Reservoir computing for the modelization of mixed effects in longitudinal data

François Plessier

4th February 2025

Introduction
Context of the study
The data

Mixed Models

Reservoir Computing

MixedML approach

Context of the study

Context: longitudinal health data with **inter-individual heterogeneity** and **noise in the measures**.

Parametric models such as mixed effect models are commonly used.

- Understanding of the underlying mechanisms.
- However, they are sensitive to the specification.

We consider Reservoirs Computing models:

- Neural networks adapted for time series.
- Able to model complex (time) relationships from "naive specifications".
- But... they are "black boxes" (prediction only).

The data

We consider a regression problem. Synthetic datasets are used, so we can control:

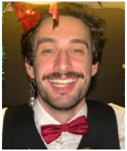
- the complexity of the relationship between the variables,
- the inter-individual heterogeneity,
- the noise on the measurements.

The data

We consider a regression problem. Synthetic datasets are used, so we can control:

- the complexity of the relationship between the variables,
- the inter-individual heterogeneity,
- the noise on the measurements.

All the work on datasets has been done by Arthur.



The data

We generate 100 training datasets (replicas), and 1 test dataset.

Each dataset has:

- repeated measures on $t \in [0, 25]$ time periods
- for $i \in [1, 500]$ individuals

The data

We generate 100 training datasets (replicas), and 1 test dataset.

Each dataset has:

- repeated measures on $t \in [0, 25]$ time periods
- for $i \in [1, 500]$ individuals

Each model is trained on each one of the 100 replicas.
The resulting 100 models are then evaluated on the test dataset.
We'll look at the predictions distribution and the MSE.

$$MSE = \frac{1}{N_r * N_i * N_t} \sum_{r=1}^{100} \sum_{i=1}^{500} \sum_{t=5}^{25} (\hat{y}_{i,t} - y_{r,i,t})^2$$

The data

We use 7 time-dependant covariates:

$$
\begin{aligned}
X_{1,i}(t) =& \alpha^1_{0,i} + \alpha^1_{1,i} t \\
X_{2,i}(t) =& \alpha^2_{0,i} + \alpha^2_{1,i} log(t+1) \\
X_{3,i}(t) =& \alpha^3_{0,i} + \alpha^3_{1,i} t^2 \\
X_{4,i}(t) =& \alpha^4_{0,i} + \alpha^4_{1,i} e^{-0.002t}
\end{aligned}
\qquad
\begin{aligned}
X_{5,i}(t) =& \frac{\alpha^5_{0,i}}{1 + e^{-\alpha^5_{1,i} t}} \\
X_{6,i}(t) =& max(0, \alpha^6_{0,i} + \alpha^6_{1,i} \cdot t^2) \\
X_{7,i}(t) =& \frac{\alpha^7_{0,i}}{1 + e^{-\alpha^7_{1,i} t}}
\end{aligned}
$$

where all $\alpha^k_{k,i}$ are generated from normal distributions.

We also use a categorical covariate: $X_{8,i} = 1$ if i is even else 0[1]

[1] yes, it is "$1 + i \ mod \ 2$"

The data

We generate 1 variable of interest **without** individual effects:

$$\begin{aligned} Y^*_{fixed,i} =& \gamma_0 + \gamma_1 \cdot X_{2,i}(t) \cdot X_{5,i}(t) + \\ & \gamma_2 \cdot X_{4,i}(t) \cdot X_{7,i}(t) + \gamma_3 \cdot X_{6,i}(t) \cdot X_{8,i} \end{aligned}$$

where all γ_k are generated for **all** individuals from uniform distributions.

We generate 1 variable of interest **with** individual effects:

$$\begin{aligned} Y^*_{mixed,i} =& \gamma_{0,i} + \gamma_{1,i} \cdot X_{2,i}(t) \cdot X_{5,i}(t) + \\ & \gamma_{2,i} \cdot X_{4,i}(t) \cdot X_{7,i}(t) + \gamma_{3,i} \cdot X_{6,i}(t) \cdot X_{8,i} \end{aligned}$$

where all $\gamma_{k,i}$ are generated for **each** individuals from normal distributions.

The data

... and we generate the corresponding noisy variables:

$$Y_{fixed,i} = Y^*_{fixed,i} + \epsilon_{fixed,i}(t)$$
$$Y_{mixed,i} = Y^*_{mixed,i} + \epsilon_{mixed,i}(t)$$

where all $\epsilon_{k,i}(t)$ are generated from normal distributions.

The data

... and we generate the corresponding noisy variables:

$$
\begin{aligned}
Y_{fixed,i} &= Y^*_{fixed,i} + \epsilon_{fixed,i}(t) \\
Y_{mixed,i} &= Y^*_{mixed,i} + \epsilon_{mixed,i}(t)
\end{aligned}
$$

where all $\epsilon_{k,i}(t)$ are generated from normal distributions.

The trainings are done on the noisy variables.
The MSE is computed on both the noiseless and noisy variables.

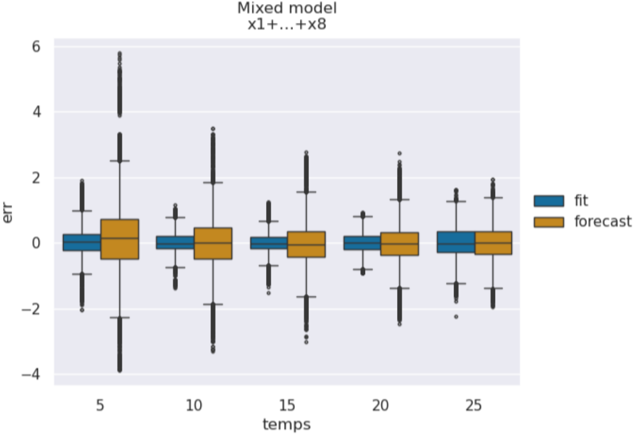
Mixed Models: modification of the prediction method)

For a fair comparison between the Mixed Models and the Reservoir Models, we consider a forecasting method, where the predictions for t_i is done using only the data from $t < t_i$:

```
forecast <- function(model, data) {
  temps <- unique(data[, TSTEP])
  x_labels <- model$Xnames
  data[, INTRCPT] <- 1
  # initialization with the (intercept) marginal effect
  pred <- as.vector(predictY(model, newdata = data, marg = TRUE)$pred)
  for (t in temps[-1]) {
    prev_data <- data[data[TSTEP] < t, ]
    ui <- predictRE(model, prev_data)
    # addition of the random effects
    reffects <- rowSums(data[data[TSTEP] == t, x_labels] * ui[, x_labels])
    pred[data[TSTEP] == t] <- pred[data[TSTEP] == t] + reffects
  }
  return(pred)
}
```

Mixed Models: modification of the prediction method

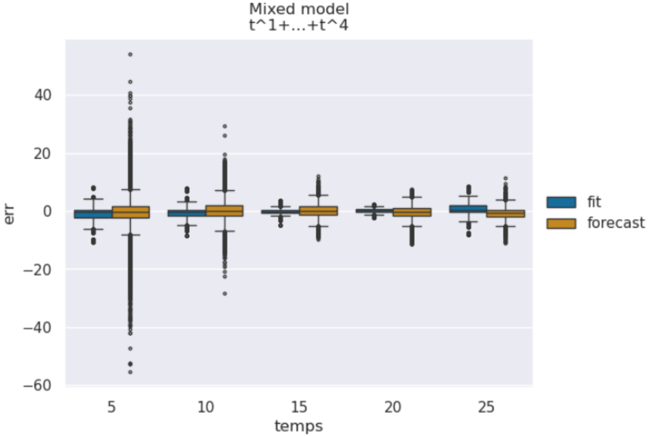
Impact of this method when considering the Mixed Effect datasets:



$$err_{r,t} = \frac{1}{N_i}\sum_{i=1}^{500}(\hat{y}_{i,t} - y_{r,i,t})$$

Mixed Models: modification of the prediction method

Impact of this method when considering the Mixed Effect datasets:



$$err_{r,t} = \frac{1}{N_i} \sum_{i=1}^{500} (\hat{y}_{i,t} - y_{r,i,t})$$

Analysis of the Mixed Models results.

Here are the MSE for the fixed effects dataset:

	dataset	train-fixed		test-fixed	
	target	w/o noise	w/ noise	w/o noise	w/ noise
Linear model	x2*x5+x4*x7+x6*x8	0.00027	1.0	0.00026	0.99
	x1+...+x8	21.	22.	20.	21.
Mixed model	x1+...+x8	0.35	1.3	0.36	1.3
	t^1+...+t^4	3.2	4.2	3.1	4.1

- Impact of the specifications
- The Mixed Models use the random effects compensates the incorrect specification

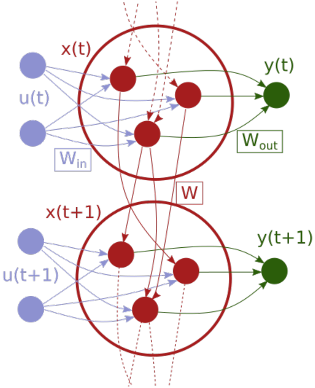
Analysis of the Mixed Models results.

Here are the MSE for the fixed effects dataset:

	dataset	train-mixed		test-mixed	
	target	w/o noise	w/ noise	w/o noise	w/ noise
Mixed model	x2*x5+x4*x7+x6*x8	0.18	1.2	0.17	1.2
	x1+...+x8	0.46	1.5	0.45	1.4
	t^1+...+t^4	8.4	9.4	8.9	10.

- Impact of the specifications
- The Mixed Models are efficient for correcting the noise in the observations.

Reservoir Computing



$x(t)$:	state of the Reservoir
W_{in}:	fixed input weights
W:	fixed recurrent weights with random initialization (*Echo State Property*)
W_{out}:	trainable output weights (Ridge regression)

$$y(t) = W_{out}\; x(t)$$

$$x(0) = 0 \;\; ; \;\; x(t+1) = (1-\alpha)\; x(t) \; + \; \alpha\, f(u(t+1), x(t))$$

$$f(u(t+1), x(t)) = \tanh(W_{in}\, u(t+1) \; + \; W\, x(t))$$

Training using Reservoir Computing

- "Warm-up" steps ($N = 5$) are used so the reservoir "forgets" the initialization step.
- Use of a Reservoirs ensemble from multiple ($N = 5$) random seeds: $y_{pred} = 1/N_{seeds} \sum_{seeds} y_{pred,seed}$
- Hyper-parameters optimization.

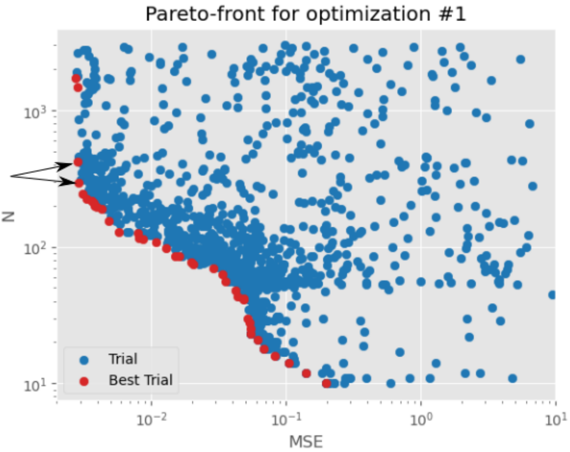
Hyper-parameters (HPs) optimization

We use Optuna to optimize the main hyper-parameters of the reservoir models (Bayesian optimization):

- the *number of neurons* of the reservoir
- the *leaking rate* (α) controlling the information retention in reservoir neurons over time
- the *spectral radius* of the reservoir, impacting its stability and memory capacity
- the *input scaling*, which is the gain applied to the input to the reservoir
- the coefficient of the *Ridge penalization* of the read-out layer.

Hyper-parameters optimization: Pareto front

To pickup the best set of HPs, we plot the Pareto front between the MSE on the validation set, and the number of neurons.



Hyper-parameters optimization

We use 2 optimization steps:

1. one optimization using replica #1 as the train set and replica #2 as the validation set.
2. one optimization using replica #2 as the train set and replica #1 as the validation set.

The final HPs are calculated as the mean of each optimization's HPs.

Reservoir Models results

Here are the MSE for the fixed effects dataset.

	dataset	train-fixed		test-fixed	
	target	w/o noise	w/ noise	w/o noise	w/ noise
Linear model	x2*x5+x4*x7+x6*x8	0.00027	1.0	0.00026	0.99
Mixed model	x1+...+x8	0.35	1.3	0.36	1.3
	t^1+...+t^4	3.2	4.2	3.1	4.1
Reservoir model	x2*x5+x4*x7+x6*x8	0.0046	1.0	0.0033	0.99
	x1+...+x8	0.035	0.99	0.074	1.1

- More robust to the "specifications" (input).
- Also efficient at correcting the noise.

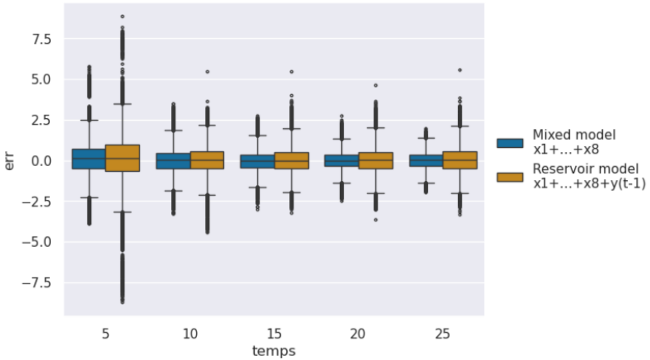
Reservoir Models results

Here are the MSE for the mixed effects dataset.

	dataset	train-mixed		test-mixed	
	target	w/o noise	w/ noise	w/o noise	w/ noise
Mixed model	x2*x5+x4*x7+x6*x8	0.18	1.2	0.17	1.2
	x1+...+x8	0.46	1.5	0.45	1.4
	t^1+...+t^4	8.4	9.4	8.9	10.
Reservoir model	x2*x5+x4*x7+x6*x8	2.9e+02	2.9e+02	2.5e+02	2.6e+02
	x2*x5+x4*x7+x6*x8+y(t-1)	0.52	1.5	0.57	1.6
	x1+...+x8+y(t-1)	0.61	1.6	0.67	1.7

- It seems that we might need to use $y(t-1)$ in the input.
- Very close to the Mixed Models, on a dataset tailored for them.

Reservoir Models results



- Higher variance for the Reservoir (Ensemble) Model.
- Need to study a bigger Ensemble Model.

MixedML approach

This approach combines a Machine Learning (ML) model and a Mixed Effect model.

The training is done so that:

- the ML model is trained to predict the fixed effects.
- the Mixed Effect model is trained to predict the random/individual effects.

```
random_hlme <- hlme(
  data_residual ~ 1,
  random = ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8,
  idiag = TRUE,
  data = data_train,
  subject = SUBJECT,
  var.time = TSTEP
)
```

MixedML approach

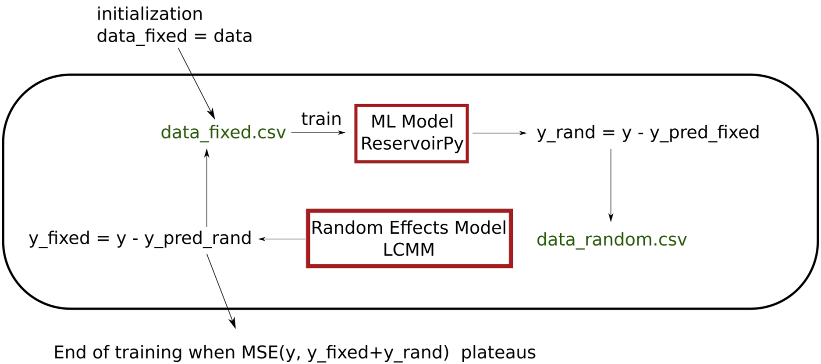
The training is iterative.
This approach combines a Machine Learning (ML) model and a Mixed Effect model.

The training is done so that:

- the ML model is trained to predict the fixed effects.
- the Mixed Effect model is trained to predict the random/individual effects.

```
random_hlme <- hlme(
  data_residual ~ 1,
  random = ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8,
  idiag = TRUE,
  data = data_train,
  subject = SUBJECT,
  var.time = TSTEP
)
```

MixedML approach: a lovely illustration



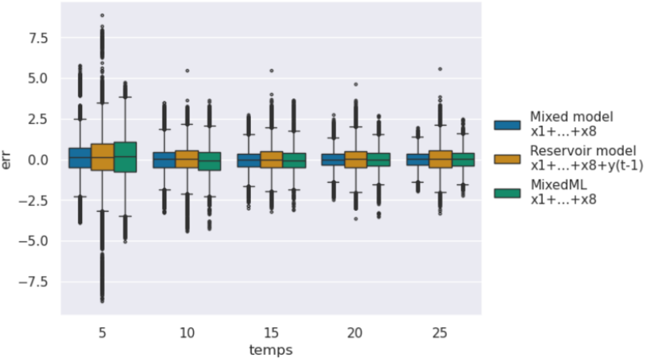
MixedML approach: results

Here are the MSE for the mixed effects dataset.

	dataset	train-mixed		test-mixed	
	target	w/o noise	w/ noise	w/o noise	w/ noise
Mixed model	x1+...+x8	0.46	1.5	0.45	1.4
Reservoir model	x1+...+x8+y(t-1)	0.61	1.6	0.67	1.7
MixedML	x1+...+x8	0.68	1.7	0.64	1.6

I haven't said my last word...

MixedML approach: results



- The variance is better than the Reservoir (Ensemble) Model.

Thank you!