Data-driven Metamodels for Failure Analysis of Power Electronic Modules

Mehdi Ghrabli* , Emanuel Aldea, Ludovic Chamoin, Mounira Bouarroudj

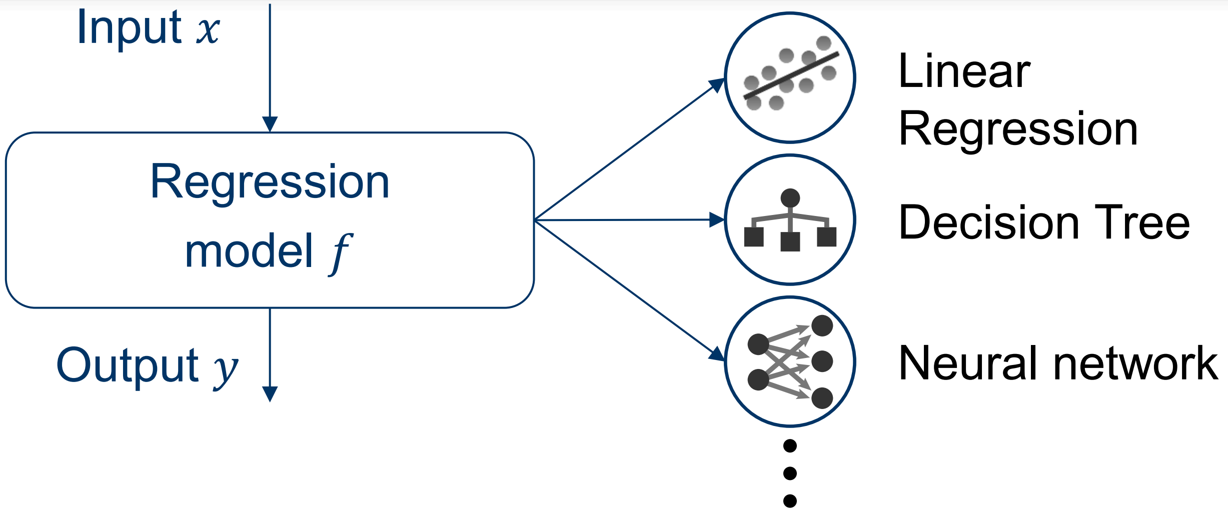
CNRS, SATIE Laboratory & LMPS Laboratory

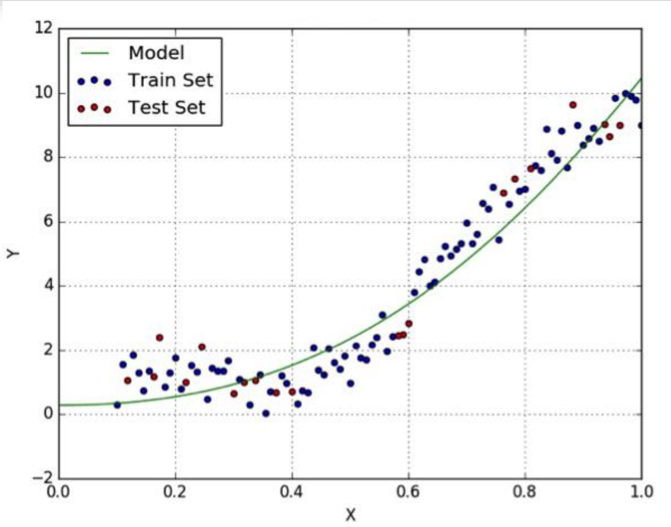
Bordeaux (France), October 6-9, 2025

Objectives

- Emulate *numerical simulations* using *data-driven models* trained on simulated data:
  - Accelerate computational workflows using machine learning (ML).
  - Optimize data selection while maximizing accuracy using advanced techniques (Active learning, residual errors, etc.).
- ML as a solution for *challenges* in power electronic modules' (PEM) *reliability assessment:*
  - Enable accurate long-term health state estimation (high cycle counts) as a more robust alternative for classical models.
  - Exploit the generalization capabilities of ML models for predictions on unseen data.
- Establish good practices for *rigorous evaluation* of ML models.

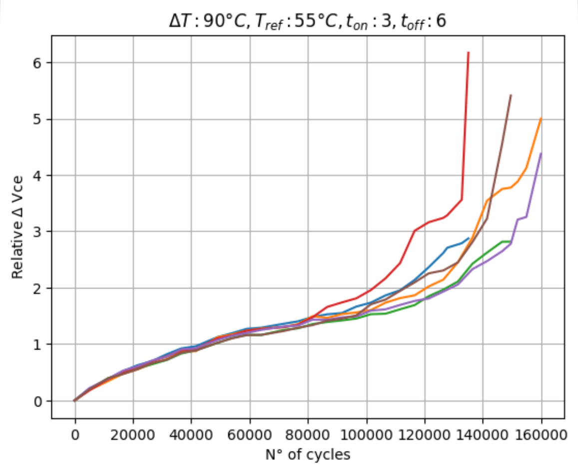
Regression in Machine Learning

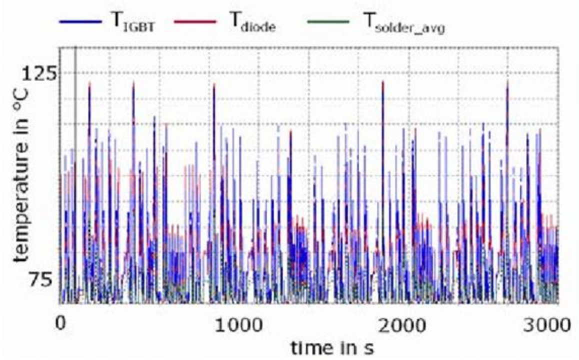


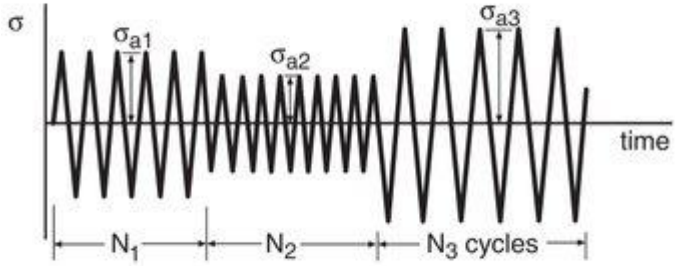


- **Fast inference** once trained.
- **Developed field:** well-established methods, libraries, benchmarks.
- **Expressive modeling capacity:** Can represent complex, nonlinear, and high-dimensional relationships beyond the reach of analytical or empirical models.
- **Interpretability**: Many models operate as black boxes, making results harder to explain.
- **Data dependence**: Performance strongly depends on the quantity and quality of training data.
- **Model selection:** choosing an appropriate model requires a good understanding of the problem and the data available.

Problems in PEM Reliability



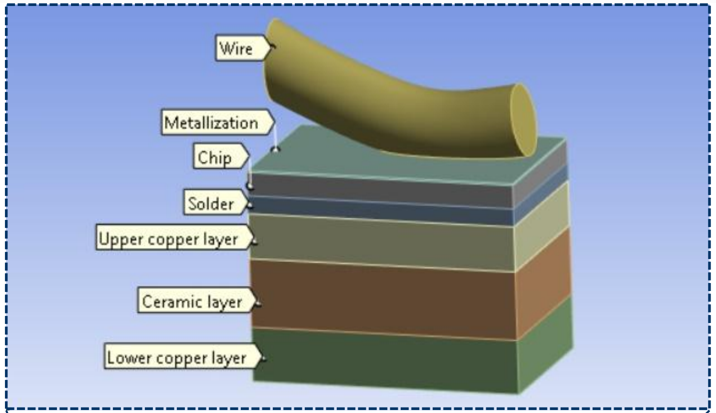


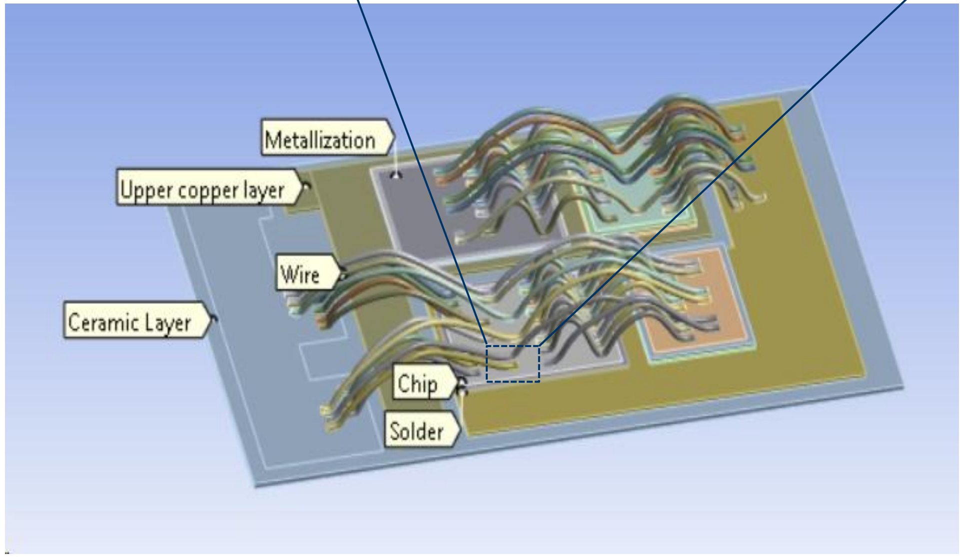


- **High computational cost due to long lifetimes:**
  - Prediction models require *frequent updates* on the PEM's state from *costly simulations.*
- **Mission profile complexity due to its stochastic nature:**
  - Experimental data is generated using regular, uniform loading cycles. This *simplified data* does not account for the *arbitrary load variations* seen in real-world operations.
- **Limitations of classical models:** *Linearity* and *order-independence assumptions* in models based on Rainflow counting, Palmgren-Miner rule, Look-up tables etc.

Simulations for Bond wire Degradation

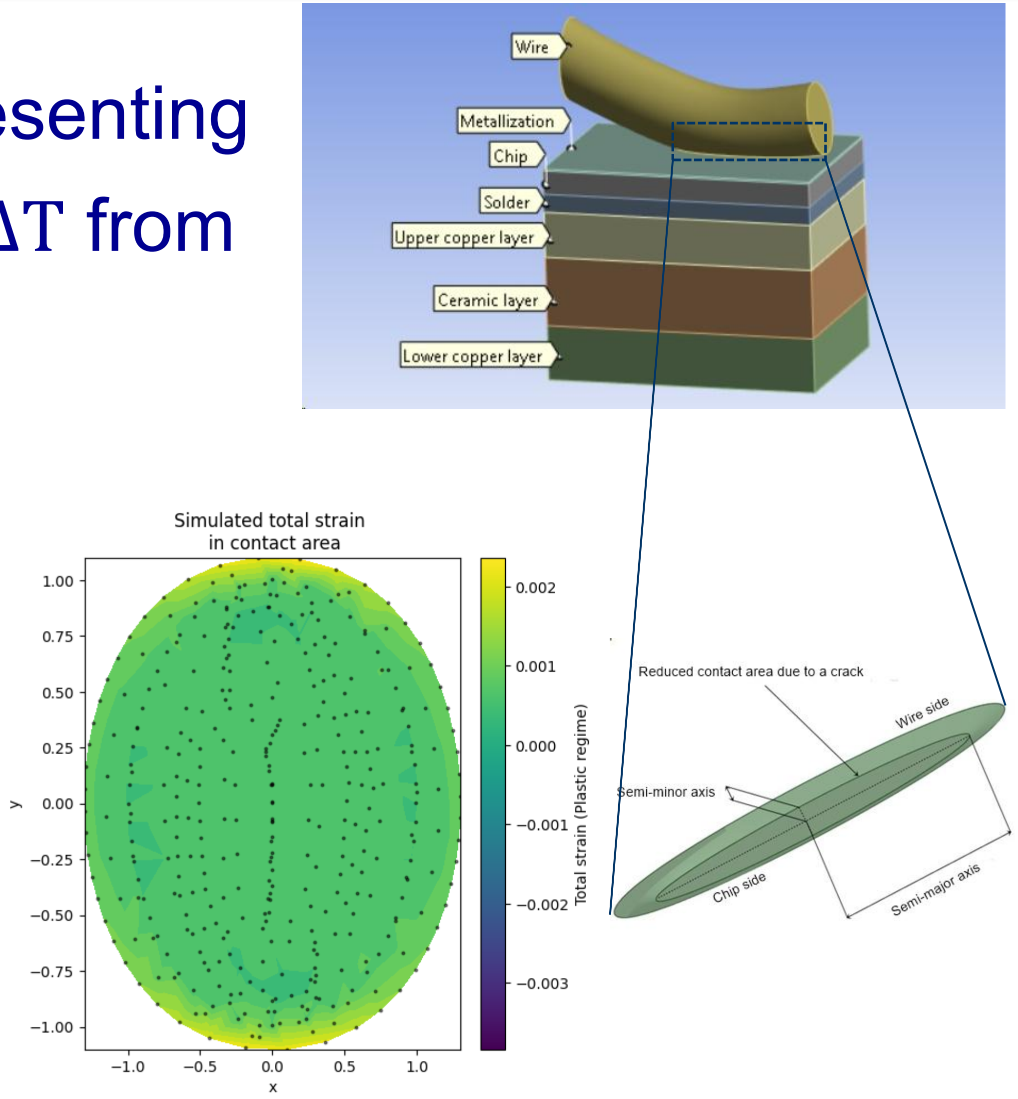
- Obtain the mechanical state of a PEM presenting a **crack** l_c, due to a **temperature swing** $\Delta\mathrm{T}$ from the module's self heating:
  - Inject a *current* I (load).
  - Obtain a *temperature profile* using a strong thermo-electric coupling.
  - Retrieve *mechanical descriptors* $(\sigma, \epsilon, u, etc.)$ using a weak thermo-mechanical coupling.





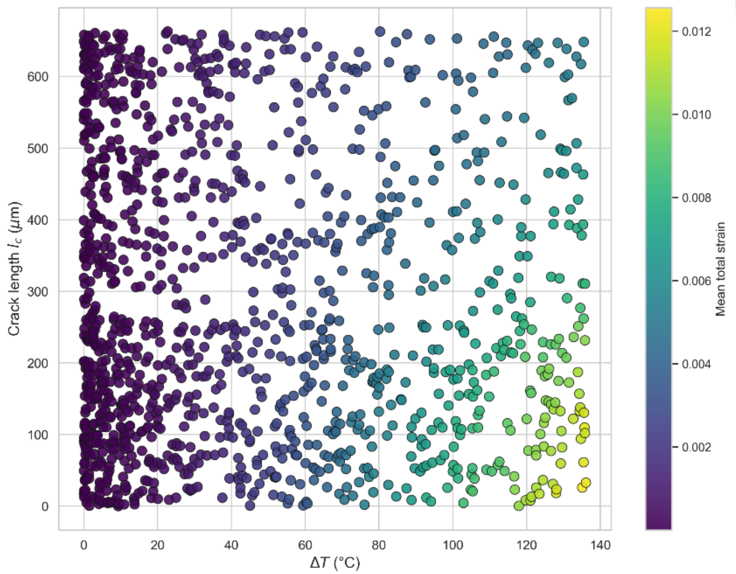
Simulations for Bond wire Degradation

- Obtain the mechanical state of a PEM presenting a **crack** l_c, due to a **temperature swing** $\Delta\mathrm{T}$ from the module's self heating:
  - Inject a *current* I (load).
  - Obtain a *temperature profile* using a strong thermo-electric coupling.
  - Retrieve *mechanical descriptors* $(\sigma, \epsilon, u, etc.)$ using a weak thermo-mechanical coupling.



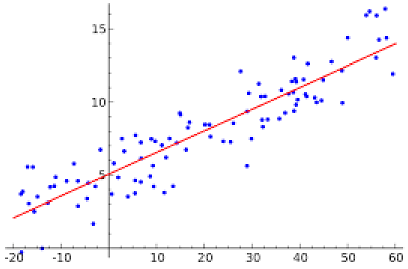
Data Generation

Temperature variation ΔT	Crack length l_c	Mech node coordinate x	Mech node coordinate y	Equivalent total strain ε



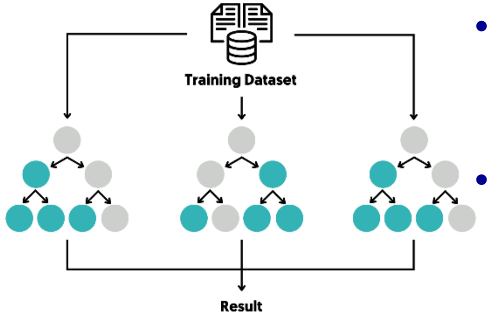
- Module state input values $(\Delta T, l_c)$ chosen using random uniform sampling.
- Measurement points (x, y) are generated according to the meshing.
- $\approx$ 3 minutes per simulation.
- Outputs: strain ϵ, stress σ, displacement u (shear, directional, principal..).

ML Models for Simulations



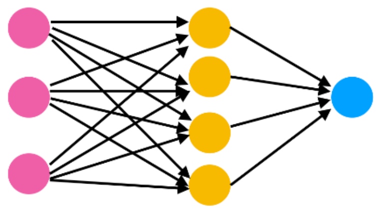
Regularization

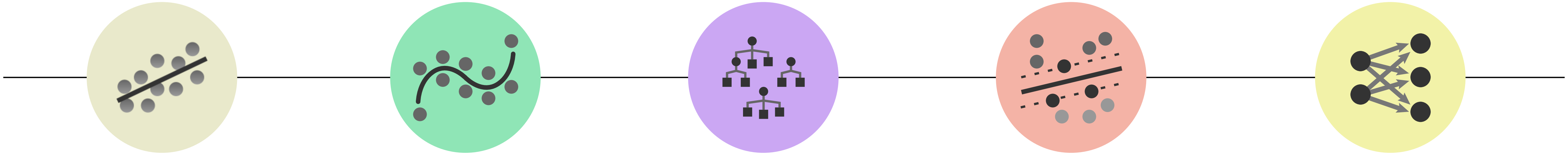
- Improved linear models, prevent overfitting by penalizing complexity.
- Includes: *Ridge, LASSO, ElasticNet.*



Non-parametric models

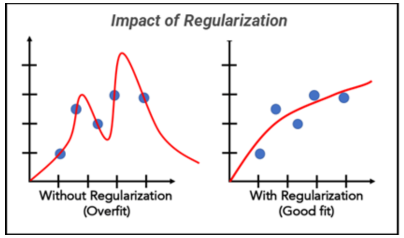
- Rely directly on the data structure, minimal assumptions.
- Includes*: Support Vector Regression, k-Nearest neighbors.*





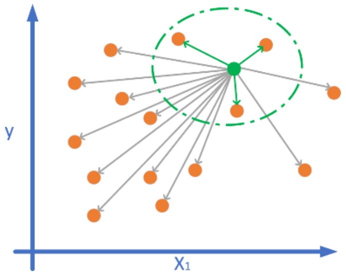
Linear models

- Fast but limited, assume a straight-line relationship.
- Includes: *Linear Regression + Polynomial Regression* (degree 2,3,4…).



Tree-based models

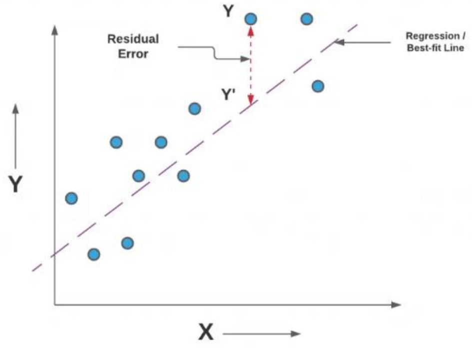
- Flexible, split data into decision rules, good interpretability.
- Includes: *Decision Tree, Random Forest, Bagging, Boosting.*



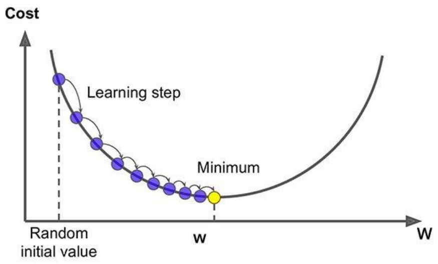
Neural Networks

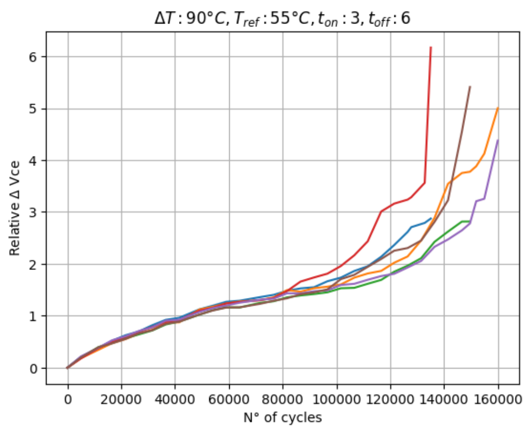
- Complex, layered models, effective for capturing nonlinear behaviors, require more data.

Evaluation Metrics



- Statistical metrics
  - ***Mean Squared Error*** *(MSE)* and ***Mean Absolute Error*** *(MAE)* measure how close the model's predictions are to actual values (lower is better).
  - ***Coefficient of Determination*** *(R^2)* indicates how well the model explains the data (higher, closer to 1, is better).

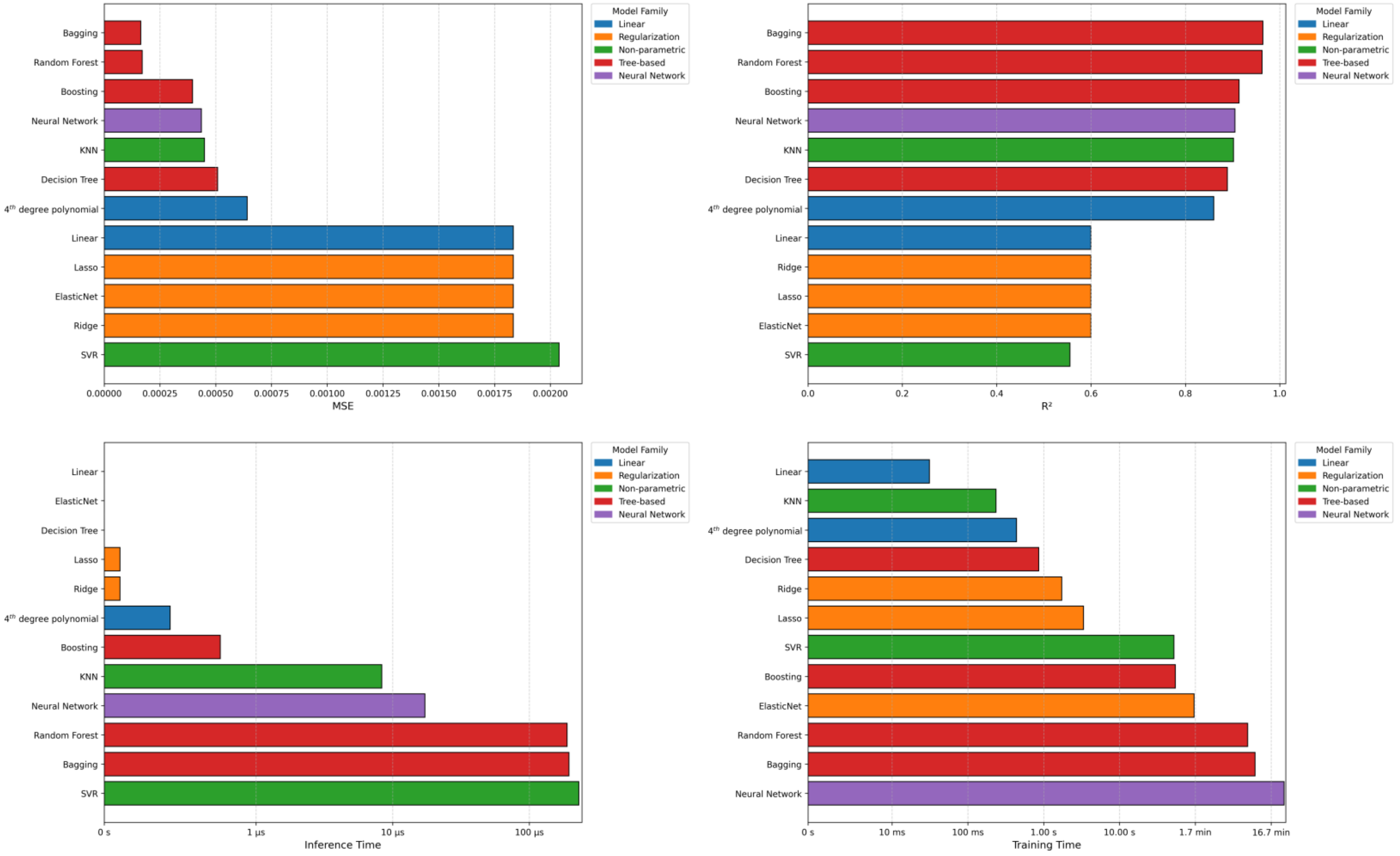




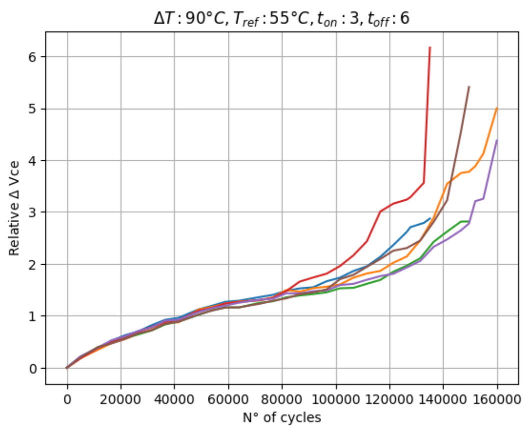
- Computation time
  - ***Fitting time****:* duration to calibrate the model parameters, done only *<u>once</u>* before use.
  - ***Inference time****:* time required for the model to produce predictions, happens *<u>frequently</u>* in a reliability assessment model.

Regression Results

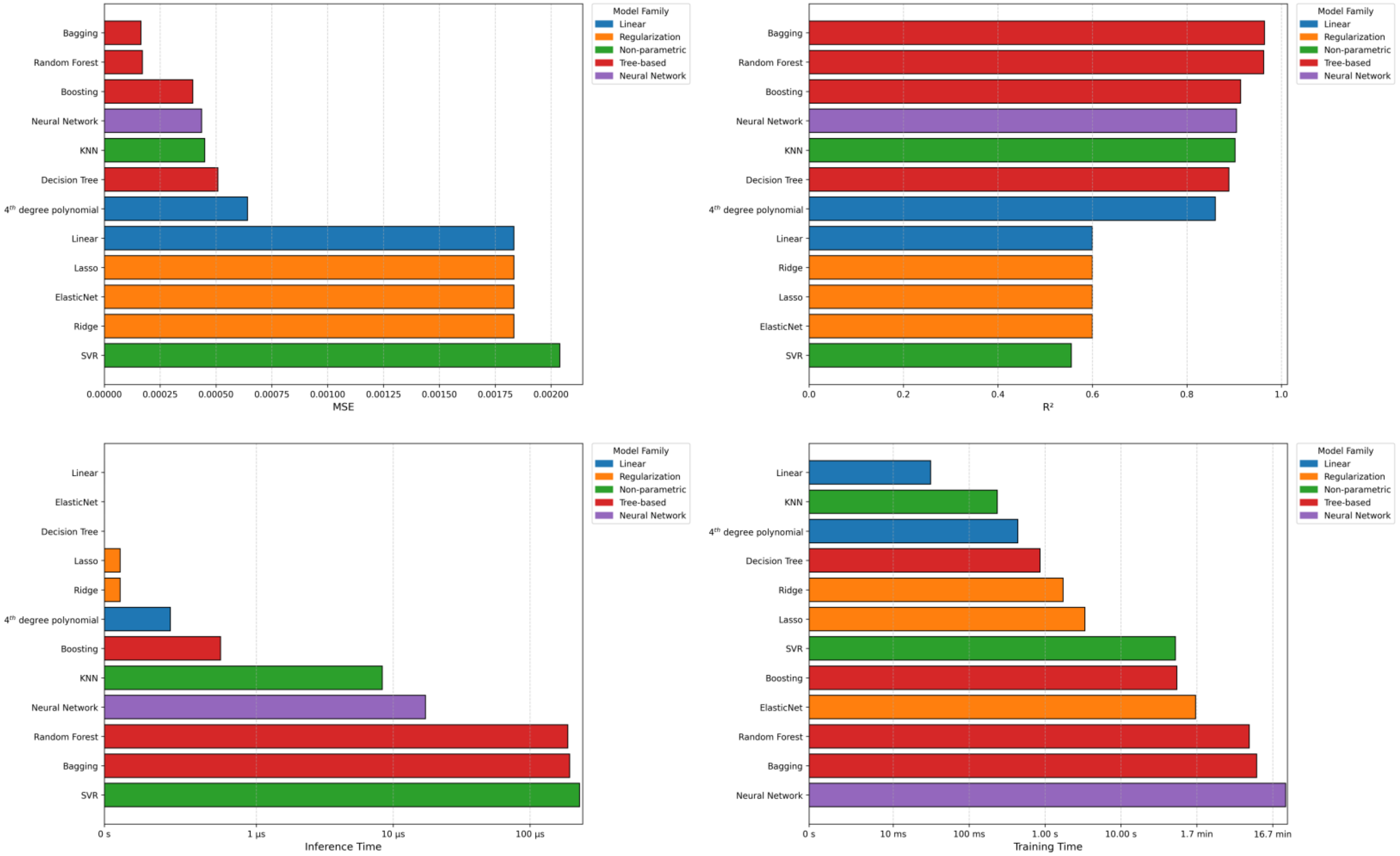
- Inference times for all models are *much faster than numerical simulations:* **10^6 times faster**.
- *<u>Tree-based models</u>* (**Bagging**), *kNN, and Neural Networks* achieve *high accuracy* with very low errors and R^2 values close to 1.
- *<u>Linear models</u>* (including regularization and polynomial models) ***fail to capture the complexity*** of the phenomenon, despite their fast computation times.



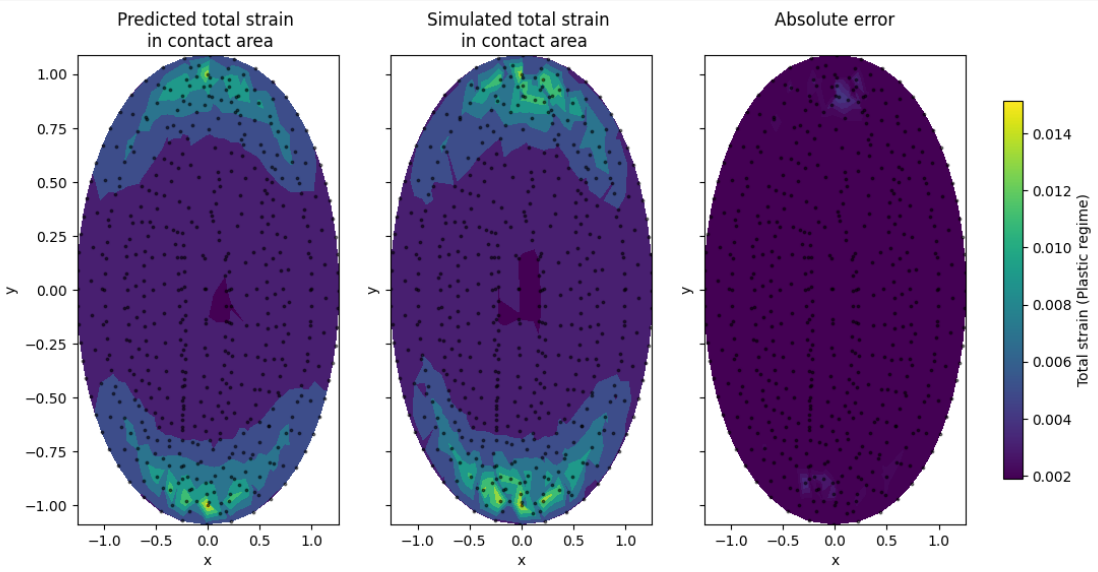
Regression Results

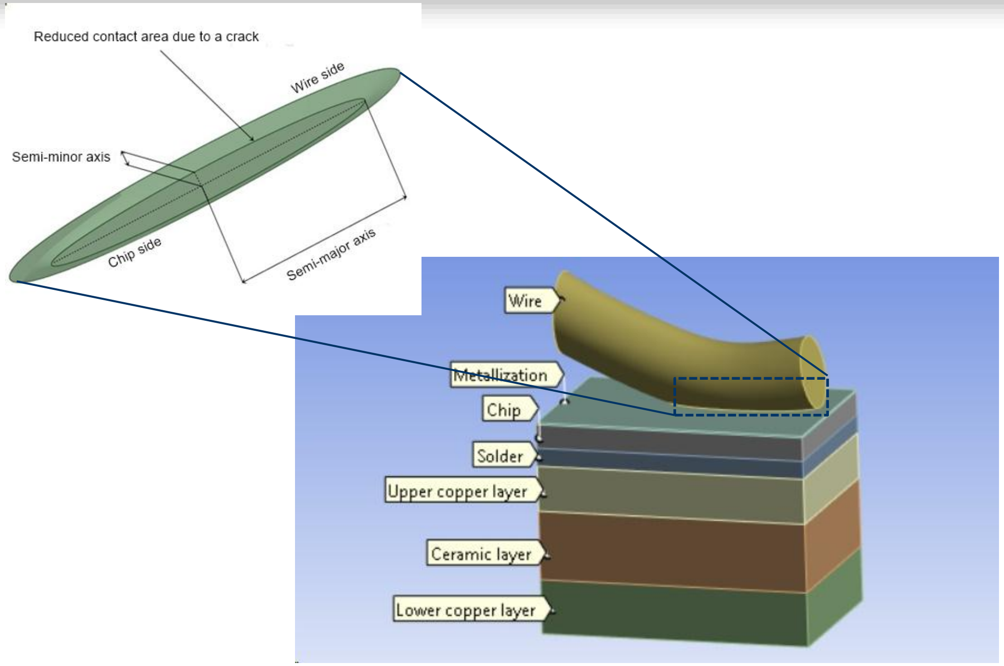


- Autoregressive (cycle-by-cycle) failure assessment for mission profiles containing 10^6 loads would require ≈<u>*230 seconds*</u> to compute using ML surrogate models, as opposed to <u>$\approx 5 \cdot 10^4$ *hours*</u> using numerical simulations.
- Failure assessment over the <u>*full lifetime*</u> using ML surrogate models (230s) is comparable to a <u>*single simplified simulation*</u> (180s)



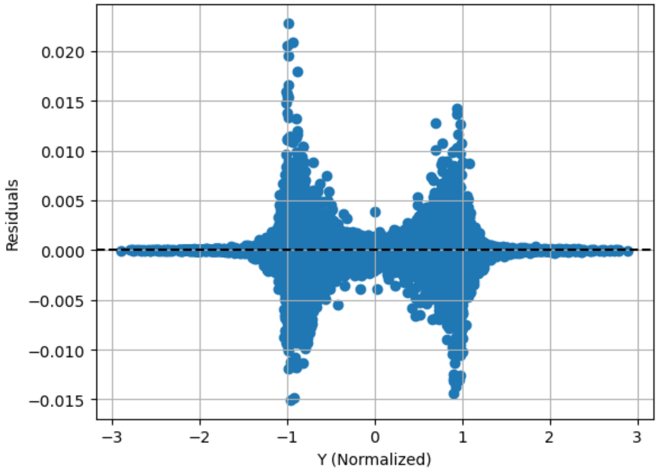
Visualizing Simulation Results and Errors

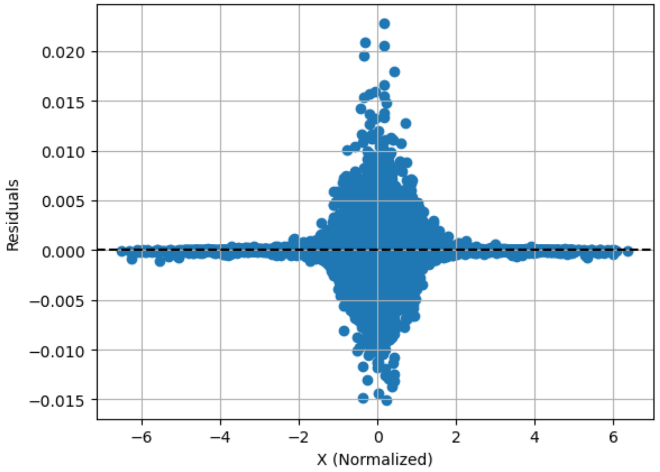


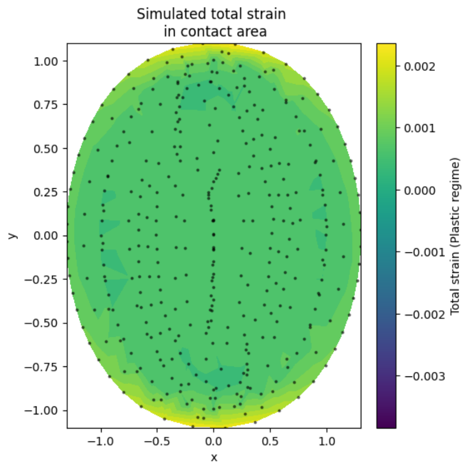


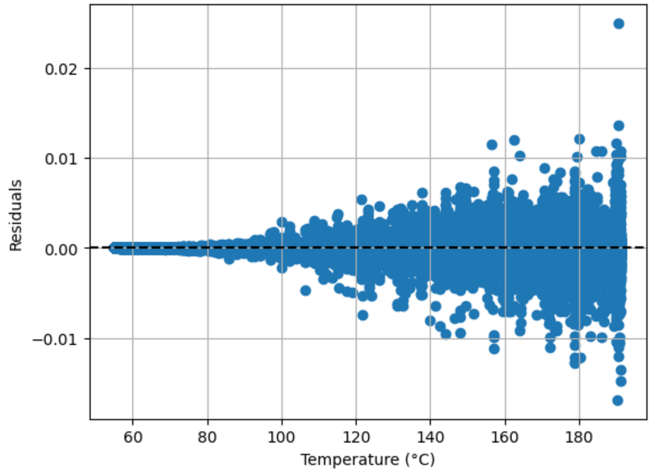
- Comparison between the **predicted** total strain (left) and the total strain **simulated** using finite element simulations (middle) in the contact area between the metallization and the wire.
- The absolute **error** (right) highlights the *accuracy* of the model (low absolute error) and the *reproduction of the phenomenon* (peak strain at the contact tips).

Residual Error Analysis





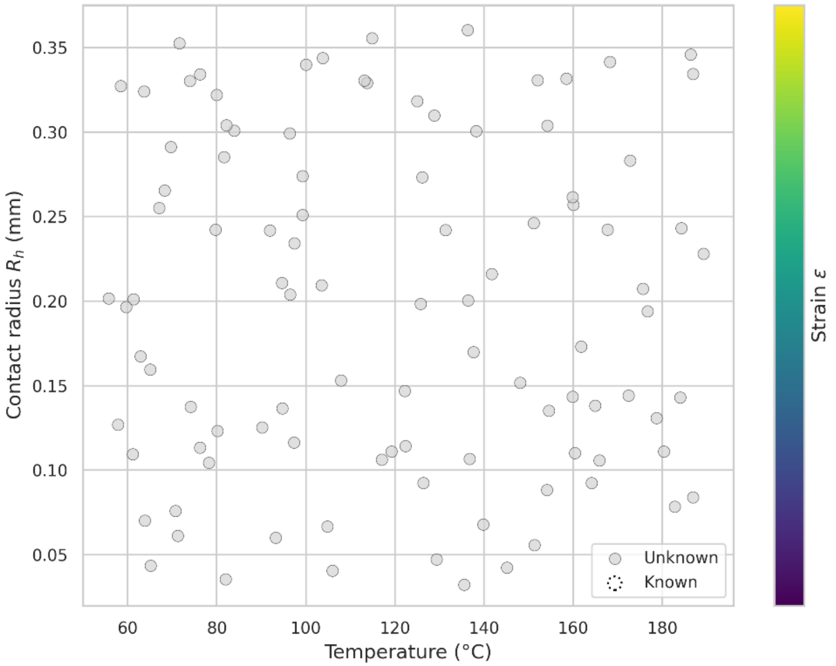




- **Residuals** r measure the difference between the prediction $\hat{\varepsilon}$ and the true value ε: $\boldsymbol{r} = \boldsymbol{\varepsilon} - \widehat{\boldsymbol{\varepsilon}}$
- Ideally, residual values are uniformly scattered around the **horizontal axis** $\boldsymbol{y} = \boldsymbol{0}$ (no bias).
- **Patterns** (high variance) indicate possible *improvements* (mesh refinements, data augmentation, feature generation):
  - Residuals with respect to *normalized* geometric features indicate uncertainty for $x = 0$ and $y = \pm 1$ (crack tips where the phenomenon is concentrated).

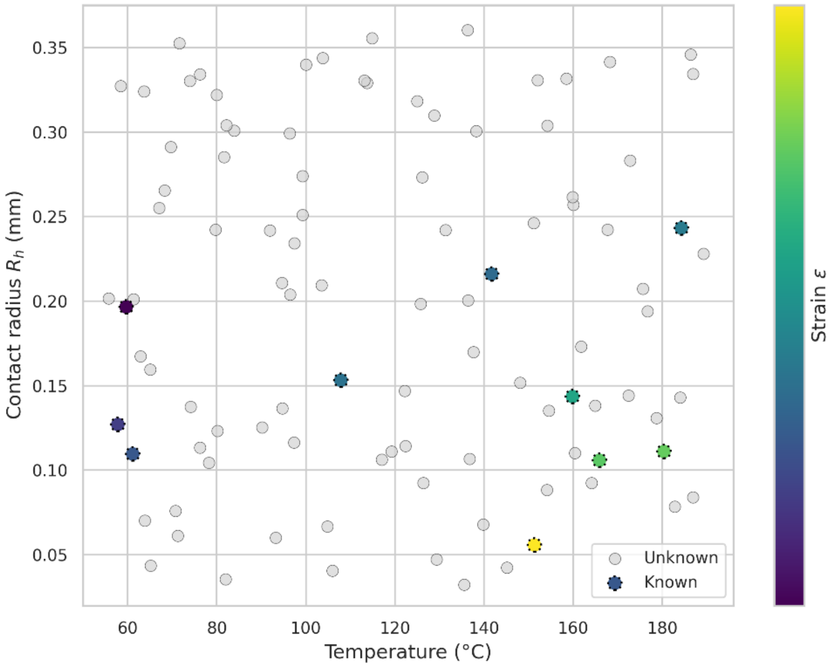
Optimizing the Simulation Protocol: Active learning

- **How to choose which simulations are most relevant to use?**
- Classical random sampling is *not optimal*: costly when data generation is slow/expensive
- Use the model's **uncertainty**:
  - If a model is uncertain in a candidate simulation's outcome, then carrying out the experiment and training on it would *remove* the model's *uncertainty*, making it more *robust for similar unseen scenarios*.
- Goal: find the m **most informative** simulations to use for training among n **possible simulations**.



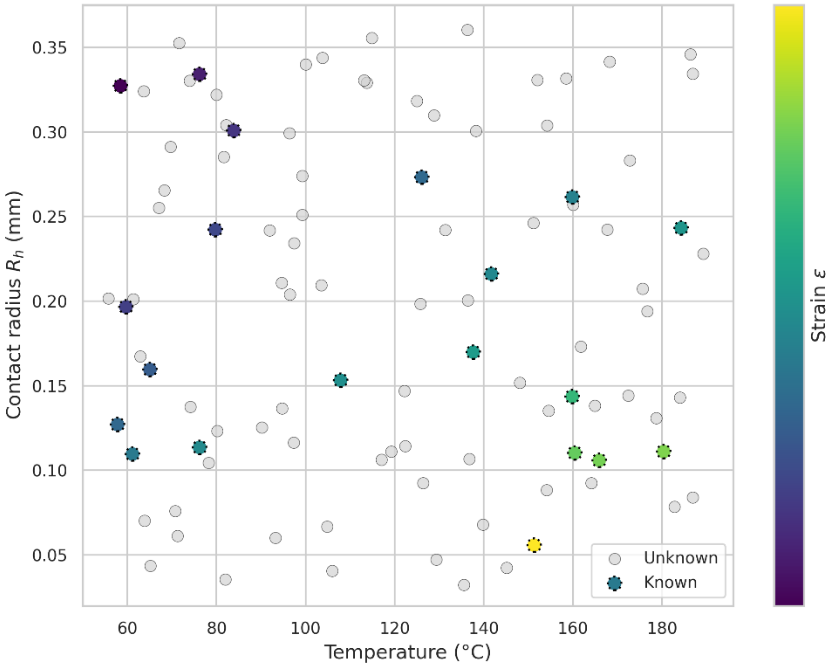
Optimizing the Simulation Protocol: Active learning

- **How to choose which simulations are most relevant to use?**
- Classical random sampling is *not optimal*: costly when data generation is slow/expensive
- Use the model's **uncertainty**:
  – If a model is uncertain in a candidate simulation's outcome, then carrying out the experiment and training on it would *remove* the model's *uncertainty*, making it more *robust for similar unseen scenarios*.
- Goal: find the m **most informative** simulations to use for training among n **possible simulations**.



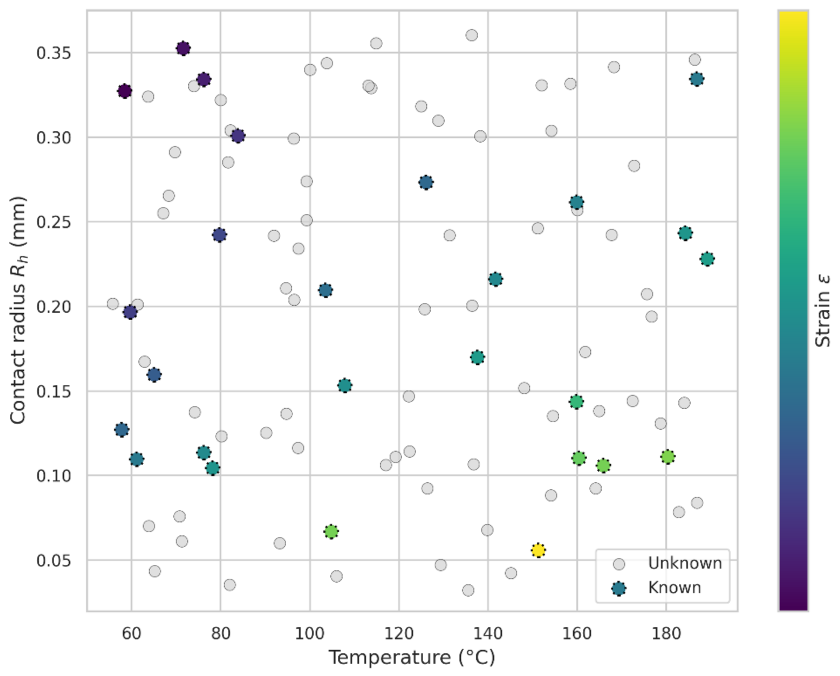
Optimizing the Simulation Protocol: Active learning

- **How to choose which simulations are most relevant to use?**
- Classical random sampling is *not optimal*: costly when data generation is slow/expensive
- Use the model's **uncertainty**:
  - If a model is uncertain in a candidate simulation's outcome, then carrying out the experiment and training on it would *remove* the model's *uncertainty*, making it more *robust for similar unseen scenarios*.
- Goal: find the m **most informative** simulations to use for training among n **possible simulations**.



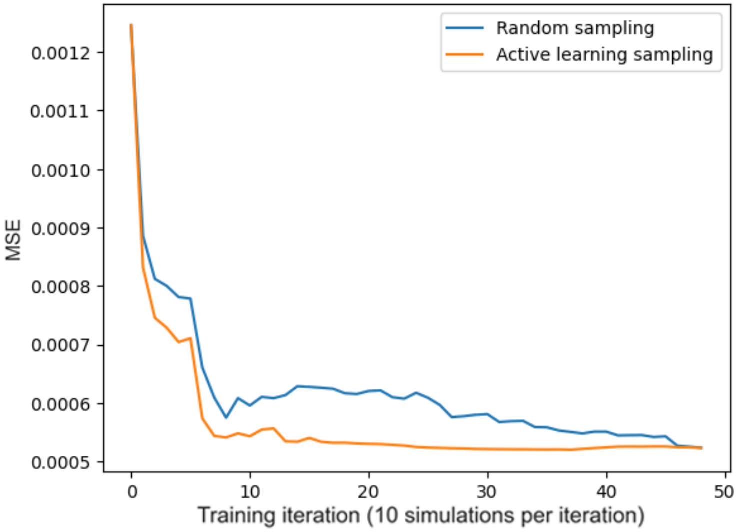
Optimizing the Simulation Protocol: Active learning

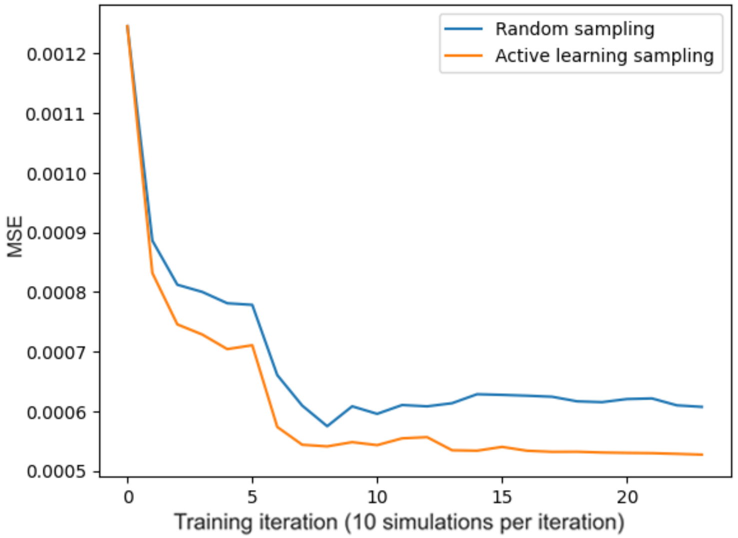
- **How to choose which simulations are most relevant to use?**
- Classical random sampling is *<u>not optimal</u>*: costly when data generation is slow/expensive
- Use the model's **uncertainty**:
  - If a model is uncertain in a candidate simulation's outcome, then carrying out the experiment and training on it would *<u>remove</u>* the model's *<u>uncertainty</u>*, making it more *<u>robust for similar unseen scenarios</u>*.
- Goal: find the m **most informative** simulations to use for training among n **possible simulations**.



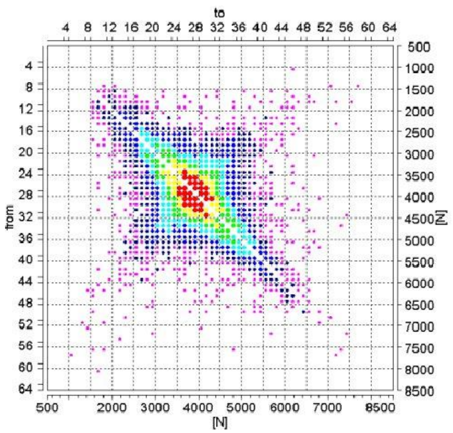
Optimizing the Simulation Protocol: Active learning

- We compare accuracies of data selection using **active learning** and **classic random sampling**.
- When *all training data is exhausted*, both methods reach the same accuracy, with active learning having the *faster convergence rate*.
- When *limiting training data* to 50% of all candidate simulations, active learning achieves *better accuracy*, highlighting the true advantages of active learning by *intelligently selecting the most informative simulations*.





ML-based Reliability model



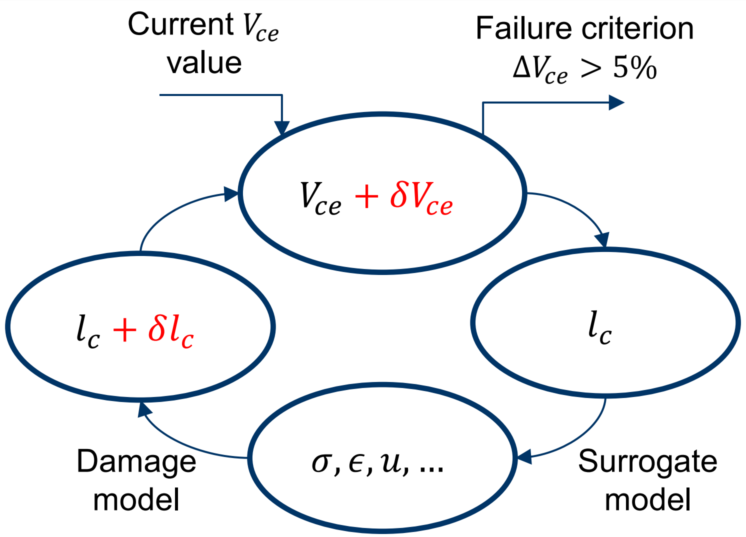
Palmgren-Miner rule:

$$D = \sum_{i=1}^{N} \frac{n_i}{N_i}$$

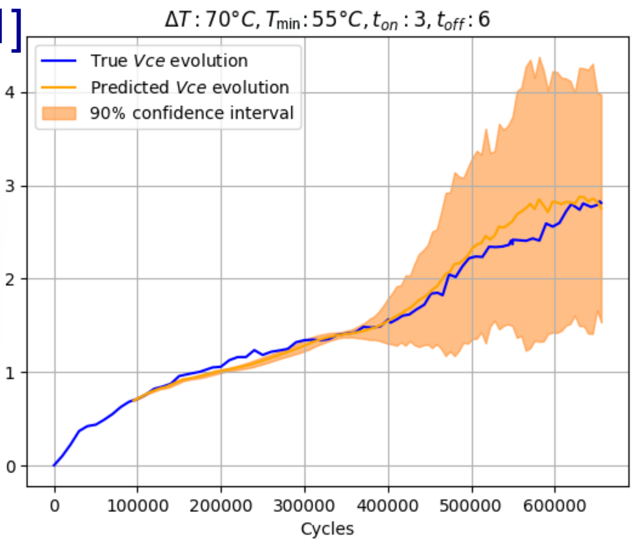
n_i: number of cycles of level i in the loading profile

N_i: number of cycles to failure of level i

D: total damage accumulated



[1]



- **Fast computations** using surrogate models allow for a **recursive** (cycle-by-cycle) **damage estimation**.
- Recursive methods preserve *order information* and *cycle interactions*, as opposed to order-independent tools (Rainflow counting matrix, Palmgren-Miner linear damage rule, etc.).

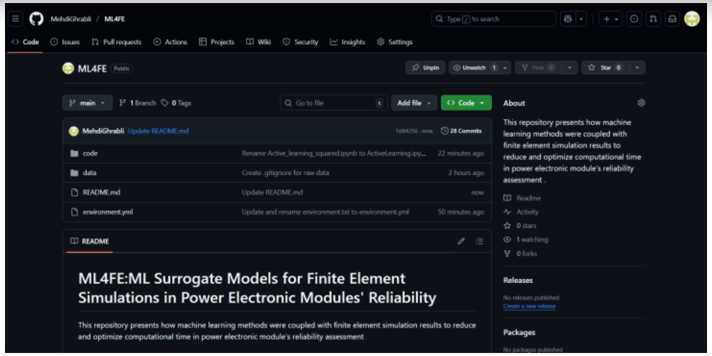
[1] Physics-informed Markov chains for remaining useful life prediction of wire bonds in power electronic modules, Microelectronics Reliability

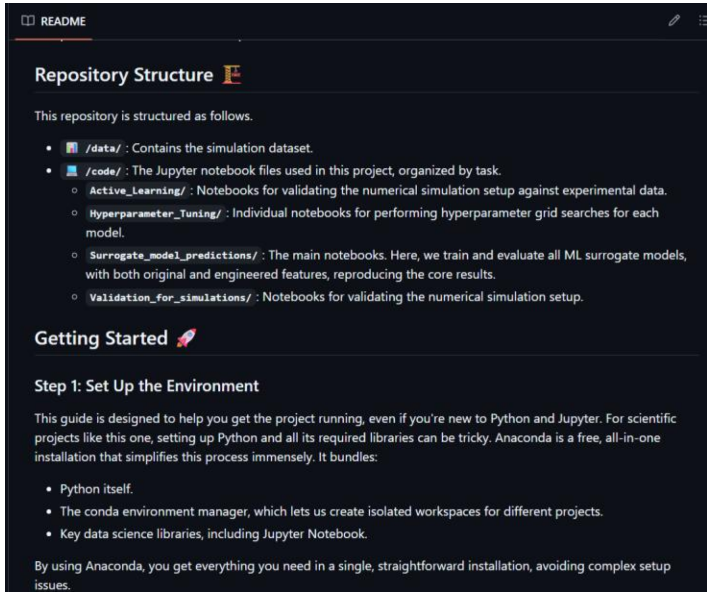
Conclusion

- **Comprehensive analysis** of ML methods as surrogate models for finite element simulations via evaluation metrics, residuals, and learning behavior.
- Advanced methods to optimize the pipeline through **efficient data selection**.
- Using ML Accelerates numerical simulations ($\mathbf{10^6}$ **times faster**) which enables more realistic and accurate remaining useful life estimation frameworks.
- A shift from classical order-independent methods towards more **robust modeling**.

Open Sourcing with GitHub

- The work is hosted on **GitHub**, an open-source platform for collaborative development.
- The repository contains the **code** and the simulated **dataset** to validate the presented findings, utilize the models, or extend this work.
- A comprehensive **documentation** is provided detailing the implementation, setup procedures and the workflow.





ML surrogate models: github.com/MehdiGhrabli/ML4FE

RUL estimation model: github.com/MehdiGhrabli/PIMC4RUL

Thank you for your attention

Contact: mehdi.ghrabli1@ens-paris-saclay.fr

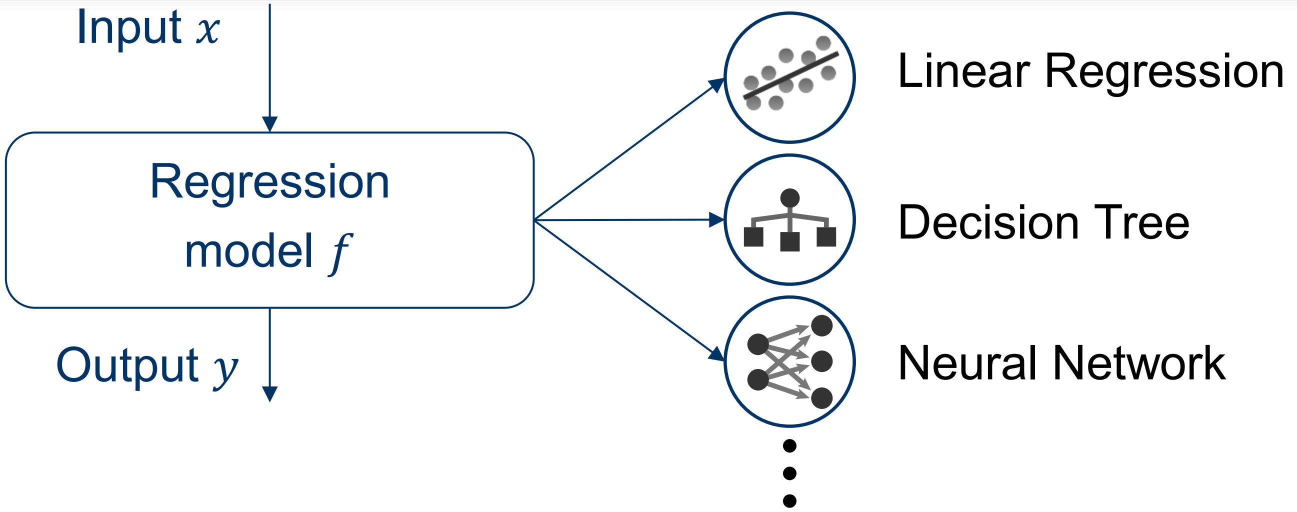
ML surrogate models: github.com/MehdiGhrabli/ML4FE

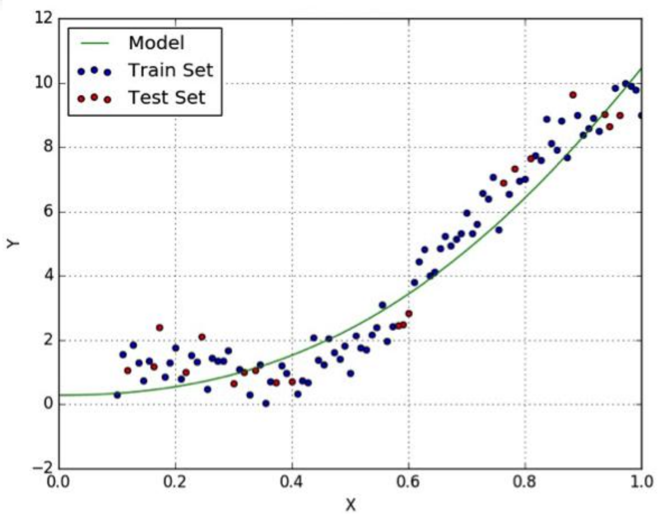
RUL estimation model: github.com/MehdiGhrabli/PIMC4RUL

References

- Jaiswal, A., et al. (2020). Introduction to Linear Regression and Polynomial Regression. Medium.
- Dornic, N., (2019). Elaboration et comparaison de deux modèles de durée de vie des fils d’interconnexion des modules de puissance, l’un basé sur les déformations et l’autre sur les dégradations. Université Paris-Saclay.
- Thoben, M., et al. (2008). From vehicle drive cycle to reliability testing of power modules for hybrid vehicle inverter.
- . Ali, A., et al. (2020). Mechanical behavior of the materials: Fatigue stress-life method of smooth members.
  Segura, T. A., et al. (2020). Lasso & Ridge Regression in 200 Words. Data Science, One Question at a Time.
  Jeline, J., et al. (2025). Random Forest Algorithm. Medium.
- Qaddoura, R., et al. (2022). Temporal prediction of traffic characteristics on real road scenarios in Amman. Journal of Ambient Intelligence and Humanized Computing.
- Logan, S., et al. (2017). Understanding the Structure of Neural Networks. Becoming Human: Artificial Intelligence Magazine.
- Encord, et al. (2025). Mean Square Error (MSE). Encord Computer Vision Glossary.
- Guliyev, J., et al. (2020). Explanation of Gradient Descent Optimization Algorithm on Linear Regression example. Analytics Vidhya.
- Azrulhisham, E. A., et al. (1970). Service loads prediction in the reliability characterization of automotive rear axle system.

Regression in Machine Learning



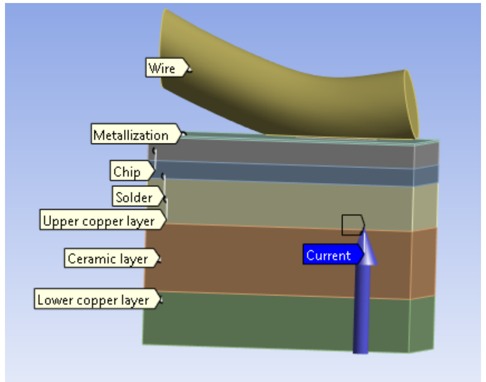


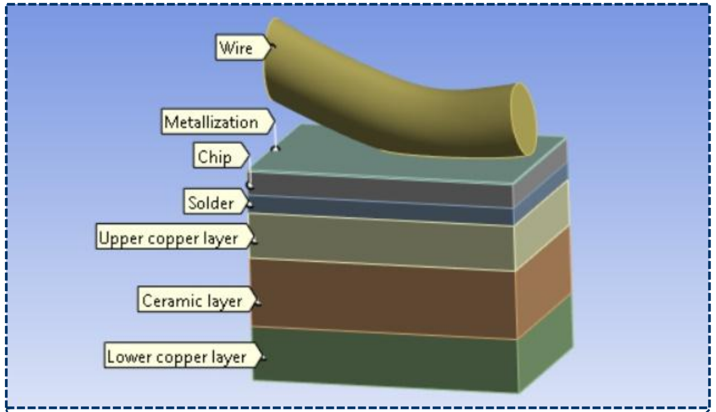
- Goal: Learn a function f that predicts an output y from an input x
- How: use training data (x_{train}, y_{train}) to calibrate f and predict unseen data $y_{test} = f(x_{test})$
- Instead of specifying equations based on physical laws, we let the data define the functional relationship.

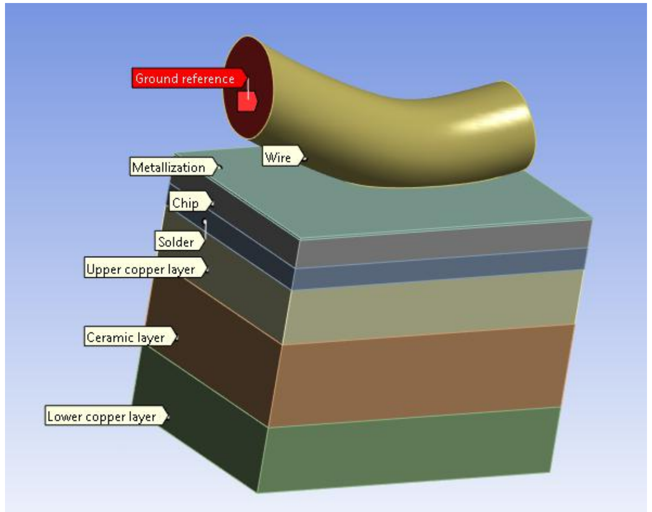
Electrical Boundary Conditions

A voltage $0V$ is imposed on the surface $\delta_W\Omega$ of the wire connecting the module to the rest of the circuit.

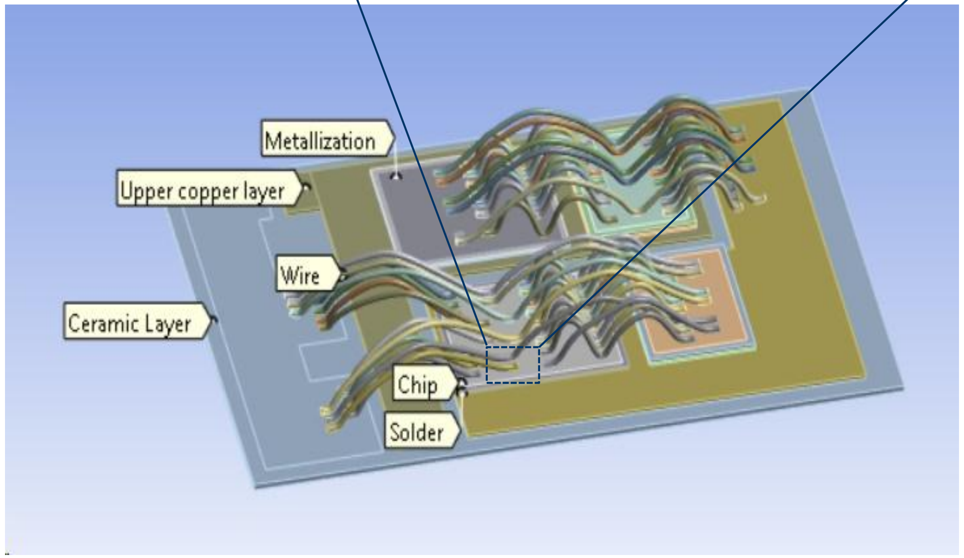
A current I is applied to the lower face of the upper copper layer Ω_{UCB}.





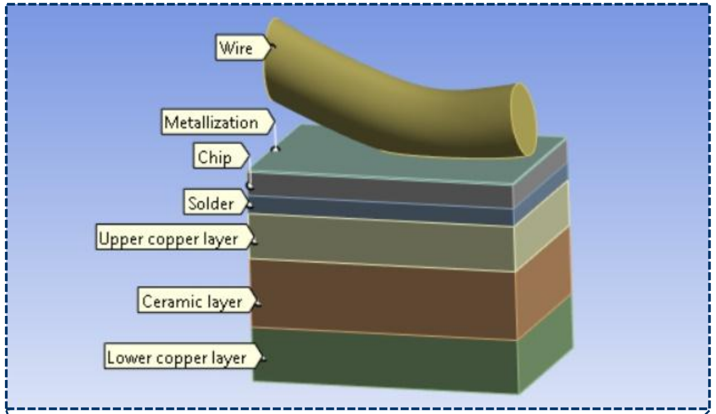


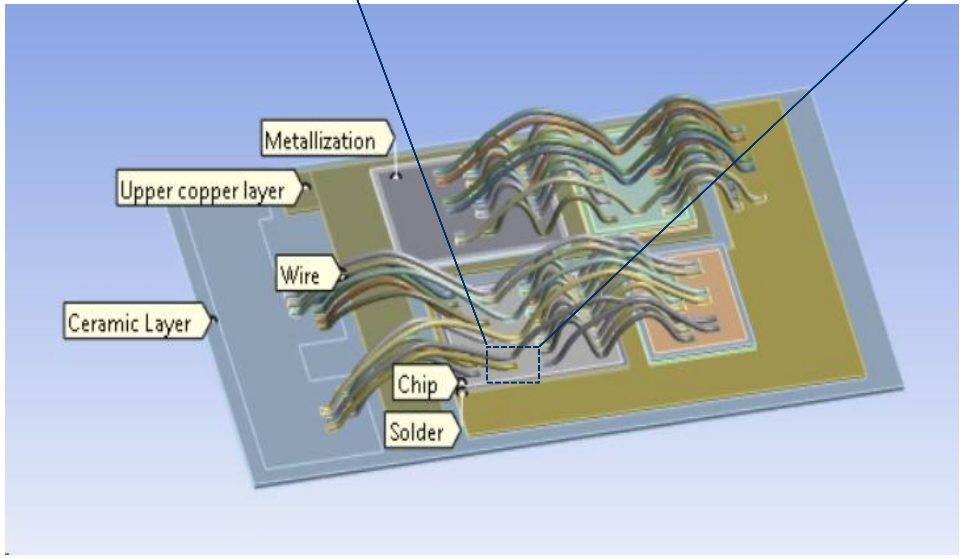
Physics	Electric
Unknowns	Potential V Current density j
Equilibrium	$div(\boldsymbol{j}) = 0$
Constitutive law	$\boldsymbol{j} + \sigma(T)\cdot \boldsymbol{grad}(V) = \boldsymbol{0}$
Boundary conditions	$V = 0$ on $\delta_w\Omega$ $\boldsymbol{j}\cdot\boldsymbol{z} = \frac{I_{ref}}{S_{UCB}}$ on $\delta_{UCB}\Omega$

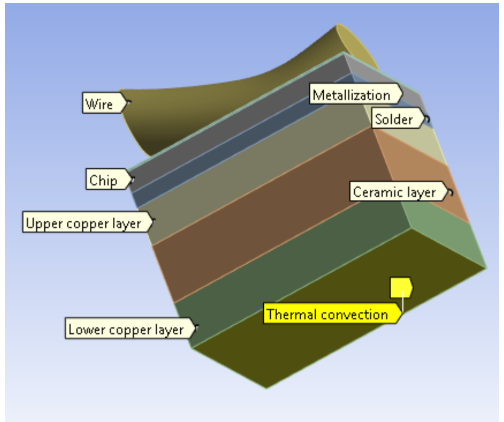


Thermal Boundary Conditions

Thermal boundary conditions on the bottom face of the lower copper layer $\delta_{LCB}\Omega$ to simulate cooling using a coefficient of thermal convection h and a cooling temperature $T_0 = 55\,°C$.





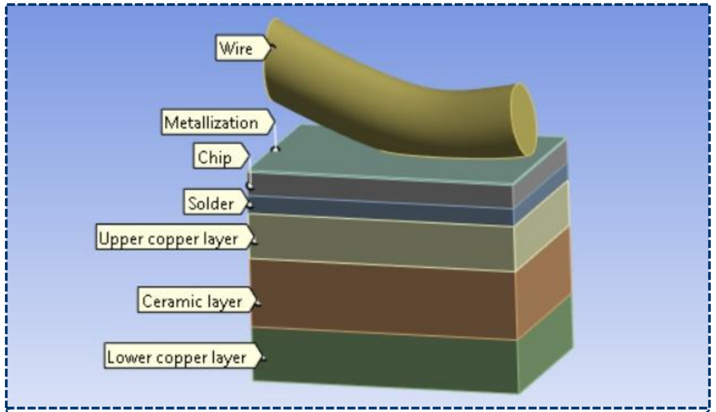


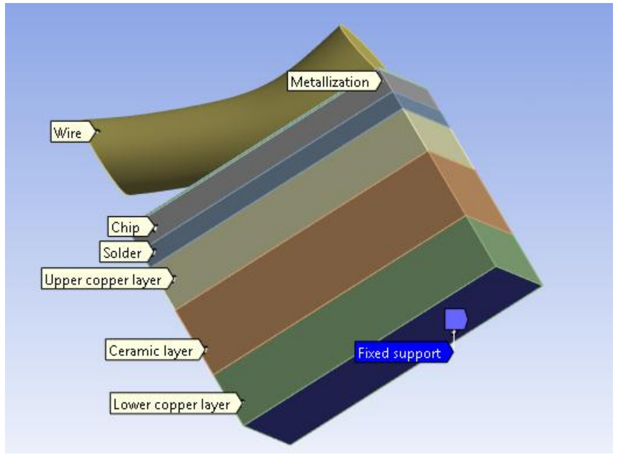
Physics	Thermal
Unknowns	Temperature T Heat flux q
Equilibrium	$-div(\boldsymbol{q}) + p_J = 0$ $p_J = \sigma(T)\|\boldsymbol{grad}(V)\|^2$
Constitutive law	$\boldsymbol{q} + k \cdot \boldsymbol{grad}(T) = 0$
Boundary conditions	$T = T_0$ on $\delta_{LCB}\Omega$ $\boldsymbol{q} \cdot \boldsymbol{z} + hT = 0$ on $\delta_{LCB}\Omega$

Mechanical Boundary Conditions

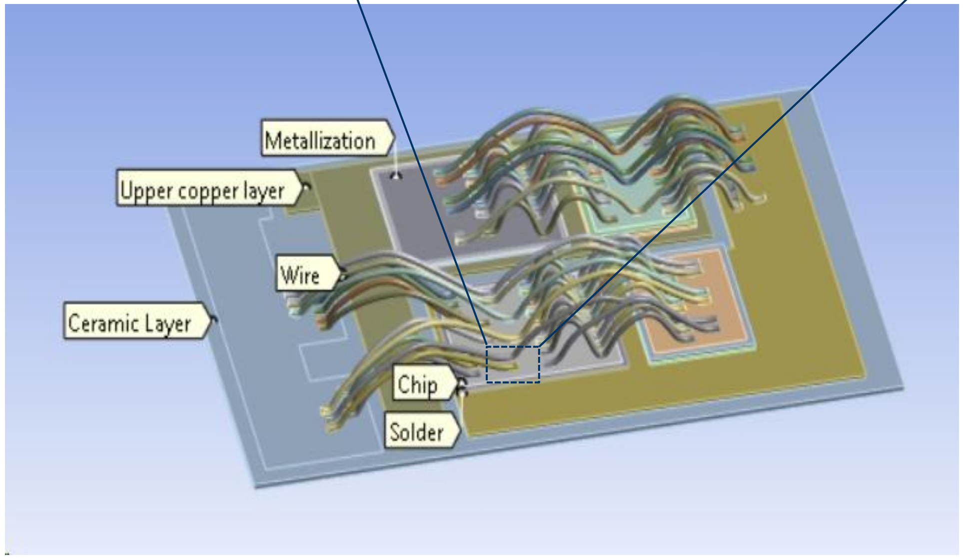
A Fixed support boundary condition is imposed on the bottom face of the lower copper layer$\delta_{LCB}\Omega$ as the module's basis du module de puissance.

Plasticity is modeled in copper Ω_C and aluminum Ω_A using Chaboche kinematic hardening and bilinear kinematic hardening, respectively.



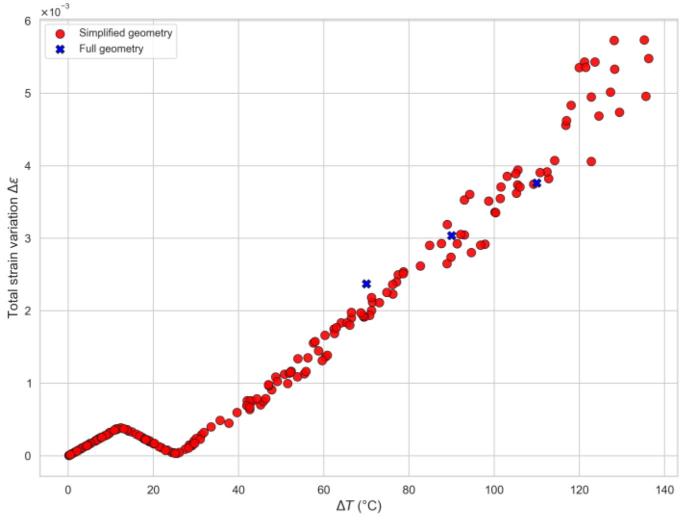


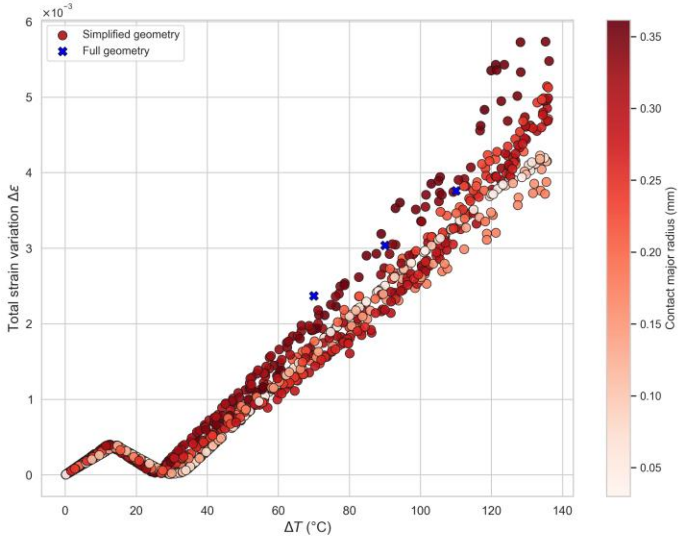
Physics	Mechanical
Unknowns	Displacement u Stress $\underline{\underline{\sigma}}$ Strain $\underline{\underline{\varepsilon}}$
Equilibrium	$\boldsymbol{div}\left(\underline{\underline{\sigma}}\right) = 0$
Constitutive law	$\underline{\underline{\sigma}} = \mathcal{K} : (\underline{\underline{\varepsilon}}^e(\boldsymbol{u}))$ in Ω_L $\underline{\underline{\sigma}} = f_A(\underline{\underline{\dot{\varepsilon}}})$ in Ω_A $\underline{\underline{\sigma}} = f_C(\underline{\underline{\dot{\varepsilon}}})$ in Ω_C $\underline{\underline{\varepsilon}}^e(\boldsymbol{u}) = grad_s(\boldsymbol{u}) -$ $\alpha_{th}(T - T_0)\underline{\underline{I_d}} - \underline{\underline{\varepsilon}}^p(\boldsymbol{u})$
Boundary conditions	$\boldsymbol{u} = \boldsymbol{0}$ on $\delta_{LCB}\Omega$ $\underline{\underline{\sigma}} \cdot \boldsymbol{n} = \boldsymbol{0}$ on $\delta\Omega \backslash \delta_{LCB}\Omega$



Validation of Simulations

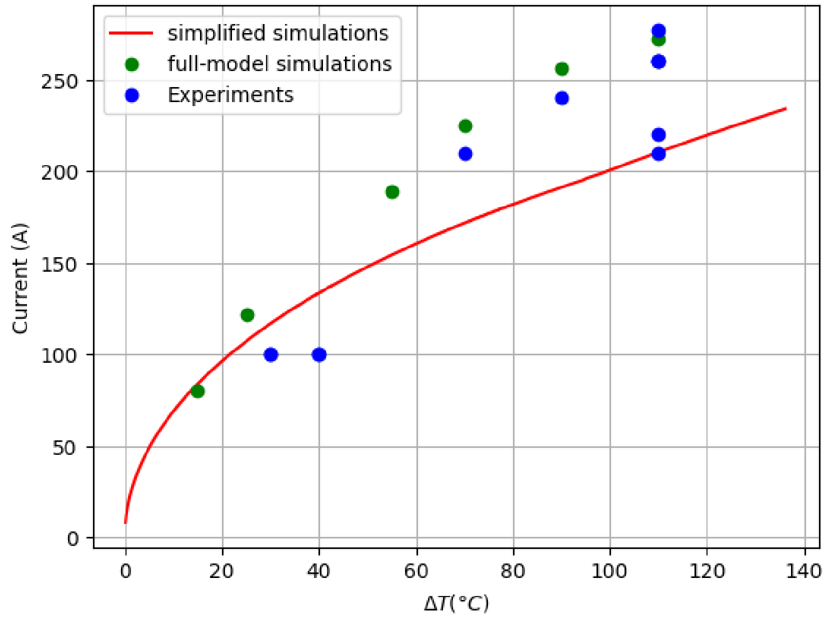
- The MMD (Maximum Mean Discrepancy) statistical test was conducted to validate our simulation results to reference data
- Reference data are taken from a previous work where simulations were performed on a *healthy module* with a *complete geometry*
- The test against the three reference points yielded a p-value of 0.28, well above the standard significance level (0.05): no statistically-significant difference between our model and the reference.
- As the reference data were measured on a healthy module (0% crack length), we refined the test for more accurate comparison
- Using simulation data presenting a crack length smaller than 15%, 10% and 5% of the total length, we recorded p-values of 0.306, 0.3231 and 0.2961 respectively
- This test is limited by the small size of the reference data.





Choice of Loading Current

- The simulation current I_{sim} was chosen to generate a predefined range of temperature variations ($\left[I_{sim}^{(1)}, I_{sim}^{(2)}\right] \Leftrightarrow [\Delta T^{(1)}, \Delta T^{(2)}]$). This approach ensures the resulting data effectively spans the necessary thermal conditions for training our surrogate models.
- Due to the model's reduced geometry, a direct current comparison is misleading. We established an equivalent full-model current $I_{eq} = \alpha I_{sim}$ by scaling simulation current with the geometric reduction factor α.
- On average, our equivalent current I_{eq} is slightly lower than reference values. This is an expected outcome attributed to two main factors:
  - **Boundary Conditions:** A higher cooling base temperature was used in our simulations, which requires less self-heating (and thus lower current) to achieve the same ΔT.
  - **Geometric range:** The model focuses exclusively on the primary hot-spot (wire-metallization contact), neglecting cooler peripheral regions. This concentrates heat, reducing the power required to reach the target temperature compared to a full model where heat dissipates over a larger area.

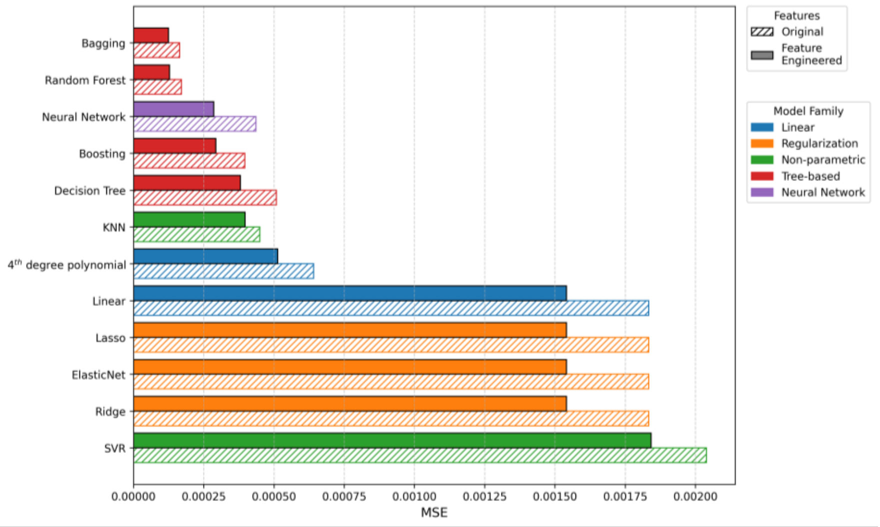


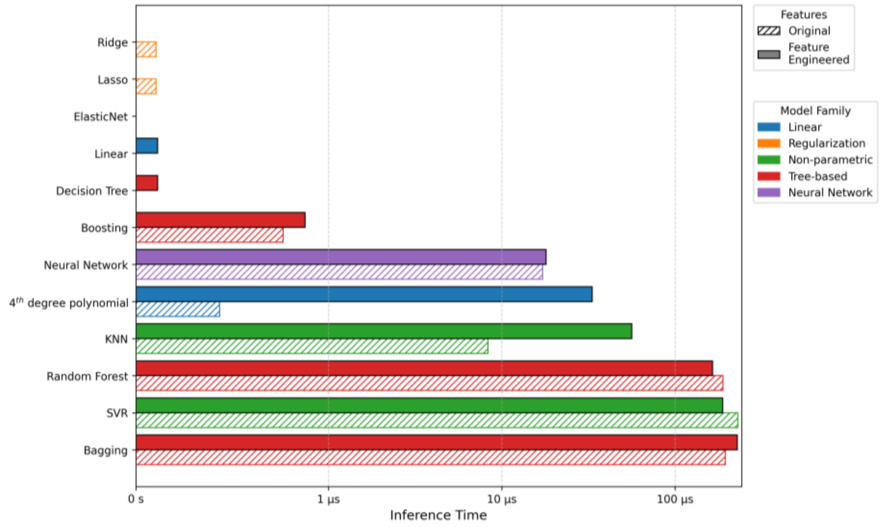
Feature Engineering

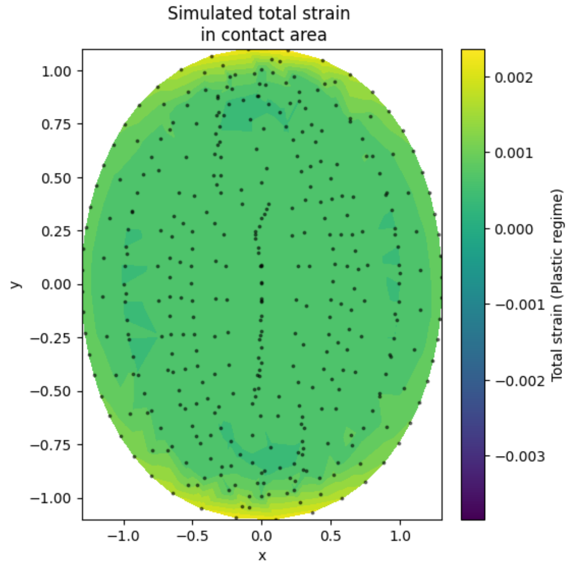
- Inspired by the geometry of the contact, quadratic features (e.g.,x^2, y^2, xy) were engineered from the initial inputs to capture the quadratic aspect of the problem, improve accuracy, and accelerate convergence.

$$\frac{(x - x_1)^2}{a^2} + \frac{(y - y_1)^2}{b^2} = 1$$

- While adding these higher-order terms significantly improves accuracy, it increases the dimensionality of the feature space, creating a crucial trade-off between predictive performance and computational cost.

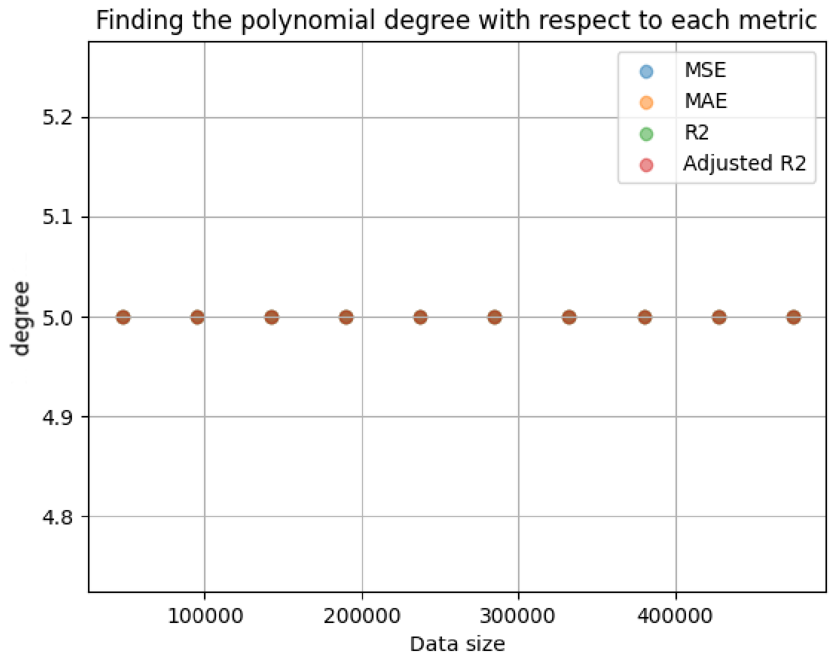






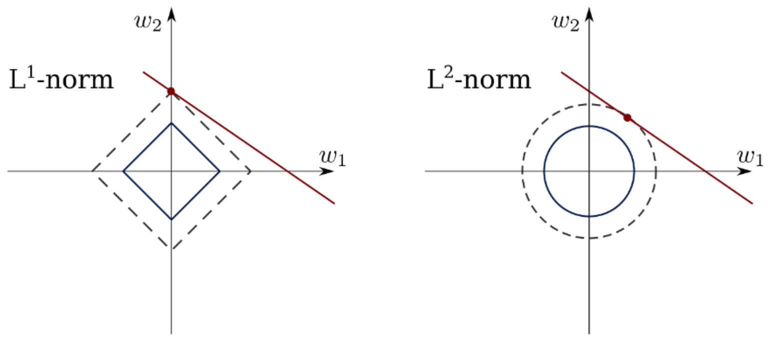
Hyperparameter Selection: Polynomial Regression

- The optimal polynomial model's degree was chosen via *grid search*
  - We calculate performance metrics (MSE, MAE, R^2, and adjusted R^2) corresponding to models with degrees 2,3,4, and 5
- Multiple data sizes were chosen for robustness
- The *optimal* polynomial model is the *degree 5* polynomial (highest degree in the grid)
- Higher degree polynomials are more *expressive*, but risk *overfitting* in some cases
- We use *degree 4* polynomials as a *middle-ground* solution



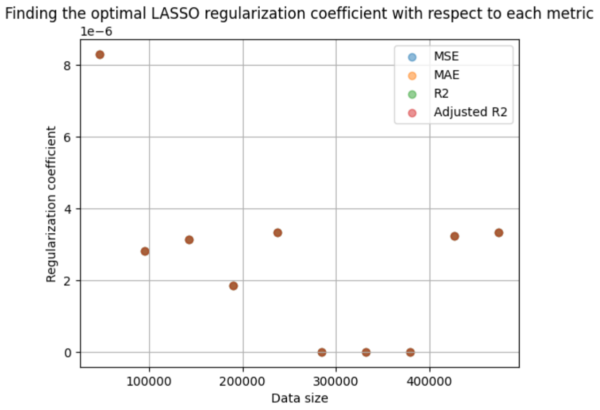
Hyperparameter Selection: Regularization

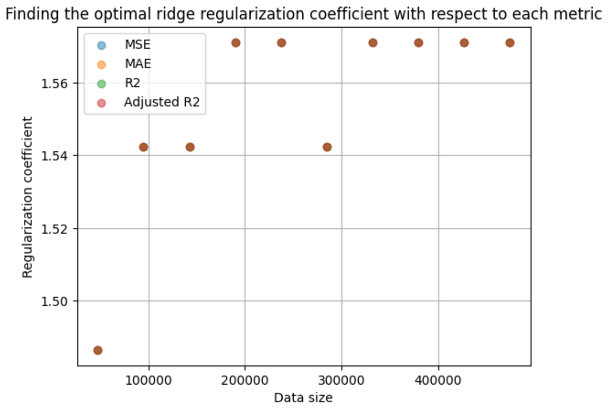
- LASSO: $\min_{\beta}\{\frac{1}{N}||y - X\beta||_2^{\ 2} + \lambda_l||\beta||_1\}$
  - λ_l: penalty term, as it increases, more coefficients are driven toward zero, promoting sparsity
- Ridge regression: $\min_{\beta}\{\frac{1}{N}||y - X\beta||_2^2 + \lambda_r||\beta||_2^2\}$
  - λ_r: penalty term, as it increases the model becomes simpler, with smaller weights
- ElasticNet: $\min_{\beta}\{\frac{1}{N}||y - X\beta||_2^2 + \lambda_1||\beta||_1 + \lambda_2||\beta||_2^2\}$
  - $\min_{\beta}\{\frac{1}{N}||y - X\beta||_2^2 + \lambda(\alpha||\beta||_1 + (1-\alpha)||\beta||_2^2)\}$
  - λ: Controls the overall strength of the regularization (Ridge and Lasso)
  - α: Controls the mix between Lasso and Ridge penalties

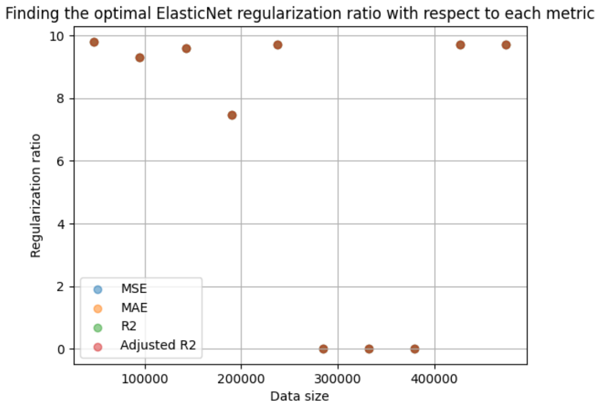


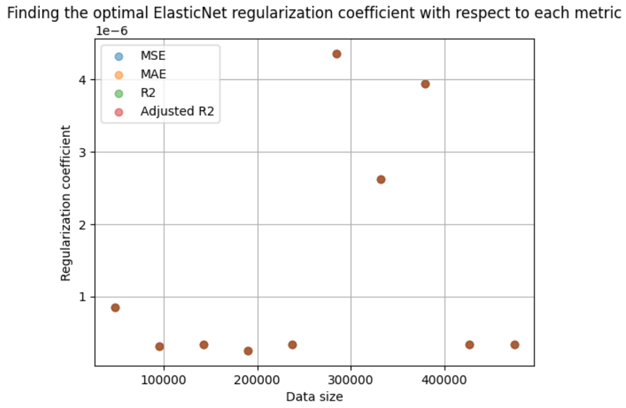
Hyperparameter Selection: Regularization

- Hyperparameters for each regularization model are chosen using a grid search
- Optimal penalty term for <u>*LASSO*</u> $\approx 3 \times 10^{-6}$, indicating that regularization has a <u>*negative effect*</u> on overall accuracy
  - Removing one or more input value is detrimental to the model
- <u>*Ridge*</u> regression simplifies the model by reducing its weights, <u>*yet no substantial increase in accuracy*</u> is observed compared to the original model
- Analyzing optimal hyperparameters for the <u>*ElasticNet*</u> model confirms <u>*Ridge*</u> regression's <u>*superiority*</u> over LASSO for this application



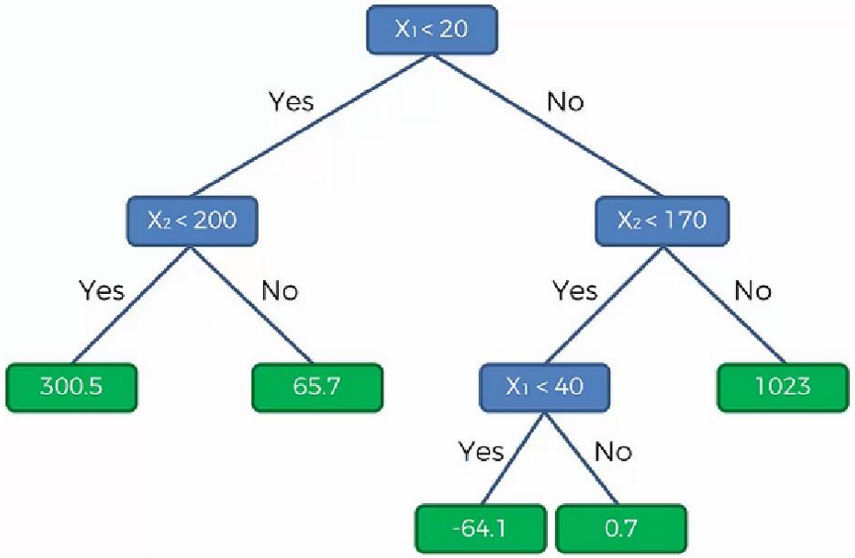


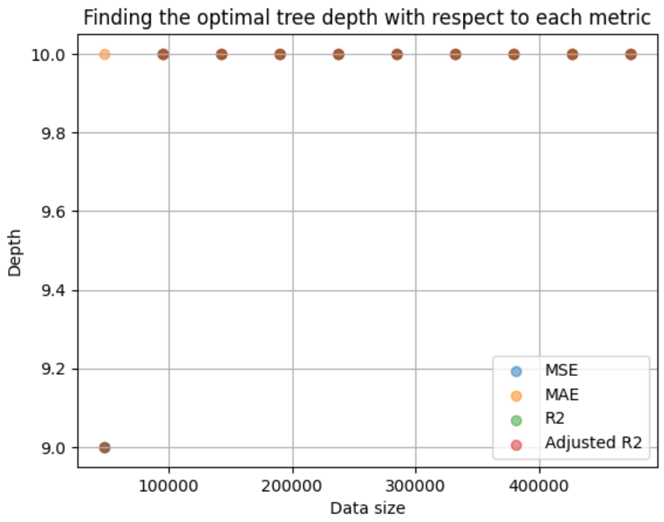




Hyperparameter Selection: Decision Trees

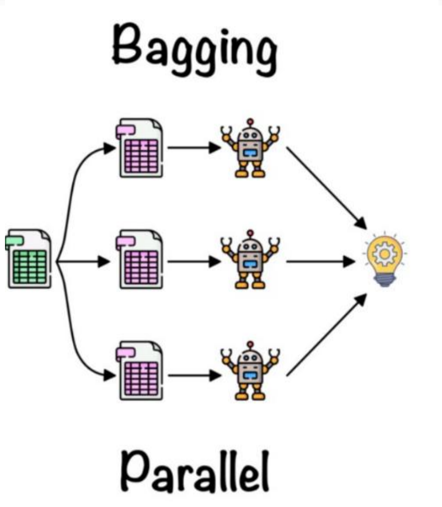
- Decision Trees predict a target value by learning simple decision rules inferred from the data features.
- They recursively split the input space into regions based on a single feature $x_1, x_2, \ldots$
- The *maximum depth* of the tree controls its *complexity*:
  - A shallow tree underfits (high bias)
  - A deep tree may overfit (high variance)
- Maximum depth is chosen using grid search, with depths varying from 5 to 10
- The optimal max size corresponds to the maximum value in the grid since our data does not present a source of noise that may cause overfitting due to the deterministic nature of the simulations

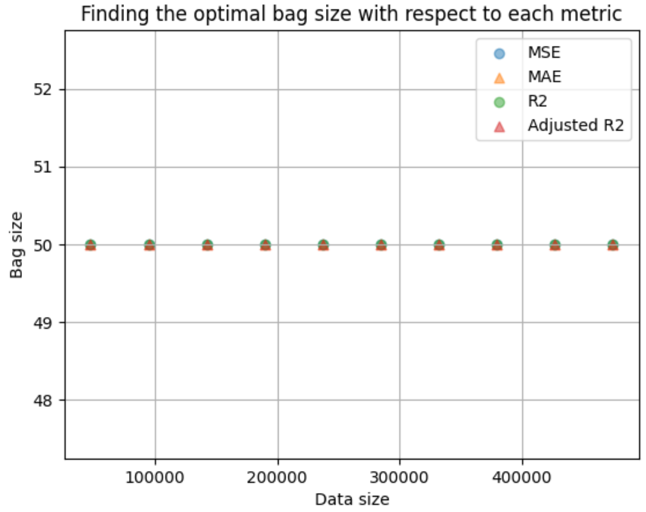




Hyperparameter Selection: Bagging

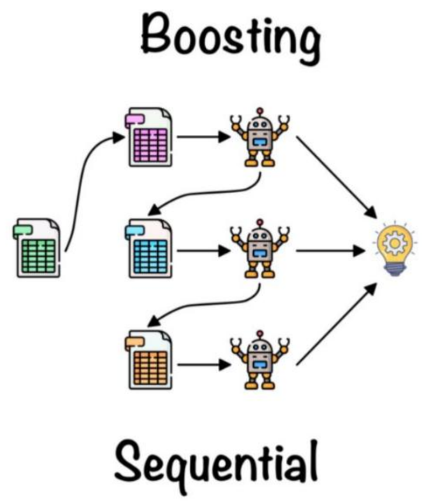
- Bagging (Bootstrap Aggregating) is an ensemble technique that combines multiple Decision Trees trained on different random subsets of the data to reduce variance and improve predictive performance.
- Each tree is trained on a bootstrap sample (random subset of the dataset, sampled with replacement)
- Final prediction is made by averaging the outputs of all trees
- The *number of Bootstrap Aggregated trees* (Bag size) is chosen using grid search on the set $\{10,20,30,40,50\}$
- The optimal Bag size corresponds to the maximum value in the grid *50* as increasing the size leads to lower variance (no risk of overfitting)
- *No restrictions* were made on the *maximum depth* of each tree

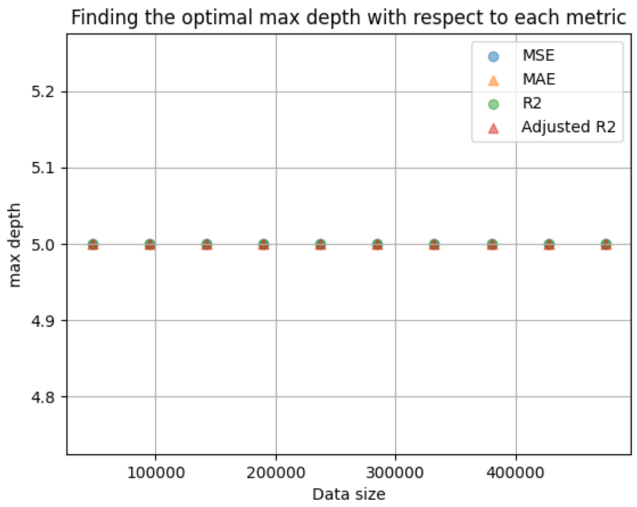




Hyperparameter Selection: Boosting

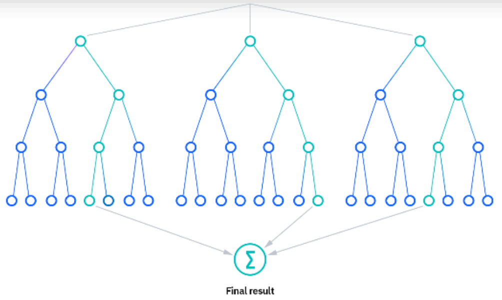
- Boosting builds an ensemble of *weak learners* (shallow Decision Trees) in a sequential manner, where each tree tries to correct the mistakes made by the previous ones.
- Each new tree is trained on the residuals (errors) of the ensemble so far.
- The predictions are weighted sums of the individual learners.
- The total number of trees is set to 100, with a 0.1 learning rate, which determines the contribution of each tree
- The *depth* of the trees is chosen using grid search for values in $\{1,2,3,4,5\}$
- The optimal depth is equal to 5 the maximum value on the grid
  - Shallow trees are used to learn simple patterns and correct only coarse errors. Shallow trees are preferred to avoid overfitting on noisy data
  - Deep trees can capture fine-grained residual patterns quickly and model complex interactions in the data

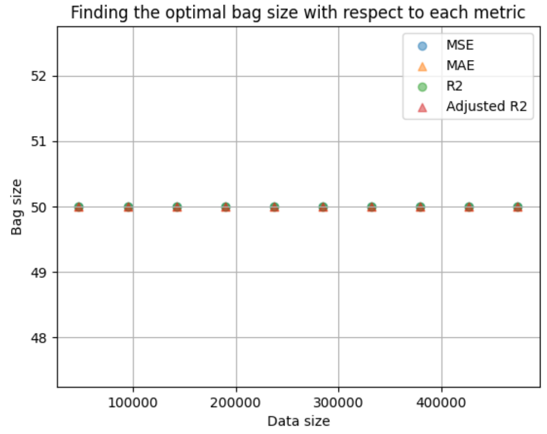


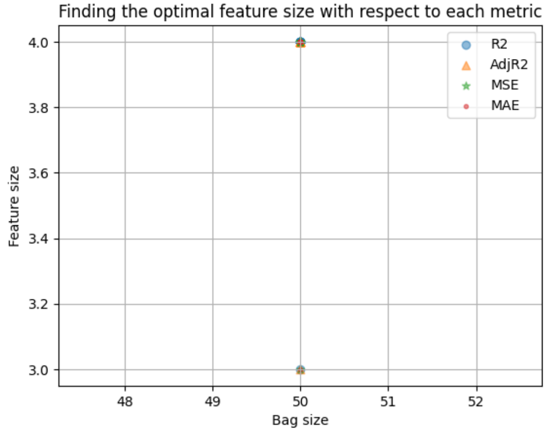


Hyperparameter Selection: Random Forest

- A Random Forest is an ensemble learning method that builds many Decision Trees and averages their predictions (similarly to Bagging)
- At each split in each tree, a random subset of features is considered (not all), which decorrelates trees and increases robustness.
- The size of the random subset of features is chosen using grid search for values varying from 1 (minimum) to 4 (maximum)
  - The optimal subset size is equal to 4, which corresponds to using all features
  - This means that decorrelation does not increase accuracy, due to the low risk of overfitting (deterministic data)
  - Random Forests with maximal subset size is equivalent to bagging
- Similarly to Bagging, the optimal number of trees used is chosen using grid search, which similar results (the higher the better)

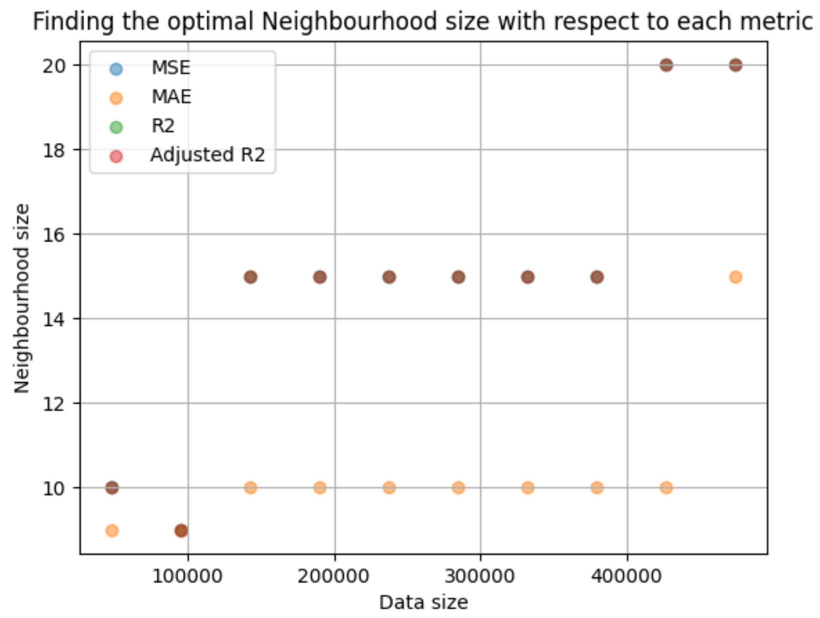






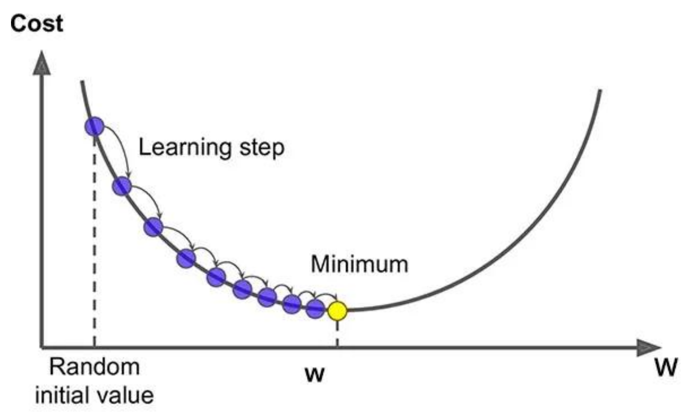
Hyperparameter Selection: Non-parametric

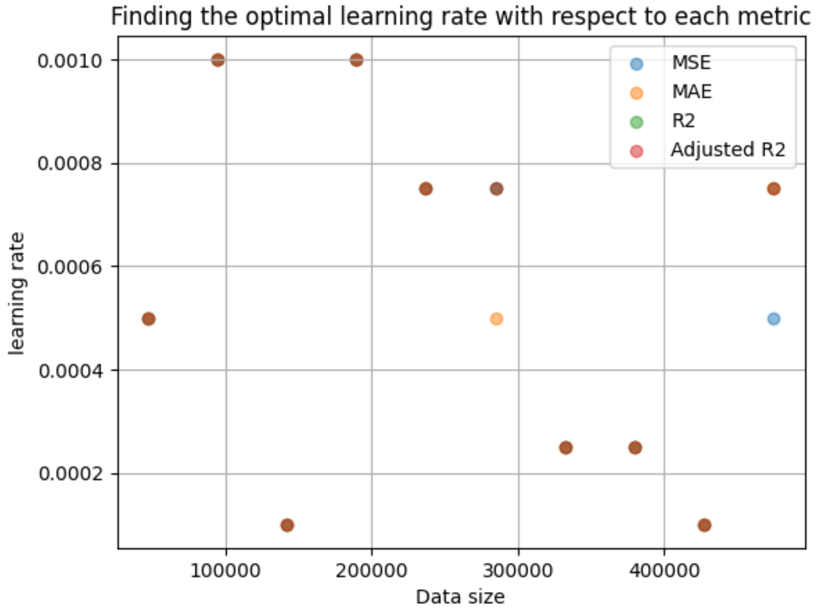
- K-NN regressors predict an output y from an input x by *aggregating* the outputs of the *nearest K points* to x in the training data, hence the name
- K is chosen via grid search
  - Models with *low* K values tend to *overfit*, presenting a small bias and a high variance
  - Models with *high* K values tend to *underfit*, as it smooths out the output by aggregating irrelevant (far neighbors) values
- We use a *neighborhood size of 15* following the grid search's results



Hyperparameter Selection: Neural Networks

- Hyperparameters chosen for the neural network based on empirical tuning
  - Architecture: 2 Hidden layers of size 50
  - Activation: ReLU
  - Optimizer: Adam
  - Loss function: Mean squared error (MSE)
- Learning rate selected via grid search
- The learning rate controls the step size during training, determining the magnitude of weight updates at each iteration



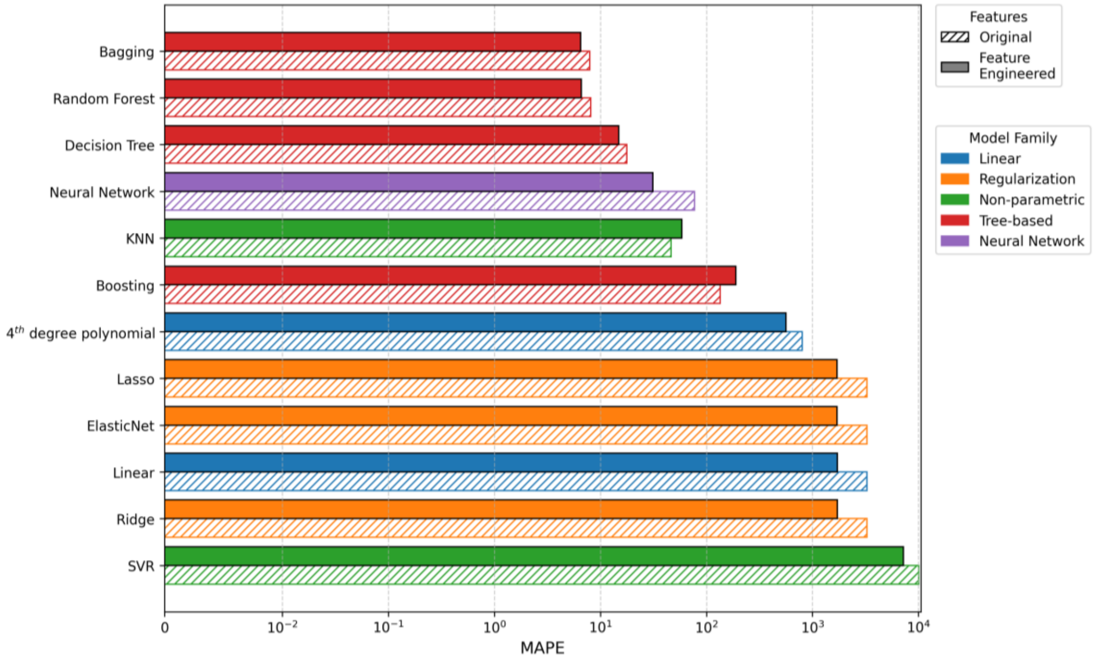


Percentage Errors

"On average, what is the percentage error of the predictions?"

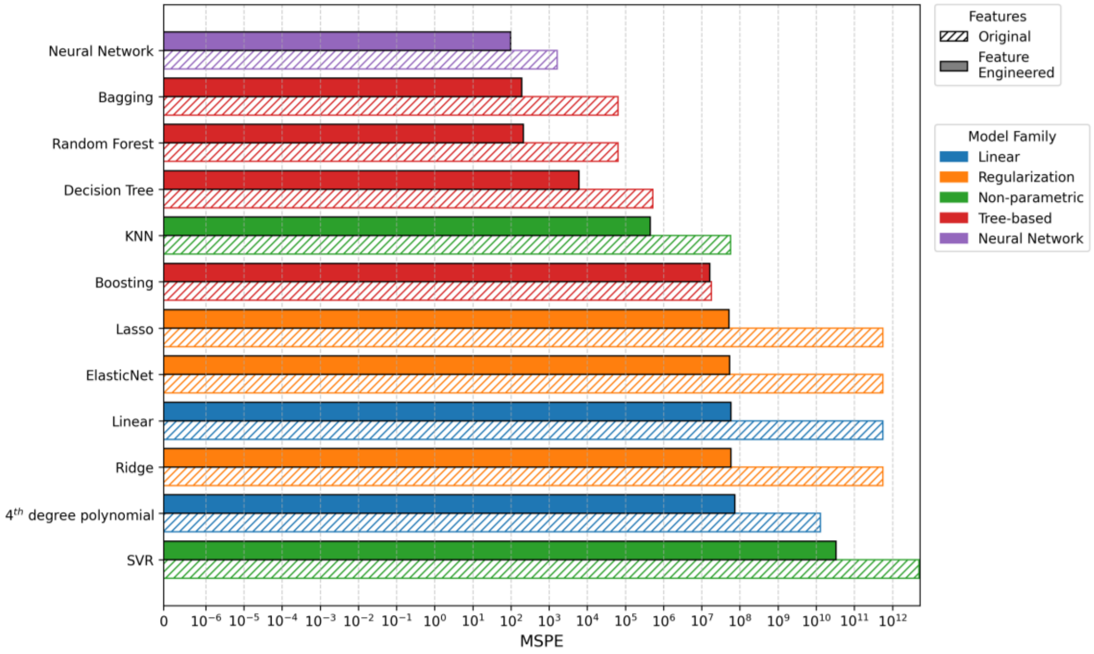
- Mean Absolute Percentage Error (MAPE) measures the average absolute difference between the predicted $\hat{\varepsilon}$ and actual values ε.

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \frac{|\varepsilon_i - \hat{\varepsilon}_i|}{\varepsilon_i}$$



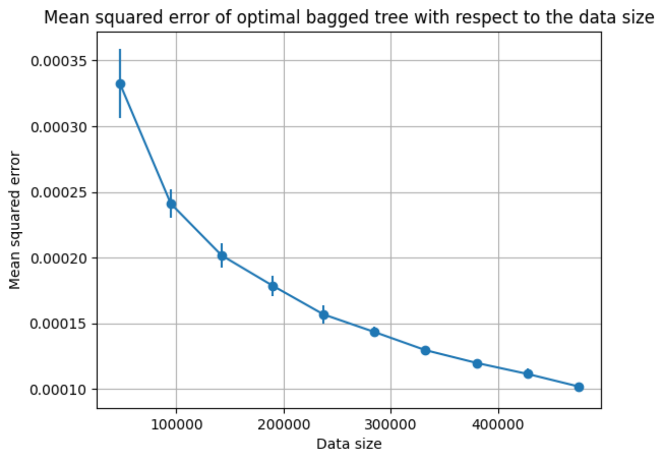
- Mean Squared Percentage Error (MSPE) measures the average squared difference between the predicted $\hat{\varepsilon}$ and actual values ε.

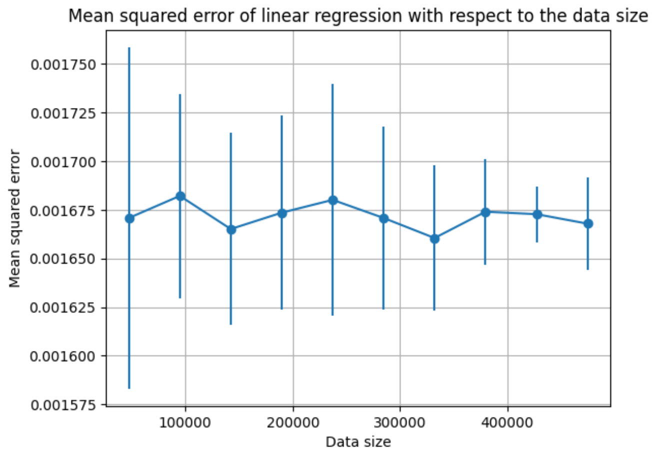
$$MSPE = \frac{100\%}{n} \sum_{i=1}^{n} \left(\frac{\varepsilon_i - \hat{\varepsilon}_i}{\varepsilon_i}\right)^2$$



Analysis: Learning Behavior and Consistency

- Statistical consistency is the capability of a model to become increasingly accurate and reliable with more data
  - A consistent model has a decreasing error with respect to data size
- Analyzing learning behavior helps decide data collection cost vs. model benefit.
- Tree-based models (e.g., bagging) are consistent, presenting a clear decreasing error with respect to data size, as opposed to linear models



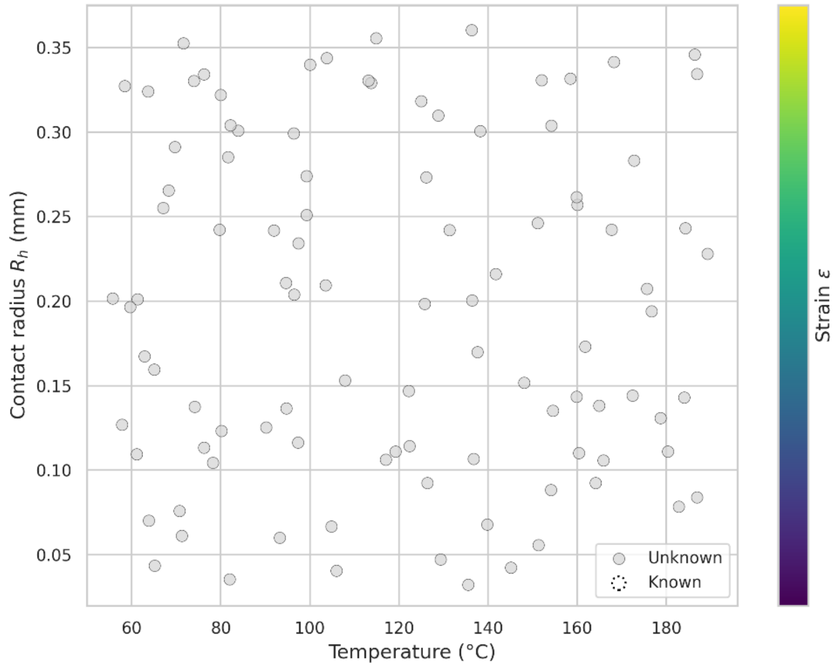


Optimizing the Simulation Protocol: Active learning

- How to choose which simulations are most relevant to use?
- Classical random sampling is not optimal: costly when data generation is slow/expensive
- Use the model's uncertainty:
  - If a model is uncertain in a candidate simulation's outcome, then carrying out the experiment and training on it would remove the model's uncertainty, making it more robust for similar unseen scenarios.
  - The higher the model's uncertainty, the more relevant the simulation.
  - Time to calculate the model's uncertainty of one simulation = inference time (μs)

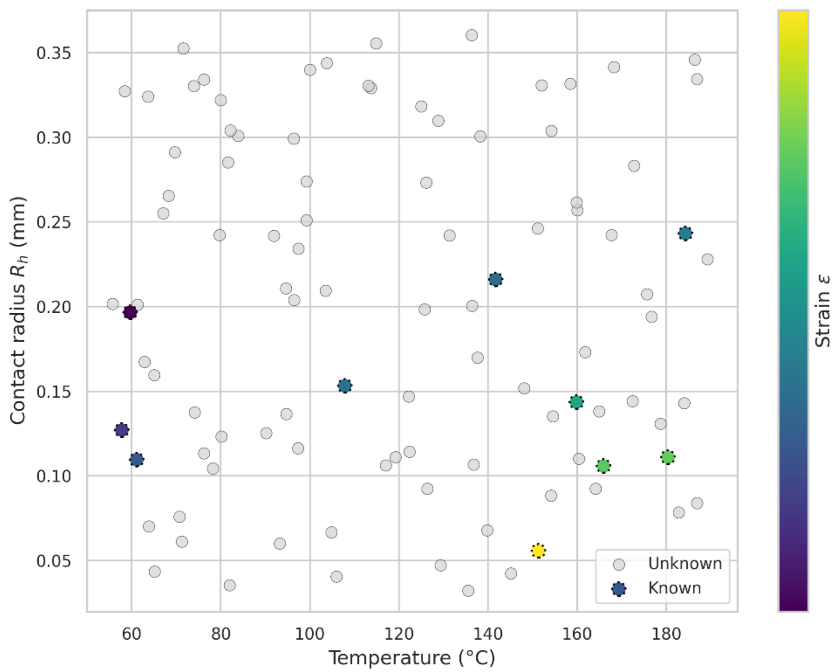
Optimizing the Simulation Protocol: Active learning

- Goal: find the best m simulations to use for training among n possible simulations
- As the model learns, the model updates its uncertainty.
- Use a batched approach: select B simulations at a time (Top B batches by uncertainty value)
- After each batch i, update the uncertainties
- Computation cost for uncertainty update at step i:$(n - iB) \times T_{inference}$



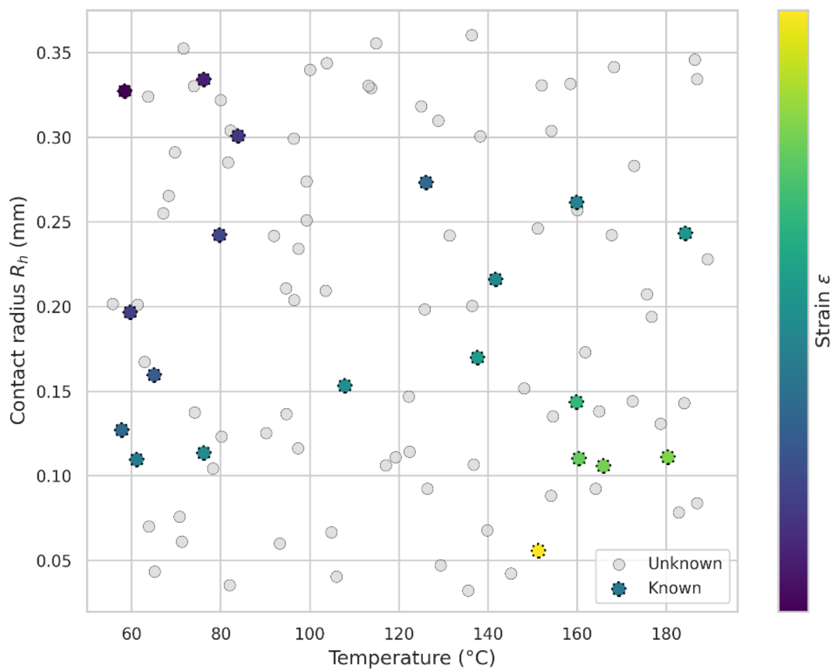
Optimizing the Simulation Protocol: Active learning

- Goal: find the best m simulations to use for training among n possible simulations
- As the model learns, the model updates its uncertainty.
- Use a batched approach: select B simulations at a time (Top B batches by uncertainty value)
- After each batch i, update the uncertainties
- Computation cost for uncertainty update at step i:$(n - iB) \times T_{inference}$



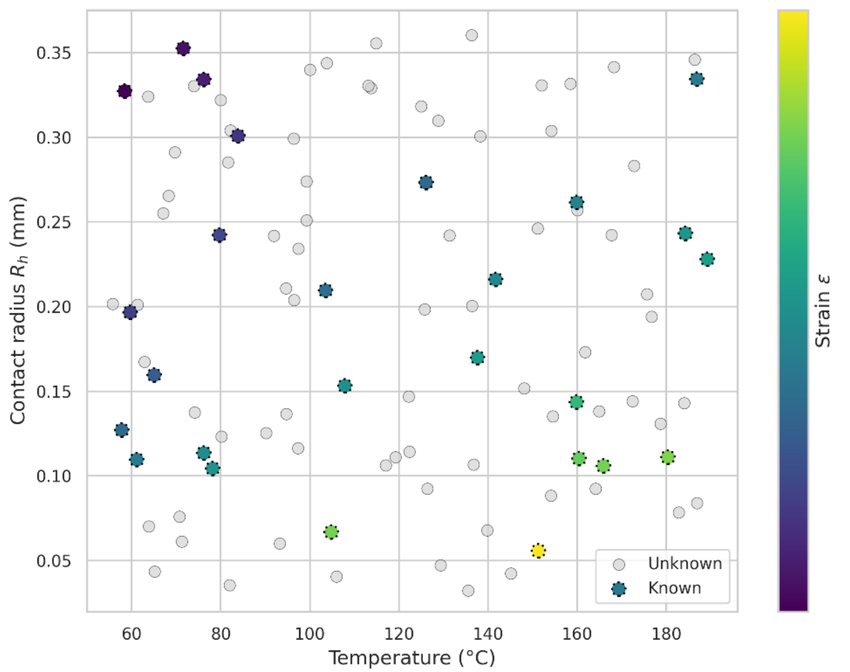
Optimizing the Simulation Protocol: Active learning

- Goal: find the best m simulations to use for training among n possible simulations
- As the model learns, the model updates its uncertainty.
- Use a batched approach: select B simulations at a time (Top B batches by uncertainty value)
- After each batch i, update the uncertainties
- Computation cost for uncertainty update at step i:$(n - iB) \times T_{inference}$

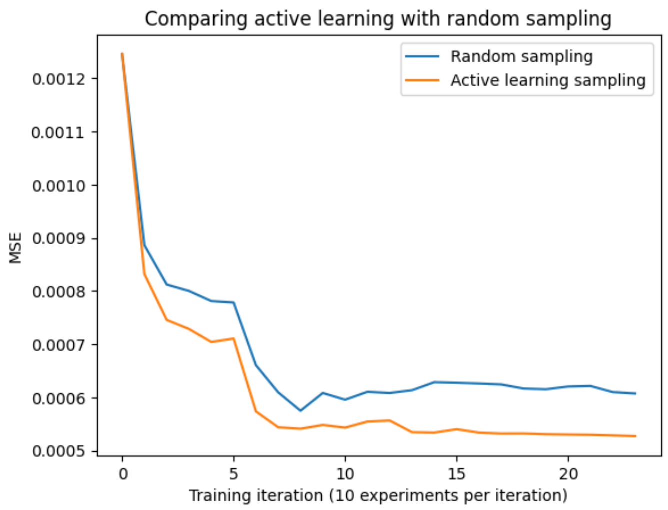


Optimizing the Simulation Protocol: Active learning

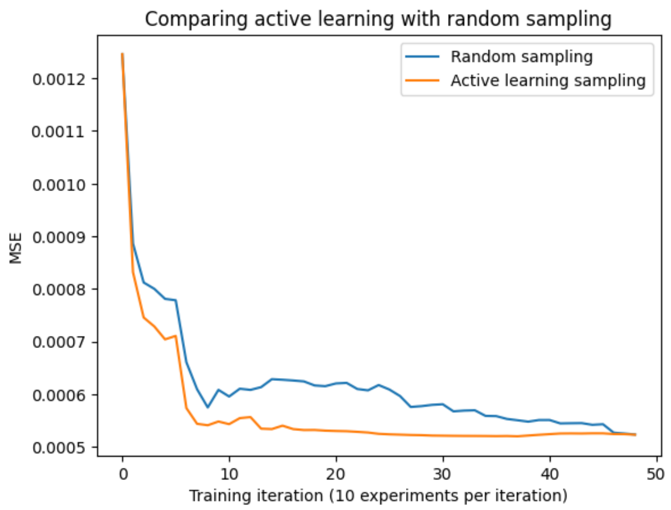
- Goal: find the best m simulations to use for training among n possible simulations
- As the model learns, the model updates its uncertainty.
- Use a batched approach: select B simulations at a time (Top B batches by uncertainty value)
- After each batch i, update the uncertainties
- Computation cost for uncertainty update at step i: $(n - iB) \times T_{inference}$



Optimizing the Simulation Protocol: Active learning



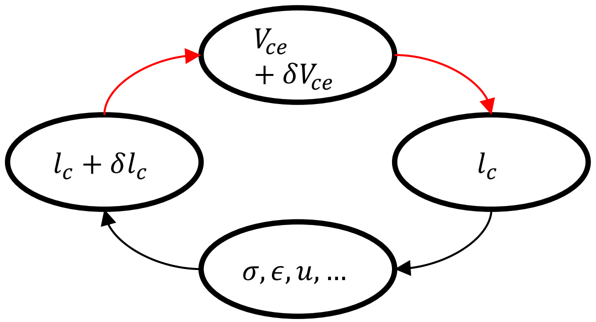
- 50 burn in experiments to initialize the model
- 10 iterations per batch
- 500 candidate simulations

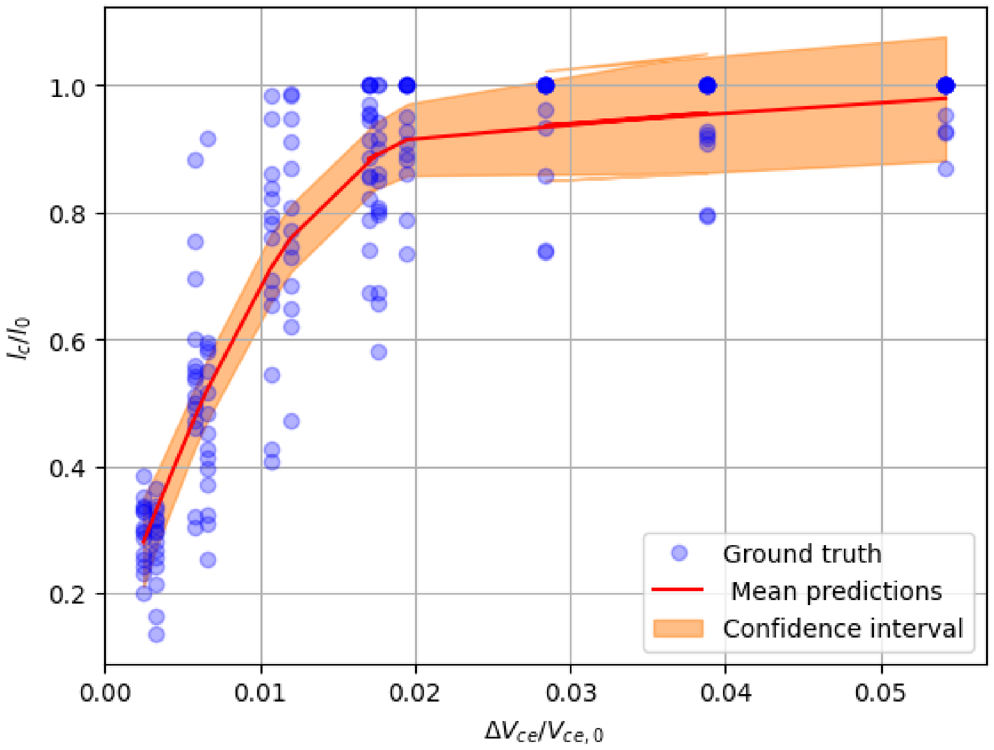


- Using 50% of the pool (left plot: 250 simulations ⇔25 iterations) active learning outperforms classical random sampling
- Using 100% of the pool (right), both models converge to the same accuracy as training data is the same, but convergence is faster with active learning

Autoregressive RUL Estimation Model

Estimating crack length using gaussian process regression (GPR)





Using a kernel function $k(.,.)$, we calculate three matrices: training data covariance $\mathcal{K}$, test data covariance $\mathcal{K}_{**}$, train-test covariance $\mathcal{K}_*$

$$\mathcal{K} = \begin{bmatrix} k(x^1, x^1) & \cdots & k(x^1, x^n) \\ \vdots & \ddots & \vdots \\ k(x^n, x^1) & \cdots & k(x^n, x^n) \end{bmatrix}$$

$$\mathcal{K}_* = \begin{bmatrix} k(x^1, x_*^1) & \cdots & k(x^1, x_*^m) \\ \vdots & \ddots & \vdots \\ k(x^n, x_*^1) & \cdots & k(x^n, x_*^m) \end{bmatrix}$$

$$\mathcal{K}_{**} = \begin{bmatrix} k(x_*^1, x_*^1) & \cdots & k(x_*^1, x_*^m) \\ \vdots & \ddots & \vdots \\ k(x_*^m, x_*^1) & \cdots & k(x_*^m, x_*^m) \end{bmatrix}$$

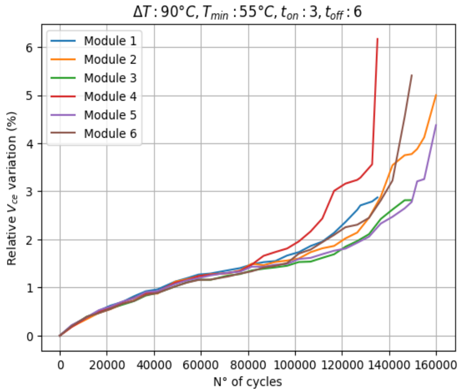
Use the calculated matrices to predict results based on the GPR assumption

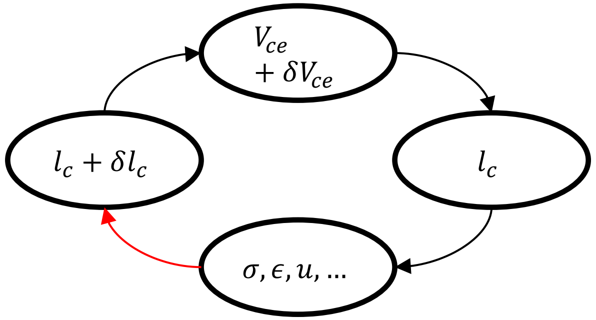
$L_{c_*} = (l_{c_*}^1, \ldots, l_{c_*}^m)$: $L_{c_*} \sim \mathcal{N}(\mu_*, \sigma_*^2)$

$$\mu_* = \mathcal{K}_*^T \left(\mathcal{K} + \sigma_{noise}^2 I\right)^{-1} L_C \qquad \sigma_*^2 = \mathcal{K}_{**} - \mathcal{K}_*^T \left(\mathcal{K} + \sigma_{noise}^2 I\right)^{-1} \mathcal{K}_*$$

Autoregressive RUL Estimation Model

Modeling degradation dynamics using health indicator transitions

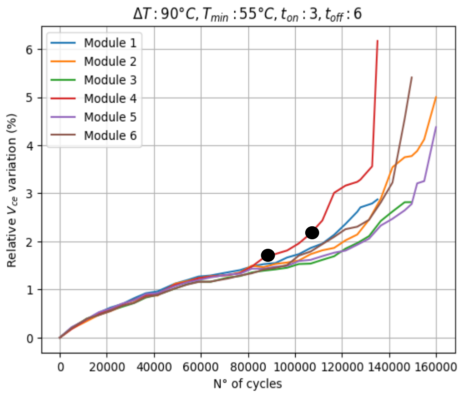


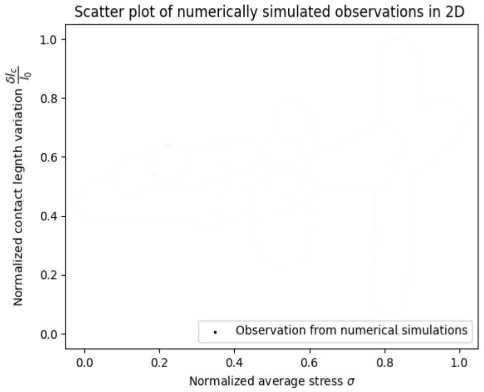


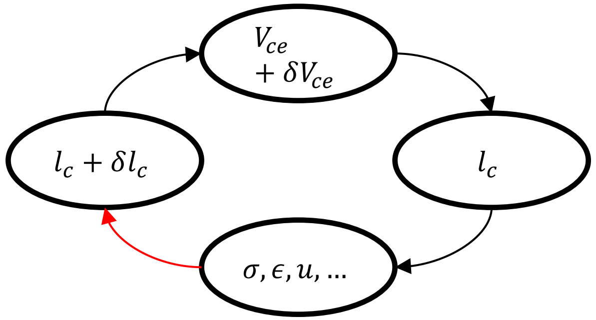
[1] Physics-informed Markov chains for remaining useful life prediction of wire bonds in power electronic modules, Microelectronics Reliability

Autoregressive RUL Estimation Model

Modeling degradation dynamics using health indicator transitions



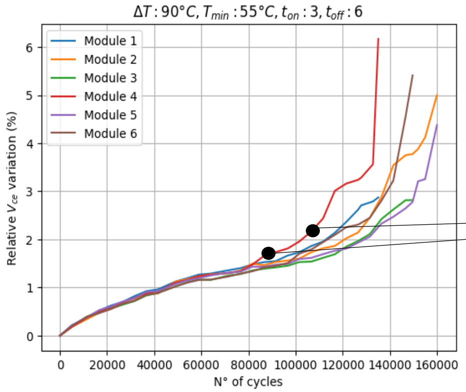


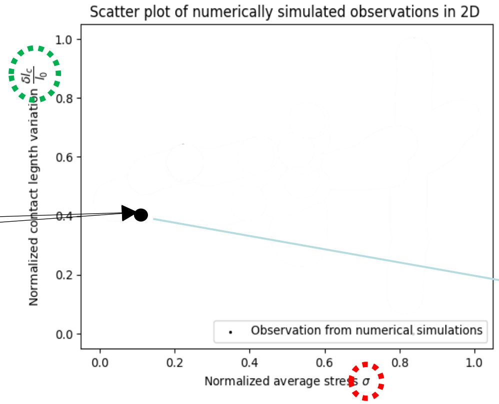


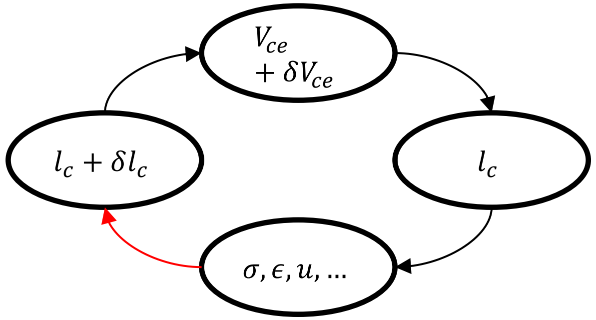
[1] Physics-informed Markov chains for remaining useful life prediction of wire bonds in power electronic modules, Microelectronics Reliability

Autoregressive RUL Estimation Model

Modeling degradation dynamics using health indicator transitions







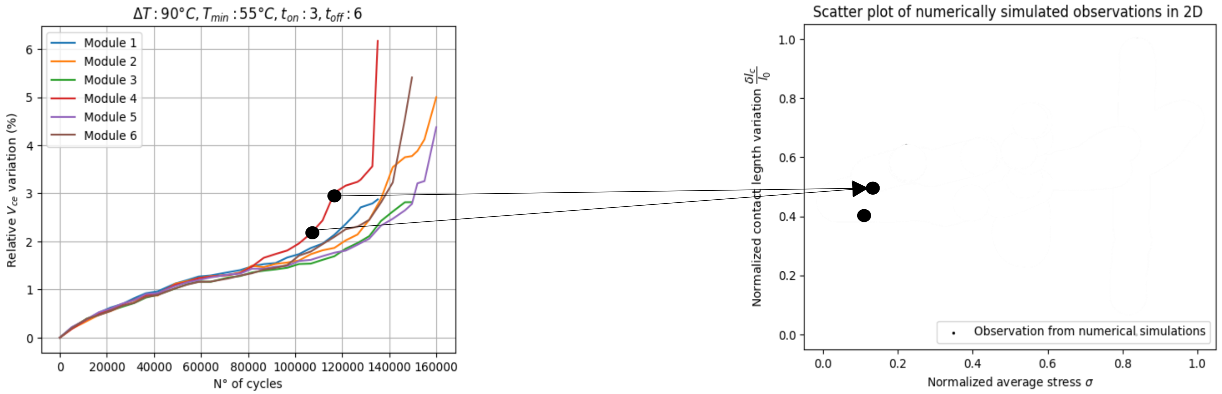
- load $(\Delta T, t_{on}, \ldots) \Rightarrow \sigma$
- Initial state $V_{ce}^1 \Rightarrow$ Initial state l_c^1
- Final state $V_{ce}^2 \Rightarrow$ Final state l_c^2

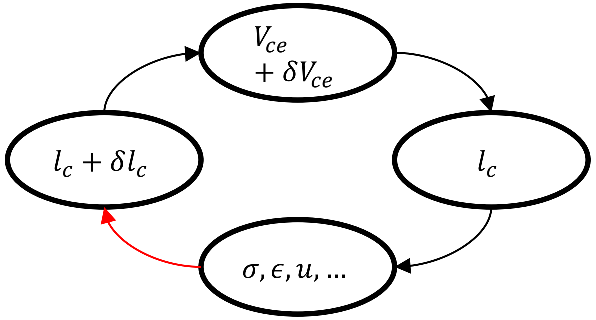
$\delta l_c = l_c^2 - l_c^1$

[1] Physics-informed Markov chains for remaining useful life prediction of wire bonds in power electronic modules, Microelectronics Reliability

Autoregressive RUL Estimation Model

Modeling degradation dynamics using health indicator transitions





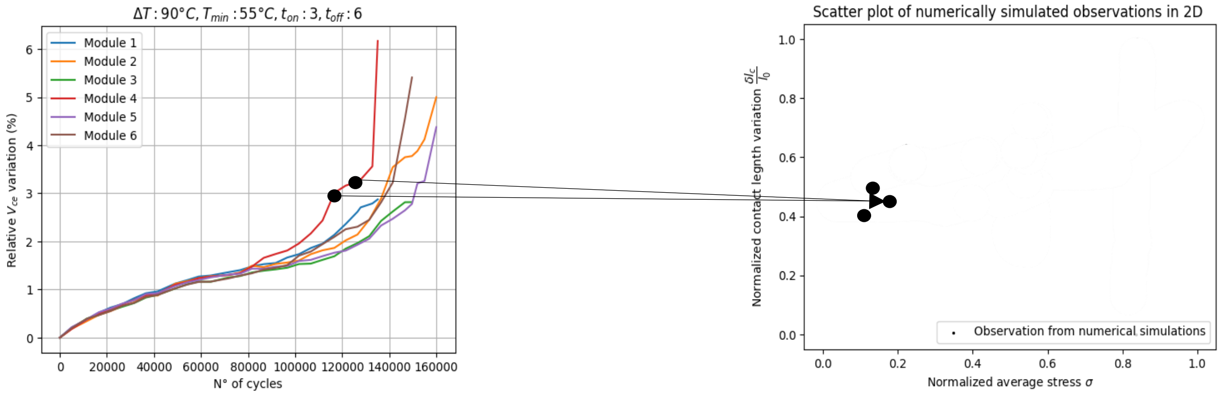
- ❖ load $(\Delta T, t_{on}, \dots) \Rightarrow \sigma$
- ❖ Initial state V_{ce}^1 $\Rightarrow$ Initial state l_c^1
- ❖ Final state V_{ce}^2 $\Rightarrow$ Final state l_c^2

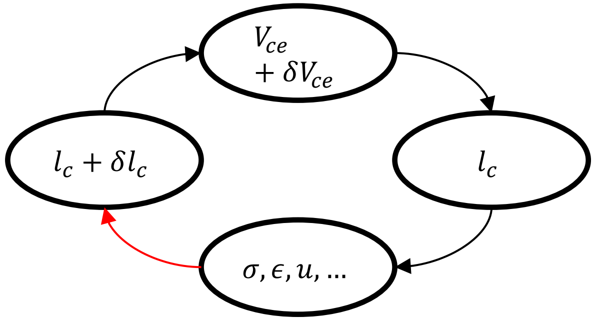
$\delta l_c = l_c^2 - l_c^1$

[1] Physics-informed Markov chains for remaining useful life prediction of wire bonds in power electronic modules, Microelectronics Reliability

Autoregressive RUL Estimation Model

Modeling degradation dynamics using health indicator transitions





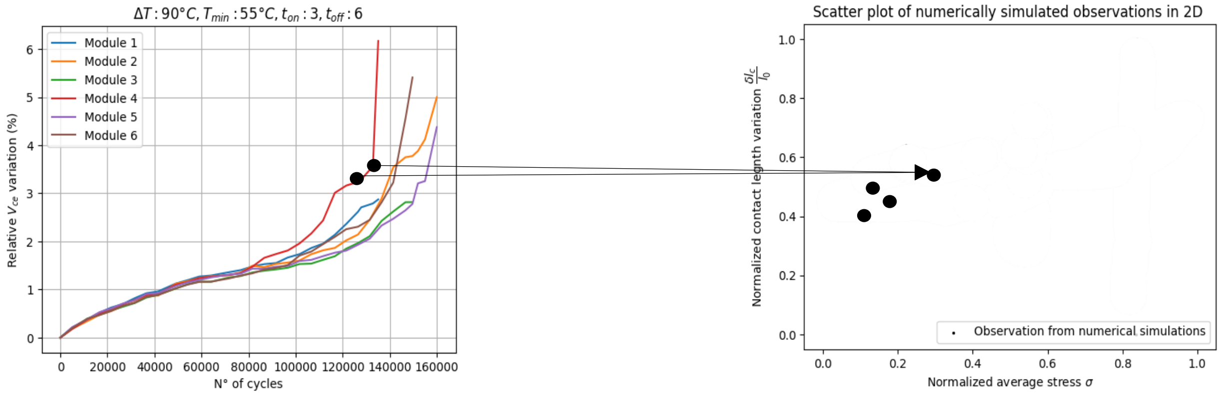
- ❖ load $(\Delta T, t_{on}, \ldots) \Rightarrow \sigma$
- ❖ Initial state $V_{ce}^1 \Rightarrow$ Initial state l_c^1
- ❖ Final state $V_{ce}^2 \Rightarrow$ Final state l_c^2

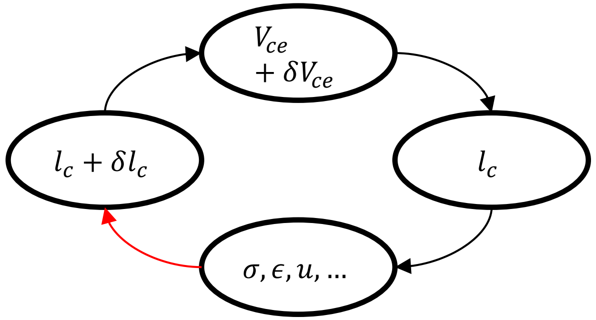
$\delta l_c = l_c^2 - l_c^1$

[1] Physics-informed Markov chains for remaining useful life prediction of wire bonds in power electronic modules, Microelectronics Reliability

Autoregressive RUL Estimation Model

Modeling degradation dynamics using health indicator transitions





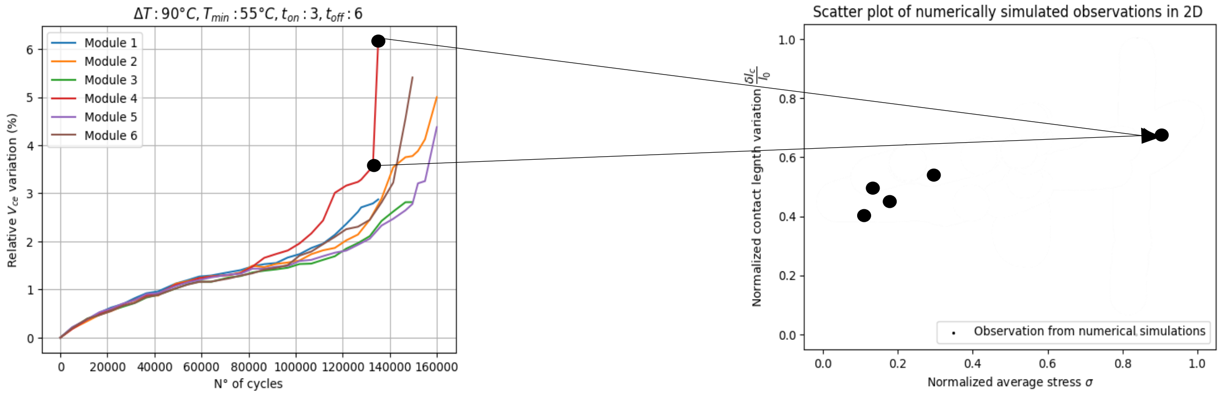
- load $(\Delta T, t_{on}, \ldots) \Rightarrow \sigma$
- Initial state $V_{ce}^1 \Rightarrow$ Initial state l_c^1
- Final state $V_{ce}^2 \Rightarrow$ Final state l_c^2

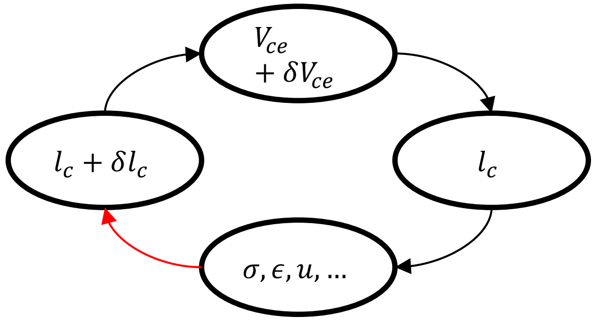
$\delta l_c = l_c^2 - l_c^1$

[1] Physics-informed Markov chains for remaining useful life prediction of wire bonds in power electronic modules, Microelectronics Reliability

Autoregressive RUL Estimation Model

Modeling degradation dynamics using health indicator transitions





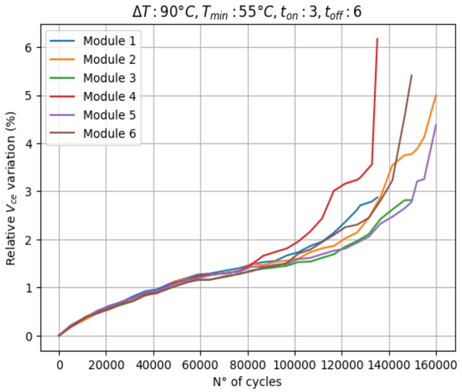
- ❖ load $(\Delta T, t_{on}, \ldots) \Rightarrow \sigma$
- ❖ Initial state $V_{ce}^1 \Rightarrow$ Initial state l_c^1
- ❖ Final state $V_{ce}^2 \Rightarrow$ Final state l_c^2

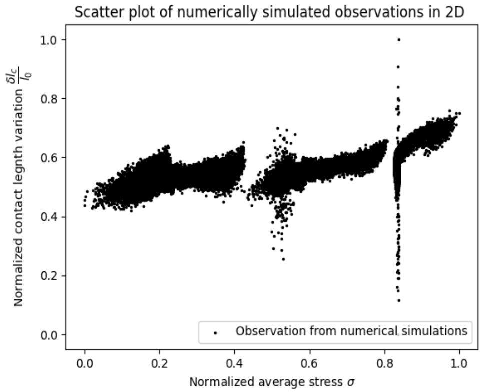
$\delta l_c = l_c^2 - l_c^1$

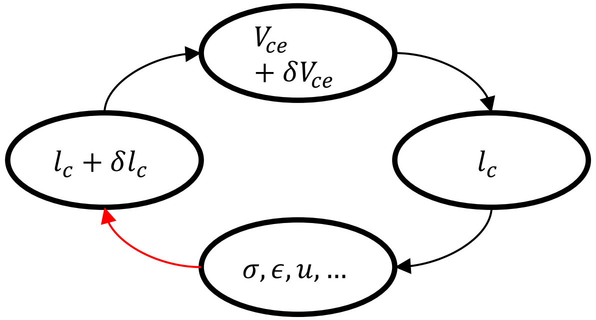
[1] Physics-informed Markov chains for remaining useful life prediction of wire bonds in power electronic modules, Microelectronics Reliability

Autoregressive RUL Estimation Model

Modeling degradation dynamics using health indicator transitions







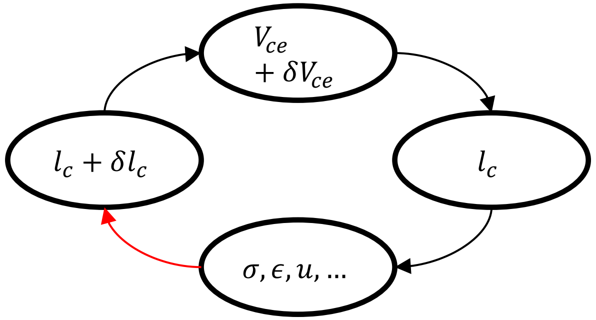
- ❖ load $(\Delta T, t_{on}, \ldots) \Rightarrow \sigma$
- ❖ Initial state V_{ce}^1 $\Rightarrow$ Initial state l_c^1
- ❖ Final state V_{ce}^2 $\Rightarrow$ Final state l_c^2

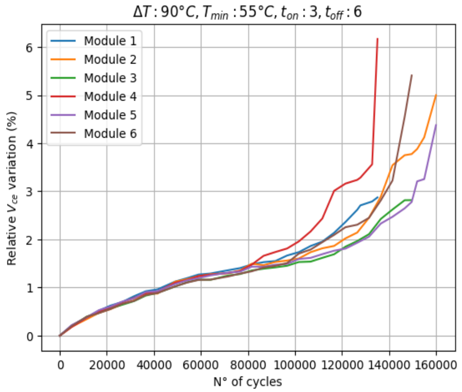
$\delta l_c = l_c^2 - l_c^1$

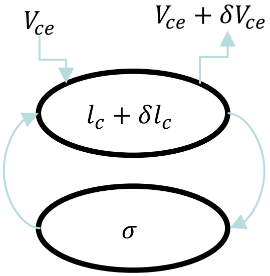
[1] Physics-informed Markov chains for remaining useful life prediction of wire bonds in power electronic modules, Microelectronics Reliability

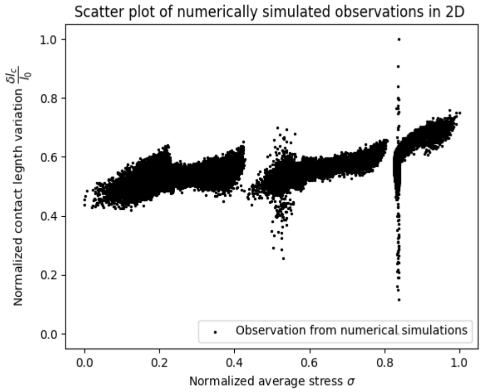
Autoregressive RUL Estimation Model

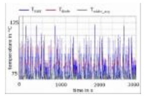
Modeling degradation dynamics using health indicator transitions

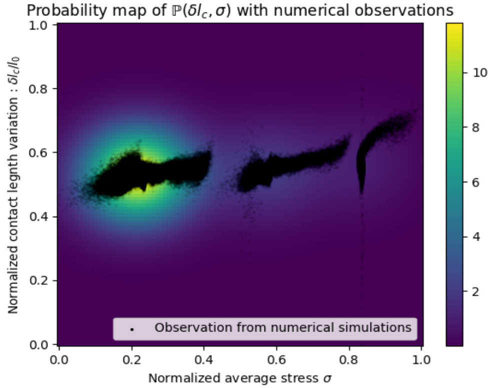










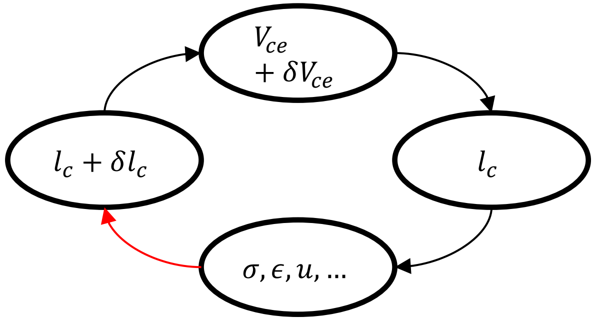


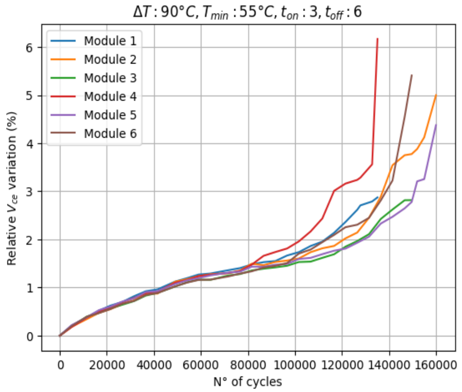
Probability distribution $P(\delta l_c, \sigma)$ calculated using Kernel Density Estimation

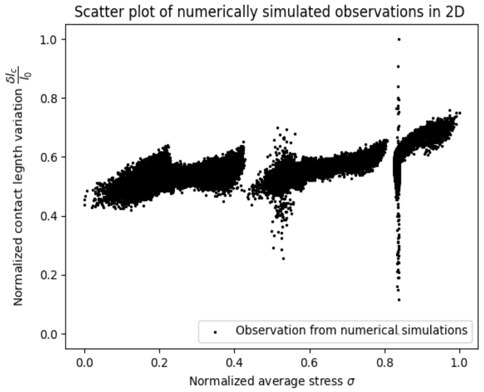
[1] Physics-informed Markov chains for remaining useful life prediction of wire bonds in power electronic modules, Microelectronics Reliability

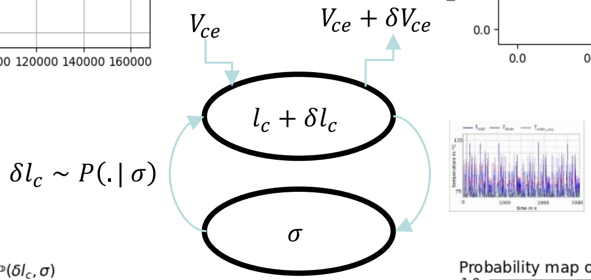
Autoregressive RUL Estimation Model

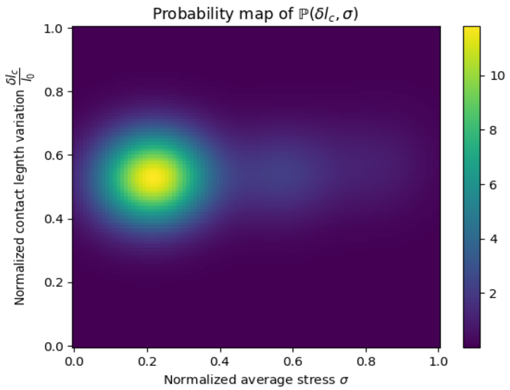
Modeling degradation dynamics using health indicator transitions

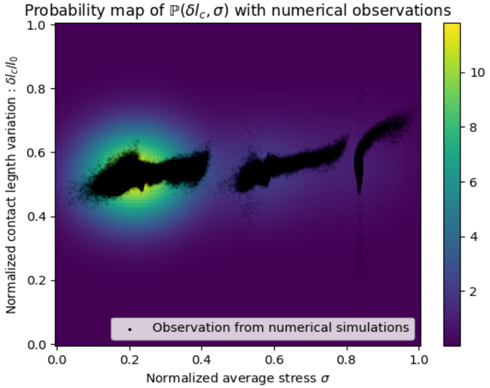








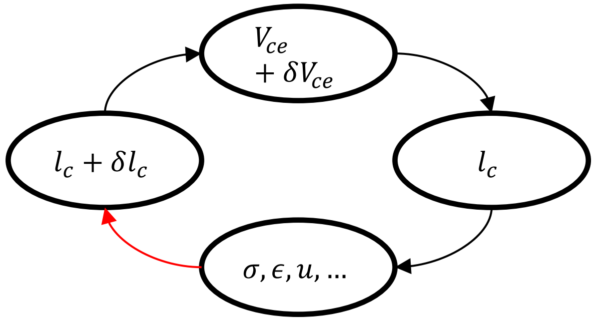


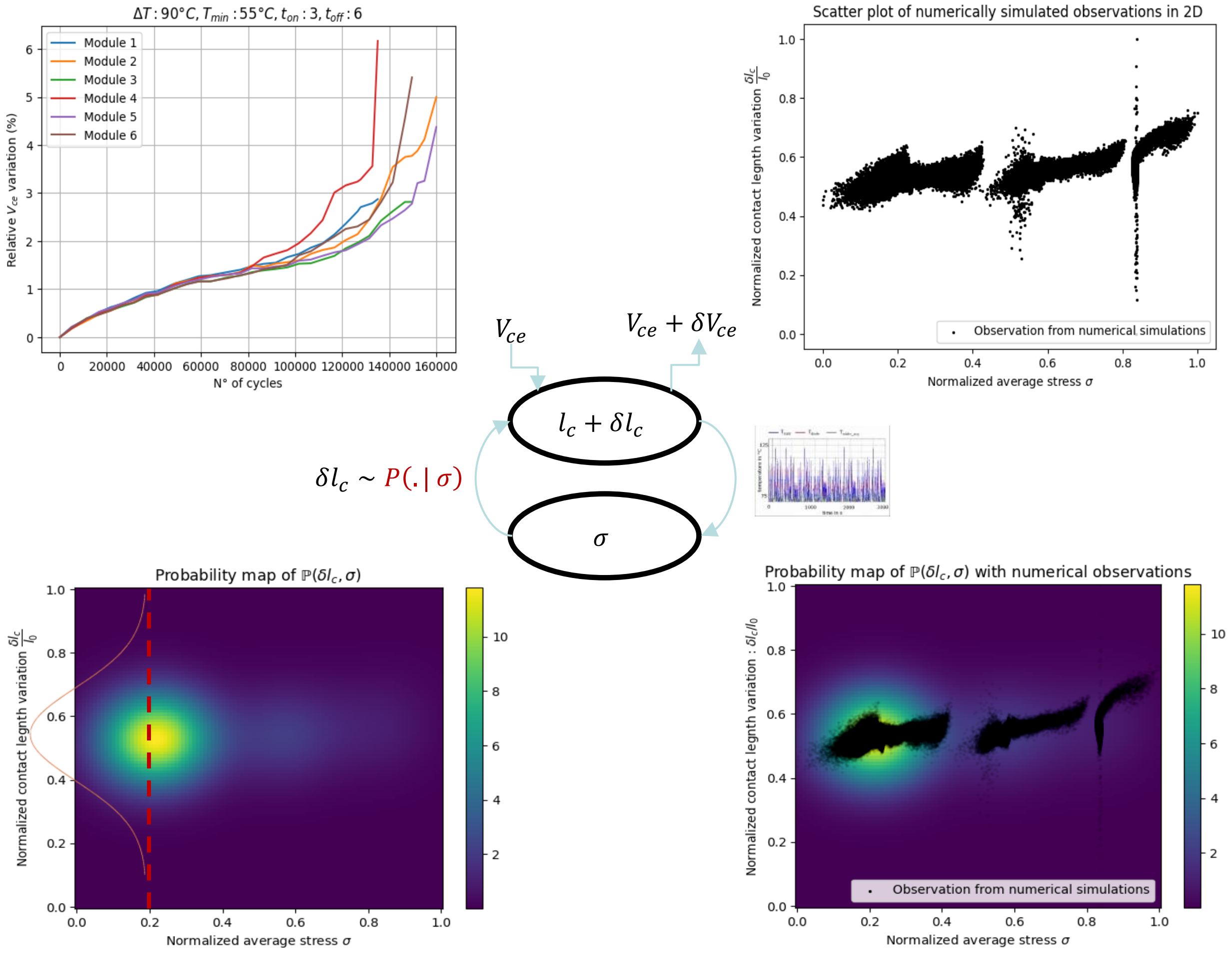


[1] Physics-informed Markov chains for remaining useful life prediction of wire bonds in power electronic modules, Microelectronics Reliability

Autoregressive RUL Estimation Model

Modeling degradation dynamics using health indicator transitions

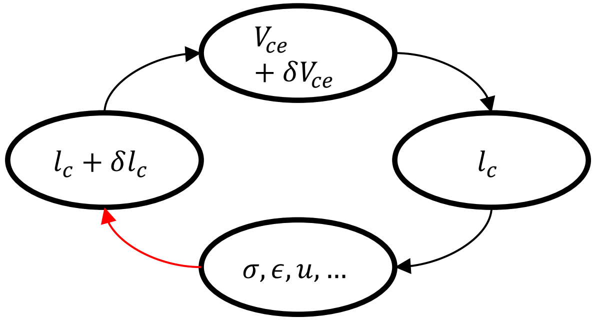


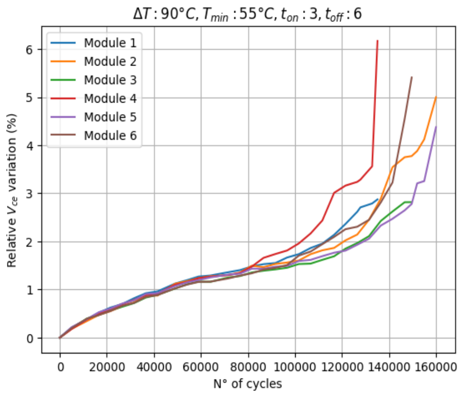


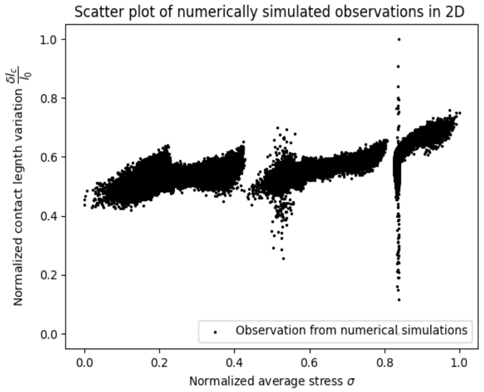
[1] Physics-informed Markov chains for remaining useful life prediction of wire bonds in power electronic modules, Microelectronics Reliability

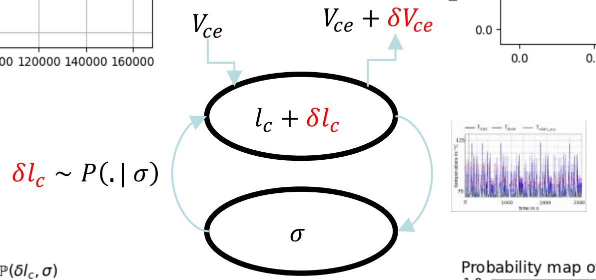
Autoregressive RUL Estimation Model

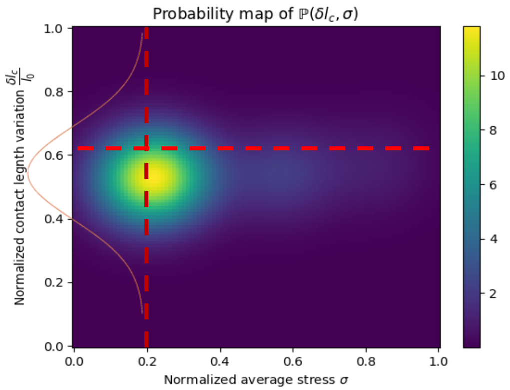
Modeling degradation dynamics using health indicator transitions

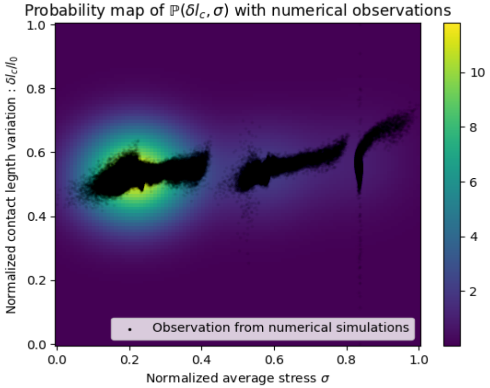










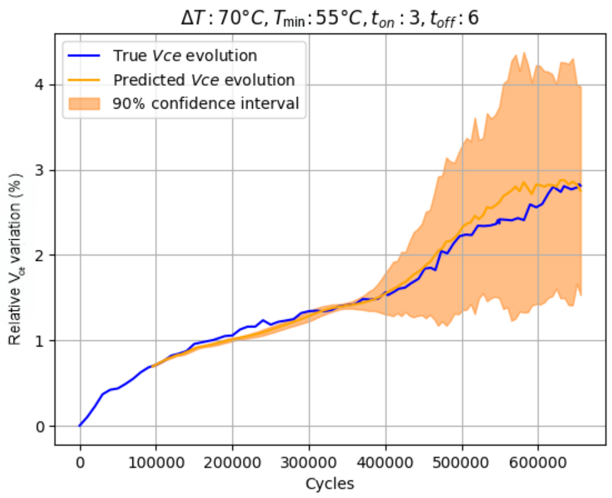


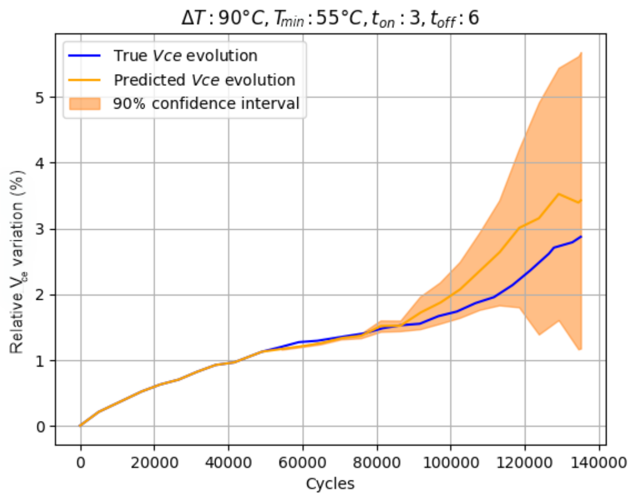
[1] Physics-informed Markov chains for remaining useful life prediction of wire bonds in power electronic modules, Microelectronics Reliability

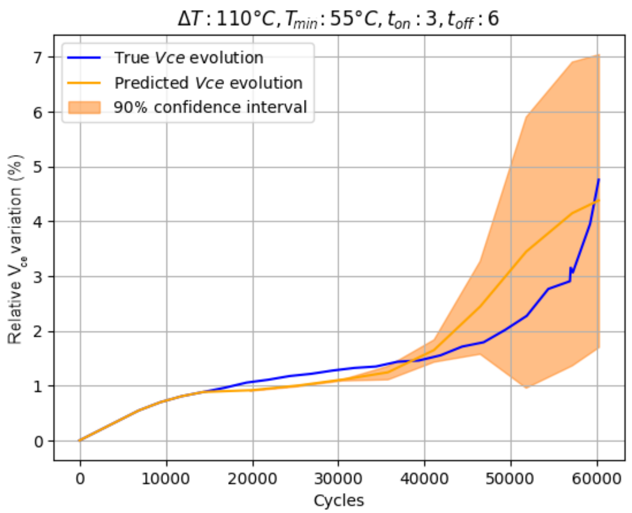
Autoregressive RUL Estimation Model

In-distribution predictions

[2] Hybrid modeling for remaining useful life prediction in power module prognosis







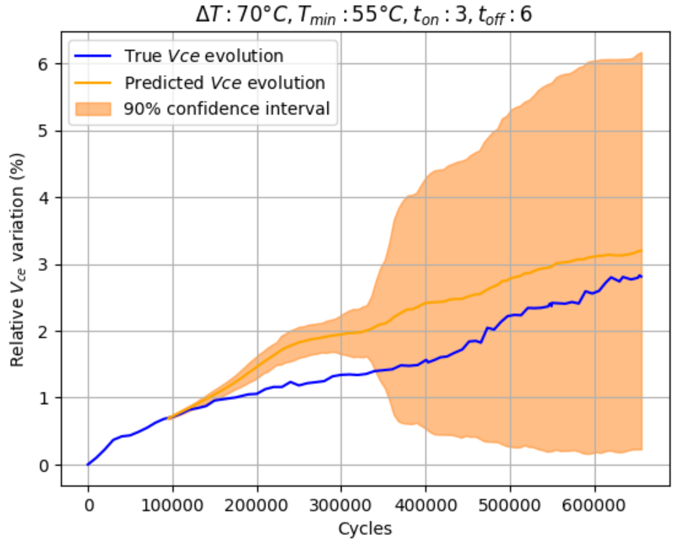
ΔT (°C)	RBT	RMSBT
70	0,702	0,637
90	0,802	0,802
110	0,509	0,509

- **RBT**: Relative bounding time, varies from 0 (worst) to 1 (best). Indicates the fit of the confidence interval
- **RMSBT**: Relative maximum strict bounding time. varies from 0 (worst) to 1 (best). Indicates the time proportion after which the confidence interval predicts failure

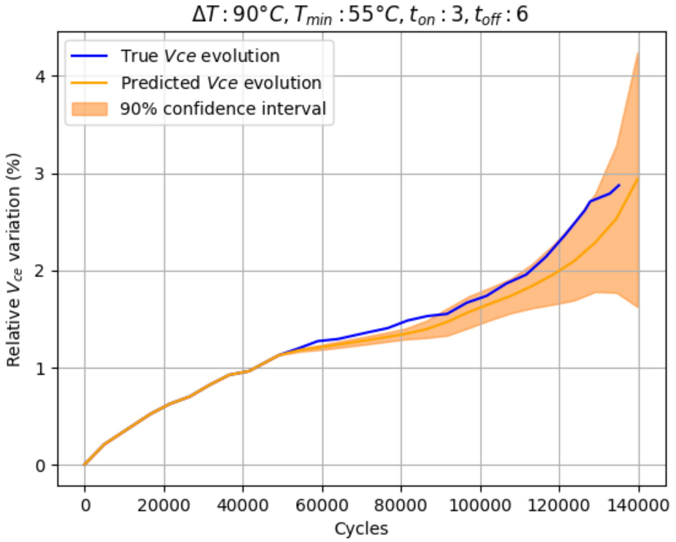
[1] Physics-informed Markov chains for remaining useful life prediction of wire bonds in power electronic modules, Microelectronics Reliability

Autoregressive RUL Estimation Model

Out-of-distribution predictions



Extrapolation



Interpolation

ΔT (°C)	RBT	RMSBT
70 (extrapolation)	0,615	0,563
90 (interpolation)	0,574	0,574

[1] Physics-informed Markov chains for remaining useful life prediction of wire bonds in power electronic modules, Microelectronics Reliability