Machine Learning Analysis on the JET Pedestal Database

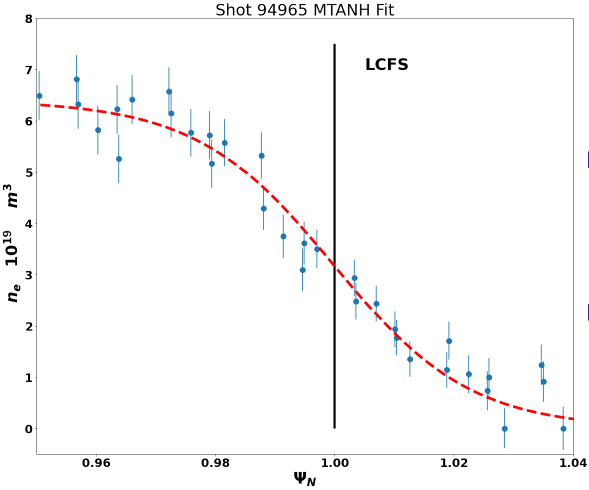
or how I learned to worry about high n_e^{ped}

9 June, 2021

Overview

Table of Contents

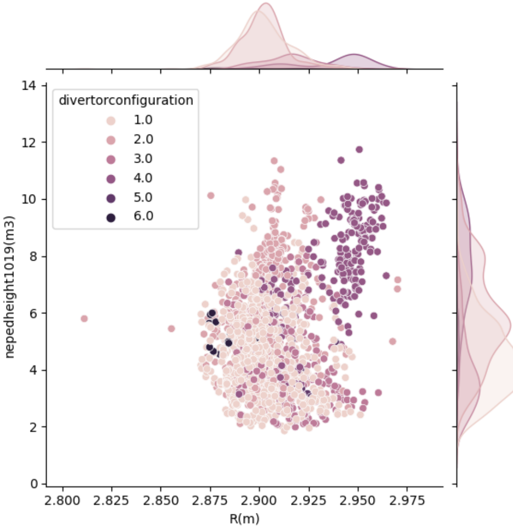
JET pedestal database contains 3000+ entries of H-mode, pre-ELM plasmas



- Pedestal parameters determined from mtanh fit
- Focus on n_e^{ped} height

Use only main engineering parameters to predict n_e^{ped}

Eng. Param	Domain
I_P [MA]	$[0.81, 4.48]$
B_T [MW]	$[0.97, 3.68]$
a [m]	$[0.83, 0.97]$
R [m]	$[2.8, 2.975]$
δ [-]	$[0.16, 0.48]$
M_{eff} [-]	$[1.0, 2.18]$
P_{NBI} [MW]	$[10^{-3}, 32.34]$
P_{ICRH} [MW]	$[0, 7.96]$
P_{TOT} [MW]	$[3.4, 38.22]$
V_P [m^3]	$[58.3, 82.19]$
q_{95} [-]	$[2.42, 6.04]$
Γ [10^{22} e/s]	$[0, 15.5]$
H [-]	$[0, 0.18]$
P_{SD} [10^{22} e/s]	$[0, 1000]$
DC [-]	$[VV \cdots]$



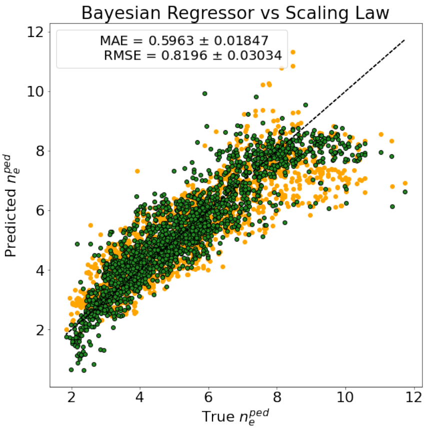
Take only deuterium shots, no RMPs, kicks, or pellets, and remove R, P_{SD}, DC from input space

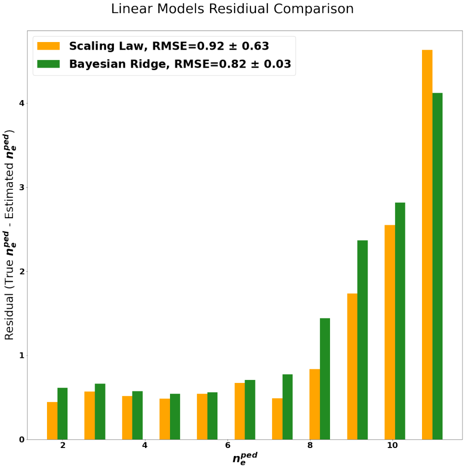
- Shafranov Shift$\rightarrow$ remove R
- P_{SD} varies too much.
- Only numerical columns used as input
- HRTS Validated

Eng. Param	Domain
I_P [MA]	$[0.81, 4.48]$
B_T [MW]	$[0.97, 3.68]$
a [m]	$[0.83, 0.97]$
$\cancel{R \text{ [m]}}$	$\cancel{[2.8, 2.975]}$
δ [-]	$[0.16, 0.48]$
M_{eff} [-]	$[1.0, 2.18]$
P_{NBI} [MW]	$[10^{-3}, 32.34]$
P_{ICRH} [MW]	$[0, 7.96]$
P_{TOT} [MW]	$[3.4, 38.22]$
V_P [m^3]	$[58.3, 82.19]$
q_{95} [-]	$[2.42, 6.04]$
Γ [10^{22} e/s]	$[0, 15.5]$
H [-]	$[0, 0.18]$
$\cancel{P_{SD} \text{ [}10^{22}\text{e/s]}}$	$\cancel{[0, 1000]}$
$\cancel{DC \text{ [–]}}$	$\cancel{[VV \cdots]}$

Table of Contents

Utilizing more input parameters yeilds higher prediction quality than existing scaling law





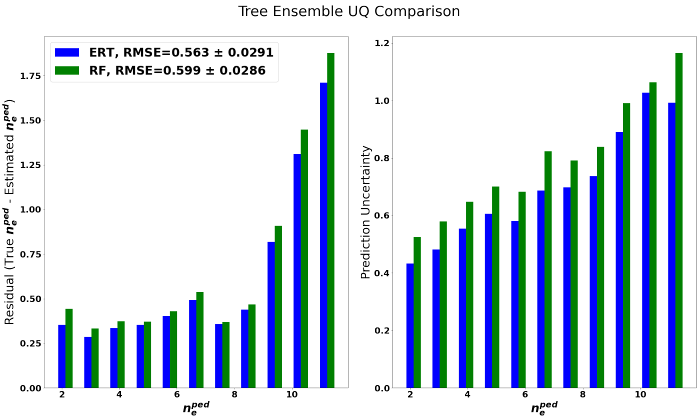
$$n_e^{ped} = (9.9 \pm 0.3) I_p^{1.24\pm0.19} P_{TOT}^{-0.34\pm0.11} \delta^{0.62\pm0.14} \Gamma^{0.08\pm0.04} M_{eff}^{0.2\pm0.2} \quad (1)$$

Including more input params reduces interpretability

- Prediction uncertainty normally distributed between 1.6 - 1.8

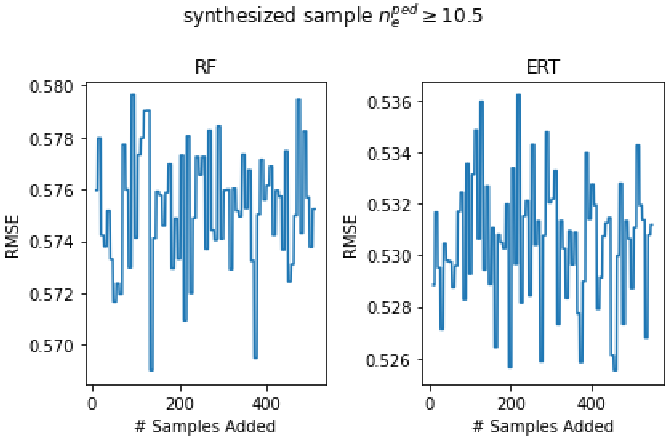
Feature	μ	σ^2
I_p	0.15	0.06
B_T	0.956	0.072
a	2.966	0.479
δ	12.95	0.154
V_P	-0.05	0.007
q_{95}	-1.064	0.0542
P_{NBI}	-1.911	0.0546
P_{ICRH}	-1.976	0.0561
P_{TOT}	1.926	0.0557
Γ	0.125	0.007
H	-4.016	0.374
M_{eff}	1.369	0.053

Non-linear models like Random Forests and Extreme Randomized Trees outperform linear models



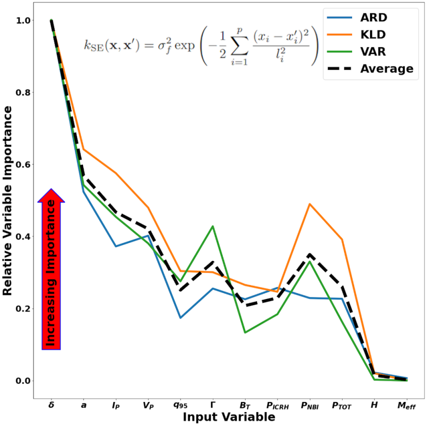
Uncertainty determined from std of each decision tree pred.

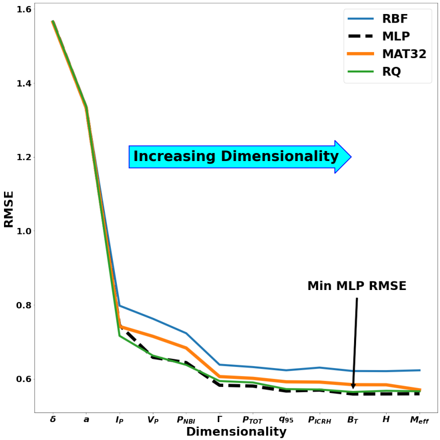
Meta-modeling has no effect on RFs and ERTs



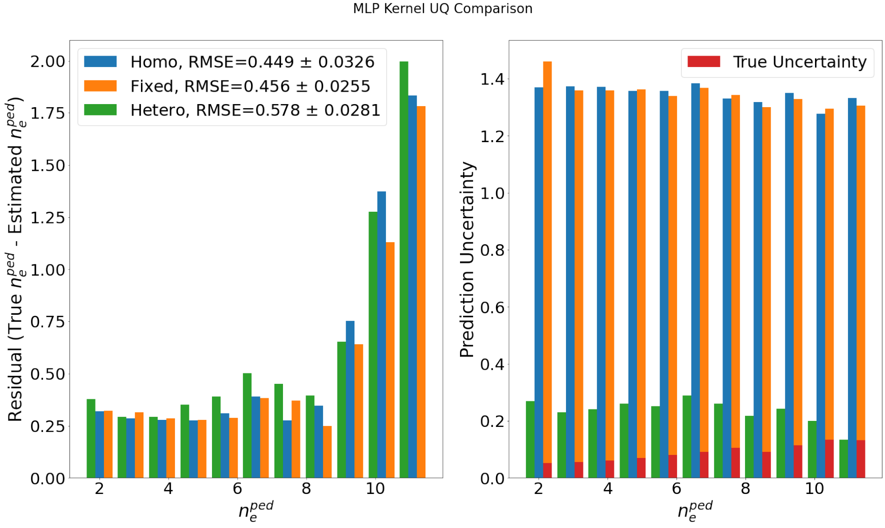
Goes on to 2000+ entries

Can reduce dimensionality for Gaussian Processes using sensitivity analysis $\rightarrow$ remove H, M_{eff} from input space

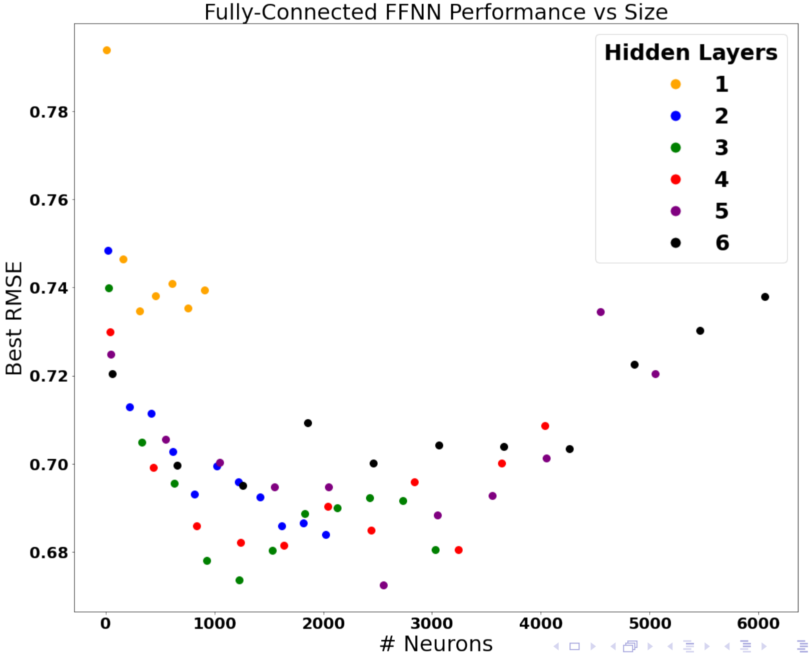




GPs outpreform RFs and ERTs, and heteroscedastic models can capture local uncertainty



Shallow Artifical Neural Networks outperform larger nets



Ensembling for prediction uncertainty → increases with size of ensemble

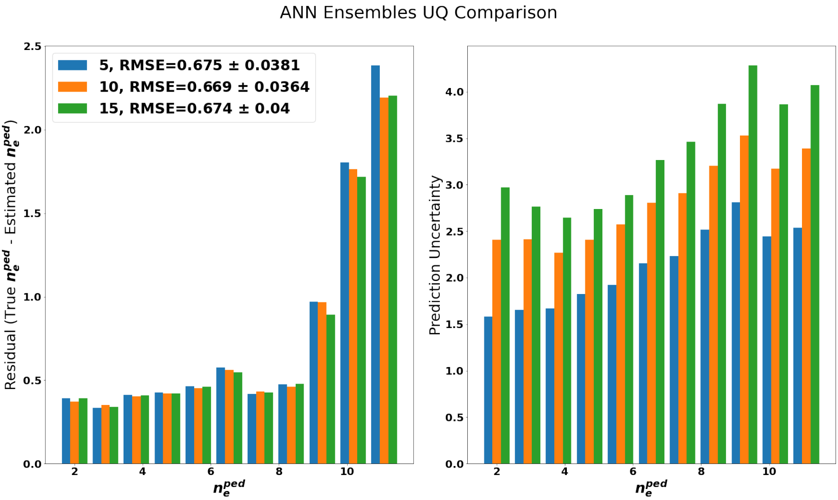


Table of Contents

Conclusion

- Non-linear models outpreform linear models
- RFs and ERTs work well as black box models
- Heteroscedastic GPs can map local uncertainty
- $n_e^{ped} \geq 9$ is elusive to non-linear models

Future Work

- Fit models on varied subsets of database
- UQ of main engineering parameters
- Ideas?