Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Do radial basis function (RBF) units and validated model selection help robust Takagi–Sugeno fuzzy identification? A comparative study

Achraf TELMOUDI

Achraf.telmoudi@ieee.org

Tunis University

Jaouher Chrouta

Tunis University

Imen Saidi

Tunis El Manar University

Research Article

Keywords:

Posted Date: November 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11586570/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Do radial basis function (RBF) units and validated model selection help robust Takagi–Sugeno fuzzy identification? A comparative study

Abstract

Robust Takagi–Sugeno (TS) fuzzy identification builds local linear models whose memberships resist outliers. This study asks whether adding robust radial basis function (RBF) units to such a rule base, and selecting models by cross-validation, helps, and how the result compares with well-tuned generic learners. None of the components is new. Sixteen models are compared on 13 simulated, semi-simulated and measured applications under a deployable protocol (rule weights computed from the input only), with paired replications, ranks, Wilcoxon tests with Holm correction and bootstrap effect sizes. The joint Huber fit of the rule base C2 with anisotropic RBF units and validated ridge priors (HYB1+) lowers the median error by 5.5 % (bootstrap interval -15.0 to -0.3 %), significantly on nine applications and with a significant increase on three, and by up to 41 % on contaminated low-dimensional nonlinear series. HYB1+ is not superior to generic learners: its mean rank (8.2) lies between those of support-vector regression (6.6) and random forest (9.5), and a linear model is the most accurate in free-run simulation of the cascaded tanks. Tuning random forest, boosting and a neural network by cross-validation does not change the ranking. A selection and stacking ensemble has the best mean rank (2.9) but is favoured by construction, retains a single fuzzy model in 3 % of replications, and is within 2 % of a selector restricted to generic learners on 10 of 13 applications. Benchmarks are evaluated in one-step prediction on decimated data, and the ablation uses eight replications.

Keywords: Takagi–Sugeno fuzzy models; fuzzy c-regression; outliers; radial basis function networks; model selection

1 Introduction and literature review

Identification of nonlinear dynamical systems from measured data is a prerequisite for model-based control, prediction and fault detection. Takagi–Sugeno (TS) fuzzy models [1, 2] approximate a nonlinear system by blending local linear models, and are attractive because each local model can be analysed and controlled with linear tools. The rules of a TS model are commonly obtained by fuzzy c-regression (FCRM) [3], which partitions the data according to the fit of c linear models. The models of this paper are assessed in one-step prediction and, on three benchmarks, in free-run simulation; the motivation by control and simulation applies to the rule bases and their hybrids, not to the stacks selected by HYB-SEL, which have no local-linear form.

Measured data are rarely clean. Sensor faults, communication errors and unmodelled events produce outliers, and a model that is identified by least squares or by fuzzy clustering is sensitive to them. Fuzzy c-regression has two well-known weaknesses in this setting: the memberships are not robust, so that a few outliers can attract a rule, and the memberships are computed from the output error, which is not available when the model is used to predict. The general problem is therefore to obtain a local-linear model that is accurate on contaminated data *and* usable as a predictor, that is, whose rule weights depend on the input only.

Possibilistic clustering [4] and noise clusters [5] reduce the first weakness. The robust possibilistic c-regression model (RPCRM) [6] combines the fuzzy and possibilistic memberships of the regression error and adds a noise cluster, and a modified FCRM [7] mixes the distance to the cluster centre and the prediction error with a coefficient λ. These ideas leave open several practical questions, which define the problem treated in this paper:

1. How much does a criterion that combines the distance to the centre and the absolute prediction error, with weights $\alpha + \beta = 1$, improve RPCRM, and where does the improvement come from?

2. Is the improvement due to the criterion, or to the way the rule weights are computed at prediction time?

3. Can the rule base be complemented by robust RBF units [8], and does the combination remain robust to outliers?

4. Can the choice between a rule base, its hybrid and generic learners be made reliably from the training data alone?

5. On which real measurements does any of this help?

To place these questions in the existing work, the next paragraphs review the literature.

We review the literature along three questions: how local models are built, how outliers are handled, and whether the model can be used for prediction without the output. Fuzzy and local-model identification starts with the TS structure [1, 2], and general views of nonlinear identification are given in [9, 10]. Alternatives with different data requirements are kernel methods [11, 12], tree ensembles [13, 14], and, more recently, deep and physics-informed models [15–21]. Table 1 compares the families that are directly relevant here.

Table 1: Comparison of families of identification methods. "Outliers" is the explicit mechanism, "Deployable" tells whether prediction needs only x, "Local linear" whether local linear models (quasi-LPV form) are obtained, "n/a not applicable.

Family and references	Training principle	Outliers	Deployable	Local linear	Cost in N
TSK-FCM, ANFIS [1, 22]	clustering, then least squares	none	yes	yes	linear
Fuzzy c-regression [3]	fit-driven memberships	none	no (uses y)	yes	linear
Noise cluster, possibilistic [4, 5]	noise cluster, typicality	explicit	clustering only	n/a	linear
Robust FCRM, RPCRM [6, 23–26]	noise cluster, kernel metric, M-estimators	explicit	mostly no	yes	linear
Type-2 TS, type-2 FCRM [27–31]	interval memberships	implicit	depends	yes	linear
Mixture of experts and regressions [32–35]	EM likelihood	Student-t	yes (gating)	yes	linear (unbounded likelihood)
RBF and fuzzy-equivalent networks [8, 36–38]	linear output layer, or gradient	Huber (robust variant) [39, 40]	yes	no	linear to cubic
GPR, SVR, trees [11–14]	kernel or averaging	ε-loss, averaging	yes	no	cubic (GPR), quadratic (SVR), $N \log N$ (trees)

Table 2: Chronology of representative contributions on Takagi–Sugeno identification.

Year	Works	Contribution
1985–88	[1, 2]	TS model; structure identification of fuzzy models
1991–93	[3–5, 22]	noise cluster; fuzzy c-regression; possibilistic clustering; ANFIS
2009	[26]	hybrid robust approach for TSK modelling with outliers
2011	[7]	modified FCRM mixing distance and prediction error
2015–17	[23, 24]	robust fuzzy c-regression; robust kernel clustering for nonlinear identification
2019–20	[6, 25, 29]	robust interval type-2 FCRM; robust FCRM; RPCRM with a possibilistic regression criterion
2020	[37, 38]	functional equivalence of TSK systems to mixtures of experts and neural networks; gradient training of TSK systems
2022–24	[30, 31, 41, 42]	interval type-2 identification (modified and improved type-2 FCRM); sparse TS identification; hybrid TS representations
2024–26	[43, 44]	TS modelling and control (activated sludge); state estimation for uncertain fuzzy systems
This study	–	hybridisation of a robust rule base with robust RBF units; deployable comparison with literature learners

Table 2 lists representative contributions on robust and type-2 TS identification in chronological order, with the scope stated by the title of each cited work.

Three observations follow. First, robust TS identification has been developed almost entirely at the level of the clustering criterion [6, 23, 24, 29]; the memberships use the regression error, so the reported errors are those of a fit with access to y, and what remains at deployment is not addressed. Second, mixtures of experts [32] and TSK systems [37] are structurally equivalent, but the probabilistic version has a likelihood that is unbounded when the variances are free [45, 46]; robust regression tools [39, 40] have not been combined with a robust rule base and RBF units in the TS literature reviewed here. Third, comparisons with generic learners [11–13, 47, 48] are rare in this literature, although tree ensembles and kernel machines are strong on medium-size tabular data.

TS models are also the basis of stability analysis and control design [49], including robust tracking with unmeasurable premise variables [50], manipulator control [51, 52], process control [43], state estimation [44] and data-driven models of batteries [53]; recent work covers sparse structures [41], hybrid representations [42] and evolving and neuro-fuzzy methods [54]. The quality of the identified local models therefore matters beyond prediction, and interpretable models are argued for in [55].

From the literature to the proposal. The review leaves a gap: robust TS rule bases, type-1 or type-2, are built from criteria that use the output and are rarely complemented by a robust nonlinear correction, and they are rarely compared with generic learners under a deployable protocol. Our contribution is the proposal and the evaluation as a whole: (i) the hybridisation of a robust rule base with robust RBF units (RRBF) fitted jointly (Section 2.2); (ii) the supporting elements, namely a combined distance–error criterion with noise cluster, a learned antecedent that needs only x, a global Huber refit and a gradient-tuned base; and (iii) the assessment of its relevance and performance, with paired replications, an ablation and a comparison with literature learners on several applications, including measured data. The evaluation is part of the contribution: it shows where the hybridisation helps and where it does not.

Objective. The objective of this study is to determine, by experiment under a deployable protocol, whether a robust TS rule base complemented by robust radial basis function (RBF) units gives a more accurate and more robust predictor than the rule bases of the literature, and how it compares with well-tuned generic learners, and on which kind of data they do not.

Methodology. We build robust rule bases (a combined distance–error criterion with noise cluster, a Gaussian antecedent learned from the input only, a global Huber refit of the consequents and a gradient-tuned variant), hybridise them with robust RBF units fitted jointly with a Huber loss, and compare the result with generic learners, including a cross-validated selection and stacking ensemble used as a reference. The models are analysed mathematically, described by flowcharts and algorithms, and compared on simulated, semi-simulated and measured data, with paired replications, rank statistics, paired tests and multiple-comparison control.

Contributions. (i) a positioning with respect to the literature on robust TS identification, by robustness mech-

anism, deployability and cost (Tables 1 and 2), which is not a systematic review; (ii) the joint Huber fit of a robust rule base with robust RBF units, an extended version HYB1+ with anisotropic units and validated priors, and their rationale (Section 2); (iii) a reference ensemble, HYB-SEL, that selects and stacks models from the training data only, used to measure what the fuzzy components add to generic learners, and a previous-error and sign correction for one-step dynamic prediction (Sections 2.6 and 2.7); (iv) flowcharts, algorithms, complexity and a mathematical analysis of standard properties (Section 3 and Appendix B); (v) a deployable comparison of 16 models, including RPCRM, a gradient-tuned TSK base and Gaussian-process, support-vector, tree-based, neural and Huber baselines, on 13 applications, with a conditional ranking in which three of the generic learners are tuned by cross-validation and a free-run simulation on the cascaded tanks and the EMPS data (Sections 4 and 5); (vi) an explicit account of advantages and failures (Section 6). None of the building blocks is new: the equivalence of TSK systems and RBF networks, Huber regression, cross-validation with the one-standard-error rule and stacking are established, and the combined criterion is a variant of the modified FCRM [7]. What the study adds is their combination in a deployable form, the evidence on where the combination helps, and the report of where it does not.

Expected results. We formulate two hypotheses, tested in the experiments: H1, the RBF hybridisation reduces the error of a robust rule base on contaminated nonlinear problems; H2, the hybrid HYB1+ is competitive with generic learners under the same protocol. We also expect the answer to depend on the data: the study aims at stating where the proposals help and where they do not, not at a method that is better in general.

Organisation of the paper. Section 2 presents the rule bases, the RBF hybrids and the selection procedure. Section 3 gives the flowcharts, the algorithms and the complexity. Section 4 describes the applications, the compared models and the statistical protocol, and Section 5 the results. Section 6 discusses the evidence and the limitations, and Section 7 concludes. Appendix A describes the alternative objective, the global Huber refit and the gradient-tuned rule base, and Appendix B gives the mathematical analysis.

2 Proposed approach: rule bases, radial basis function hybrids and validated model selection

Notation: c rules, memberships μ_{ik}, centres v_i, parameters θ_i, fuzzifier m.

2.1 Background: TS model, c-regression clustering and RBF networks

TS model and its linear-in-parameters (quasi-LPV) form. Let $x \in \mathbb{R}^p$ be the regressor (for dynamic systems, lagged outputs and inputs) and $\bar{x} = [x^T \ 1]^T$. A TS model with c rules predicts

$$\hat{y}(x) = \sum_{i=1}^{c} w_i(x)\, \theta_i^T \bar{x} = \Theta^T \varphi(x), \qquad \varphi(x) = \left[w_1(x)\bar{x}^T, \ldots, w_c(x)\bar{x}^T\right]^T, \tag{1}$$

with $\sum_i w_i(x) = 1$. For fixed antecedent weights w_i, (1) is linear in $\Theta = [\theta_1^T, \ldots, \theta_c^T]^T$. If x contains lagged outputs and inputs, (1) is a state-dependent ARX model

$$y(t) = \sum_{k=1}^{n_a} a_k(\xi_t)\, y(t-k) + \sum_{k=1}^{n_b} b_k(\xi_t)\, u(t-k) + c_0(\xi_t), \quad a_k(\xi) = \sum_i w_i(\xi)\theta_{i,k}, \tag{2}$$

i.e. a quasi-LPV model, which at a frozen ξ_t is a linear time-invariant model. We use this only as a description: global stability of (2) is not established by the experiments.

Clustering-based identification: FCRM, PCRM, RPCRM and MFCRM. FCRM minimises

$$J_{\mathrm{FCRM}} = \sum_{k=1}^{N} \sum_{i=1}^{c} \mu_{ik}^m E_{ik}^2, \qquad E_{ik} = |y_k - \theta_i^T \bar{x}_k|. \tag{3}$$

RPCRM [6] introduces a noise cluster at constant distance δ and uses memberships built from the fuzzy and possibilistic parts of the regression error, so that points far from every local model receive a low total membership. The local parameters are obtained by weighted least squares. The modified FCRM [7] replaces the

squared error by

$$\lambda E_{ik}^2 + (1-\lambda)\, d_{ik}^2, \qquad d_{ik} = \|x_k - v_i\|, \tag{4}$$

where d_{ik} is the distance to the centre of rule i. In this paper, "MFCRM-like" denotes our own simplified re-implementation of that idea, with the robust scales defined below and without noise cluster; it is not the original code.

Interval type-2 models. Interval type-2 TS models and interval type-2 FCRM [27–31] replace the fuzzifier by an interval and add a type reduction; they are part of the literature reviewed in Section 1 but are not part of the comparison of this paper.

RBF networks and their robust (Huber) variant RRBF. An RBF network predicts

$$\hat{y}(x) = \beta_0 + \sum_{j=1}^{J} \beta_j \exp\Big(-\frac{\|x - C_j\|^2}{2s^2}\Big), \tag{5}$$

which is linear in β; centres C_j are given by k-means and β by regularised least squares [8]. The robust variant (RRBF) estimates β with the Huber loss [39] by iteratively reweighted least squares and selects the width and ridge by cross-validation on the mean absolute error.

2.2 Rationale of the proposed hybridisation

The proposal is to complement a robust rule base with robust RBF units. Three arguments motivate it, and each is tested later rather than assumed. (i) *Complementary approximation.* A few local linear rules describe the large-scale structure with few parameters, but leave a bias where the curvature inside a rule is strong; RBF units correct local nonlinearity but, used alone, need many units and follow outliers. TSK systems and RBF networks are functionally related [36,37], so the combination is not redundant only if the units are placed where the rules are weak and are fitted jointly with the rules. (ii) *Use of the robust partition.* The noise cluster identifies the points that no rule explains. Placing the RBF centres on non-noise points only, and fitting with the Huber loss, prevents the correction from following outliers [5, 39, 40]. (iii) *Convex joint fit.* For a fixed antecedent the joint problem is strictly convex (Proposition 2), which adds capacity without a non-convex training stage; the ridge prior towards the local parameters keeps the rules interpretable. The hypotheses tested are: H1, the hybrid improves its rule base on contaminated, low-dimensional nonlinear problems; H2, the hybrid HYB1+ is competitive with generic learners. H1 is expected to fail on nearly linear or high-dimensional data, and H2 on data whose operating conditions differ from those of training.

2.3 Combined distance–error criterion (DE) with noise cluster

The squared distance of point k to rule i is

$$D_{ik}^2 = \alpha\Big(\frac{d_{ik}}{s_d}\Big)^2 + \beta\Big(\frac{E_{ik}}{s_E}\Big)^2, \quad \alpha + \beta = 1, \tag{6}$$

where s_d and s_E are robust scales computed once on the training set (median distance to the median of x, and MAD of y) so that α does not depend on units. The objective keeps the noise cluster,

$$J = \sum_k \sum_i \mu_{ik}^m D_{ik}^2 + \delta^2 \sum_k \Big(1 - \sum_i \mu_{ik}\Big)^m, \quad \delta^2 = \kappa\, \mathrm{med}_k \min_i D_{ik}^2, \tag{7}$$

and is minimised by alternating the updates

$$\mu_{ik} = \frac{(1/D_{ik}^2)^{1/(m-1)}}{\sum_j (1/D_{jk}^2)^{1/(m-1)} + (1/\delta^2)^{1/(m-1)}}, \tag{8}$$

$$v_i = \frac{\sum_k \mu_{ik}^m x_k}{\sum_k \mu_{ik}^m}, \tag{9}$$

$$\theta_i = \Big(\sum_k \mu_{ik}^m \bar{x}_k \bar{x}_k^T\Big)^{-1} \sum_k \mu_{ik}^m \bar{x}_k y_k. \tag{10}$$

With $\alpha = 0$ the criterion reduces to RPCRM-type regression memberships. We use $\kappa = 9$; $\alpha \in \{0, 0.1, 0.3, 0.6, 0.9\}$ is chosen by two-fold cross-validation.

Rule weights at prediction time: distance rule versus learned Gaussian antecedent. The memberships (7) depend on y, which is unknown at prediction. Two choices of $w_i(x)$ are compared. *Distance rule*: normalised inverse distances to the centres v_i. *Learned Gaussian antecedent*:

$$w_i(x) = \frac{\pi_i \mathcal{N}(x; \mu_i, \Sigma_i)}{\sum_j \pi_j \mathcal{N}(x; \mu_j, \Sigma_j)}, \tag{11}$$

with π_i, μ_i, Σ_i estimated from the final memberships and a shrunk covariance

$$\Sigma_i \leftarrow 0.8\,\Sigma_i + 0.2\,\frac{\mathrm{tr}(\Sigma_i)}{p} I. \tag{12}$$

This is a normalised radial-basis activation, and we call it "+Gauss".

2.4 Hybrid of the rule base with robust RBF units: HYB1

HYB1 (joint basis). Each rule is split into up to three sub-units by k-means on its non-noise points (points whose total membership exceeds 0.5), giving Gaussian units φ_j whose centres therefore ignore outliers. The model

$$\hat{y}(x) = \sum_i w_i(x)\, \theta_i^T \bar{x} + \sum_j \beta_j \varphi_j(x) \tag{13}$$

is fitted jointly with the Huber loss, a ridge prior pulling θ toward the local parameters and a ridge on β; the width multiplier and the ridge are selected by two-fold cross-validation on the mean absolute error. HYB1-X denotes the hybrid built on base model X $\in$ {C2, GT-TSK}.
All models, once the antecedent and units are fixed, are linear in Θ and β; for HYB1,

$$\hat{y} = [\Theta; \beta]^T [\varphi(x); \psi(x)], \tag{14}$$

and the Huber estimate is an iteratively reweighted least-squares solution. Centres, covariances, widths and α remain nonlinear parameters.

2.5 Extended hybrid HYB1+: anisotropic units and validated ridge priors

HYB1 uses isotropic units, three sub-units per rule and two-fold cross-validation, which under-fits smooth regressions and over-selects units on small samples. HYB1+ keeps the joint model (13) and changes the construction of the units and the selection of the hyper-parameters. Each rule is split by k-means on its non-noise points into $k \in \{3, 8\}$ sub-units; unit j has centre C_j and a diagonal width given by the standard deviation σ_{jl} of its sub-cluster in each input l (floored at 5% of the global standard deviation), so that

$$\psi_j(x) = \exp\Big(-\frac{1}{2} \sum_{l=1}^{p} \Big(\frac{x_l - C_{jl}}{s\,\sigma_{jl}} \Big)^2 \Big), \qquad s \in \{0.5, 1, 2\}. \tag{15}$$

The coefficients are obtained by the Huber IRLS of HYB1, with a ridge prior λ_p towards the local parameters of the rule base and a ridge λ_b on the unit coefficients. The grid is $\lambda_p \in \{0.03, 0.3\}$ and $\lambda_b \in \{10^{-2}, 1, 10^4\}$; the value $\lambda_b = 10^4$ reduces the hybrid to its rule base and is the safe fallback. The configuration $(k, s, \lambda_p, \lambda_b)$ is chosen by three-fold cross-validation on the mean absolute error with the one-standard-error rule [56]: among the configurations whose error exceeds the best one by at most the standard error of the paired differences, the one with the strongest ridge λ_b, then the smallest k, is retained. This favours simple models and is the main safeguard against the over-fitting observed with HYB1 on the wind and robot data.

2.6 Reference ensemble HYB-SEL: validated model selection and stacking

A single hybrid is not the best model on every data set, and the literature learners remain more accurate on several (Section 5). HYB-SEL therefore selects, by cross-validation on the training set only, among a small set

of candidates that includes the proposed hybrids and generic learners. The candidates, in increasing order of complexity, are: Huber linear regression, the rule base C2, HYB1, HYB1+, a random forest, and two stacked predictors. The first stacked predictor, Stack3, is the convex combination

$$\hat{y}_{\mathrm{Stack3}}(x) = \omega_1 \hat{y}_{\mathrm{HYB1+}}(x) + \omega_2 \hat{y}_{\mathrm{SVR}}(x) + \omega_3 \hat{y}_{\mathrm{GPR}}(x), \qquad \omega \geq 0,\ \ \textstyle\sum_r \omega_r = 1, \tag{16}$$

with ω on a grid of step 0.1; the second, StackL1, adds the random forest and the Huber regression and obtains ω by linear programming. In both cases ω minimises the out-of-fold mean absolute error [57, 58]

$$\omega^{\star} = \arg\min_{\omega} \frac{1}{N} \sum_{k=1}^{N} \left| y_k - \sum_r \omega_r \hat{y}_r^{(-f(k))}(x_k) \right|, \tag{17}$$

where $\hat{y}_r^{(-f(k))}$ is the prediction of learner r trained without the fold $f(k)$ containing sample k. With the out-of-fold absolute errors $e_j(k)$ of candidate j and $j^{\star}$ the candidate of smallest mean, the selected model is the first candidate, in the order above, that satisfies

$$\overline{e_j - e_{j^{\star}}} \ \leq \ \frac{\mathrm{sd}\,(e_j - e_{j^{\star}})}{\sqrt{N}}. \tag{18}$$

The test set is never used, the five folds are drawn from the training set with the replication seed, and all candidates are refitted on the full training set after the selection. Fig. 1 shows the procedure. The selection is a safeguard, not a new estimator: HYB-SEL inherits the accuracy of its candidates, and it can be better than the best single candidate only through the stacked combinations.

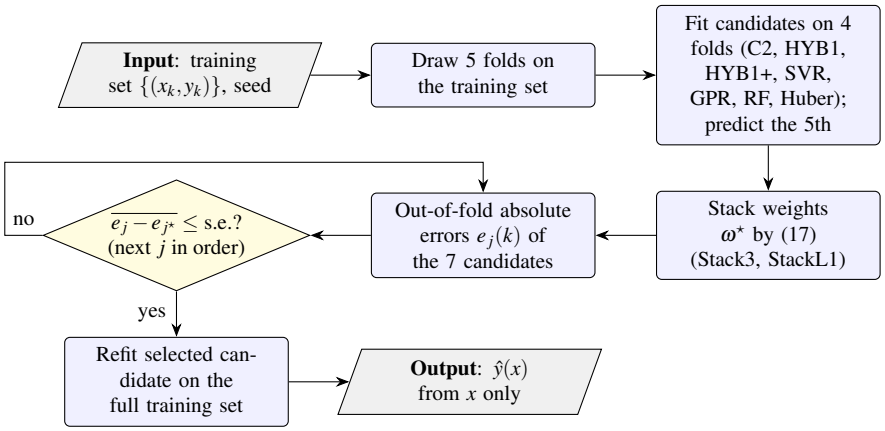

Fig. 1: Selection procedure of HYB-SEL. The diamond is the one-standard-error test (18), applied to the candidates in increasing order of complexity

2.7 Previous-error and sign term for one-step dynamic prediction

For one-step prediction of a dynamic system, the output measured L steps before ($L = 1$, and L equal to the horizon for the Mackey–Glass series) is available when the prediction is made. With the residual $e(j) = y(j) - \hat{y}(j)$ of the selected model, the corrected prediction is

$$\hat{y}_{\mathrm{ES}}(k) = \hat{y}(k) + \gamma e(k-L) + \delta\, \mathrm{sgn}\big(\bar{e}_M(k)\big), \qquad \bar{e}_M(k) = \frac{1}{M} \sum_{j=0}^{M-1} e(k-L-j), \tag{19}$$

with $\mathrm{sgn} \in \{-1, 0, +1\}$. The sign term gives a bounded bias correction from the last M observed errors, which is less sensitive to a single spike than the error itself. The coefficients γ, δ are fitted by Huber regression on the out-of-fold residuals of the selected model; the window $M \in \{3, 5, 10\}$ and the form (error, sign, or both) are chosen on the last 40% of the training sequence, and the correction is kept only if it lowers the validation error. It is activated only when the Ljung–Box test [59] on the out-of-fold residuals, with five lags, rejects whiteness at the 1% level; otherwise the prediction is left unchanged. This term is not available in a deployed predictor that has no access to the recent measured output, and it changes the protocol: results with the term are reported

separately and labelled HYB-SEL+ES.

3 Flowcharts, algorithms and computational complexity of the proposed approaches

3.1 Flowcharts and algorithms: DE fit, learned antecedent, C2 refit, HYB1, HYB1+ and HYB-SEL

The models are described by one global flowchart and by one algorithm per model, in the order in which a model is built: the rule base is fitted by the DE criterion, its antecedent is learned so that predictions use the input only, the consequents are refitted globally, RBF units are added by the hybrids HYB1 and HYB1+, and HYB-SEL chooses among the resulting models. The flowchart of Fig. 2 gives the order of the operations (rectangles are processing steps, parallelograms are inputs and outputs, diamonds are tests); the algorithms give the exact steps and the quantities computed at each of them. The selection procedure of HYB-SEL has its own flowchart in Section 2.6.

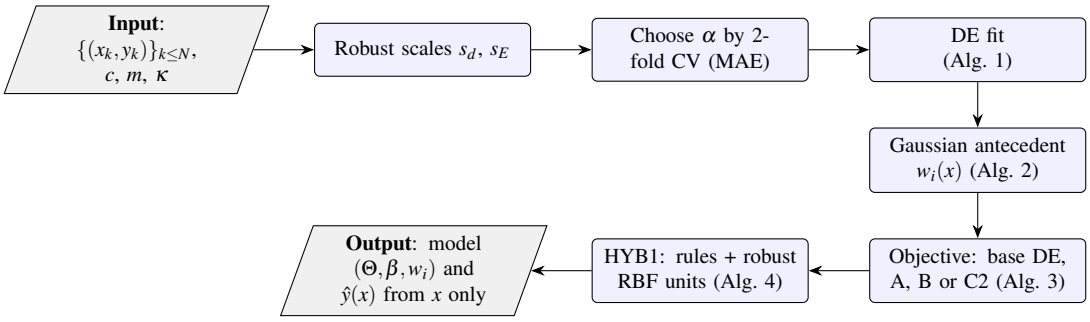

Fig. 2: Global pipeline. Rectangles are processing steps, parallelograms are the input and output; the only decision (diamond) is the one-standard-error test of Fig. 1

DE fit. Algorithm 1 minimises the criterion (7) by alternating the membership update (8), the centre update (9) and the weighted least-squares update (10) of the local parameters. The scales s_d and s_E of (6) are robust (median-based), so that the distance and error terms are comparable; the noise distance δ^2 is recomputed from the data at each iteration. The loop stops when the relative change of J is below ε or after $T_{\max}$ iterations.

Algorithm 1 DE fit (one value of α)

Require: $\{(x_k, y_k)\}$, c, m, κ, α, ε, $T_{\max}$
compute s_d (median distance to the median of x) and s_E (MAD of y)
$V \leftarrow k$-means++ centres [60]; $\theta_i \leftarrow$ local least squares on the nearest points
for $t = 1, \ldots, T_{\max}$ **do**
  $D_{ik}^2 \leftarrow \alpha(\|x_k - v_i\|/s_d)^2 + (1-\alpha)(|y_k - \theta_i^T \bar{x}_k|/s_E)^2$
  $\delta^2 \leftarrow \kappa \,\mathrm{med}_k \min_i D_{ik}^2$; update μ_{ik} in closed form
  $v_i \leftarrow \sum_k \mu_{ik}^m x_k / \sum_k \mu_{ik}^m$; $\theta_i \leftarrow$ weighted LS with weights μ_{ik}^m
  if $|J^{(t)} - J^{(t-1)}| < \varepsilon J^{(t-1)}$ **then break**
  end if
end for
return U, V, Θ

The algorithm is run for each value of α of the grid and the value is chosen by cross-validation on the training data (Section 4); its cost is given in Table 3.

Learned antecedent. The memberships of Algorithm 1 depend on the measured output through E_{ik} and cannot be used for prediction. Algorithm 2 therefore fits, from the final memberships, one Gaussian per rule (weight π_i, mean μ_i, shrunk covariance Σ_i as in (12)) and computes the rule weights of a new input by (11), from x only.

Algorithm 2 Learned antecedent and deployable prediction

Require: U, V from Algorithm 1
1: $\pi_i \leftarrow \sum_k \mu_{ik} / \sum_{k,j} \mu_{jk}$; $\mu_i \leftarrow$ weighted mean; $\Sigma_i \leftarrow$ shrunk weighted covariance
2: **for** each test point x **do**
3: $w_i(x) \leftarrow \pi_i \mathcal{N}(x; \mu_i, \Sigma_i) / \sum_j \pi_j \mathcal{N}(x; \mu_j, \Sigma_j)$; $\hat{y} \leftarrow \sum_i w_i(x) \theta_i^T \bar{x}$
4: **end for**

This step is the one that makes the predictions deployable; it needs no iteration and its cost is that of c covariance estimates.

C2 refit. The local parameters obtained by weighted least squares are optimal for the memberships, not for the final prediction error. Algorithm 3 refits all consequents jointly, with the weights of the learned antecedent, by minimising the Huber loss of the global model with a ridge term (26), solved by iteratively reweighted least squares; the ridge scale is relative to the trace of $\Phi^T \Phi$ as in (20). The memberships and the antecedent are then refreshed with the new residuals, and the procedure is repeated $R = 3$ times.

Algorithm 3 C2: global robust refit

Require: base DE model, $\lambda_{\text{rel}} = 0.3$, $R = 3$
1: **for** $r = 1, \ldots, R$ **do**
2: build $\varphi(x_k)$ from the current antecedent; $\lambda \leftarrow \lambda_{\text{rel}} \operatorname{tr}(\Phi^T \Phi)/q$
3: solve (26) by IRLS (Huber, MAD scale), starting at Θ_0
4: update the memberships with the new residuals and refresh the antecedent
5: **end for**

The cost is dominated by the solution of a system of size cq at each reweighting (Table 3).

HYB1. Algorithm 4 builds the hybrid (13) on a base rule model. Gaussian units are placed on the non-noise points of each rule, so that outliers do not move their centres, and the coefficients of rules and units are fitted jointly by the Huber estimate (28), which is linear in the parameters for fixed units. The width multiplier and the ridge are selected by two-fold cross-validation on the mean absolute error, and the model is refitted on all data with the selected pair.

Algorithm 4 HYB1

Require: base model (A, B or C2), grids $\mathcal{G}_s$ (width multiplier) and $\mathcal{G}_\lambda$ (ridge)
1: units: for each rule, k-means (≤ 3 clusters) on its non-noise points
2: **for** $(s, \lambda) \in \mathcal{G}_s \times \mathcal{G}_\lambda$ **do**
3: fit (28) on each fold by IRLS; record the validation MAE
4: **end for**
5: refit (28) on all data with the best (s, λ)

The grids are small, so the training cost is that of a few joint fits per fold; the limitation of this version is the isotropic width and the small number of folds, which HYB1+ addresses.

HYB1+. Algorithm 5 extends HYB1 with the anisotropic units of (15), a larger set of configurations and a one-standard-error rule. Among the configurations whose cross-validated error is within one standard error of the best, it keeps the one with the strongest ridge on the unit coefficients and then the fewest units, which favours the rule base when the units bring no gain; the value $\lambda_b = 10^4$ reduces the model to its rule base.

Algorithm 5 HYB1+ (anisotropic units, one-standard-error selection)

Require: base model C2; grids $k \in \{3, 8\}$, $s \in \{0.5, 1, 2\}$, $\lambda_p \in \{0.03, 0.3\}$, $\lambda_b \in \{10^{-2}, 1, 10^4\}$
1: **for** $k \in \{3, 8\}$ **do**
2: units: k-means on the non-noise points of each rule; per-input widths σ_{jl} (floor 5%)
3: **end for**
4: **for** each configuration $(k, s, \lambda_p, \lambda_b)$ **do**
5: fit (28) on each of 3 folds by Huber IRLS; record the validation absolute errors
6: **end for**
7: keep the configurations within one standard error [56] of the best; take the largest λ_b, then the smallest k
8: refit (28) with this configuration on all training data

The 36 configurations are evaluated on three folds, which is the main part of the cost of HYB1+ (Table 3).

HYB-SEL. Algorithm 6 implements the selection of Section 2.6 (Fig. 1). Out-of-fold predictions of the base learners give the out-of-fold absolute errors of all candidates, including the two stacked predictors whose

weights solve (17); the first candidate, in increasing order of complexity, whose error is within one standard error of the best (18) is retained and refitted on the full training set. The optional correction (19) is added only for one-step prediction and only if the residuals of the selected model are autocorrelated.

Algorithm 6 HYB-SEL: validated selection among candidates, and optional ES correction

Require: training set, seed, candidates $\mathscr{C}$ ordered by complexity: Huber, C2, HYB1, HYB1+, RF, Stack3, StackL1

1: draw 5 folds; for each fold f and each base learner (C2, HYB1, HYB1+, SVR, GPR, RF, Huber) fit on the other folds and predict fold f
2: compute the stack weights by (17) (grid for Stack3, linear programme for StackL1) and the out-of-fold absolute errors $e_j(k)$ of all candidates
3: $j^\star \leftarrow \arg\min_j \overline{e_j}$; select the first $j \in \mathscr{C}$ such that (18) holds
4: refit the selected candidate (and its base learners) on the full training set
5: **if** ES term requested **and** Ljung–Box rejects whiteness of the out-of-fold residuals **then**
6: fit (γ, δ) by Huber regression; choose M and the form on the last 40% of the sequence; keep the term only if the validation error decreases
7: **end if**
8: **return** the predictor $x \mapsto \hat{y}(x)$ (with (19) if the term is kept)

The test set takes no part in any step. The cost of HYB-SEL is that of fitting each base learner five times and the selected one once more; it is dominated by the Gaussian-process regression, which is cubic in the number of training points (Table 3).

Models A and B follow Algorithm 1 with the likelihood-based updates of Appendix B and with per-rule scales $\sigma_{d,i}, \sigma_{E,i}$ in line 4, respectively.

3.2 Time and memory complexity, and measured training times

Let N be the number of training points, p the input dimension, $q = p+1$, c the number of rules, $J \le 3c$ the number of RBF units, G the size of a hyper-parameter grid, $F = 2$ the number of folds and T the number of iterations.

Table 3: Time and memory complexity (training), and cost of one prediction.

Model	Training time	Memory	Prediction
DE, one α	$O\big(T\,c\,(Nq^2+q^3)\big)$	$O(Nc+cq^2)$	$O(cp^2)$
DE with CV over α	$O\big(G_\alpha F\,T\,c\,(Nq^2+q^3)\big)$	same	$O(cp^2)$
Gaussian antecedent	$O(Ncp^2+cp^3)$	$O(cp^2)$	$O(cp^2)$
B	as DE	as DE	$O(cp^2)$
C2	DE $+\,O\big(RI\,(N(cq)^2+(cq)^3)\big)$	$O(N\,cq+(cq)^2)$	$O(cp^2)$
HYB1	base $+\,O\big(GF\,I\,(N(cq+J)^2+(cq+J)^3)\big)$	$O(N(cq+J))$	$O(cp^2+Jp)$
HYB1+	HYB1 with $G=36$ configurations, $F=3$, and $k \le 8$ units per rule ($J \le 8c$)	$O(N(cq+J))$	$O(cp^2+Jp)$
HYB-SEL	5 folds $\times$ (C2, HYB1, HYB1+, SVR, GPR, RF, Huber) + refit of the selected candidate	dominated by GPR $O(N^3)$	that of the selected candidate
RBF / RRBF	$O\big(GF\,I\,(NJ^2+J^3)\big)$	$O(NJ)$	$O(Jp)$
GPR	$O(N^3)$ per likelihood evaluation	$O(N^2)$	$O(N)$ (mean)
SVR (libsvm)	between $O(N^2)$ and $O(N^3)$	$O(N^2)$	$O(n_{sv}p)$
Random forest, B trees	$O\big(BN\log N \cdot m\big)$, m features per split	$O(BN)$	$O(B\log N)$
Huber linear	$O(INq^2)$	$O(Nq)$	$O(q)$

Here I is the number of IRLS iterations and $R = 3$. The quantities that grow quickly are cq (the number of consequent parameters, 114 for $c = 6$ and the robot with $p = 18$) and N: for the proposed models the cost is linear in N and at most cubic in cq, whereas GPR is cubic in N. Measured times (median of 2–3 runs, single thread, simulated data with 10% outliers, $c = 6$, $p = 4$) agree with these orders: the fits of DE, A, B and C2 grow roughly linearly in N (from 0.2–0.3 s at $N = 250$ to 0.6–0.8 s at $N = 2000$), HYB1 adds 0.05–0.16 s, whereas GPR goes from 0.10 s to 16.4 s and SVR with cross-validation to 9.8 s; the fuzzy models are thus competitive in cost for $N \ge 1\,000$, but not for very small sets.

4 Experimental protocol: applications, compared models, replications and statistics

4.1 Applications: simulated, semi-simulated and measured data

Table 4 lists the 13 applications of the primary analysis: eight of the initial study, three simulated systems added to test the conclusions on new cases (Narendra–Li, Lorenz and a Wiener–Hammerstein system), and two measured configurations (a cascaded-tanks configuration with outliers added to the estimation output, and nano-drone flight data). The *deployable* protocol is used throughout: rule weights at test time are computed from the input x only. Memberships computed from the measured test output would give errors that cannot be obtained in prediction, and are not used. Test targets are noise-free for the simulated and semi-simulated cases and are the measurements for the measured cases.

Several configurations were examined and are not part of the primary analysis. The rule base DE+Gauss, the interval type-2 rule bases and their hybrids, and the gas-furnace and plain cascaded-tanks configurations BJ2 and CT were left out after their results were known; the diabetes data (a static regression problem), Silverbox and the EMPS data in one-step prediction were left out because they do not discriminate between the models (on Silverbox all models are within 6 % of each other, and one-step errors on EMPS are almost exact for every model). These choices were made after seeing results; the plain cascaded-tanks configuration, on which model selection by internal cross-validation fails because the test sequence differs from the training sequence, is reported in free-run simulation in Section 5.

Table 4: Applications. MG: Mackey–Glass [61] ($\tau = 17$, predict $x(t+6)$); BJ10: Box–Jenkins gas furnace [62]; NL: Narendra–Li [63]; LZ: Lorenz [64]; WH: simulated Wiener–Hammerstein; CT-o: cascaded tanks [65]; ND: nano-drone. c: number of rules; Rep.: replications; lags in the regressor: output/input.

Label	Description	Train/test	c	Rep.
MG-c	MG series (Runge–Kutta), no noise	500/500	6	50
MG-n	MG with Gaussian noise ($\sigma = 0.05$)	500/500	6	50
MG-o	MG, noise $\sigma = 0.02$ and 3 % spikes of amplitude 0.5 in training	500/500	6	50
pH-o	pH neutralisation process (2 500 samples), noise $\sigma = 0.05$ and 5 % output outliers in training	625/1 250	6	50
2-D	Nonlinear function of two inputs, 10 % outliers and 3 % leverage points in training	300/1 000	6	50
BJ10	Gas furnace, 4 output lags and 6 input lags, noise $\sigma = 0.1$ added	145/145	4	50
Wind	Wind speed at Tunis-Carthage airport (METAR reports 2012–2016, 13 961 samples, compiled by the authors; forecasting methods reviewed in [66]), one step ahead	600/1 200	6	50
Robot	KUKA KR300 R2500 industrial robot (measured), inverse dynamics, 18 inputs, torque of axis 2 [67]	1 000/3 636	8	30
NL	Narendra–Li NARX system, inputs $y(k-1), y(k-2), u(k-1), u(k-2)$; noise $\sigma = 0.05$ and 3 % outliers; random-amplitude input for training, sum of sines for test	500/500	6	50
LZ	Lorenz x-component, predict $x(t+4)$ from three lags; noise $\sigma = 0.05$ and 3 % outliers (series scaled to unit variance)	500/500	6	50
WH	Simulated Wiener–Hammerstein system (linear filter, tanh saturation, linear filter), noise and 3 % outliers in training	500/500	6	50
CT-o	Cascaded tanks (measured), official estimation/validation split, 2 lags of output and input, with 3 % outliers added to the estimation output only	1 022/1 022	6	15
ND	Nano-quadrotor flight data (measured), vertical velocity, 10 Hz, 2 output lags and the four motor speeds; training on one of twelve flights, test on three different flights	≈600/1 944	6	15

4.2 Compared models

The 16 models are: three reference models from the literature (linear least squares, TSK-FCM and RPCRM [6]); two rule bases of this study (C2 and the gradient-tuned GT-TSK); six literature learners (GPR [11], SVR [12], random forest [13], gradient boosting [14], a multilayer perceptron and Huber regression [39]); three hybrids of this study (HYB1-C2, HYB1-GT-TSK and HYB1+); and two reference ensembles, Stack3 (convex combination of HYB1+, SVR and GPR) and HYB-SEL (cross-validated selection among candidates, Section 2.6). The rule bases and the hybrids are defined in Section 2 and Appendix A. The learners are scikit-learn [68] implementations; GPR (RBF plus white-noise kernel) and SVR (grid on C, γ, ε by cross-validation) are tuned on

the training data, whereas the random forest (200 trees), gradient boosting (absolute-error loss), the multilayer perceptron (2×64 units, early stopping) and Huber regression use fixed settings in the main ranking; a conditional ranking in which the random forest (leaf size, features), gradient boosting (learning rate, leaf nodes) and the multilayer perceptron (architecture, weight decay) are tuned by three-fold cross-validation on the training data, with eight replications per application, is given in Section 5. c, m and κ are not tuned for any model.

4.3 Replications, hyper-parameter selection, statistics and tests

Baseline results for several identification benchmarks are collected in [69]; our protocol follows the recommendations on paired comparisons over several data sets [70]. Each application is replicated 50 times (30 for the robot and 15 for CT-o and the nano-drone, for computing time). The 13 applications come from 11 distinct underlying data sets or systems (MG-c, MG-n and MG-o share one series, and BJ10 and CT-o are fixed records), so the applications are not independent. The replications change the noise and outlier realisations (simulated cases), the training window (wind), the training flight (nano-drone) and the initialisation. For the cascaded tanks the official split is fixed and the replications only change the added outliers (CT-o). All methods share the same data and seeds in each replication, so that comparisons are paired. The fuzzifier is $m = 2$ ($m = 1.6$ for BJ), $\kappa = 9$, $\nu = 4$, and α, the width and the ridge are chosen by cross-validation on the training data only; the test set never takes part in a selection. The error measure is the root-mean-square error (RMSE) on the test set.

Pairs of models are compared by the two-sided Wilcoxon signed-rank test [71] on the paired replications, with Holm correction [72] over the comparisons of each table, which are fixed in advance. Replications are not independent samples of the underlying system (the gas furnace and the official cascaded-tanks split are fixed data), so the p-values describe the sensitivity of the methods to noise, initialisation and training window, not generalisation to other processes. The error of a model that selects among learners (HYB-SEL, the stacks) is favoured by construction, because its candidates are also compared, and we report it as such. The minimum and maximum over the replications are given as descriptive statistics only, because the best of R replications is not obtainable in practice and favours unstable methods.

Software, hardware and data. All computations were run in Python 3.13 with NumPy 2.5, SciPy 1.18 and scikit-learn 1.9, on a two-core Intel Xeon processor at 2.1 GHz with 7 GB of memory. The replications are indexed by their random seed, which fixes the noise and outlier realisations, the training window and the initialisation. The public benchmark files (cascaded tanks, EMPS, nano-drone) are used under the terms of their providers and are not redistributed.

5 Results: errors, tests and behaviour on simulated and measured applications

All errors are test RMSE under the deployable protocol of Section 4, averaged over the replications, for the 16 models and 13 applications of the primary analysis. The ES correction is not included except in Section 5.6. Two points organise the section: the effect of the RBF hybridisation on its rule base (Section 5.2), and the position of the hybrid with respect to generic learners, tuned and untuned (Sections 5.3 and 5.4).

5.1 Mean errors and ranks on the 13 applications

Table 5 gives the mean RMSE of the 16 models. No model is best everywhere: a different model wins on almost every application. Table 6 gives the rank of each model on each application, computed on the mean RMSE (1 is the lowest error, ties share the mean rank), the type of each model, and the mean rank over all applications, over the eight simulated or semi-simulated applications, over the five measured ones and over the applications without BJ10.

Table 5: Mean test RMSE of the 16 models (best in bold). (a) Eight applications of the initial study. (b) Added applications, with the mean rank of each model over the 13 applications (lower is better). Stack3: convex combination of HYB1+, SVR and GPR. *: model proposed in this study.

Model	MG-c	MG-n	MG-o	pH-o	2-D	BJ10	Wind	Robot
Linear LS	0.0960	0.1057	0.1343	0.1389	0.8304	0.5119	1.863	2.087
TSK-FCM	0.0193	0.0529	0.1028	0.1008	0.4248	0.5270	**1.836**	2.071
RPCRM	0.0630	0.0851	0.1263	0.1409	0.7594	0.5248	1.912	2.080
C2*	0.0145	0.0534	0.0808	0.1000	0.2575	0.5289	1.920	2.197
GT-TSK	0.0141	0.0545	0.0735	0.0809	0.1724	0.5255	1.926	2.173
HYB1-C2*	0.0131	0.0526	0.0760	0.0927	0.1963	0.5276	1.928	2.259
HYB1-GT-TSK*	0.0127	0.0526	0.0724	0.0781	0.1516	0.5228	1.956	2.233
HYB1+*	0.00976	0.0524	0.0741	0.0858	0.1510	0.5273	1.941	2.299
Stack3*	**0.0015**	0.0510	0.0644	0.0747	0.0592	0.5298	1.904	2.120
HYB-SEL*	0.0015	0.0503	0.0579	0.0677	0.0582	0.5237	1.898	1.182
GPR	**0.0015**	**0.0503**	0.0688	0.1008	0.2867	0.5538	1.935	1.936
SVR	0.00691	0.0508	0.0625	**0.0660**	**0.0577**	0.6079	1.950	2.024
Random forest	0.0177	0.0579	0.0694	0.1587	0.3815	1.073	1.915	**1.177**
Grad. boosting	0.0128	0.0581	**0.0572**	0.0946	0.1664	0.9634	1.975	1.189
MLP	0.0103	0.0528	0.0828	0.1101	0.4315	0.9553	1.945	2.094
Huber linear	0.0993	0.1071	0.1375	0.1418	0.8309	**0.5099**	1.869	2.115

Model	NL	LZ	WH	CT-o	ND	Mean rank
Linear LS	1.375	0.9701	0.3732	0.0986	0.1095	10.69
TSK-FCM	1.201	0.7801	0.3715	0.0947	0.2147	10.00
RPCRM	1.137	1.015	0.3952	0.0767	0.1182	10.62
C2*	0.9559	0.8828	0.3812	0.0909	0.1137	9.92
GT-TSK	0.7491	0.5783	0.3608	0.0933	0.1812	8.54
HYB1-C2*	0.8410	0.7922	0.3746	0.0916	0.1233	9.46
HYB1-GT-TSK*	0.7222	0.5707	0.3590	0.0932	0.1823	7.77
HYB1+*	0.8124	0.7214	0.3738	0.0859	0.1221	8.15
Stack3*	0.6738	0.5404	0.3444	0.0903	0.1641	5.42
HYB-SEL*	0.6639	**0.4800**	0.3399	0.0792	0.1490	**2.92**
GPR	0.6922	0.5403	0.3477	0.1840	0.2683	7.81
SVR	0.6906	0.6936	0.3384	0.1268	0.2468	6.62
Random forest	0.7337	0.4935	0.3222	0.1902	0.2538	9.46
Grad. boosting	**0.6571**	0.4868	**0.2937**	0.1453	0.3533	8.00
MLP	0.7515	0.5255	0.3427	0.1241	0.1836	9.85
Huber linear	1.196	1.002	0.3987	**0.0753**	**0.0915**	10.77

Table 6: Rank of each of the 16 models on each of the 13 applications (rank 1: lowest mean test RMSE, best in bold; models with identical errors share the lower rank, here Stack3 and GPR on MG-c, while the mean ranks use the mean of the tied ranks, as in the Friedman test) and mean rank over all applications, over the eight simulated or semi-simulated applications (sim.), over the five measured applications (meas.) and without BJ10. Type: ours, model of this study; literature, learner from the literature; recent, recent fuzzy method re-implemented by us; baseline, fuzzy or linear reference. Models are sorted by mean rank.

			Rank per application													Mean rank			
Rank	Model	Type	MG-c	MG-n	MG-o	pH-o	2-D	BJ10	Wind	Robot	NL	LZ	WH	CT-o	ND	all 13	sim. (8)	meas. (5)	no BJ10
1	HYB-SEL	ours	3	2	2	2	2	4	4	2	2	**1**	4	3	7	**2.92**	**2.25**	**4.00**	**2.83**
2	Stack3	ours	**1**	4	4	3	3	11	5	11	3	6	6	5	8	5.42	3.81	8.00	4.96
3	SVR	literature	4	3	3	**1**	**1**	13	14	5	4	9	3	13	13	6.62	3.50	11.60	6.08
4	HYB1-GT-TSK	ours	7	7	7	4	5	3	15	14	6	7	8	8	10	7.77	6.38	10.00	8.17
5	GPR	literature	**1**	**1**	5	10	10	12	11	4	5	5	7	15	15	7.81	5.56	11.40	7.46
6	Gradient boosting	literature	8	13	**1**	8	6	15	16	3	**1**	2	**1**	14	16	8.00	5.00	12.80	7.42
7	HYB1+	ours	5	5	9	6	4	8	12	16	10	10	12	4	5	8.15	7.62	9.00	8.17
8	GT-TSK	recent (re-implemented)	10	11	8	5	7	6	9	12	8	8	9	9	9	8.54	8.25	9.00	8.75
9	HYB1-C2	ours	9	6	10	7	8	9	10	15	11	12	13	7	6	9.46	9.50	9.40	9.50
10	Random forest	literature	12	12	6	16	11	16	7	**1**	7	3	2	16	14	9.46	8.62	10.80	8.92
11	MLP	literature	6	8	12	12	13	14	13	9	9	4	5	12	11	9.85	8.62	11.80	9.50
12	C2	ours	11	10	11	9	9	10	8	13	12	13	14	6	3	9.92	11.12	8.00	9.92
13	TSK-FCM	fuzzy / linear baseline	13	9	13	11	12	7	**1**	6	15	11	10	10	12	10.00	11.75	7.20	10.25
14	RPCRM	fuzzy / linear baseline	14	14	14	14	14	5	6	7	13	16	15	2	4	10.62	14.25	4.80	11.08
15	Linear LS	linear baseline	15	15	15	13	15	2	2	8	16	14	11	11	2	10.69	14.25	5.00	11.42
16	Huber linear	literature	16	16	16	15	16	**1**	3	10	14	15	16	**1**	**1**	10.77	15.50	3.20	11.58

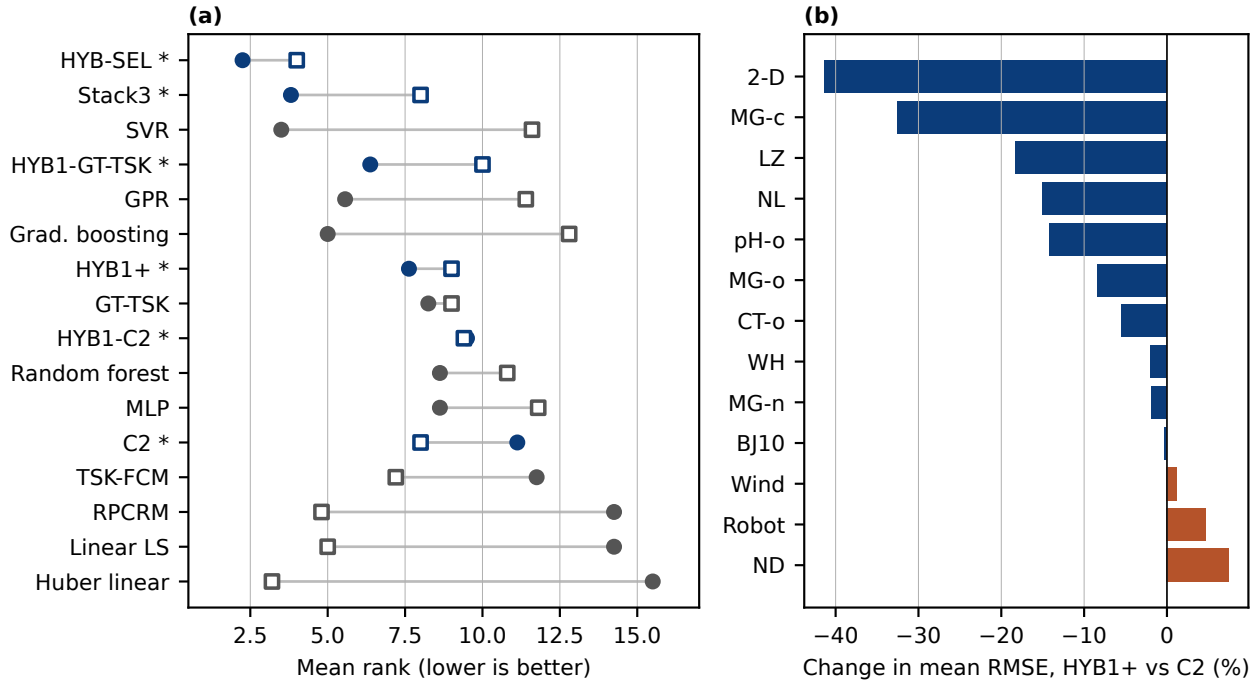

Fig. 3: **(a)** Mean rank of the 16 models on the eight simulated or semi-simulated applications (dots) and the five measured ones (squares); models of this study are marked * and drawn in blue. **(b)** Change in the mean RMSE of HYB1+ with respect to its rule base C2 on each application (blue bars: lower error; hatched orange bars: higher error)

Mean rank. The ensemble HYB-SEL has the lowest mean rank (2.92), followed by Stack3 (5.42), SVR (6.62), HYB1-GT-TSK (7.77), GPR (7.81), gradient boosting (8.00) and HYB1+ (8.15); the hybrid HYB1+ is ahead of its rule base C2 (9.92) and of RPCRM (10.62). Two caveats apply. HYB-SEL and Stack3 combine SVR and GPR, which are also ranked, so their rank cannot be much worse than that of their best candidate and the comparison is favourable to them by construction. The rank also depends on the type of application: on the eight simulated applications HYB-SEL has mean rank 2.25, SVR 3.50, gradient boosting 5.00 and HYB1+ 7.62, whereas on the five measured ones HYB-SEL has 4.00, Huber regression 3.20, RPCRM 4.80 and linear least squares 5.00, and SVR (11.6), GPR (11.4) and gradient boosting (12.8) are among the worst. Simulated and measured applications therefore favour different families (Fig. 3a).

Global test. The Friedman test rejects equal ranks ($\chi^2_{15} = 38.9$, $p = 6.6 \times 10^{-4}$). With HYB-SEL as control, the rank-difference test with Holm correction [70] finds HYB-SEL better than 13 of the 15 other models (adjusted p from 4×10^{-4} to 0.036) and cannot separate it from SVR ($p = 0.096$) or Stack3 ($p = 0.18$). These p-values are optimistic, because the applications are not independent (Section 4); the effect sizes of the next subsections are more informative.

5.2 Effect of the RBF hybridisation on the rule base

The extended hybrid HYB1+ reduces the mean error of its rule base C2 by at least 1 % on nine of the 13 applications. The largest reductions are on the 2-D function (−41 %), MG-c (−33 %), LZ (−18 %), NL (−15 %) and pH-o (−14 %). It is within ±1 % of C2 on BJ10, and worse on the wind data (+1 %), the robot (+5 %) and the nano-drone (+7 %). The median change over the applications is −5.5 % (Fig. 3b; 95 % bootstrap interval over applications −15.0 to −0.3 %). With a paired Wilcoxon test and Holm correction over the 13 applications, the error of HYB1+ is significantly lower than that of C2 on nine applications and significantly higher on three (wind, robot and nano-drone, by 1 to 7 %). The hybrid is therefore useful for nonlinear problems of low dimension, and the validated ridge prior prevents large losses on the others.

Table 7 relates the gain to the contribution of the RBF units. The root-mean-square of the RBF part of the output, divided by the standard deviation of the output, is 0.61 on NL, 0.51 on LZ, 0.32 on the 2-D function, 0.30 on the robot and 0.15 on MG-o, and at most 0.04 elsewhere; the relative change of the local parameters caused by the joint fit is 0.45 on NL, 0.26 on LZ, 0.13 on the 2-D function and WH, and at most 0.06 on the other applications. The relation with the gain is not one-to-one. On MG-c and pH-o the RBF part is small

relative to the output (0.04) but the error of C2 is small too, so a small correction gives a large relative gain (-33 and $-14\,\%$); on the robot the share is 0.30 and the error increases ($+5\,\%$); and on BJ10, WH and the nanodrone the RBF part is essentially zero. The relative error reduction and the RBF share are therefore different measures, and the table should be read as a description of what the units do, not as an explanation of the gain. Where the RBF share is large (NL, LZ) the local linear parameters are modified by 26 to 45 % and the rules are a less faithful local description than in the rule base alone.

Table 7: Contribution of the RBF units in HYB1+ (three replications per application): RMS of the RBF part divided by the output standard deviation, relative change of the local parameters caused by the joint fit ($\Delta\theta$), and change in the mean RMSE with respect to C2 (%).

Application	RBF share	$\Delta\theta$	Selected α	Change vs C2 (%)
MG-c	0.04	0.01	0.1	-32.5
MG-n	0.00	0.01	0.5	-1.8
MG-o	0.15	0.06	0.1	-8.4
pH-o	0.04	0.01	0.7	-14.2
2-D	0.32	0.13	0.6	-41.4
BJ10	0.00	0.00	0.0	-0.3
Wind	0.04	0.04	0.9	+1.1
Robot	0.30	0.06	0.0	+4.6
NL	0.61	0.45	0.2	-15.0
LZ	0.51	0.26	0.8	-18.3
WH	0.00	0.13	0.4	-2.0
CT-o	0.01	0.01	0.8	-5.5
ND	0.00	0.02	0.2	+7.4

5.3 HYB1+ against generic learners

Against the learners of the literature, HYB1+ is competitive but not superior. The median change of its mean RMSE is $-9.4\,\%$ with respect to the random forest (bootstrap interval -50.8 to $+10.7\,\%$), $-9.2\,\%$ with respect to gradient boosting (-24.0 to $+27.2\,\%$) and $-5.4\,\%$ with respect to the multilayer perceptron (-30.7 to $+8.1\,\%$), with the intervals including zero; it is $+4.3\,\%$ with respect to GPR (-14.8 to $+17.4\,\%$) and $+10.4\,\%$ with respect to SVR (-0.4 to $+18.6\,\%$), which are lower by more than 1 % on 7 and 9 of the 13 applications. The hybrid is lower than the linear and fuzzy references by $-25.6\,\%$ (linear least squares), $-28.0\,\%$ (Huber regression) and $-28.6\,\%$ (RPCRM) in median, but these medians come from the nonlinear simulated applications; on the five measured ones the linear models are as good or better (Table 6). The rule base C2 alone is $-13.0\,\%$ below RPCRM in median (interval -36.0 to $+0.4\,\%$) and $-11.9\,\%$ below Huber regression (-41.2 to $+3.7\,\%$).

Tuned competitors. The comparison above uses fixed settings for the random forest, gradient boosting and MLP. Tuning them by three-fold cross-validation on the training data (eight replications per application, on all 13 applications) changes their mean RMSE by a median of $-1.5\,\%$ (random forest), $-0.7\,\%$ (gradient boosting) and $+0.7\,\%$ (MLP); the changes range from $-29\,\%$ to $+12\,\%$ for the random forest on single applications and are within $\pm 7\,\%$ for the other two. In the conditional ranking, in which the tuned values replace the fixed ones (Table 8), the mean rank of HYB1+ is unchanged (8.38), that of the random forest improves from 9.38 to 8.77, and the order of the other models is unchanged apart from exchanges of less than one rank. Tuning these three learners with the same budget therefore does not change the conclusion; c, m and κ of the fuzzy models, and the settings of the other models, were not tuned (Section 6).

Table 8: Mean rank over the 13 applications with fixed and cross-validation-tuned random forest, gradient boosting and MLP (first eight replications of each application).

Model	Fixed settings	Tuned RF/GB/MLP
HYB-SEL	3.15	3.23
Stack3	5.23	5.31
SVR	6.88	6.96
HYB1-GT-TSK	7.85	7.85
Gradient boosting	7.92	7.92
GPR	8.15	8.08
HYB1+	8.38	8.38
GT-TSK	8.46	8.46
Random forest	9.38	8.77
TSK-FCM	9.31	9.31
HYB1-C2	9.15	9.31
MLP	9.46	9.54
C2	9.92	10.00
Huber linear	10.65	10.65
Linear LS	11.00	11.08
RPCRM	11.08	11.15

5.4 Free-run simulation on the cascaded tanks and the EMPS data

Table 9 gives the RMSE of recursive simulation on the test data divided by the standard deviation of the test output (the predicted outputs are fed back and only the inputs are measured). The ranking differs from one-step prediction. On the official clean cascaded tanks the linear model is the most accurate (0.34), followed by HYB1+ (0.47), Huber regression (0.51) and C2 (0.52); HYB-SEL (0.66) is behind them, because it mostly selects stacks of kernel learners, which degrade in recursion (SVR 0.96). With outliers in the estimation output (CT-o) the linear model is again the best (0.37), followed by RPCRM (0.50) and Huber regression (0.53), and C2 (0.58), HYB-SEL (0.58) and HYB1+ (0.62) are behind the robust linear and fuzzy references. On EMPS, HYB1+ (0.076) and GPR (0.081) are the most accurate, ahead of HYB-SEL (0.146), C2 (0.24) and linear least squares (0.42). The hybrid is thus the best simulator on EMPS and second on the clean cascaded tanks, but not on CT-o, and no model dominates. These values use our own preprocessing (decimation, two lags, 1 000 training points), with five replications for the official cascaded tanks and three for the others, and are not comparable with published free-run results for these benchmarks.

Table 9: Free-run RMSE divided by the standard deviation of the test output (mean over the replications; lowest value in bold). HYB1-GT: HYB1-GT-TSK.

Benchmark	Rep.	LS	Huber	TSK-FCM	RPCRM	SVR	GPR	C2	HYB1+	HYB1-GT	HYB-SEL
CT (official, clean)	5	**0.337**	0.514	0.592	0.617	0.962	0.662	0.523	0.471	1.422	0.662
CT-o (outliers in estimation)	3	**0.369**	0.528	0.657	0.496	0.903	1.172	0.581	0.621	0.647	0.584
EMPS	3	0.423	2.649	0.435	0.507	1.619	0.081	0.236	**0.076**	0.218	0.146

5.5 Reference ensemble HYB-SEL: what it selects and when it fails

HYB-SEL is a reference, not a proposed estimator. The median change of its mean RMSE is $-7.3\,\%$ with respect to SVR (interval -37.5 to $-1.1\,\%$), $-11.2\,\%$ with respect to GPR (-38.9 to $-2.3\,\%$), $-18.3\,\%$ with respect to HYB1+ (-33.5 to $-4.1\,\%$) and only $-1.6\,\%$ with respect to Stack3 (-10.1 to $-1.3\,\%$). Over the 560 replications of the 13 applications, HYB-SEL retains a stack in 88 % (Stack3 51 %, the L1-weighted stack 37 %), Huber regression in 9 %, the rule base alone in 2 %, HYB1 in 1 % and HYB1+ in one replication: a single fuzzy model is retained in 3 % of the replications.

Ablation. To isolate the contribution of the fuzzy components, the same procedure (five-fold out-of-fold errors, one-standard-error rule, weighted and L1-weighted stacks) was run with generic candidates only (Huber regression, random forest, SVR, GPR and their stacks) on all 13 applications, with eight replications each (Table 10). The generic selector is within 2 % of HYB-SEL on 10 of the 13 applications (median difference 0.0 %). HYB-SEL is lower by 3.7 % on MG-o and by 20.1 % on the nano-drone, and higher by 5.1 % on pH-o. The fuzzy components therefore bring no measurable gain to the ensemble on most applications, and the gain on the nano-drone, where the hybrid enters the selected stacks, rests on eight replications of a single flight configuration.

Table 10: Ablation: mean RMSE of the selector restricted to generic candidates and of HYB-SEL (same eight replications), and change of HYB-SEL (%).

Application	Replications	Generic selector	HYB-SEL	Change of HYB-SEL (%)
MG-c	8	0.0015	0.0015	+0.0
MG-n	8	0.0502	0.0500	-0.3
MG-o	8	0.0651	0.0627	-3.7
pH-o	8	0.0649	0.0683	+5.1
2-D	8	0.0575	0.0575	+0.0
BJ10	8	0.5225	0.5225	+0.0
Wind	8	1.953	1.934	-1.0
Robot	8	1.125	1.125	-0.0
NL	8	0.6547	0.6668	+1.9
LZ	8	0.5033	0.5022	-0.2
WH	8	0.3384	0.3423	+1.2
CT-o	8	0.0773	0.0788	+1.8
ND	8	0.1917	0.1531	-20.1

Failure under distribution shift, and blocked validation. The selection fails on the nano-drone, where the error of HYB-SEL is 63 % above that of Huber regression: the test uses flights different from the training flight, and the stack gives a large weight to GPR or SVR, which are accurate under cross-validation inside the training data and, as a plausible explanation that we did not test separately, extrapolate poorly. Replacing the random folds by contiguous blocks (ten replications) lowers the error of HYB-SEL from 0.146 to 0.115 on the nano-drone (-22 %) and from 0.081 to 0.074 on CT-o (-9 %), but it remains 28 % above Huber regression (0.090) on the nano-drone, and still selects the L1-weighted stack in 8 of 10 replications. Blocked validation reduces the failure; it does not remove it. When 3 % outliers are added to the estimation output (CT-o), the stack is no longer selected in most replications (Huber in 10 of 15) and HYB-SEL is 5 % above the best model.

5.6 Previous-error and sign correction (ES)

The ES correction is a Huber-fitted linear model of the last residuals, with or without a sign feature, chosen by validation and applied only when a Ljung–Box test detects autocorrelated residuals. Compared with HYB-SEL without correction (six replications), it changes the error by -17.6 % on WH, -7.1 % on CT-o, -9.8 % on the nano-drone and -0.6 % on LZ. A fixed autoregressive residual model without the sign feature gives -21.2 % on WH, -0.3 % on CT-o, -9.3 % on the nano-drone and 0.0 % on LZ. The sign feature thus helps clearly only on CT-o, where the autoregressive model alone does not help, and it is worse than the autoregressive model on WH. The correction requires the measured output of the previous steps at prediction time, so it applies to one-step prediction and not to free simulation. Dispersion over the replications (mean, standard deviation, median, minimum and maximum of the RMSE of HYB-SEL for each application) is given in Online Resource 1.

5.7 Sensitivity to c, m and κ, and to the antecedent of RPCRM

With the number of rules changed by ± 2, the fuzzifier set to 1.5 or 3 and the noise-cluster constant κ set to 4 or 16 (one parameter at a time), the mean RMSE of C2 and HYB1+ changes by between -8 % and $+9$ % with respect to the default setting (NL, WH, CT-o and EMPS, five replications). On the Narendra–Li series the error decreases by up to 8 % with other values, so the default values are not optimal. Giving RPCRM the learned Gaussian antecedent changes its error by -10 % (NL), $+1$ % (WH), -3 % (EMPS) and $+6$ % (CT-o), and it remains 13 % and 3 % above C2 on the first two; the antecedent explains part of the difference with RPCRM on the Narendra–Li series only.

6 Discussion and limitations

6.1 Overall synthesis: which models gain and which lose

Table 11 summarises the 13 applications for each model: its mean rank, the number of applications on which it is best, in the top three (rank 3 or better) and in the bottom three (rank 14 or worse), the last two split between the eight simulated and the five measured applications.

Table 11: Overall evaluation of the 16 models over the 13 applications: mean rank, number of applications with rank 1, in the top three and in the bottom three (rank ≥ 14), and the split of the top-three and bottom-three counts between simulated (sim.) and measured (meas.) applications. Models with identical errors share the lower rank. Models are sorted by mean rank.

Model	Mean rank	Best (rank 1)	Top 3	Bottom 3 (rank ≥ 14)	Top 3 sim. (of 8)	Top 3 meas. (of 5)	Bottom 3 sim./meas.
HYB-SEL	2.92	1	9	0	7	2	0/0
Stack3	5.42	1	4	0	4	0	0/0
SVR	6.62	2	5	1	5	0	0/1
HYB1-GT-TSK	7.77	0	1	2	0	1	0/2
GPR	7.81	2	2	2	2	0	0/2
Gradient boosting	8.00	3	5	4	4	1	0/4
HYB1+	8.15	0	0	1	0	0	0/1
GT-TSK	8.54	0	0	0	0	0	0/0
HYB1-C2	9.46	0	0	1	0	0	0/1
Random forest	9.46	1	3	4	2	1	1/3
MLP	9.85	0	0	1	0	0	0/1
C2	9.92	0	1	1	0	1	1/0
TSK-FCM	10.00	1	1	1	0	1	1/0
RPCRM	10.62	0	1	7	0	1	7/0
Linear LS	10.69	0	3	6	0	3	6/0
Huber linear	10.77	3	4	8	0	4	8/0

Models that gain most. The ensemble HYB-SEL is in the top three on nine of the 13 applications and never in the bottom three, which is its main strength, but it is favoured by construction and its advantage is concentrated on the simulated applications (seven of nine). Among single models, no model dominates. SVR, GPR and gradient boosting win on the simulated series (SVR is in the top three on five of the eight, gradient boosting on four) and fall to the bottom three on several measured ones (gradient boosting on four of five). Huber regression and linear least squares show the opposite pattern: they are in the top three on four and three of the five measured applications, including the best model on BJ10, CT-o and the nano-drone for Huber regression, and in the bottom three on eight and six of the eight simulated applications respectively.

Models that lose most. RPCRM is in the bottom three on seven of the eight simulated applications and in the top three on one measured one, which confirms that a robust rule base built on the measured output alone is not competitive on nonlinear simulated series. The random forest is in the bottom three on four applications, mostly measured, and the multilayer perceptron never reaches the top three.

Position of the proposed models. HYB1+ is never the best model and never in the top three, and it is in the bottom three once (the robot). Its ranks lie between 4 and 16, mostly between 5 and 12, with a mean rank of 8.15. Compared with its own rule base C2 (mean rank 9.92), it has a lower mean error on 10 of the 13 applications, and it has a lower mean error than the linear and RPCRM references on 8 of 13, than the multilayer perceptron on 9, gradient boosting on 8 and the random forest on 7, but than GPR on only 5 and SVR on 4. The hybridisation improves the rule base consistently, but the improved rule base remains a middle-ranked predictor in this study. The rule base C2 is in the top three on one application (the nano-drone) and the gradient-tuned hybrid HYB1-GT-TSK is the best single fuzzy model in mean rank (7.77), without being in the top three more than once.

Balance. In short, the generic kernel and boosting learners and the ensemble win on simulated, contaminated nonlinear series; robust linear models win on measured data and in free-run simulation of the cascaded tanks; the proposed hybrid improves its rule base and is competitive with tree-based and neural learners, but is not the most accurate model on any of the 13 applications. The proposed components therefore justify themselves by what they add to a robust rule base and by the structure they keep, not by a ranking in which they are first.

6.2 What the evidence supports

Hypothesis H1: hybridisation of the rule base. The joint Huber fit of a robust rule base with robust RBF units lowers the error of the rule base on contaminated nonlinear problems of low dimension (up to 41 % for HYB1+ on the 2-D function and 33 % on MG-c), and, with the validated ridge prior, loses at most 7 % elsewhere. Its median effect over the 13 applications is $-5.5\,\%$, with a bootstrap interval (-15.0 to $-0.3\,\%$) that is close to zero, significant gains on nine applications and significant losses on three (wind, robot and nano-drone). H1 is therefore supported on the nonlinear simulated series, where the gains are largest, and on CT-o, and not

supported on the wind data, the robot and the nano-drone, which is the expected pattern for nearly linear, high-dimensional or shifted data. The measurements of Table 7 show that the relation between the share of the RBF part and the gain is not one-to-one, so the gain cannot be read as the size of the correction alone.

Interpretability of the hybrid. The hybrid keeps the local linear rules and the quasi-LPV form of the rule base, and the validated ridge prior keeps the local parameters close to those of the rules. Where the RBF part is small (BJ10, the nano-drone, and most applications with a share of at most 0.04) the model is essentially its rule base. Where it is large (NL and LZ) the local parameters change by 26 to 45 %, so the rules are a less faithful local description, and the interpretability argument holds for the hybrid only to the extent that the RBF share is small. The quasi-LPV form is a representation; no stability or identifiability analysis is carried out, and none is claimed.

Hypothesis H2: comparison with generic learners. HYB1+ is competitive with random forest, gradient boosting and the multilayer perceptron, with and without cross-validated tuning of these learners, and its mean rank (8.15) lies between those of gradient boosting (8.00) and the random forest (9.46). It is not superior to SVR and GPR, which are better on most simulated applications. On the measured applications, linear and robust-linear models rank as high as the hybrid, and in free-run simulation of the cascaded tanks a linear model is the most accurate; in free run the hybrid is the best model on EMPS and second on the clean cascaded tanks. H2 is therefore supported for tree-based and neural learners and not for kernel learners on simulated data, and the case for the hybrid over generic learners is its fuzzy structure (local linear models, quasi-LPV form) and its competitive accuracy on contaminated low-dimensional nonlinear series, not a general gain in accuracy.

Role of the antecedent. A learned Gaussian antecedent, which computes the rule weights from the input only, makes the rule bases deployable, since memberships that depend on the measured output cannot be used in prediction. In the check on four applications, giving RPCRM this antecedent changed its error by -10 to $+6$ % (Section 5), so its contribution to the difference between RPCRM and the proposed rule bases is established on part of the data only.

Selection ensemble. The ensemble HYB-SEL has the best mean rank, but it retains a single fuzzy model in 3 % of replications, and a selector restricted to generic learners is within 2 % of it on 10 of 13 applications. Its advantage over the generic selector rests on the nano-drone and, to a smaller extent, MG-o, and it is obtained with eight replications. The ensemble serves here as a reference for the accuracy that can be reached from the training data, not as a recommended estimator, and it has no local-linear interpretation. Simulated and measured applications favour different families (Fig. 3a): kernel and boosting learners on simulated series, robust linear models on measured ones, so that a practitioner gains more from validating a small set of model families on the data at hand than from a single family.

Relation to the literature. The rule bases of the literature are built from criteria that use the measured output, and their reported advantages are mostly obtained against other fuzzy models. Under a deployable protocol and against generic learners, the picture is more modest: the robust criteria and the noise cluster help the rule base on contaminated series, the RBF correction adds a further gain on some of them, and the resulting model is competitive with, but not better than, well-established learners. The study does not show that the proposed components are necessary to reach the accuracy of the ensemble, and it does not show that they are useless: it locates the conditions (contaminated, low-dimensional, nonlinear data) in which they help.

6.3 Limitations and failures

The first limitation concerns the data and the design of the comparison. Only five of the 13 applications are measured, and on them simple models (Huber regression, linear least squares, RPCRM) are as good as the hybrid or better, so the favourable ranks of the ensemble come mainly from the simulated and semi-simulated cases. The 13 applications come from 11 underlying data sets, three of them being variants of one series, and the measured sets are fixed (single split, or a single training flight), so the replications measure sensitivity to noise, initialisation and training window and the p-values are descriptive. Several configurations were examined and left out after their results were known (Section 4), and C2, the grids of HYB1+, the candidate set of HYB-SEL and the gate of the ES correction were designed after pilot runs; the added applications (NL, LZ, WH, CT-o, ND) were run once with the final settings and are the more independent evidence, but the study should be repeated on new data. Only the random forest, gradient boosting and the multilayer perceptron were tuned, with eight replications, the same number as in the ablation; c, m and κ and the settings of the other models were not, and a sensitivity check on four applications gave changes of -8 to $+9$ %. GT-TSK and the modified-FCRM-like

criterion are our own implementations, not the original codes.

The second limitation concerns the evaluation protocol and the behaviour under shift. The benchmarks are decimated, use two lags and are mainly evaluated in one-step prediction, not in free-run simulation on the official test data, so the results cannot be compared with published baselines [69]; the free-run simulation covers the cascaded tanks and EMPS only. The ensemble HYB-SEL is a one-step predictor whose selected stacks have no local-linear form, so the motivation by simulation and control applies to the rule bases and the hybrids. The previous-error correction needs the measured output of the previous steps and cannot be used in free simulation; its sign feature helps clearly on CT-o only, and a fixed autoregressive residual model is better on WH. On the nano-drone, HYB-SEL is 63 % above the best model because cross-validation inside the training data selects a stack that does not transfer to the operating data (different flights); blocked validation reduces the error by 22 % but does not remove the failure, and a check of covariate shift between training and operating data is needed before such a procedure is used where the operating range is not covered by the training data.

7 Conclusion and perspectives

Two claims are supported by the evidence. First, the joint Huber fit of a robust rule base with robust RBF units (HYB1+) lowers the error of the rule base on contaminated, low-dimensional nonlinear series, but its median effect over the 13 applications is small ($-5.5\,\%$, bootstrap interval from -15.0 to $-0.3\,\%$), it is significantly worse on three applications, and the largest reductions (up to 41 %) are on simulated cases. Second, HYB1+ is competitive with random forest, gradient boosting and a multilayer perceptron, also when these are tuned by cross-validation, but it is not more accurate than SVR and GPR on most simulated cases, and linear models are as good on the measured ones and in free-run simulation of the cascaded tanks. A selection and stacking ensemble has the best mean rank, but it is favoured by construction, it retains a single fuzzy model in 3 % of replications, and a selector restricted to generic learners is within 2 % of it on 10 of 13 applications; it also fails under distribution shift on the nano-drone, which blocked validation reduces without removing. The evidence for a gain from the fuzzy components in the ensemble therefore rests on one measured configuration and on eight replications. Perspectives are a detection of this shift and a selection that accounts for it, a free-simulation version of the previous-error correction, multi-step prediction, tuning of c, m and κ under the same budget as the competitors, and validation on further measured benchmarks with genuine outliers.

A Global Huber refit and gradient-tuned rule base

A.1 Global Huber refit (C2)

Global robust re-fit. After a DE fit, all consequents are re-estimated together by minimising the Huber loss of the final output $\sum_i w_i(x)\theta_i^T \bar{x}$, with a ridge penalty $\lambda \|\theta - \theta_{\text{local}}\|^2$, where

$$\lambda = \lambda_{\text{rel}} \frac{\text{tr}(\Phi^T \Phi)}{q}; \tag{20}$$

memberships are then updated and the procedure is repeated three times. C uses $\lambda_{\text{rel}} = 10^{-2}$. C2 uses $\lambda_{\text{rel}} = 0.3$; it was defined *after* C was observed to over-fit on the wind data, so C2 is a post-hoc variant (Section 6).

A.2 Gradient-tuned extension of the rule base

Gradient-tuned robust TSK (GT-TSK) and HYB1-GT. Following the idea of training TSK systems by mini-batch gradient descent with uniform regularisation [38], the centres, diagonal widths, rule biases and consequents of the DE+Gauss model are refined by Adam on the Huber loss, with a penalty $\lambda_u \sum_i (\bar{w}_i - 1/c)^2$ on the mean normalised firing strengths and a ridge prior towards the initial consequents; early stopping uses 20% of the training set. HYB1-GT applies the joint fit (28) on the refined rule base. This is our own implementation of the idea, not the original code.

B Mathematical analysis: criteria, standard propositions and quasi-LPV representation

B.1 Problem setting: contaminated data and deployable prediction

Let $\mathscr{D} = \{(x_k, y_k)\}_{k=1}^N$ with

$$y_k = f(x_k) + e_k, \qquad e_k \sim (1-\varepsilon)\,\mathscr{N}(0,\sigma^2) + \varepsilon\,\mathscr{H}, \tag{21}$$

where $\mathscr{H}$ is a wide or heavy-tailed law. We look for a model of the form (1) with $w_i \geq 0$ and $\sum_i w_i = 1$, minimising the out-of-sample error on the *clean* target f using only x at prediction time (deployable prediction). The training problem has two groups of unknowns: the antecedent $\{w_i\}$ (centres, covariances) and the consequents Θ. For fixed $\{w_i\}$ the model is linear in Θ.

B.2 Combined criterion with noise cluster: alternating minimisation (Proposition 1)

With the notation of (6)–(7), define $U = (\mu_{ik})$, $V = (v_i)$ and Θ. The feasible set for U is

$$\mathscr{U} = \Big\{\mu_{ik} \geq 0,\ \sum_i \mu_{ik} \leq 1\Big\}, \tag{22}$$

the slack $\mu_{0k} = 1 - \sum_i \mu_{ik}$ being the membership in the noise cluster.

The two results of this appendix are standard: monotonicity of block-coordinate descent (Proposition 1) and convexity with half-quadratic majorisation of Huber regression (Proposition 2 [40]). They are stated to fix the conditions under which the algorithms are monotone and well defined; they are not claimed as new, and they do not explain the empirical differences between the models.

Proposition 1 (Alternating minimisation for fixed δ^2)**.** *Let $m > 1$, $\alpha \in [0,1]$, $\delta^2 > 0$ fixed and $D_{ik}^2 > 0$. (a) For fixed V, Θ the minimiser of J over $\mathscr{U}$ is the closed form* (8)*. (b) For fixed U, Θ the minimiser over V is given by* (9) *if $\alpha > 0$. (c) For fixed U, V the minimiser over θ_i is the weighted least-squares estimate with weights μ_{ik}^m if $\beta > 0$. Hence one sweep (a)→(b)→(c) does not increase J, and the sequence $J^{(t)}$ converges.*

Proof. (a) For fixed k, J restricted to $(\mu_{1k}, \ldots, \mu_{ck})$ is a convex function on the convex set $\{\mu \geq 0, \sum_i \mu_i \leq 1\}$: each term $\mu_i^m D_{ik}^2$ is convex and so is $\delta^2(1 - \sum_i \mu_i)^m$ for $m > 1$. The stationarity conditions

$$m\mu_i^{m-1} D_i^2 = m\delta^2 \Big(1 - \sum_j \mu_j\Big)^{m-1} \tag{23}$$

give

$$\mu_i = \Big[\frac{\delta^2}{D_i^2}\Big]^{1/(m-1)} \Big(1 - \sum_j \mu_j\Big); \tag{24}$$

summing over i and solving for $1 - \sum_j \mu_j$ gives the closed form. (b) and (c) are quadratic in v_i and θ_i respectively, because D_{ik}^2 is the sum of $\alpha\|x_k - v_i\|^2/s_d^2$ and $\beta(y_k - \theta_i^T \bar{x}_k)^2/s_E^2$ (with the squared-distance form of d_{ik} and E_{ik}), and each term involves one block only. Each block step is an exact minimisation, hence $J^{(t+1)} \leq J^{(t)}$; $J \geq 0$ gives convergence of $J^{(t)}$. □

Remark. Proposition 1 concerns the values of J and not the iterates; convergence of the iterates to a stationary point needs further conditions. In our implementation δ^2 is recomputed at each sweep from the current D (a data-dependent constant), so the monotonicity holds exactly only between two updates of δ^2; we observed convergence in all runs but do not claim a proof for this variant.

B.3 Global Huber refit of the consequents (C, C2)

Given $\{w_i\}$, let

$$\varphi(x) = \big[w_1(x)\bar{x}^T, \ldots, w_c(x)\bar{x}^T\big]^T \tag{25}$$

and Θ_0 the local estimate. C2 solves

$$\min_{\Theta} \sum_{k=1}^{N} \rho_H\big(y_k - \Theta^T \varphi(x_k)\big) + \lambda \|\Theta - \Theta_0\|^2, \qquad \rho_H(r) = \begin{cases} r^2/2, & |r| \le \kappa_H s \\ \kappa_H s|r| - \kappa_H^2 s^2/2, & \text{else} \end{cases} \tag{26}$$

where s is a MAD scale of the residuals and $\kappa_H = 1.345$. Unlike the local fits, (26) measures the error of the *final output*, and the ridge term controls the departure from the local solution ($\lambda_{\text{rel}} = 0.3$ for C2).

B.4 Joint rule–RBF hybrid (HYB1): convexity and IRLS (Proposition 2)

Let

$$\psi(x) = [\varphi_1(x), \ldots, \varphi_J(x)]^T, \qquad \varphi_j(x) = \exp\Big(-\frac{\|x - C_j\|^2}{2s_j^2}\Big), \tag{27}$$

and $\gamma = [\Theta; \beta]$. HYB1 solves

$$\min_{\gamma} \sum_{k} \rho_H\big(y_k - \gamma^T [\varphi(x_k); \psi(x_k)]\big) + \lambda_\theta \|\Theta - \Theta_0\|^2 + \lambda_\beta \|\beta\|^2. \tag{28}$$

Proposition 2 (Convexity and IRLS)**.** *For $\lambda_\theta, \lambda_\beta > 0$ the objective* (28) *is strictly convex and has a unique minimiser; the iteratively reweighted least-squares (IRLS) iteration with weights $\omega_k = \psi_H(r_k)/r_k$ decreases it monotonically.*

Proof. ρ_H is convex and the composition with a linear map is convex; the ridge terms are strongly convex, so the sum is strictly convex and coercive, which gives existence and uniqueness. For IRLS, $\rho_H(r) \le \rho_H(r_t) + \frac{\omega_t}{2}(r^2 - r_t^2)$ because $\rho_H(\sqrt{u})$ is concave in u (half-quadratic majorisation); minimising the quadratic majoriser is a ridge-regularised weighted least-squares problem, so each iteration decreases the objective. □

The proposition concerns the fit for *fixed* antecedent, units and hyper-parameters; the choice of the width multiplier and of $(\lambda_\theta, \lambda_\beta)$ by cross-validation is a nonconvex outer problem.

B.5 Quasi-LPV form, poles and identifiability

With the regressor

$$x_t = [y(t-1), \ldots, y(t-n_a), u(t-1), \ldots, u(t-n_b)]^T, \tag{29}$$

(2) is an ARX model whose coefficients are convex combinations of the local ones. At a frozen scheduling vector the model is linear time-invariant with poles given by the roots of

$$z^{n_a} - \sum_{k=1}^{n_a} a_k(\xi)\, z^{n_a - k} = 0. \tag{30}$$

Identifiability requires that each rule has at least $q = p + 1$ points with sufficiently exciting regressors; with small clusters the weighted normal equations are ill-conditioned, which motivates the ridge terms in C2 and HYB1. The use of lagged signals as scheduling variables is standard in identification [9]. A sufficient stability condition for TS models, such as a common quadratic Lyapunov function [49], could be checked on the local companion matrices, but was not done here.

References

[1] Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Syst. Man Cybern. 15(1), 116–132 (1985)

[2] Sugeno, M., Kang, G.T.: Structure identification of fuzzy model. Fuzzy Sets Syst. 28(1), 15–33 (1988)

[3] Hathaway, R.J., Bezdek, J.C.: Switching regression models and fuzzy clustering. IEEE Trans. Fuzzy Syst. 1(3), 195–204 (1993)

[4] Krishnapuram, R., Keller, J.M.: A possibilistic approach to clustering. IEEE Trans. Fuzzy Syst. 1(2), 98–110 (1993)

[5] Davé, R.N.: Characterization and detection of noise in clustering. Pattern Recognit. Lett. 12(11), 657–664 (1991)

[6] Jabeur Telmoudi, A., Soltani, M., Chaouech, L., Chaari, A.: Parameter estimation of nonlinear systems using a robust possibilistic c-regression model algorithm. Proc. Inst. Mech. Eng. Part I: J. Syst. Control Eng. 234(1), 134–143 (2020). doi:10.1177/0959651818756246

[7] Soltani, M., Aissaoui, B., Chaari, A., Ben Hmida, F., Gossa, M.: A modified fuzzy c-regression model clustering algorithm for T-S fuzzy model identification. In: Eighth Int. Multi-Conf. on Systems, Signals and Devices, pp. 1–6. IEEE (2011). doi:10.1109/SSD.2011.5767365

[8] Moody, J., Darken, C.J.: Fast learning in networks of locally-tuned processing units. Neural Comput. 1(2), 281–294 (1989)

[9] Ljung, L.: System Identification: Theory for the User, 2nd edn. Prentice Hall, Upper Saddle River (1999)

[10] Schoukens, J., Ljung, L.: Nonlinear system identification: a user-oriented road map. IEEE Control Syst. Mag. 39(6), 28–99 (2019)

[11] Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)

[12] Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Stat. Comput. 14(3), 199–222 (2004)

[13] Breiman, L.: Random forests. Mach. Learn. 45(1), 5–32 (2001)

[14] Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat. 29(5), 1189–1232 (2001)

[15] Ljung, L., Andersson, C., Tiels, K., Schön, T.B.: Deep learning and system identification. IFAC-PapersOnLine 53(2), 1175–1181 (2020)

[16] Beintema, G., Tóth, R., Schoukens, M.: Nonlinear state-space identification using deep encoder networks. In: Proc. 3rd Conf. Learning for Dynamics and Control, PMLR 144, 241–250 (2021)

[17] Gedon, D., Wahlström, N., Schön, T.B., Ljung, L.: Deep state space models for nonlinear system identification. IFAC-PapersOnLine 54(7), 481–486 (2021)

[18] Forgione, M., Piga, D.: dynoNet: a neural network architecture for learning dynamical systems. Int. J. Adapt. Control Signal Process. 35(4), 612–626 (2021)

[19] Pillonetto, G., Aravkin, A., Gedon, D., Ljung, L., Ribeiro, A.H., Schön, T.B.: Deep networks for system identification: a survey. Automatica 171, 111907 (2025)

[20] Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707 (2019)

[21] Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L.: Physics-informed machine learning. Nat. Rev. Phys. 3(6), 422–440 (2021)

[22] Jang, J.-S.R.: ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans. Syst. Man Cybern. 23(3), 665–685 (1993)

[23] Leśki, J.M., Kotas, M.: On robust fuzzy c-regression models. Fuzzy Sets Syst. 279, 112–129 (2015). doi:10.1016/j.fss.2014.12.004

[24] Bouzbida, M., Hassine, L., Chaari, A.: Robust kernel clustering algorithm for nonlinear system identification. Math. Probl. Eng. 2017, 2427309 (2017). doi:10.1155/2017/2427309

[25] Soltani, M., Telmoudi, A.J., Ben Belgacem, H., Chaari, A.: A new robust approach for fuzzy c-regression model identification. Neural Comput. Appl. 32(15), 11361–11371 (2020). doi:10.1007/s00521-019-04631-w

[26] Chuang, C.-C., Jeng, J.-T., Tao, C.-W.: Hybrid robust approach for TSK fuzzy modeling with outliers. Expert Syst. Appl. 36(5), 8925–8931 (2009). doi:10.1016/j.eswa.2008.11.053

[27] Mendel, J.M., John, R.I.B.: Type-2 fuzzy sets made simple. IEEE Trans. Fuzzy Syst. 10(2), 117–127 (2002)

[28] Karnik, N.N., Mendel, J.M., Liang, Q.: Type-2 fuzzy logic systems. IEEE Trans. Fuzzy Syst. 7(6), 643–658 (1999)

[29] Soltani, M., Telmoudi, A.J., Chaouech, L., Ali, S.B., Chaari, A.: Design of a robust interval-valued type-2 fuzzy c-regression model for a nonlinear system with noise and outliers. Soft Comput. 23(15), 6125–6134 (2019). doi:10.1007/s00500-018-3265-z

[30] Tsai, S.-H., Chen, Y.-W.: A novel interval type-2 fuzzy system identification method based on the modified fuzzy c-regression model. IEEE Trans. Cybern. 52(9), 9834–9845 (2022). doi:10.1109/TCYB.2021.3072851

[31] Shi, J.: TS fuzzy model identification based on an improved interval type-2 fuzzy c-regression model. J. Intell. Fuzzy Syst. 44(3), 4495–4507 (2023). doi:10.3233/JIFS-221434

[32] Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Comput. 3(1), 79–87 (1991)

[33] Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. B 39(1), 1–38 (1977)

[34] Peel, D., McLachlan, G.J.: Robust mixture modelling using the *t* distribution. Stat. Comput. 10(4), 339–348 (2000)

[35] Chamroukhi, F.: Robust mixture of experts modeling using the *t* distribution. Neural Netw. 79, 20–36 (2016)

[36] Jang, J.-S.R., Sun, C.-T.: Functional equivalence between radial basis function networks and fuzzy inference systems. IEEE Trans. Neural Netw. 4(1), 156–159 (1993)

[37] Wu, D., Lin, C.-T., Huang, J., Zeng, Z.: On the functional equivalence of TSK fuzzy systems to neural networks, mixture of experts, CART, and stacking ensemble regression. IEEE Trans. Fuzzy Syst. 28(10), 2570–2580 (2020)

[38] Cui, Y., Wu, D., Huang, J.: Optimize TSK fuzzy systems for classification problems: minibatch gradient descent with uniform regularization and batch normalization. IEEE Trans. Fuzzy Syst. 28(12), 3065–3075 (2020)

[39] Huber, P.J.: Robust estimation of a location parameter. Ann. Math. Stat. 35(1), 73–101 (1964)

[40] Holland, P.W., Welsch, R.E.: Robust regression using iteratively reweighted least-squares. Commun. Stat. Theory Methods 6(9), 813–827 (1977)

[41] Wiktorowicz, K., Krzeszowski, T.: Identification of time series models using sparse Takagi–Sugeno fuzzy systems with reduced structure. Neural Comput. Appl. 34(10), 7473–7488 (2022). doi:10.1007/s00521-021-06843-5

[42] Ben Mabrouk, A., Alanazi, A., Bassfar, Z., et al.: New hybrid model for nonlinear systems via Takagi–Sugeno fuzzy approach. AIMS Math. 9(9), 23197–23220 (2024). doi:10.3934/math.20241128

[43] Arifi, A., Bouallègue, S.: Takagi–Sugeno fuzzy-based approach for modeling and control of an activated sludge process. Int. J. Dyn. Control 12(8), 3123–3138 (2024). doi:10.1007/s40435-024-01398-4

[44] Klai, Z., Hammami, M.A., Ayari, M., Delmotte, F.: New criteria for state estimation in uncertain fuzzy dynamical control systems. Int. J. Dyn. Control 14(9), 329 (2026). doi:10.1007/s40435-026-02267-y

[45] Day, N.E.: Estimating the components of a mixture of normal distributions. Biometrika 56(3), 463–474 (1969)

[46] Kiefer, J., Wolfowitz, J.: Consistency of the maximum likelihood estimator in the presence of infinitely many incidental parameters. Ann. Math. Stat. 27(4), 887–906 (1956)

[47] Grinsztajn, L., Oyallon, E., Varoquaux, G.: Why do tree-based models still outperform deep learning on typical tabular data? In: Advances in Neural Information Processing Systems 35, Datasets and Benchmarks Track (2022)

[48] Shwartz-Ziv, R., Armon, A.: Tabular data: deep learning is not all you need. Inf. Fusion 81, 84–90 (2022)

[49] Tanaka, K., Wang, H.O.: Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach. Wiley, New York (2001)

[50] Ghorbel, H., El Hajjaji, A., Souissi, M., Chaabane, M.: Robust tracking control for Takagi–Sugeno fuzzy systems with unmeasurable premise variables: application to tank system. J. Dyn. Syst. Meas. Control 136(4), 041011 (2014). doi:10.1115/1.4026467

[51] Liu, H., Zhang, T.: Fuzzy sliding mode control of robotic manipulators with kinematic and dynamic uncertainties. J. Dyn. Syst. Meas. Control 134(6), 061007 (2012). doi:10.1115/1.4006626

[52] Bertino, A., Naseradinmousavi, P., Kelkar, A.: Analytical and experimental decentralized adaptive control of a high-degrees-of-freedom robot manipulator. J. Dyn. Syst. Meas. Control 143(7), 071007 (2021). doi:10.1115/1.4049795

[53] Rodriguez, R., Ahmadzadeh, O., Wang, Y., Soudbakhsh, D.: Data-driven discovery of lithium-ion battery state of charge dynamics. J. Dyn. Syst. Meas. Control 146(1), 011101 (2024). doi:10.1115/1.4064026

[54] Škrjanc, I., Iglesias, J.A., Sanchis, A., Leite, D., Lughofer, E., Gomide, F.: Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: a survey. Inf. Sci. 490, 344–368 (2019)

[55] Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat. Mach. Intell. 1(5), 206–215 (2019)

[56] Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning, 2nd edn. Springer, New York (2009)

[57] Wolpert, D.H.: Stacked generalization. Neural Netw. 5(2), 241–259 (1992)

[58] Breiman, L.: Stacked regressions. Mach. Learn. 24(1), 49–64 (1996)

[59] Ljung, G.M., Box, G.E.P.: On a measure of lack of fit in time series models. Biometrika 65(2), 297–303 (1978)

[60] Arthur, D., Vassilvitskii, S.: *k*-means++: the advantages of careful seeding. In: Proc. 18th ACM-SIAM Symp. Discrete Algorithms, pp. 1027–1035 (2007)

[61] Mackey, M.C., Glass, L.: Oscillation and chaos in physiological control systems. Science 197(4300), 287–289 (1977)

[62] Box, G.E.P., Jenkins, G.M.: Time Series Analysis: Forecasting and Control. Holden-Day, San Francisco (1970)

[63] Narendra, K.S., Parthasarathy, K.: Identification and control of dynamical systems using neural networks. IEEE Trans. Neural Netw. 1(1), 4–27 (1990)

[64] Lorenz, E.N.: Deterministic nonperiodic flow. J. Atmos. Sci. 20(2), 130–141 (1963)

[65] Schoukens, M., Mattsson, P., Wigren, T., Noël, J.-P.: Cascaded tanks benchmark combining soft and hard nonlinearities. In: Workshop on Nonlinear System Identification Benchmarks, Brussels (2016)

[66] Hanifi, S., Liu, X., Lin, Z., Lotfian, S.: A critical review of wind power forecasting methods—past, present and future. Energies 13(15), 3764 (2020)

[67] Weigand, J., Götz, J., Ulmen, J., Ruskowski, M.: Dataset and baseline for an industrial robot identification benchmark. TU Kaiserslautern (2022). doi:10.26204/data/5

[68] Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830 (2011)

[69] Champneys, M.D., Beintema, G.I., Tóth, R., Schoukens, M., Rogers, T.J.: Baseline results for selected nonlinear system identification benchmarks. arXiv:2405.10779 (2024)

[70] Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7, 1–30 (2006)

[71] Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bull. 1(6), 80–83 (1945)

[72] Holm, S.: A simple sequentially rejective multiple test procedure. Scand. J. Stat. 6(2), 65–70 (1979)

[73] Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. Ann. Stat. 32(2), 407–499 (2004)

[74] Wigren, T., Schoukens, J.: Three free data sets for development and benchmarking in nonlinear system identification. In: Proc. European Control Conf., pp. 2933–2938 (2013)

[75] Janot, A., Gautier, M., Brunot, M.: Data set and reference models of EMPS. In: Workshop on Nonlinear System Identification Benchmarks, Eindhoven (2019)