 EJS, Ramsay an Silverman'05 (FDA)) doesn't work

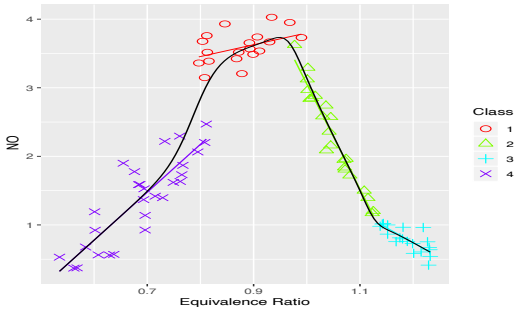
$Y_i = \beta_0 + \int_{\mathcal{T}} X_i(t)\beta(t)dt + \varepsilon_i$

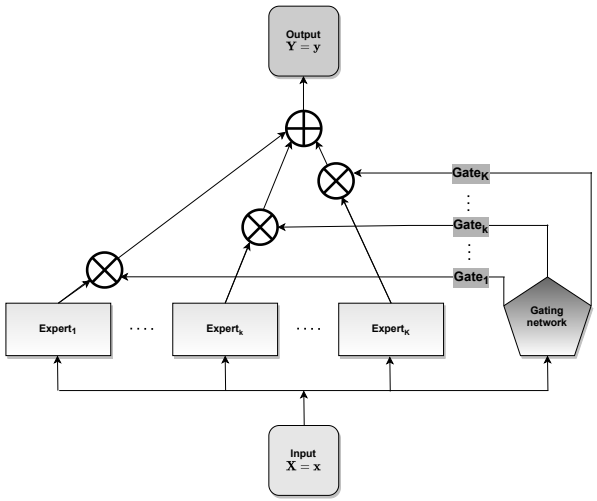
Mixtures-of-Experts to model heterogeneous data

Mixtures-of-Experts as good candidates to model a response Y given predictor.s X governed by a hidden structure accounting for heterogeneity









Schematic diagram of the neural network architecture of a K-component MoE model.

– first studied as neural networks (NNs) by Jacobs, Jordan, Nowlan, and Hinton (1991)

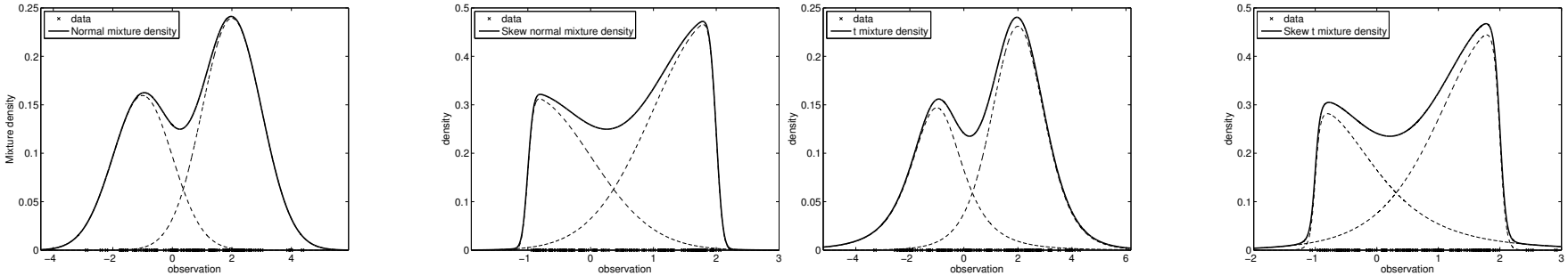
– Nguyen and Chamroukhi. Practical and theoretical aspects of mixture-of-experts modeling : An overview. WIRES : Data Mining and Knowledge Discovery Wiley Periodicals, Inc. 2018

Principled robustness in learning with MoE

- **Questionings :** Prediction (non-linear regr., classification) & clustering in presence of **Outliers**, with potentially **skewed**, **heavy-tailed** distributions
- **Answering :** Robust MoE that accommodate asymmetry, heavy tails, and outliers

$$m(y|\boldsymbol{r},\boldsymbol{x};\boldsymbol{\theta}) = \sum_{k=1}^{K} \underbrace{g_k(\boldsymbol{r};\boldsymbol{\alpha})}_{\text{Softmax Gating Network}} \underbrace{\mathcal{ST}(y;\mu(\boldsymbol{x};\boldsymbol{\beta}_k),\sigma_k,\lambda_k,\nu_k)}_{\text{Skew-}t\text{ Expert Network}}$$

kth expert : has a skew t distribution [Azzalini and Capitanio 2003]



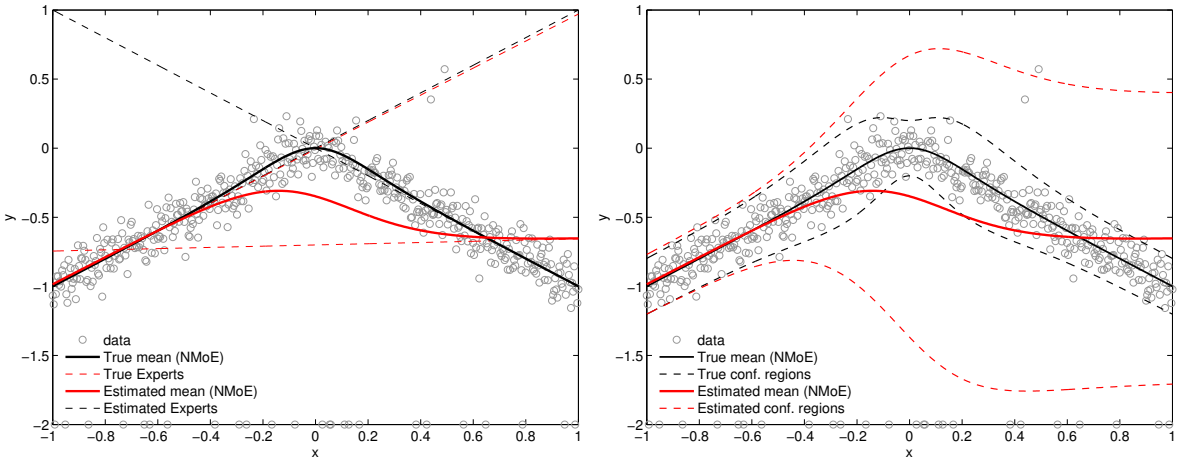
$\pi_k = [0.4, 0.6]$, $\mu_k = [-1, 2]$; $\sigma_k = [1, 1]$; $\nu_k = [3, 7]$; $\lambda_k = [14, -12]$;

Flexible and robust generalization of the standard MoE models

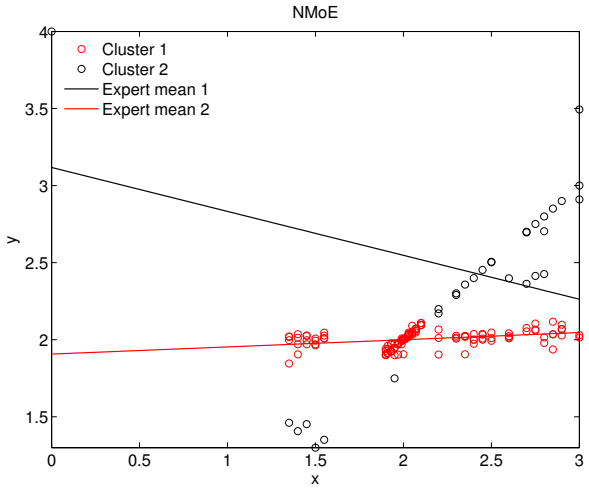
For $\{\nu_k\} \to \infty$, STMoE reduces to SNMoE ; For $\{\lambda_k\} \to 0$, STMoE reduces to TMoE.
For $\{\nu_k\} \to \infty$ and $\{\lambda_k\} \to 0$, StMoE approaches the NMoE.

Chamroukhi. Skew t mixture of experts. *Neurocomputing*, 266 :390–408, 2017.
Chamroukhi. Robust mixture of experts modeling using the t-distribution. *Neural Networks*, 79 :20–36, 2016.
Chamroukhi. Skew-normal mixture of experts. IJCNN 2016.

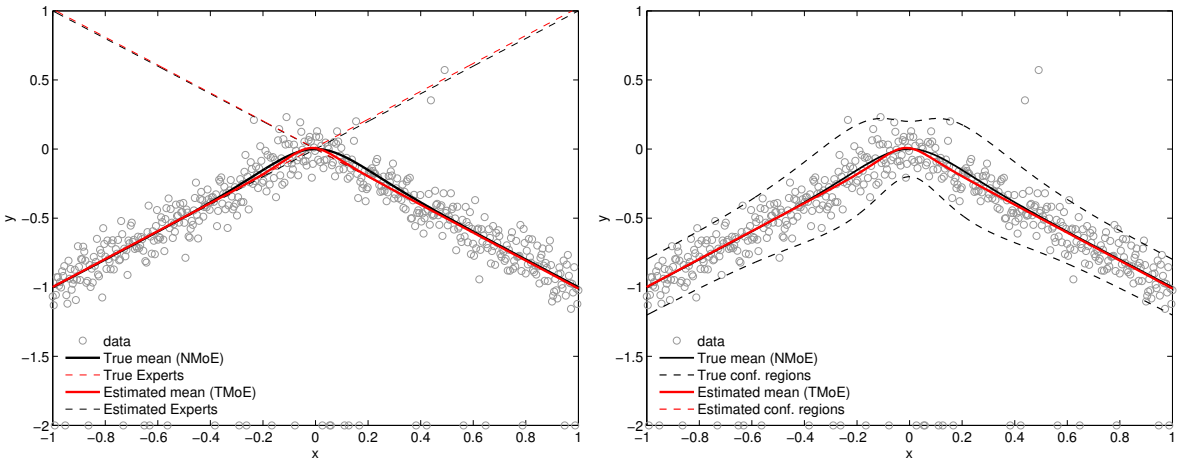
Robust learning with mixtures-of-experts models



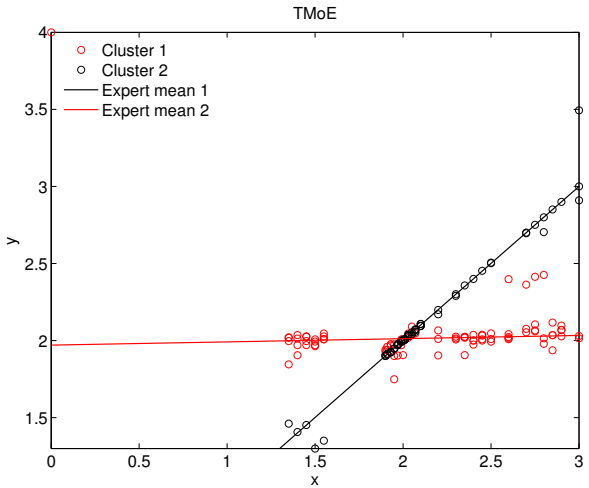
$n = 500$ observations with 5% of outliers $(x; y = -2)$: Normal fit



Tone data with 10 outliers $(0, 4)$: Normal fit



$n = 500$ observations with 5% of outliers $(x; y = -2)$: **Robust** fit



Tone data with 10 outliers $(0, 4)$: **Robust** fit

Training framework using the EM algorithm

$$\boldsymbol{\theta}^{new} \in \arg\max_{\boldsymbol{\theta}\in\Omega} \mathbb{E}[\ln L_c(\boldsymbol{\theta})|\mathcal{D}, \boldsymbol{\theta}^{old}]$$

complete log-likelihood : $\log L_c(\boldsymbol{\theta}) = \sum_{i=1}^{n}\sum_{k=1}^{K} \mathbb{I}_{\{Z_i=k\}} \log[G_k E_k]$

Training framework using the EM algorithm

$$\boldsymbol{\theta}^{new} \in \arg\max_{\boldsymbol{\theta}\in\Omega} \mathbb{E}[\ln L_c(\boldsymbol{\theta})|\mathcal{D}, \boldsymbol{\theta}^{old}]$$

complete log-likelihood : $\log L_c(\boldsymbol{\theta}) = \sum_{i=1}^{n}\sum_{k=1}^{K} \mathbb{I}_{\{Z_i=k\}} \log[G_k E_k]$

MEteorits : open-source soft. Robust learning with mixtures-of-experts models

MEteorits : **M**ixtures-of-**E**xper**T**s mod**E**ling for c**O**mplex and non-no**R**mal d**I**s**T**ributionS

▸ Github ▸ CRAN ▸ Matlab software

Available algorithms and Packages

`NMoE` : Normal Mixture-of-Experts	▸ R software	▸ Matlab software
`SNMoE` : Skew-Normal Mixture-of-Experts	▸ R software	▸ Matlab software
`tMoE` : Robust MoE using the t-distribution	▸ R software	▸ Matlab software
`StMoE` : Skew-t Mixture-of-Experts	▸ R software	▸ Matlab software

- Meteorits include sampling, fitting, prediction, clustering with each MoE model
- Non-normal mixtures (and MoE) is a very recent topic in the field

Approximation capabilities

Approximation capabilities of finite mixture distributions

Density approximation in Unsupervised Learning

- **Data** : observations $\{\boldsymbol{x}_i\}$ from $\boldsymbol{X} \in \mathbb{X} \subset \mathbb{R}^d$ of density (multimodal) $f \in \mathcal{F}$
- **Objective** : approximate the density f (and represent the data, e.g. *clustering*)
- **Solution :** Approximate f within the class $\mathcal{H}^{\varphi} = \bigcup_{K \in \mathbb{N}^{\star}} \mathcal{H}_K^{\varphi}$ of finite location-scale mixture h_K^{φ} (of K-components) of density φ (e.g., Gaussian), where

$$\mathcal{H}_K^{\varphi} = \left\{ \boxed{h_K^{\varphi}(\boldsymbol{x}) := \sum_{k=1}^{K} \pi_k \frac{1}{\sigma_k^d} \varphi\left(\frac{\boldsymbol{x} - \boldsymbol{\mu}_k}{\sigma_k}\right)}, \boldsymbol{\mu}_k \in \mathbb{R}^d, \sigma_k \in \mathbb{R}_+, \pi_k > 0 \, \forall k \in [K], \sum_{k=1}^{K} \pi_k = 1 \right\}$$

Theorem : Universal approximation of finite location-scale mixtures

(a) Given any p.d.f $f, \varphi \in \mathcal{C}$ and a compact set $\mathbb{X} \subset \mathbb{R}^d$, there exists a sequence $(h_K^{\varphi}) \subset \mathcal{H}^{\varphi}$, such that $\lim_{K \to \infty} \sup_{\boldsymbol{x} \in \mathcal{X}} |f(\boldsymbol{x}) - h_K^{\varphi}(\boldsymbol{x})| = 0$.

(b) For $p \in [1, \infty)$, if $f \in \mathcal{L}_p$ (Lebesgue p.d.f) and $\varphi \in \mathcal{L}_{\infty}$ (essentially bounded p.d.f), there exists a sequence $(h_K^{\varphi}) \subset \mathcal{H}^{\varphi}$, such that $\lim_{K \to \infty} \|f - h_K^{\varphi}\|_{\mathcal{L}_p} = 0$.

– Nguyen, Chamroukhi, Nguyen, & McLachlan (2023). Approximation of probability density functions via location-scale finite mixtures in Lebesgue spaces. *Communications in Statistics - Theory and Methods*, 52(14), 5048–5059.

Modeling with mixtures-of-experts (ME)

- **Context :** n observations $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}$ from a pair $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathbb{X} \times \mathbb{Y}$ with unknown conditional p.d.f $f \in \mathcal{F} = \left\{f : \mathbb{X} \times \mathbb{Y} \to \mathbb{R}_+ \middle| \int_{\mathbb{Y}} f(\boldsymbol{y}|\boldsymbol{x}) \, \mathrm{d}\lambda(\boldsymbol{y}) = 1, \forall \boldsymbol{x} \in \mathbb{X}\right\}$
- **High-dimensional** : $\mathbb{X} \subseteq \mathbb{R}^d$, $\mathbb{Y} \subseteq \mathbb{R}^q$, with $d, q \gg n$ and *heterogeneous* setting.
- **Objectives :** Regression ; Clustering ; Model selection
- **Solution :** Approximate f within the class of **mixtures-of-experts** :

Let φ be a p.d.f (compactly supported on $\mathbb{Y} \subseteq \mathbb{R}^q$), we define the functional classes :

- Location-scale family : $\mathcal{E}_\varphi = \left\{\varphi_q(\boldsymbol{y}; \boldsymbol{\mu}, \sigma) := \frac{1}{\sigma^q}\varphi\left(\frac{\boldsymbol{y}-\boldsymbol{\mu}}{\sigma}\right); \boldsymbol{\mu} \in \mathbb{Y}, \sigma \in \mathbb{R}_+\right\}$.
- Mixture of location-scale experts with softmax activation network : SGaME :

$$\mathcal{H}_S^\varphi = \left\{ \boxed{h_K^\varphi(\boldsymbol{y}|\boldsymbol{x}) := \sum_{k=1}^{K} g_k(\boldsymbol{x}; \boldsymbol{\gamma}) \varphi_q(\boldsymbol{y}; \boldsymbol{\mu}_k, \sigma_k)}; \quad \varphi_q \in \mathcal{E}_\varphi \cap \mathcal{L}_\infty, g_k(\cdot; \boldsymbol{\gamma}) \in \{\text{softmax}\} \right\}$$

Theorem : Approximation capabilities of isotropic mixtures-of-experts SGaME

(a) For $p \in [1, \infty)$, $f \in \mathcal{F}_p \cap \mathcal{C}$, $\varphi \in \mathcal{F} \cap \mathcal{C}$, $\mathbb{X} = [0,1]^d$, there exists a sequence $\left(h_K^\varphi\right) \subset \mathcal{H}_S^\varphi$ such that $\lim_{K \to \infty} \left\| f - h_K^\varphi \right\|_{\mathcal{L}_p} = 0$.

(b) For $f \in \mathcal{F} \cap \mathcal{C}$, if $\varphi \in \mathcal{F} \cap \mathcal{C}$, $d = 1$, there exists a sequence $\left(h_K^\varphi\right) \subset \mathcal{H}_S^\varphi$ such that $\lim_{K \to \infty} h_K^\varphi = f$ almost uniformly.

– Nguyen, Nguyen, Chamroukhi, McLachlan (2021). Approximations of conditional probability density functions in Lebesgue spaces via mixture of experts models. *Journal of Statistical Distributions and Applications*. 8, 13 (2021).

Learning with mixtures-of-experts models

Learning via the EM algorithm : $\boldsymbol{\theta}^{new} \in \arg\max_{\boldsymbol{\theta}\in\Omega} \mathbb{E}[\ln L_c(\boldsymbol{\theta})|\mathcal{D}, \boldsymbol{\theta}^{old}]$

Learning with mixtures-of-experts models

Learning via the EM algorithm : $\boldsymbol{\theta}^{new} \in \arg\max_{\boldsymbol{\theta}\in\Omega} \mathbb{E}[\ln L_c(\boldsymbol{\theta})|\mathcal{D}, \boldsymbol{\theta}^{old}]$

SaMUraiS : open source software for statistical time-series analysis

SaMUraiS : **S**t**A**tistical **M**odels for the **U**nsupe**R**vised segment**A**t**I**on of time-**S**eries

▸ Github ▸ CRAN ▸ Matlab software

Available algorithms and Packages

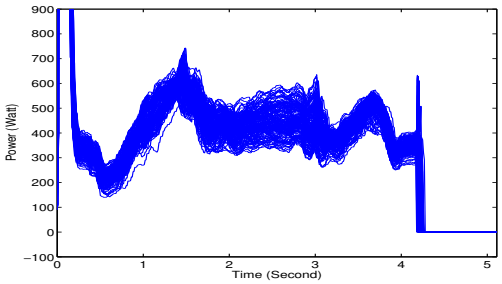
`RHLP` : Regression with Hidden Logistic Process	▸ R software	▸ Matlab software
`HMMR` : Hidden Markov Model Regression	▸ R software	▸ Matlab software
`PWR` : Piece-Wise Regression	▸ R software	▸ Matlab software
`MRHLP` : Multivariate RHLP	▸ R software	▸ Matlab software
`MHMMR` : Multivariate HMMR	▸ R software	▸ Matlab software
`MPWR` : Multivariate PWR	▸ R software	▸ Matlab software

Include estimation, segmentation, approximation, model selection, and sampling

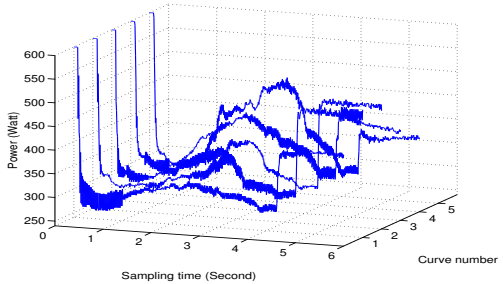
Some real-world data applications

- Heterogenous, Multimodal, High-Dimensional, Unlabeled, Possibly Massive ...
- Need for adapted analysis tools

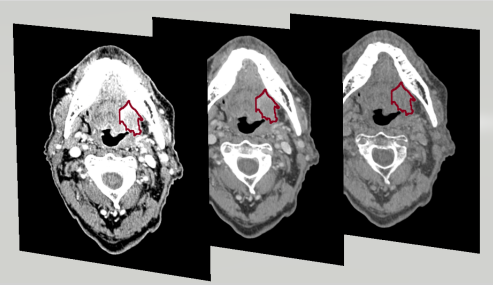
Transport : Railway switch curves diagnostic



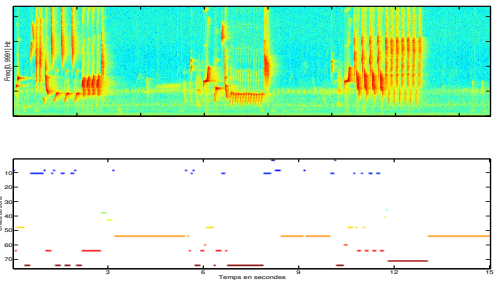
Predictive Maintenance

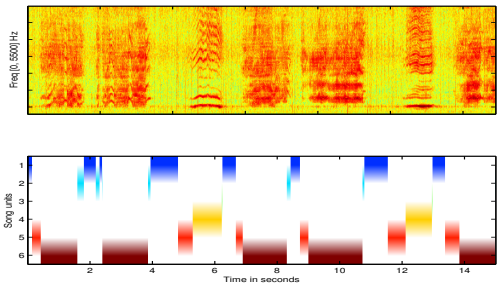


Health : Medical images

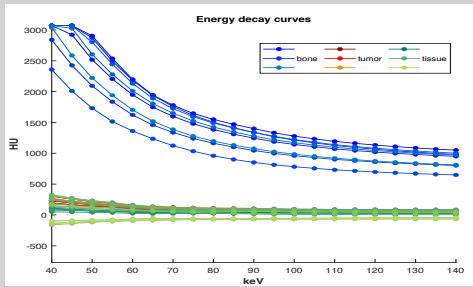


Acoustics : scene listening (marine, terrestrial)

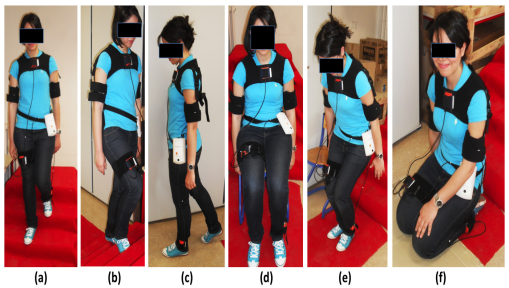




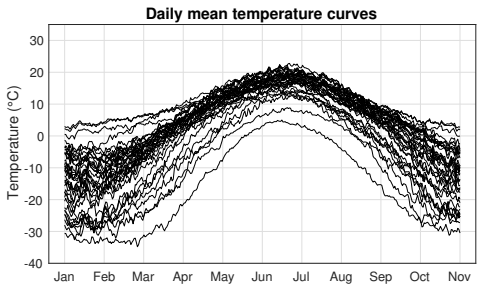
Dual-energy computed tomography

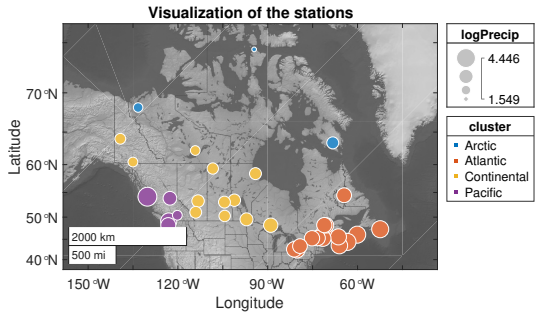


Health & Well Being : Activity recog.



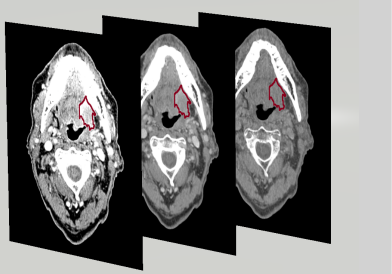
Climate/Environment : meteorological data

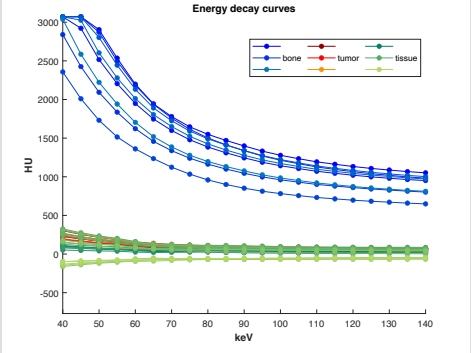


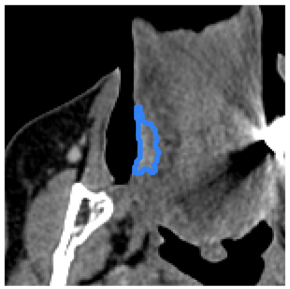


Dual-energy computed tomography (DECT) image Clustering

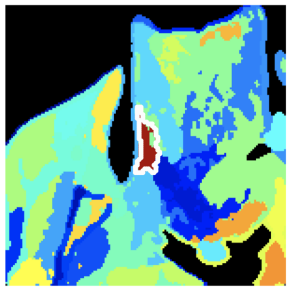
- Learning from Multimodal information in Healthcare/Radiology
- Cancer detection in Radiology : DECT clustering [Diagnostics (AI in medicine), 2022]
  Spatial mixture of functional regressions for dual-energy CT images
  $m(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{v};\boldsymbol{\theta}) = \sum_{k=1}^{K} \alpha_k(\boldsymbol{v};\boldsymbol{w}) f_k(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}_k)$ with $\alpha_k(\boldsymbol{v};\boldsymbol{w}) = \frac{w_k \phi_3(\boldsymbol{v};\boldsymbol{\mu}_k,\mathbf{R}_k)}{\sum_{\ell=1}^{K} w_\ell \phi_3(\boldsymbol{v};\boldsymbol{\mu}_\ell,\mathbf{R}_\ell)}$





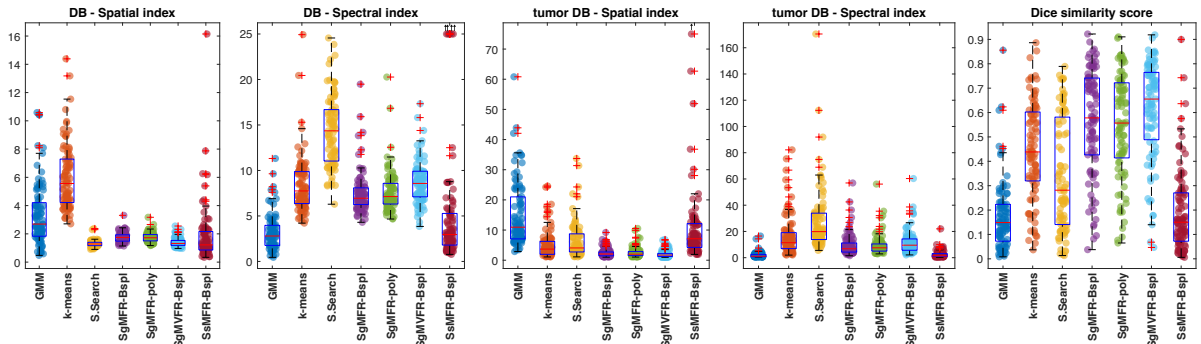


(a) Original slice



(b) Dice=0.84, DB=1.64/6.92

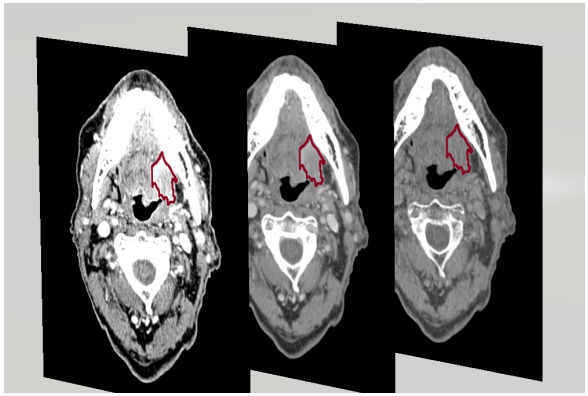
DECT multimodal Data : 3D voxels & energy levels Expert Annotation Automatic Annotation

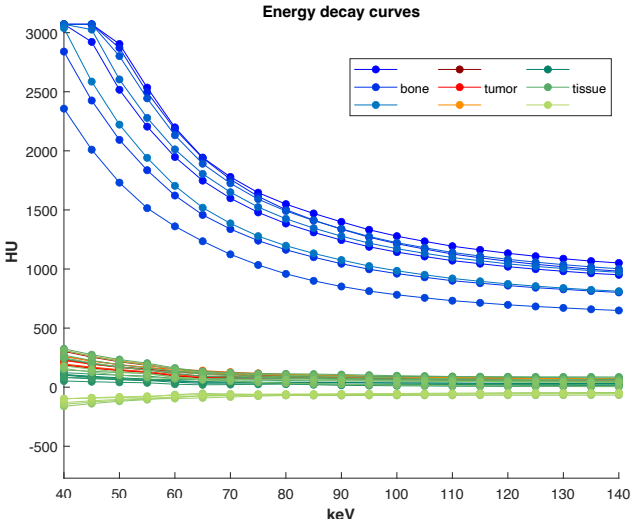


Chamroukhi, Brivet, Savadjiev, Coates and Forghani (2022). DECT-CLUST. Dual-Energy CT Image Clustering and Application to Head and Neck Squamous Cell Carcinoma Segmentation. *Diagnostics*, Vol. 12(12)

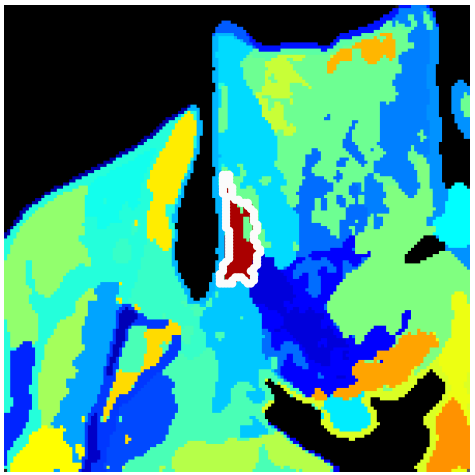
▸ Codes available on Github

Application of ML in precision medicine (Radiology)





Expert Annotation



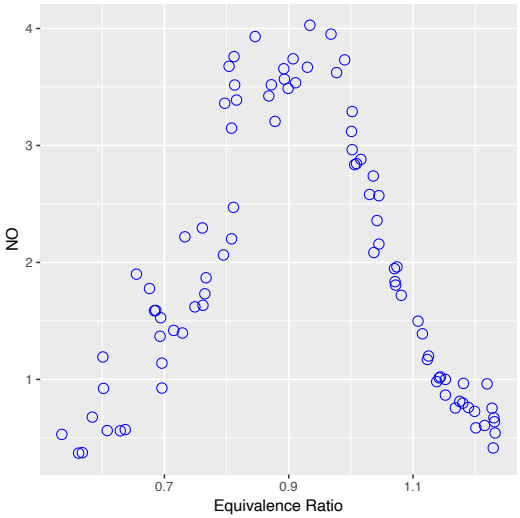
Our automatic Annotation

Chamroukhi, Brivet, Savadjiev, Coates and Forghani (2022). DECT-CLUST. Dual-Energy CT Image Clustering and Application to Head and Neck Squamous Cell Carcinoma Segmentation. *Diagnostics*, Vol. 12(12)

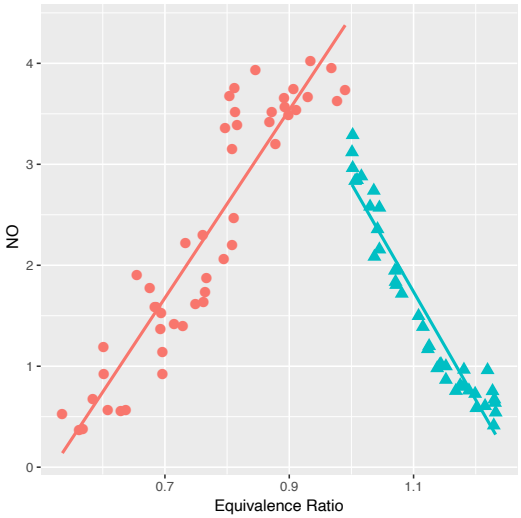
Model training and (variable/model) selection in MoE

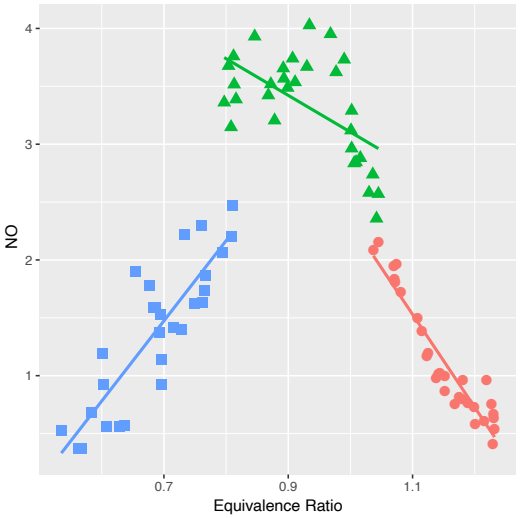
Model training and selection in MoE

(a) Raw Ethanol data set

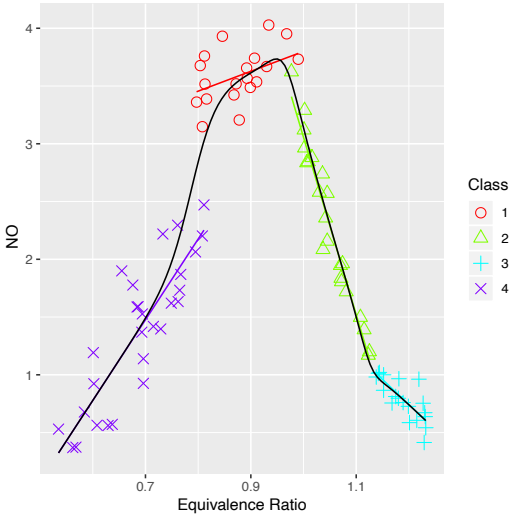


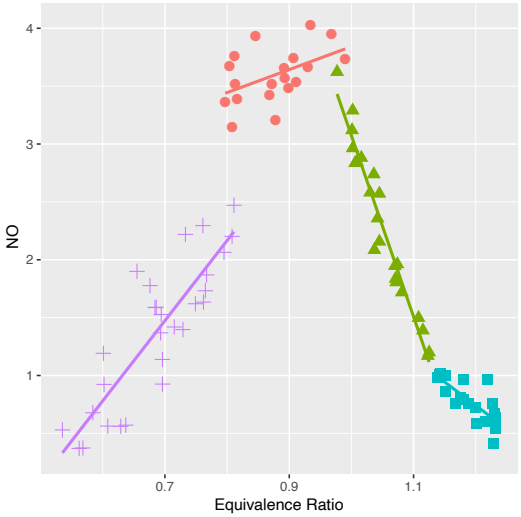
Collection of MoE models with linear mean functions characterized by 2-5 clusters

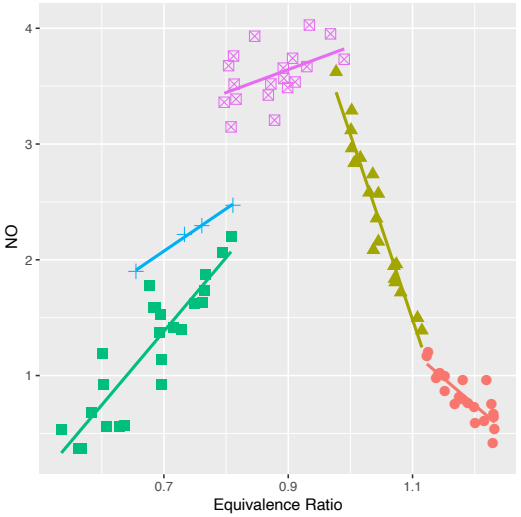




(b) Best data-driven MoE model







Learning with high-dimensional predictors

Questioning : Prediction (non-linear regr., classification) & clustering in presence of

[1.] **High-dimensional** predictors : $\boldsymbol{X}_i \in \mathbb{R}^p$ with $p \gg n$, in a heterogenous population

$\hookrightarrow$ Look for parsimonious models

Learning with high-dimensional predictors

Questioning : Prediction (non-linear regr., classification) & clustering in presence of

[1.] **High-dimensional** predictors : $\boldsymbol{X}_i \in \mathbb{R}^p$ with $p \gg n$, in a heterogenous population

$\hookrightarrow$ Look for parsimonious models

[1.] HDME : High-Dimensional Mixtures-of-Experts

- Learning : PMLE $\widehat{\boldsymbol{\theta}}_{\mathsf{n}} \in \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log h_K^{\varphi}(\boldsymbol{y}_i|\boldsymbol{x}_i;\boldsymbol{\theta}) - \ell_1(\boldsymbol{\theta})$
- $\hookrightarrow$ Lasso penalty : $\mathrm{Pen}_{\lambda}(\boldsymbol{\theta}) = \underbrace{\sum_{k=1}^{K} \lambda_k \|\boldsymbol{\beta}_k\|_1}_{\text{Experts Net.}} + \underbrace{\sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1}_{\text{Gating Net.}}$

$\hookrightarrow$ encourages sparse solutions & performs estimation and feature selection

$\hookrightarrow$ computationally attractive (Avoid matrix inversion ; univariate updates)

▸ Software Toolbox HDME on Github (GaussRMoE, LogisticRMoE, PoissonRMoE)

Learning with high-dimensional predictors

Questioning : Prediction (non-linear regr., classification) & clustering in presence of

[1.] **High-dimensional** predictors : $\boldsymbol{X}_i \in \mathbb{R}^p$ with $p \gg n$, in a heterogenous population

↪ Look for parsimonious models

[1.] HDME : High-Dimensional Mixtures-of-Experts

- Learning : PMLE $\widehat{\boldsymbol{\theta}}_{\mathsf{n}} \in \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log h_K^{\varphi}(\boldsymbol{y}_i|\boldsymbol{x}_i;\boldsymbol{\theta}) - \ell_1(\boldsymbol{\theta})$
- ↪ Lasso penalty : $\mathrm{Pen}_{\lambda}(\boldsymbol{\theta}) = \underbrace{\sum_{k=1}^{K} \lambda_k \|\boldsymbol{\beta}_k\|_1}_{\text{Experts Net.}} + \underbrace{\sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1}_{\text{Gating Net.}}$

↪ encourages sparse solutions & performs estimation and feature selection

↪ computationally attractive (Avoid matrix inversion ; univariate updates)

▸ Software Toolbox HDME on Github (GaussRMoE, LogisticRMoE, PoissonRMoE)

- Gaussian experts $\mathcal{N}_q\left(\mathbf{y}; \boldsymbol{\upsilon}_{k,d_{\boldsymbol{\Upsilon}}}(\mathbf{x}), \boldsymbol{\Sigma}_k\left(\mathbf{B}_k\right)\right)$ with $\mathbf{B} = (\mathbf{B}_k)_{k\in[K]}$: block-diagonal structures for covariance matrices

↪ A non-asymptotic result. If $\mathrm{pen}(\mathbf{m})$ is well chosen, then our PMLE behaves in a comparable manner compared to **the best (oracle) model** $\mathcal{H}_{\mathbf{m}^\star}$ in the collection

– Chamroukhi & Huynh. Regularized Maximum Likelihood Estimation and Feature Selection in Mixtures-of-Experts Models. Journal de la Société Francaise de Statistique, Vol. 160(1), pp :57–85, 2019

– Nguyen, Nguyen, Chamroukhi, Mchlachlan : Australian Joint Conference on Artificial Intelligence 2024.

– Nguyen, Chamroukhi, Nguyen, Forbes. Non-asymptotic model selection in block-diagonal mixture of polynomial experts models. arXiv :2104.08959, 2021

Measuring uncertainty in high-dimensional learning

Questioning : Prediction (non-linear regr., classification) & clustering in presence of **High-dimensional** predictors : Data $\mathcal{D}_n = (\boldsymbol{X}_i, Y_i)_{i=1}^n$ where $\boldsymbol{X}_i \in \mathbb{R}^p$ with $p \gg n$

`HDME` : High-Dimensional MoE : PMLE $\widehat{\boldsymbol{\theta}}_{\mathsf{n}} \in \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^n \log h_K^{\boldsymbol{\varphi}}(\boldsymbol{y}_i|\boldsymbol{x}_i;\boldsymbol{\theta}) - \mathsf{pen}(\boldsymbol{\theta})$

Measuring uncertainty in high-dimensional learning

Questioning : Prediction (non-linear regr., classification) & clustering in presence of **High-dimensional** predictors : Data $\mathcal{D}_n = (\boldsymbol{X}_i, Y_i)_{i=1}^n$ where $\boldsymbol{X}_i \in \mathbb{R}^p$ with $p \gg n$

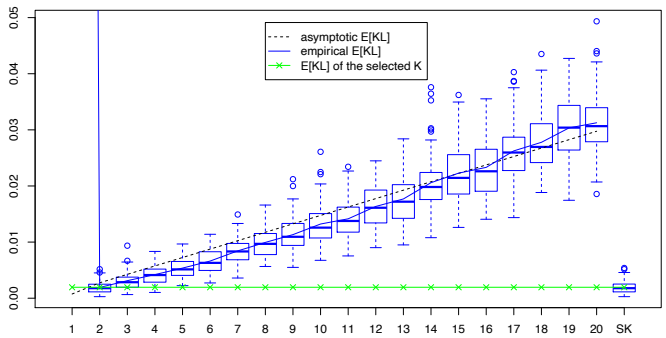
`HDME` : High-Dimensional MoE : PMLE $\widehat{\boldsymbol{\theta}}_{\mathrm{n}} \in \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^n \log h_K^{\varphi}(\boldsymbol{y}_i|\boldsymbol{x}_i;\boldsymbol{\theta}) - \mathrm{pen}(\boldsymbol{\theta})$

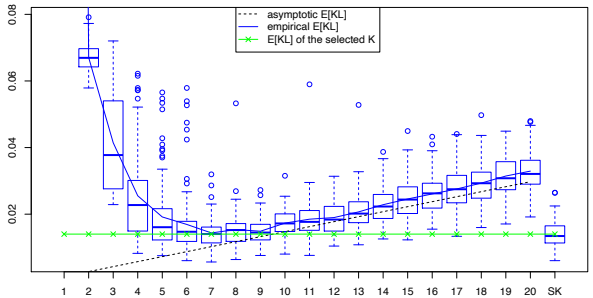
Theorem : Non-asymptotic oracle inequality for collection of MoE models

Result : $\exists$ constants C et $\kappa(\rho, C_1) > 0$ ($C_1 > 1$) s.that whenever for $\mathbf{m} \in \mathcal{M}$, $\mathrm{pen}(\mathbf{m}) \geq \kappa(\rho, C_1)\left[(C + \ln n)\dim(\mathcal{H}_{\mathbf{m}}) + z_{\mathbf{m}}\right]$, the estimator PMLE $\widehat{h}_{\widehat{\mathbf{m}}}$ satisfies

$$\mathbb{E}\left[\mathrm{JKL}_{\rho}^{\otimes \mathrm{n}}\left(f, \widehat{h}_{\widehat{\mathbf{m}}}\right)\right] \leq C_1 \inf_{\mathbf{m}\in\mathcal{M}}\left(\inf_{h_{\mathbf{m}}\in\mathcal{H}_{\mathbf{m}}} \mathrm{KL}^{\otimes \mathrm{n}}(f, h_{\mathbf{m}}) + \frac{\mathrm{pen}(\mathbf{m})}{n}\right) + \frac{\kappa(\rho, C_1) C_1 \xi}{n} + \frac{\eta + \eta'}{n}.$$

- A non-asymptotic result. If $\mathrm{pen}(\mathbf{m})$ is well chosen, then our PMLE behaves in a comparable manner compared to **the best (oracle) model** $\mathcal{H}_{\mathbf{m}^\star}$ in the collection, minimizing the risk : $\inf_{\mathbf{m}\in\mathcal{M}}\left(\inf_{h_{\mathbf{m}}\in\mathcal{H}_{\mathbf{m}}} \mathrm{KL}^{\otimes \mathrm{n}}(f, h_{\mathbf{m}}) + \frac{\mathrm{pen}(\mathbf{m})}{n}\right)$ (f is unknown).





– Nguyen, Nguyen, Chamroukhi and Forbes. A non-asymptotic approach for model selection via penalization in high-dimensional mixture of experts models. Electronic Journal of Statistics. 2022

2. Functional predictors

[2.] Learning with functional predictors

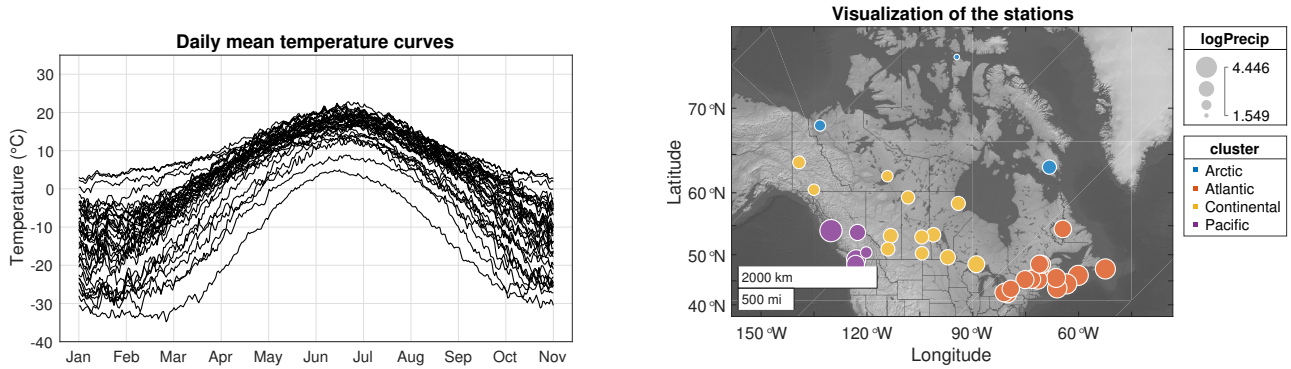


FIGURE – $n = 35$ daily mean temperature measurement curves (X_i's) in different stations (Left) and the log of precipitation values (Y_i's) visualized with the climate regions (Z_i's) (Right).

- Relate functional predictors $\{X(t) \in \mathbb{R}; t \in \mathcal{T} \subset \mathbb{R}\}$ to a scalar response $Y \in \mathcal{Y} \subset \mathbb{R}$
- Regression and classification of heterogeneous responses given functional predictors

  (1) generative functional modeling, sparsity and feature selection (high-dimension)

  (2) User guideline : keep an interpretable fit

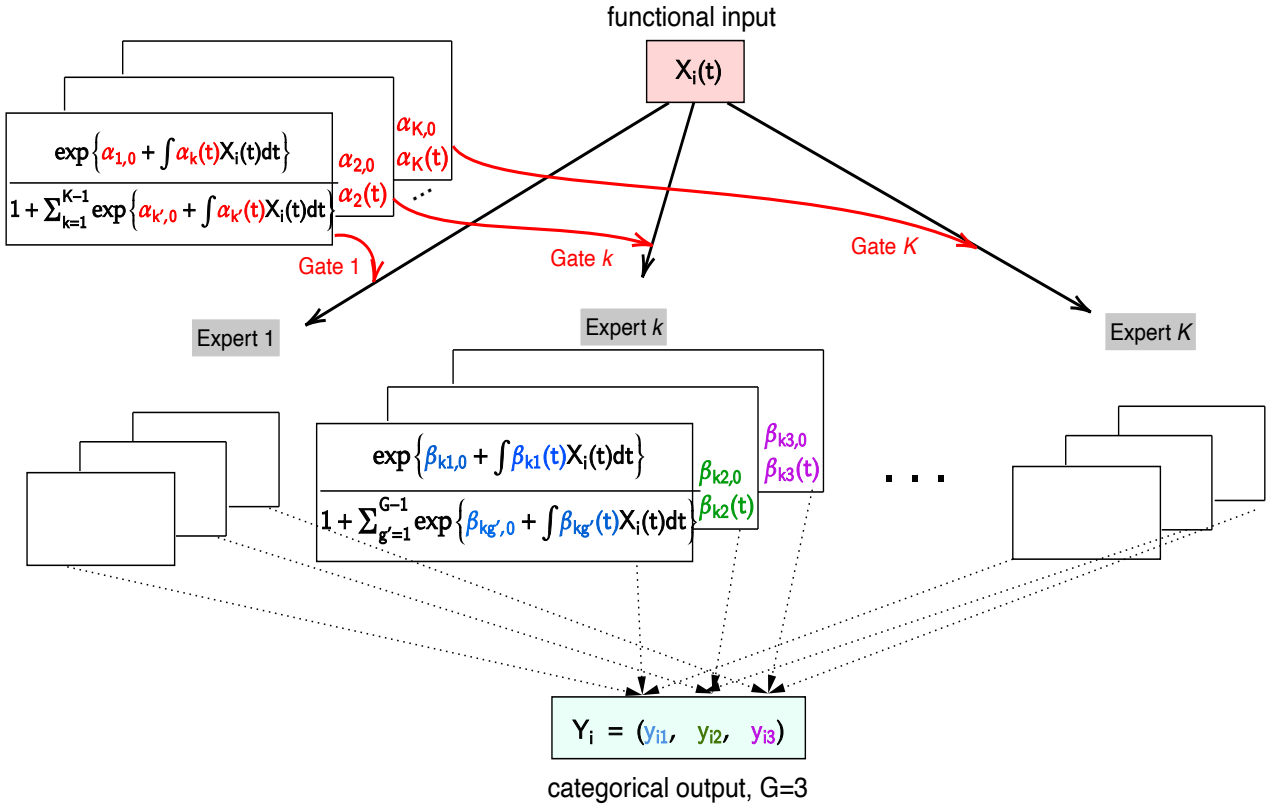
[2.] Functional Mixtures-of-Experts (and Different Learning strategies, in particular)

- $Y_i = \beta_{z_i,0} + \int_{\mathcal{T}} X_i(t)\beta_{z_i}(t)dt + \varepsilon_i$ avec $h_z(X_i(.)) = \alpha_{z_i,0} + \int_{\mathcal{T}} X_i(t)\alpha_{z_i}(t)dt$
- Lasso-type Regularized MLE w.r.t the derivatives of the $\alpha(\cdot)$ and $\beta(\cdot)$ functions

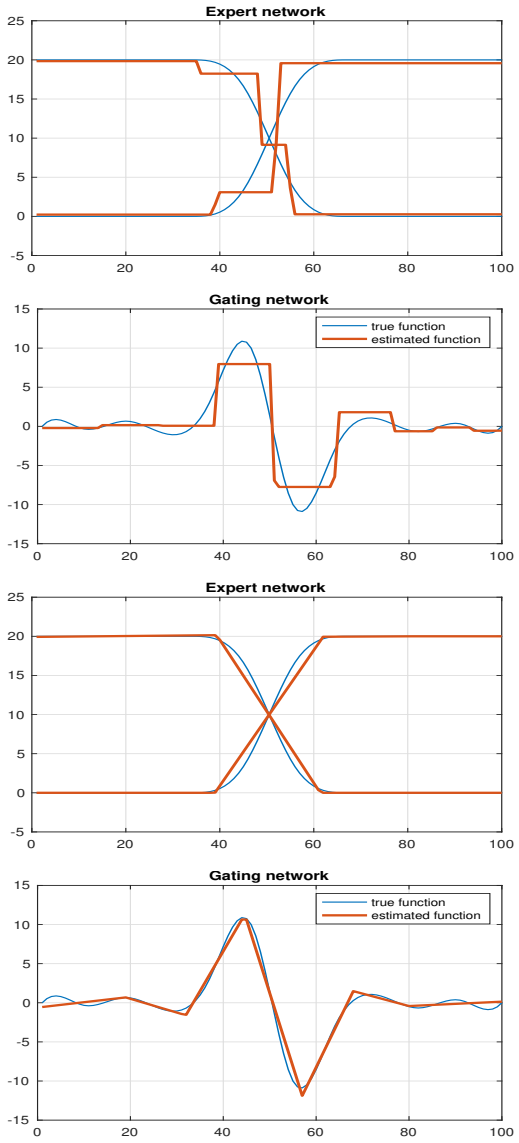
Chamroukhi, Pham, Hoang, McLachlan. Functional Mixtures-of-Experts. *Statistics and Computing* ., Vol. 34 (98), 2024

Interpretable learning with time-series inputs

Mixture-of-Experts Architecture



Interpretable fits



$Y_i = \beta_{z_i,0} + \int_{\mathcal{T}} X_i(t)\beta_{z_i}(t)dt + \varepsilon_i$ with $h_z(X_i) = \alpha_{z_i,0} + \int_{\mathcal{T}} X_i(t)\alpha_{z_i}(t)dt$

l_1-Regularized MLE w.r.t the derivatives of the $\alpha(\cdot)$ and $\beta(\cdot)$ functions

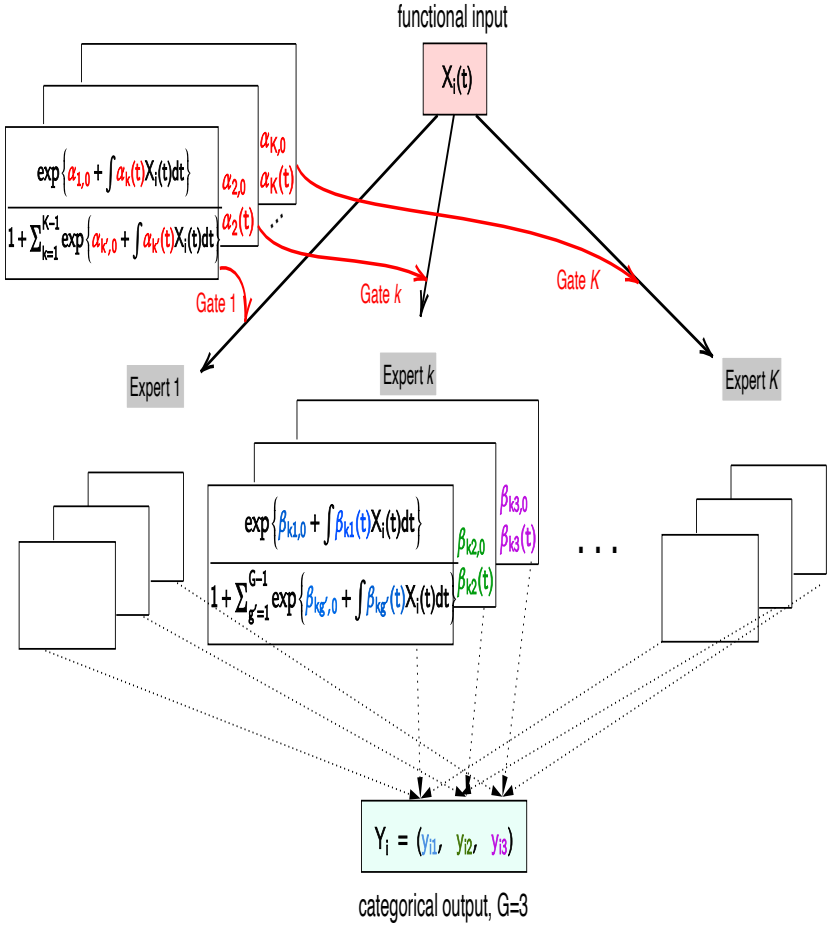
↪ produces a meaningful sparse estimates for $\beta_{z_i}(t)$ curves :

- $\beta_{z_i}^{(0)}(t) = 0$ implies that $X(t)$ has no effect on Y at t
- $\beta_{z_i}^{(1)}(t) = 0$ means that $\beta_{z_i}(t)$ is constant at t,
- $\beta_{z_i}^{(0)}(t) = 1$ shows that $\beta_{z_i}(t)$ is a linear function of t, etc.

Chamroukhi, Pham, Hoang, McLachlan. Functional Mixtures-of-Experts. *Statistics and Computing* ., Vol. 34 (98), 2024

Interpretable learning with time-series inputs

Mixture-of-Experts Architecture



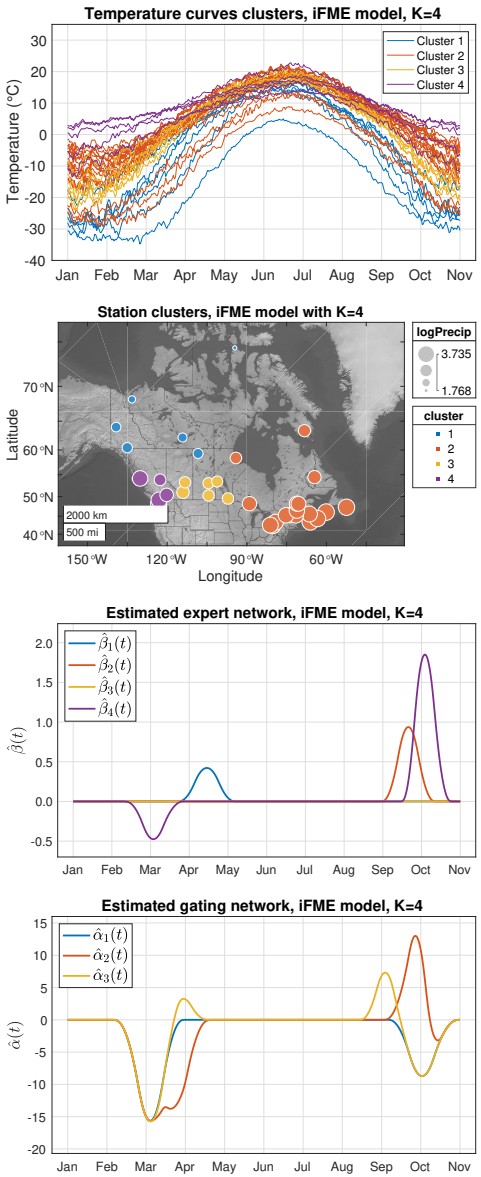
produces a meaningful sparse estimates for $\beta_{z_i}(t)$ curves :

$\beta^{(0)}_{z_i}(t) = 0$ implies that $X(t)$ has no effect on Y at t

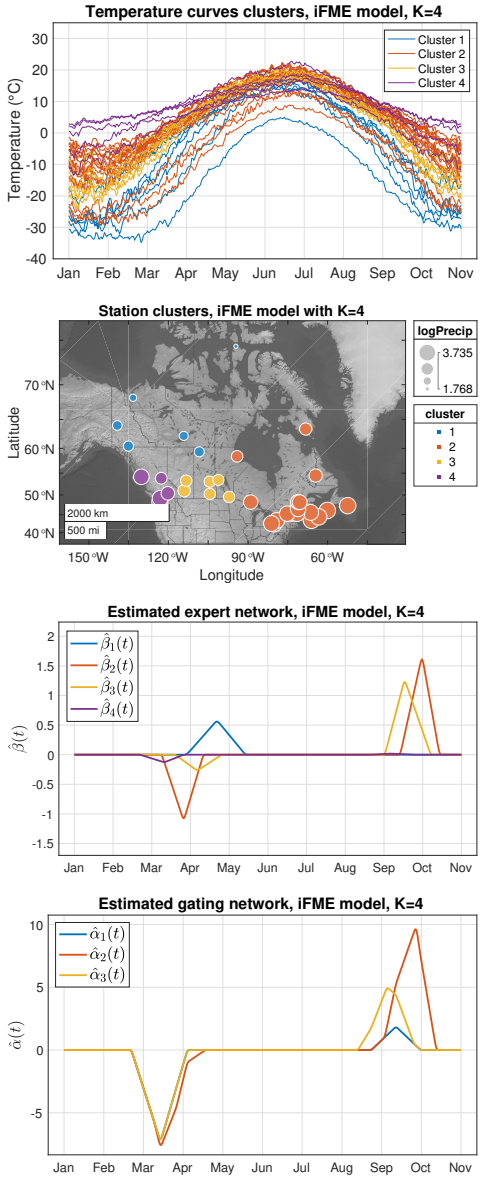
$\beta^{(1)}_{z_i}(t) = 0$ means that $\beta_{z_i}(t)$ is constant at t,

$\beta^{(0)}_{z_i}(t) = 1$ shows that $\beta_{z_i}(t)$ is a linear function of t, etc.

OUR regularization



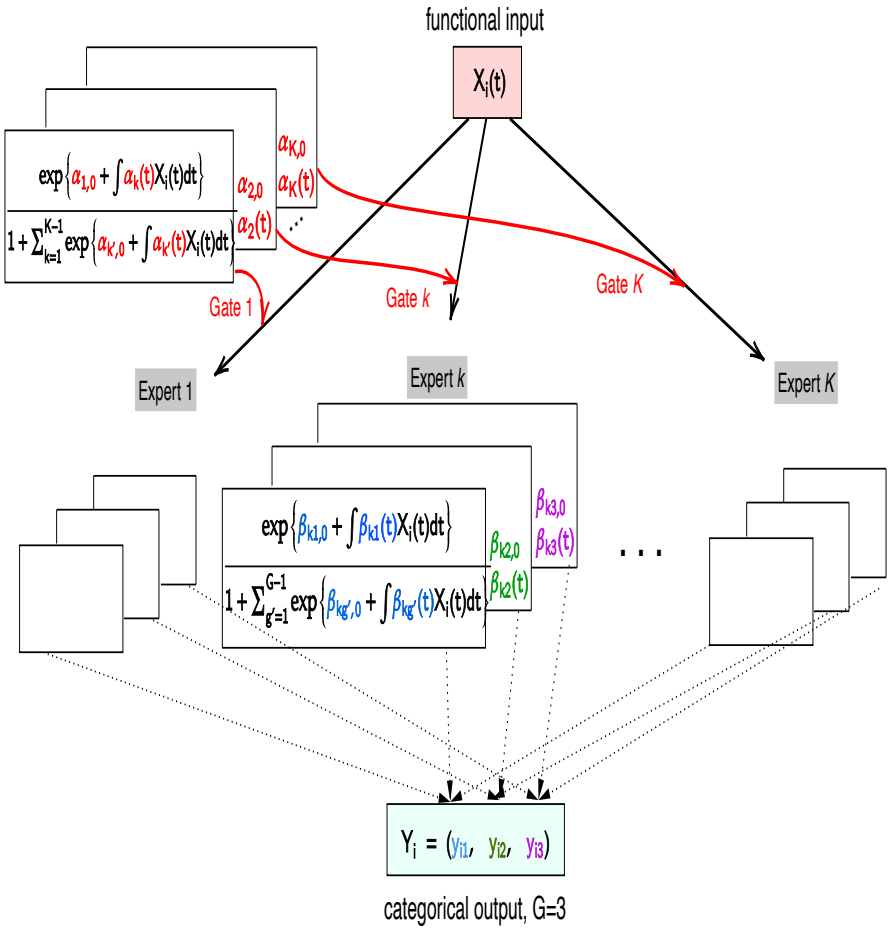
OUR regularization



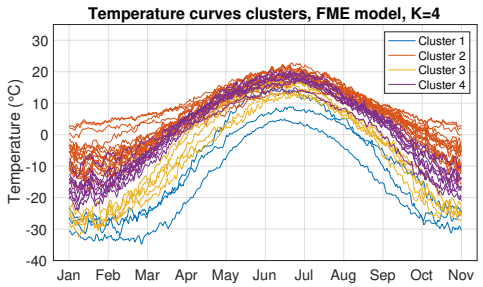
Chamroukhi, Pham, Hoang, McLachlan. Functional Mixtures-of-Experts. *Statistics and Computing* ., Vol. 34 (98), 2024

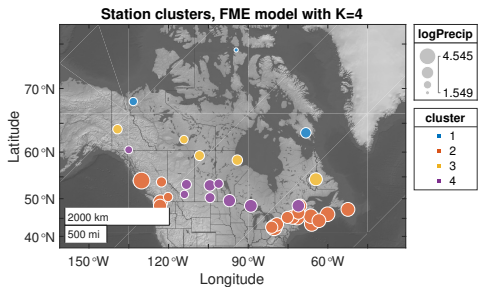
Interpretable learning with time-series inputs

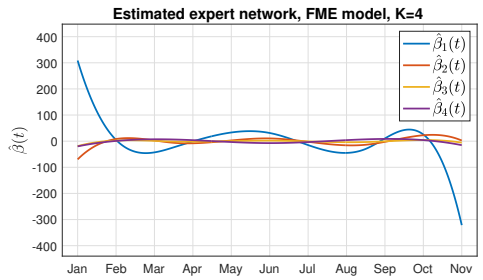
Mixture-of-Experts Architecture

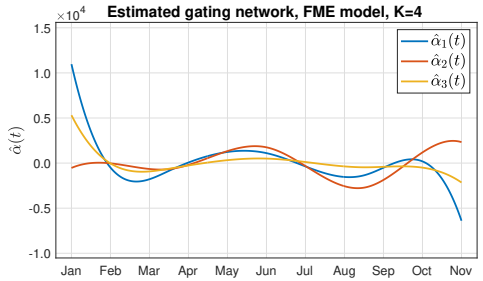


No regularization

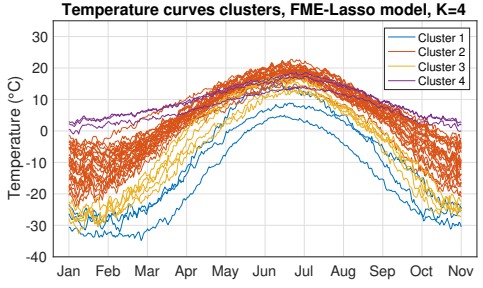


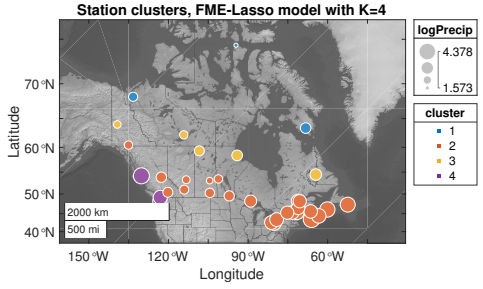


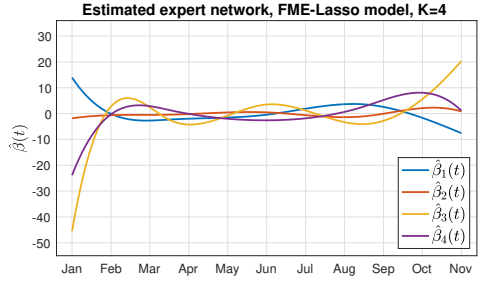


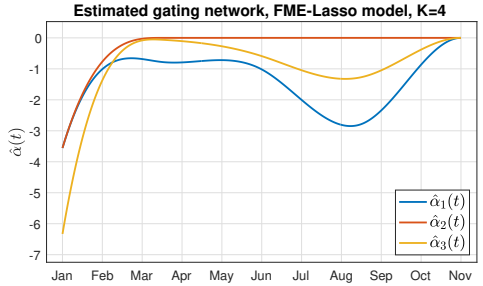


LASSO regularization









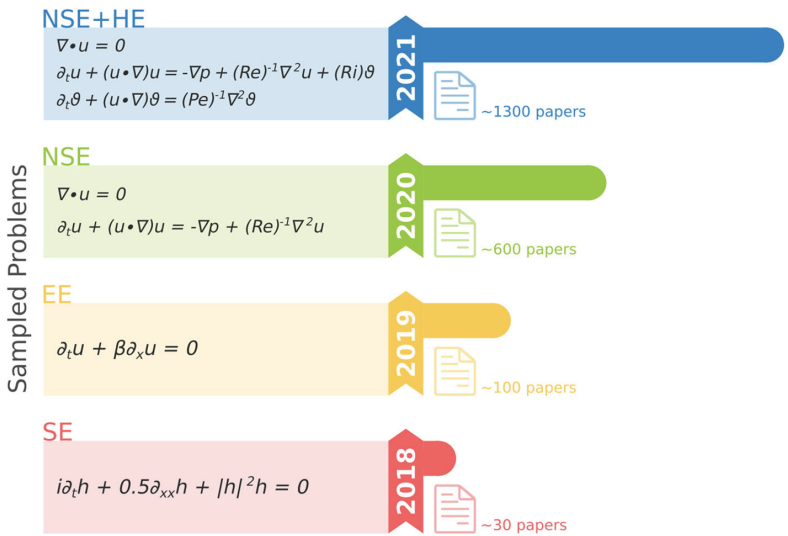
Chamroukhi, Pham, Hoang, McLachlan. Functional Mixtures-of-Experts. *Statistics and Computing* ., Vol. 34 (98), 2024

Regularization to accommodate physical priors

Physics-Informed Machine Learning : combining ML and Physics

➔ Enables prior scientific knowledge based on physics to be taken into account in data-driven machine learning methods e.g including PINNs - Physics-Informed Neural Nets (Raissi's paper in 2019)

➔ Has been *successfully and increasingly* applied to solve a wide variety of linear and nonlinear problems in physics, covering various fields like mechanics, fluid dynamics, thermodynamics, electromagnetism … including :



- Solving Navier–Stokes equations coupled with the corresponding temperature equation for analyzing heat flow convection (NSE+HE). Cai et al, 2021
- Solving incompressible Navier–Stokes equations (NSE). Jin et al., 2020.
- Solving Euler equations (EE) that model high-speed aerodynamic flows. Mao et al, 2019
- Solving the nonlinear Shrödinger Equation (SE).

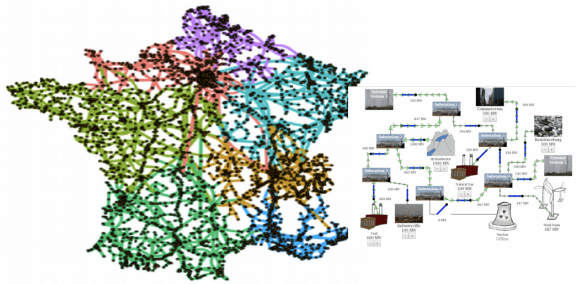
Raissi, M et al. (2019) Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. Journal of Computational Physics. 378. Online

Motivation : Some physical problems in Industry

Related to the design and supervision of complex (physical) systems

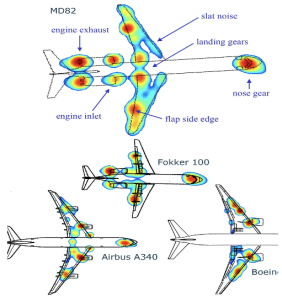
- Covering various fields in physics (mechanics, fluid dynamics, aerodynamics, electromagnetism ...)
- In a wide variety of Applications in industry, in particular in numerical **simulation**

Electricity (power grids)



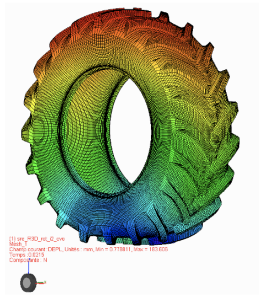
Picture from **Marot**, A., et al. (2018).

Aerodynamics



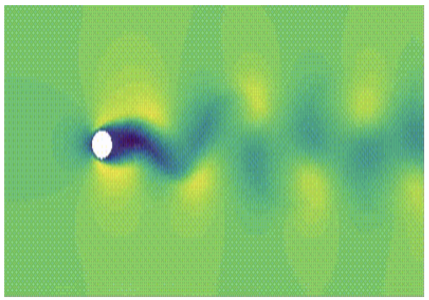
Picture form Merino-Martínez *et al. CEAS Aeronautical* Journal (2019).

Solid Mechanics pneumatics



From HSA – SystemX

Fluid Flows/Dynamics

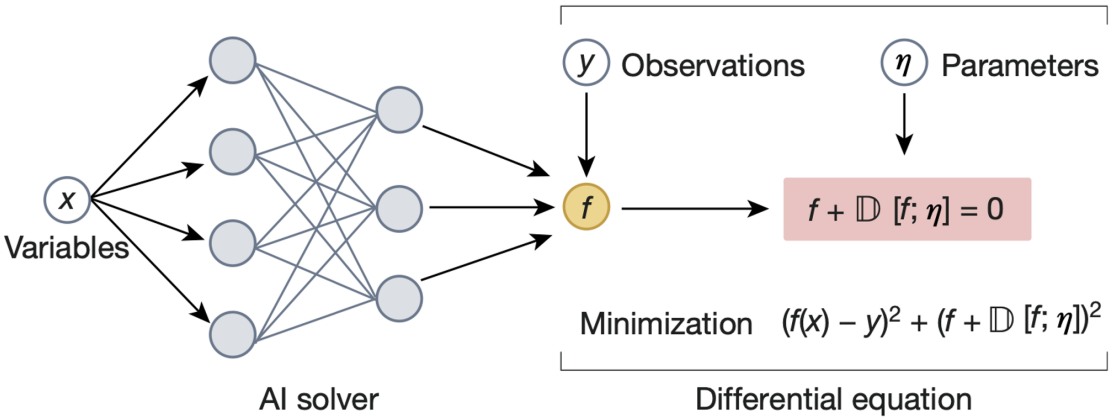


from Emmanuel Menier (PhD, LSIN/SystemX, 2024)

Domain Challenges : Physical systems that are
- Complex to model/solve analytically
- Compuationally expensive to solve numerically

eg. , Computational Fluid Dynamics – CFD, Turbulance, Flows

Hybrid ML modeling for solving Partial Differential Equations



A neural framework for solving PDEs, where

- the AI solver is a PINN trained to estimate target function ***f***.
- The derivative of ***x*** is calculated by automatically differentiating the NN's outputs.
- When the differential equation parametrized by **(*η*)** is unknown, it can be estimated by solving a loss that optimizes both the functional form of the equation and its fit to obser. ***y***.

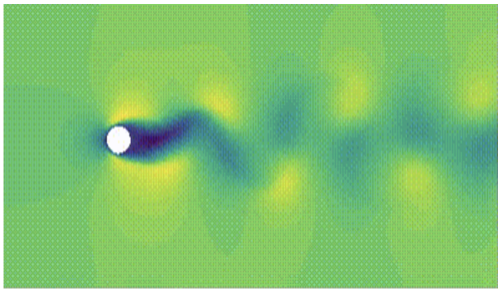
Wang & al. (2023). Scientific discovery in the age of artificial intelligence. Nature, 620. Read Online

- Eg. Learning Computational Fluid Dynamics

- Navier-Stokes Equations: fundamental partial differentials equations (**PDE**) that describe the flow of incompressible fluids.

C.L. M. H. Navier, Memoire sur les Lois du Mouvements des Fluides, Mem. de l'Acad. d. Sci.,6, 398 (1822)
C.G. Stokes, On the Theories of the Internal Friction of Fluids in Motion, Trans. Cambridge Phys. Soc., 8, (1845)

- Challenge: High-Dimensional non-linear Physical Equations



Credit: Emmanuel Menier, PhD LISN/SystemX

Regularization and Physics-Informed Neural Networks (PINNs)

Regularization view for MoE

- Add a penalty term to the loss : $\mathcal{L}(\theta) = \underbrace{\mathcal{L}_{\text{data}}(\theta)}_{\text{fit to data}} + \lambda \underbrace{\text{Pen}(\theta)}_{\text{regularization term}}$
- $\text{Pen}(\theta)$ encodes sparsity, smoothness

Physics-Informed ML as Regularization

- Integrate known physics (e.g., PDEs) into the loss :

$$\mathcal{L}(\theta) = \underbrace{\mathcal{L}_{\text{data}}(\theta)}_{\text{data fit}} + \lambda \underbrace{\mathcal{L}_{\text{physics}}(\theta)}_{\text{physics-based residuals}}$$

- $\mathcal{L}_{\text{physics}}(\theta)$ penalizes violations of physical laws, e.g., $\|\mathcal{N}_{\text{RANS}}(\mathbf{u}, p)\|^2$, where $\mathcal{N}_{\text{RANS}}$ denotes the residual of the Reynolds-Averaged Navier-Stokes equations.
- Possible $+\gamma\mathcal{L}_{\text{BC}}$ Boundary conditions loss
- Equivalent to imposing a constraint from domain knowledge

PINNs can be interpreted as a regularized learning framework with physics as a prior.

Federated Learning

Federated Learning

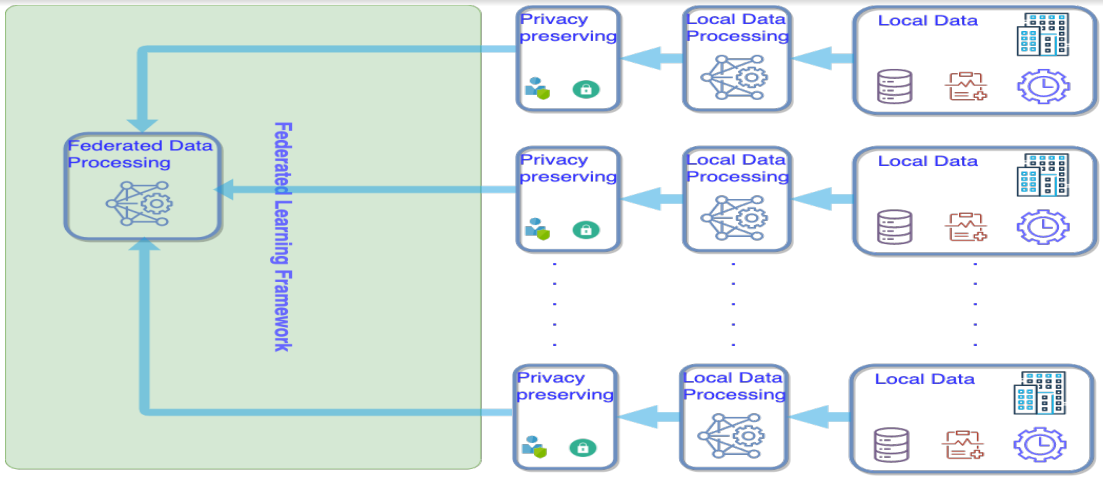
- Real-world data problems may either i) arise **massively** **or** are ii) by nature **distributed** (available on local sites) [and my involve specific domain constraints]
- Challenges : optimizing data use and transfer to reduce the need to collect, store, process and transfer large amounts of data and/or large AI models, while preserving privacy and reducing energy consumption :

[A.] Distributed Learning from massive data [New issues in learning and aggregation]

[for optimized learning processes] that require less input (data efficient AI) without degrading the estimation/prediction ↪ How to distribute/aggregate data/models

[B.] Federated Learning from data distributed by nature

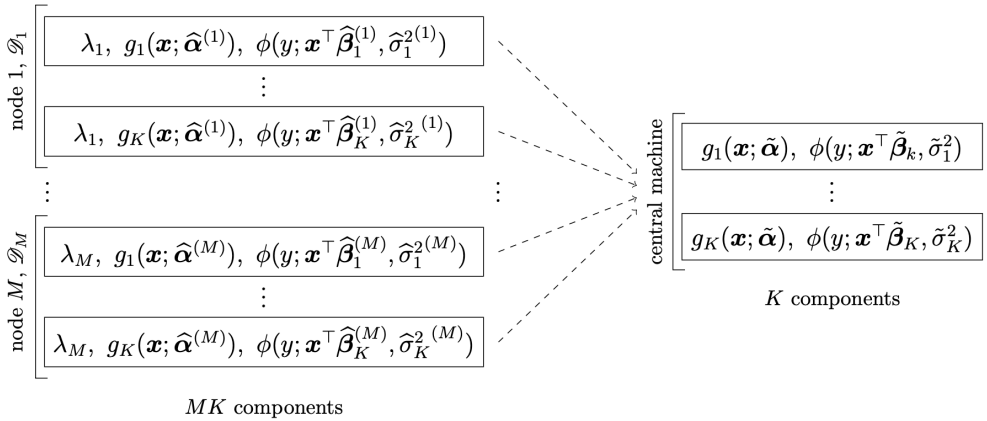
↪ How to accommodate local constraints (eg. privacy, energy, etc)



Federated Learning

Aggregating distributed mixtures-of-experts models (MoE)

- collaborative MoE for distributed (eg. large-scale data) or federated learning

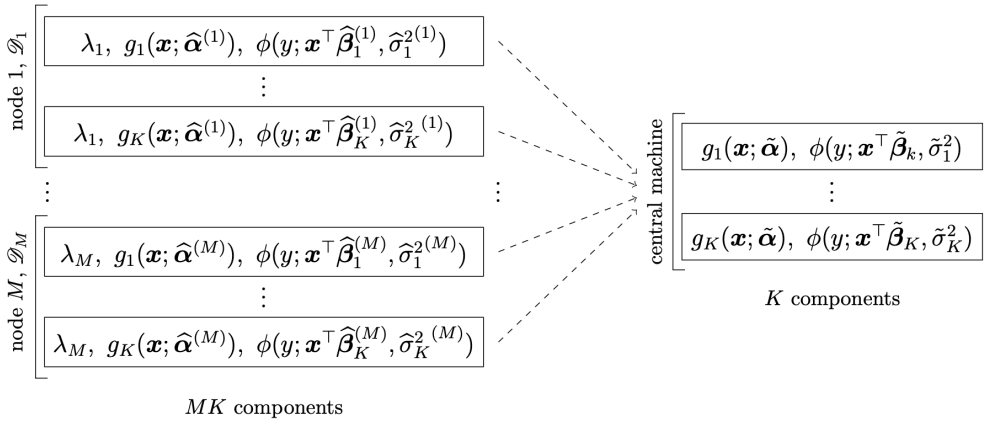


- Local estimators : $\hat{f}_m = f(\cdot|\mathbf{x},\widehat{\boldsymbol{\theta}}_m) = \sum_{k=1}^K g_k(\mathbf{x},\widehat{\boldsymbol{\alpha}}^{(m)})\phi(\cdot;\mathbf{x}^\top\widehat{\boldsymbol{\beta}}_k^{(m)},\widehat{\sigma}_k^{2\ (m)})$,
- weighted average : $\bar{f} = f(y|\mathbf{x};\bar{\boldsymbol{\theta}}) = \sum_{m=1}^M \lambda_m\hat{f}_m$ where $\lambda_m = \frac{N_m}{N}$ the sample proportion. $\bar{f}$ is good but relates MK components so not our direct target.

Federated Learning

Aggregating distributed mixtures-of-experts models (MoE)

- collaborative MoE for distributed (eg. large-scale data) or federated learning



- Local estimators : $\hat{f}_m = f(\cdot|\mathbf{x},\widehat{\boldsymbol{\theta}}_m) = \sum_{k=1}^K g_k(\mathbf{x},\widehat{\boldsymbol{\alpha}}^{(m)})\phi(\cdot;\mathbf{x}^\top\widehat{\boldsymbol{\beta}}_k^{(m)},\widehat{\sigma}_k^{2\ (m)})$,
- weighted average : $\bar{f} = f(y|\mathbf{x};\bar{\boldsymbol{\theta}}) = \sum_{m=1}^M \lambda_m\hat{f}_m$ where $\lambda_m = \frac{N_m}{N}$ the sample proportion. $\bar{f}$ is good but relates MK components so not our direct target.

↪ Reduced estimator : $\bar{f}^R = \underset{h_K\in\mathcal{M}_K}{\arg\inf}\ \rho\left(h_K, \sum_{m=1}^M \lambda_m\hat{f}_m\right)$: we seek for a K-component ME h that is closest to the MK-component ME $\bar{f} = \sum_{m=1}^M \lambda_m\hat{f}_m$ w.r.t a transportation divergence $\rho(\cdot,\cdot)$, e.g. KL.

Chamroukhi and Pham. Distributed Learning of Mixtures of Experts. *arxiv 2312.09877*, 2024
{PhD, Pham. 2022}

▸ Source codes publicly available on Github.

■ Consistency : The reduction estimator $\bar{\boldsymbol{\theta}}^R$ has a desired property that it is a consistent estimator of the true parameter $\boldsymbol{\theta}^*$ as soon as the local estimators are consistent estimators of $\boldsymbol{\theta}^*$. We have the following proposition.

Proposition (Chamroukhi and Pham 2023, ArXiv)

- A1 The dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ is an i.i.d. sample from the K-component MoE model $f(y|\mathbf{x}, \boldsymbol{\theta}^*)$, in which the parameters are ordered and initialized.
- A2 The cost function $c(\cdot, \cdot)$ is continuous in both arguments, and $c(\varphi_1, \varphi_2) \to 0$ if and only if $\varphi_1 \to \varphi_2$ in distribution.

Let $\bar{\boldsymbol{\theta}}^R$ be the parameter of the reduction density $\bar{f}^R$ defined in

$$\bar{f}^R = \underset{h_K \in \mathcal{M}_K}{\arg\inf} \; \rho\left(h_K, \sum_{m=1}^{M} \lambda_m \hat{f}_m\right)$$

with ρ being the expected transportation divergence between two mixtures h and g. Suppose assumptions A1-A2 are satisfied. Then $\bar{\boldsymbol{\theta}}^R$ is a consistent estimator of $\boldsymbol{\theta}^*$.

Federated Learning

Numerical results in Distributed clustering and Prediction

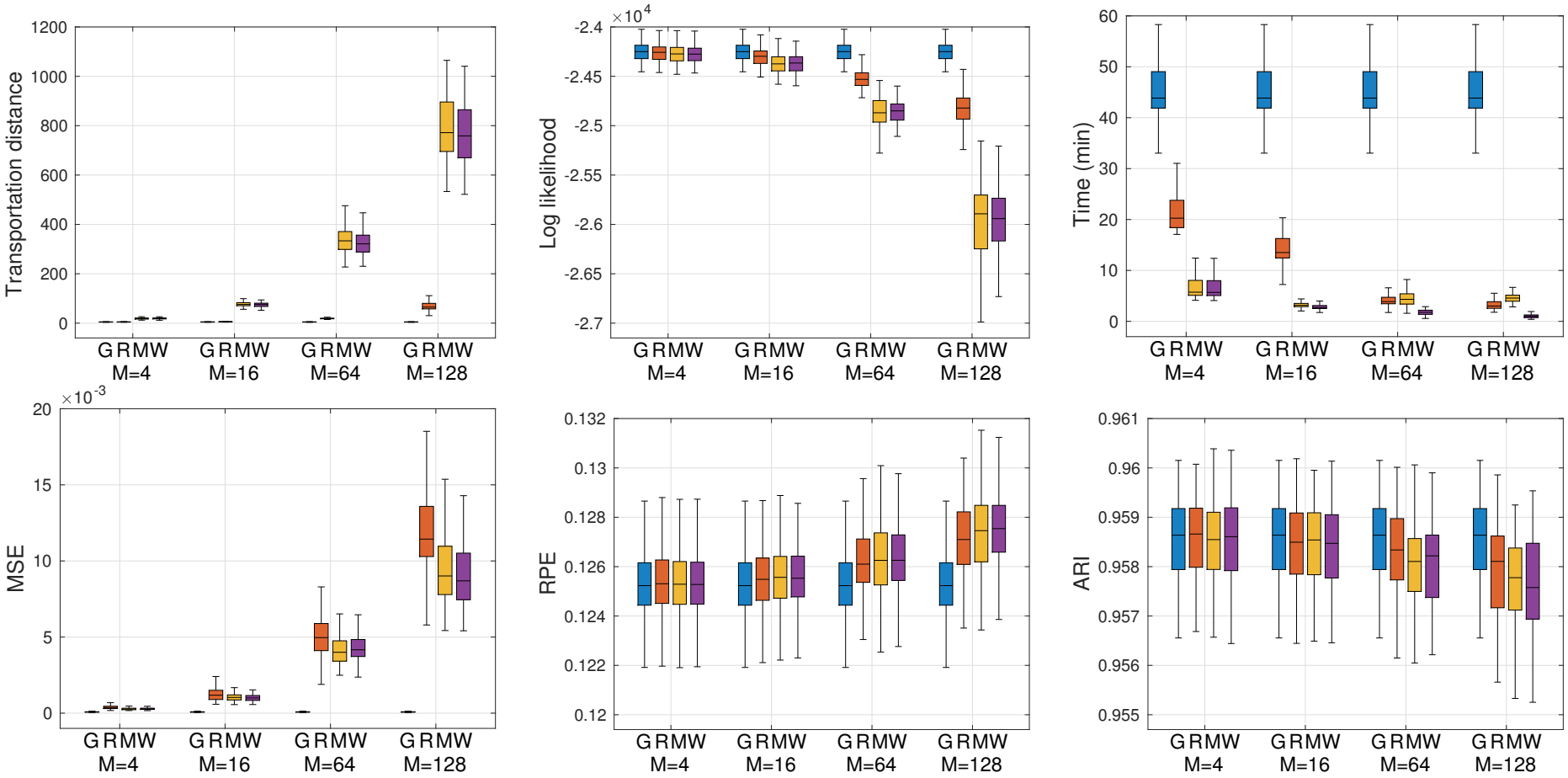


FIGURE – Performance of the Global ME (G), Reduction (R), Middle (M) and Weighted average (W) estimator for sample size $N = 10^6$ and M machines.

- Chamroukhi and Pham T. Distributed Learning of Mixtures of Experts. *arxiv 2312.09877*, 2023
{- PhD, Pham. 2022}

- Source codes publicly available on Github.

FL for spatio-temporal data forecasting in mobility

Attention mechanism : our model operates an Attention mechanism on the output of the LSTM module, i.e., the sequence $\mathcal{X}_t''$ $Q_k = \mathcal{X}_t'' W_k^Q, K_k = \mathcal{X}_t'' W_k^K, V_k = \mathcal{X}_t'' W_k^V$

$$\text{Att.}(Q_k, K_k, V_k) = \text{softmax}\left(\frac{Q_k K_k^\top}{\sqrt{d_h}}\right) V_k$$

Outputs from all G heads are concatenated and linearly projected to form a rich temporal embedding that captures complex, heterogeneous interactions

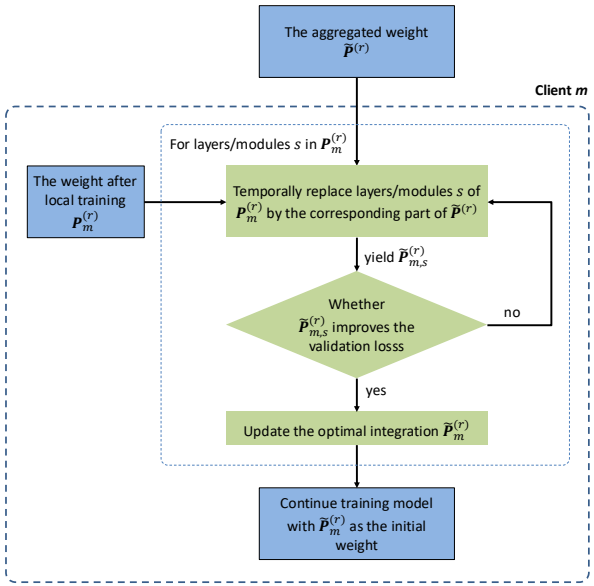


FIGURE – Client-Side Validation mechanism.

1. **Local training**
2. **Parameter sharing** : Clients send their locally trained model to the server.
3. **Server-side aggregation** : FedAveraging
4. **Client-side validation** : Each client performs for each subset of the three modules of the LSTM-DSTGCRN :
   1. **Temporary update** : Replace the local parameters with the corresponding aggregated parameters.
   2. **Validation** : Compute the validation loss using a local validation set.
   3. **Selective update** : Retain aggregated parameters if validation loss improves ; otherwise, revert to local ones.
5. **Global update** : The validated and selectively updated local models are used for the next round of local training.

– Pham, Furno, Chamroukhi and Oukhellou. Federated Dynamic Modeling and Learning for Spatiotemporal Data Forecasting. ArXiv.2503.04528, 2025

FL for Urbain Mobility

Multimodal transport demand forecasting : three real-world public datasets, which include bike and taxi demand data from New York City (NYC) and Chicago (CHI) :

Prediction results :

Table 2: Datasets description

Dataset	Period	Number of nodes
NYC-Bike[2]	From 01/04/2016 to 30/06/2016	283
NYC-Taxi[3]	From 01/04/2016 to 30/06/2016	263
CHI-Taxi[4]	From 01/04/2024 to 30/06/2024	77

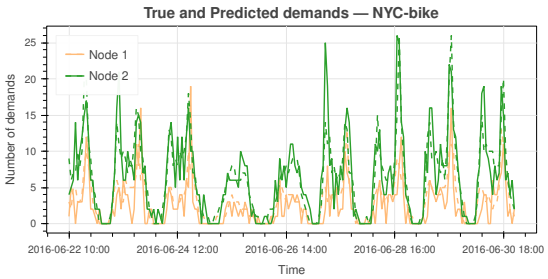
Table 3: Performance comparison of local models on transport demand datasets

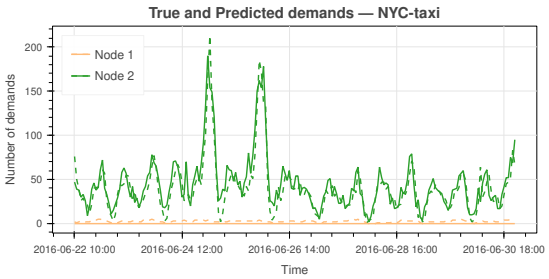
Model	NYC-Bike		NYC-Taxi		CHI-Taxi	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
GRU	2.1518	3.3612	11.7579	31.6111	2.6852	6.9351
LSTM	1.9727	3.2877	11.3981	30.5406	2.9644	7.0128
AGCRN Bai et al. (2020)	1.9793	**2.9422**	10.3327	**24.1955**	2.8162	6.8642
DSTGCRN Gong et al. (2024)	1.9593	3.0953	9.9527	26.4385	2.4335	6.6989
LSTM-DSTGCRN (ours)	1.8994	3.0450	9.8571	26.3968	2.6110	6.6846
FedGRU Liu et al. (2020)	2.4897	3.5930	11.0238	31.4011	2.7614	**6.6305**
FedLSTM Zeng et al. (2021)	2.2396	3.4365	11.5929	31.2145	2.9620	7.4549
AGCRN + Attentive Li and Liu (2024)	2.0143	3.1892	11.0276	27.3424	3.1452	8.2231
LSTM-DSTGCRN + Attentive	1.9323	3.2024	11.2389	26.1926	2.5551	6.8292
LSTM-DSTGCRN + Attentive + CSV	**1.8667**	3.4377	11.7614	28.3069	**2.3127**	8.2746
LSTM-DSTGCRN + FedAvg	1.9423	3.3253	10.3052	26.8824	3.3121	6.9915
LSTM-DSTGCRN + FedAvg with CSV	1.8677	3.1978	**9.3249**	26.1514	2.5453	7.2776

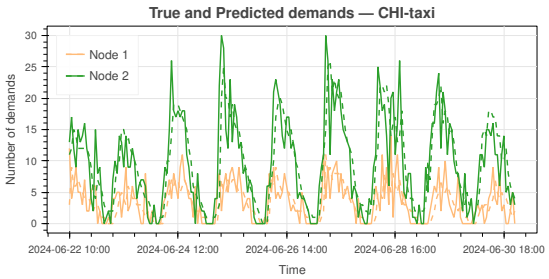
Table 4: Ablation study of LSTM-DSTGCRN model on transport demand datasets

Model	NYC-Bike		NYC-Taxi		CHI-Taxi	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Without LSTM	1.9408	3.1432	10.1305	**26.1436**	3.3061	6.9760
Without MA	1.9143	3.1485	9.9951	27.0137	2.9370	6.8889
Without LSTM and MA	1.9661	3.1658	10.6862	26.7259	2.9259	6.9825
LSTM-DSTGCRN (our)	**1.8994**	**3.0450**	**9.8571**	26.3968	**2.6110**	**6.6846**

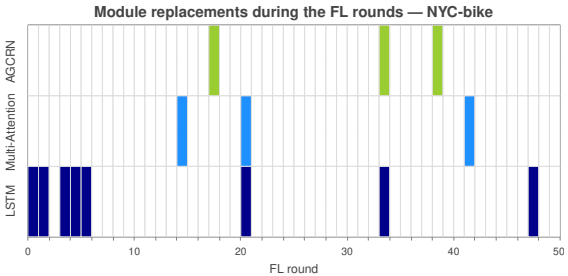
Prediction results :

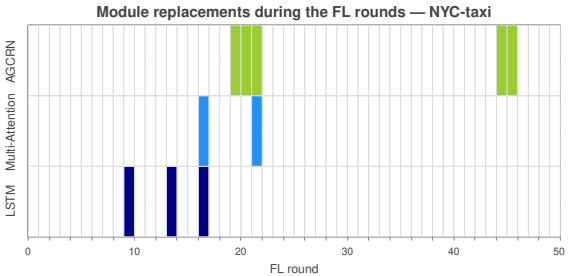


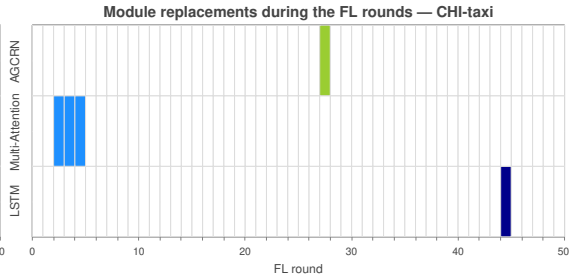




Client-Side Validation :







– Pham, Furno, Chamroukhi and Oukhellou. Federated Dynamic Modeling and Learning for Spatiotemporal Data Forecasting. ArXiv.2503.04528, 2025

FL for Urbain Mobility : Origin-Destination matrix forecasting

Table 5: Comparison of the local learning and federated learning across years and clients

Model	2021				2022			
	Lyon PT		Orange Telecom		Lyon PT		Orange Telecom	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
GRU (locally)	55.0299	134.6204	780.2223	1515.2890	47.5444	129.3204	927.7017	1645.8309
LSTM (locally)	89.1214	278.9800	788.6166	1487.3965	86.2957	284.2191	925.4053	1617.0138
LSTM-DSTGCRN (locally)	**42.3453**	**107.7701**	722.3093	1137.5153	46.7632	158.5128	607.1526	939.9001
FedGRU	96.8846	264.2408	773.3111	1525.6516	68.1231	160.2127	978.3316	1701.6402
FedLSTM	92.0171	330.3783	780.8315	1423.6266	93.6214	283.8421	1061.4070	1711.0444
LSTM-DSTGCRN + FedAvg	58.2769	127.7215	**693.0593**	**1099.7407**	44.0000	179.1663	**579.1404**	**833.5996**
LSTM-DSTGCRN + FedAvg with CSV	46.1419	123.1041	737.1721	1193.0021	**36.8111**	**89.4602**	636.3386	917.6329
LSTM-DSTGCRN + FedAvg with CSV	46.1419	123.1041	737.1721	1193.0021	**36.8111**	**89.4602**	636.3386	917.6329

(c) LyonPT and Orange Telecom OD forecasting results

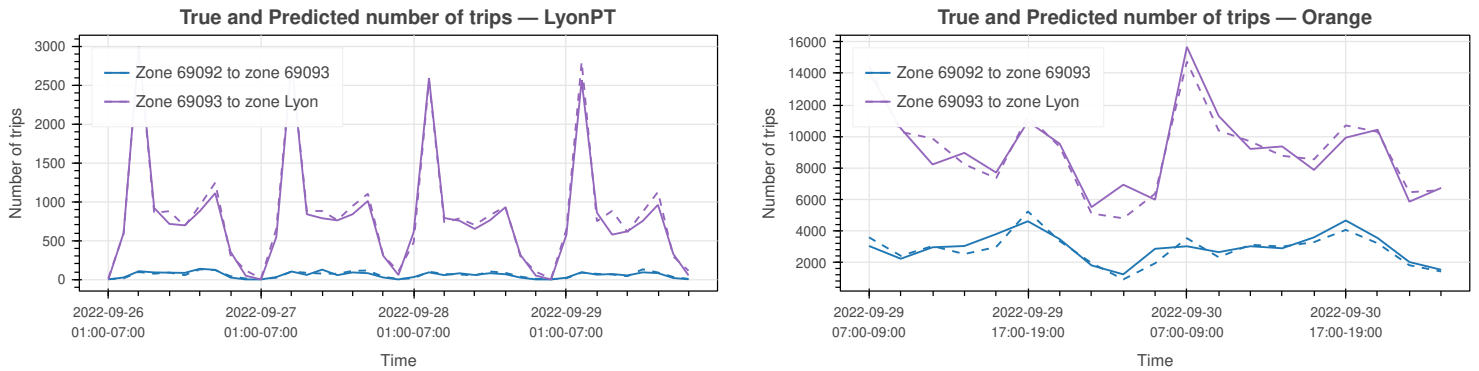
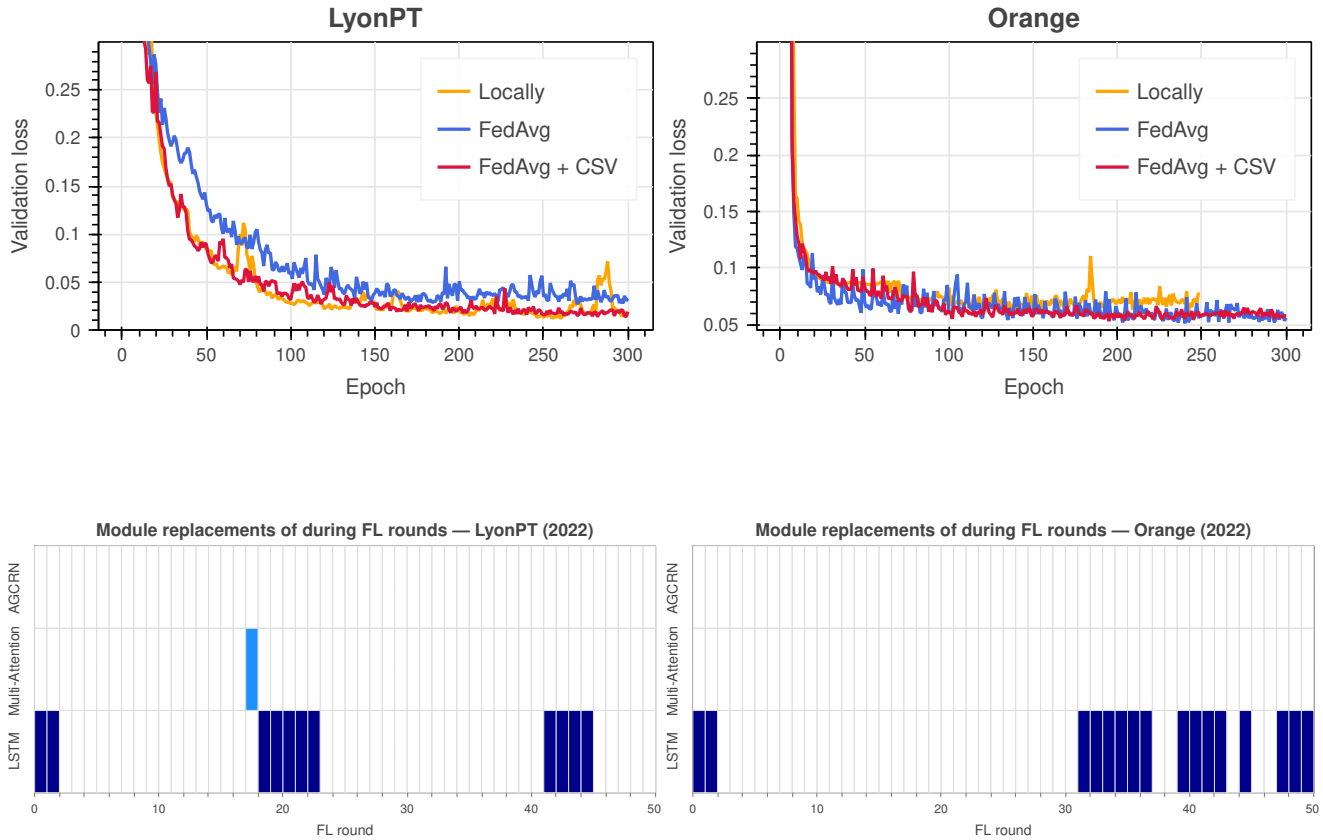


FIGURE – Forecasts given by LSTM-DSTGCRN + FedAvg with CSV at some random OD pairs.

– Pham, Furno, Chamroukhi and Oukhellou. Federated Dynamic Modeling and Learning for Spatiotemporal Data Forecasting. ArXiv.2503.04528, 2025

Comparison of local models and FL approaches on OD matrix datasets



We can see that the LSTM module is updated more frequently than the other modules. This suggests that the clients learned from each other's temporal patterns, which are more relevant to the overall model's performance.

– Pham, Furno, Chamroukhi and Oukhellou. Federated Dynamic Modeling and Learning for Spatiotemporal Data Forecasting. ArXiv.2503.04528, 2025

Final remarks

- Latent variable models are flexible and can be efficiently built upon complex distributions
- Available grounded framework tools of model training and selection
- Penalization joins hybrid (Data-Physics) Machine Learning - eg. ML to account for real-wold physical : $\hookrightarrow$ Applications in augmented physical simulation, augmented medecine
- ML models uncertainty can be casted as a feature of trustworthiness (trust by design)
- Federated learning fosters trustworthiness (privacy preserving) while reducing the need to collect, store, process and transfer large amounts of data/models

Thank you for your attention !