Principle Machine Learning Analysis on the JET-Pedestal Database, aka Nutshell

Univerisitat Leipzig

UNI LOGO

Adam Kit

15 05 2021

Contents

1 Introduction

Harnessing of controlled thermonuclear fusion on Earth is a complex multi-faceted problem; a potential solution is that of controlled magnetic confinement fusion (MCF), such as stellarators or tokamaks [reference EUROfusion road map]. The field of MCF research is entering the era of superconducting, reactor-scale, long pulse devices, such as ITER and DEMO [Progress in ITER physics basis NF 2007, Wenninger NF 2017]. These reactor-scale devices encompass a significant risk of very costly component damages in off-normal events, and, hence, the emphasis on reactor and plasma scenario design is shifting form experimental approaches to theory based predict-first and plasma flight simulator methods [FIND REFERENCE]. In order to bridge the gap between computational and experimental efforts and to be able to rapidly design reactors, there exists a need for data driven approaches to produce simplified models through the use of machine learning (ML). The topic of this thesis is to analyse and compare predictive ML tools in estimating plasma parameters, specifically the density of plasma in the edge region of tokamaks.

1.1 Why Pedestal Physics?

In tokamaks, the fusion plasma is confined in a toroidal vacuum chamber using magnetic fields. The magnetic field has components around the torus, called the toroidal component, and around the cross-section of the vacuum chamber, called the poloidal component. These are generated with magnetic coils and plasma currents. The toroidal and poloidal magnetic fields generate helical field lines that are necessary to confine the plasma. The helical field lines form nested closed flux surfaces. At the edge of the plasma, structures of the reactor wall intersect these flux surfaces, such that they become open. The region of the open field lines within which the plasma is in contact with the reactor components is called the scrape-off layer (SOL). The last flux surface that is closed is aptly named the last closed flux surface (LCFS) and the field line that separates LCFS from SOL is called the seperatrix. In present day plasma scenarios, typically the edge plasma is magnetically diverted to a separate divertor area, such that the LCFS is not directly in contact with the wall.

Nuclear fusion with net energy gain requires sufficiently high fuel pressure and confinement time, i.e., the triple product of the density, temperature and confinement time must be high enough [SOURCE]. The closely related 'Lawson criterion' states that a fusion plasma is considered 'ignited' when the rate of energy production is higher than its energy loss. An ignited plasma must be confined for long enough time, τ_E, at high enough density, n, such that the Lawson criterion is met. So far, the highest triple product achieved is through MCF devices is 1.53×10^{21}keV m / s^3 using deuterium reactions in the JT60U reactor [SOURCE], and ITER and DEMO plan to achieve triple product values of around 10^{22}keV m / s^3 through the use of deuterium-tritium reactions [SOURCE].

The maximum pressures achievable in magnetic confinement fusion (MCF) devices are limited by magnetic field strength and magnetohydrodynamic (MHD) instabilities. The pressure in tokamaks is commonly referred to as $\beta = p^2/B^t/2\mu_0$ (or β_N when β is normalized with respect to the tokamak in question), and due to the MHD instabilities has a limit known as the Troyon-Beta limit [SOURCE]. A MCF device then aims to confine a plasma which has pressure on the order of atmospheric pressure for a time τ_E on the order of seconds.

The energy confinement time is limited by turbulence of the plasma, which leads to radial transport across the flux surfaces significantly faster than would be expected based on classical or neo-classical transport [Chapter 2 of Progress in the ITER physics basis NF 2007]. Typically, the turbulence modes show critical gradient behaviour [same as above]. As a result, the radial gradients are limited near their critical value and the effective transport increases with more

heating power. This results in a reduction of τ_E with heating power[same as above], meaning that reaching $nT\tau_E$ is quite challenging considering the solution of throwing power at the problem has the opposite than desired effect. In the 1980s, a sudden transition into an enhanced confinement regime, called the high confinement mode (H-mode), was discovered in plasmas operating with the divertor configurations with neutral beam heating [Wagner PRL 1982] The H-mode confinement can break away from the stiff gradients at the edge, as self-organized shear flows at the plasma edge reduce turbulent radial fluxes, leading to the formation of a 'pedestal'. To achieve H-mode confinement, there needs a minimum amount of power flow through the edge [Martin Scaling source], which ultimately leads to about a factor of two increase of the energy confinement time and thus leading to H-mode being the baseline operational mode for future fusion devices such as ITER and DEMO.

The suppressed turbulence in the pedestal region allows the radial pressure and current gradients to grow until they trigger magnetohydrodynamic (MHD) instabilities, called edge localized modes (ELMs)[[SOURCES AD INFIN]] The current understanding is that high performance pedestal plasmas are limited by ideal MHD peeling-ballooning instabilities that trigger type-I ELMs[sources]. Pedestal plasmas are not always limited by ideal MHD peeling-ballooning instabilities. For example, a large fraction of JET H-mode plasmas operating with the ITER-like wall, including beryllium main chamber and tungsten divertor targets [reference], do not reach the ideal MHD peeling-ballooning stability threshold [Frassinetti NF 2021], therefore determining which transport phenomena cause limits in the gradients within the pedestal transport barrier is a very active topic of research[source].

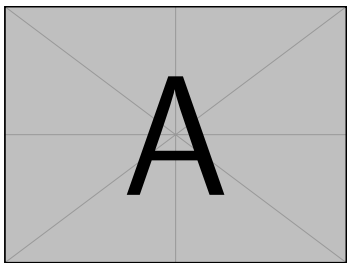
Since H-mode and edge transport barriers are key ingredients of future ITER and DEMO scenarios, predictive capability is necessary for the pedestal region in order to confidently design future fusion reactors and their operational scenarios. Simulation codes today have to make certain transport assumptions; the EPED model [SOURCE SYNDER REFERNCES] assumes that the kinetic ballooning mode (KBM) limits the radial pressure scale length. EPED has been successful in predicting pedestals in may tokamaks, but experimental observations show that the KBM assumption is not observed to describe the experiment well [LORENZO SOURCES]. Additionally, EPED takes certain plasma parameters, such as β, pedestal density n_e^{ped}, and the effective charge of the plasma Z_{eff} as inputs, and thus can not be considered as fully predictive model. Addressing this issue is EUROPED, another simulation package that is based on EPED but uses other transport models for the core and pedestal density [SOURCE SAARELMA]. In this thesis, machine learning tools for pedestal analysis and predicting pedestal quantities based on an experimental database for the JET-ILW plasma will be studied. To limit the extent of the thesis, the focus is on the pedestal density, and ML tools analysed are compared to an experimental log-linear fit published in [Lorenzo DB paper]

1.2 JET Pedestal Database

The JET pedestal database is the most comprehensive of all pedestal databases today, containing over 3000 entries [Frassinetti 2021]. Each entry corresponds to time averaged measurements of various plasma parameters over the course of 70-95% of an ELM cycle [cross check]. The measurements are done using high resolution thomson scattering (HRTS)[Frassinetti], and are then fitted using the mtanh function:

$$mtanh(r) = \frac{h_1 - h_0}{2}\left(\frac{(1+sx)e^x - e^{-x}}{e^x + e^{-x}} + 1\right) + h_0, \qquad x = \frac{p-r}{w/2} \tag{1}$$

where the pedestal height, position, width, and slope are h_1, p, w, and s respectively, and r the normalized radius Ψ_N. The fitting process is visualized in Figure 1. Since the measurements are taken near the end of the ELM cycle, the pedestal parameters should be saturated near their maximum, right before the ELM.



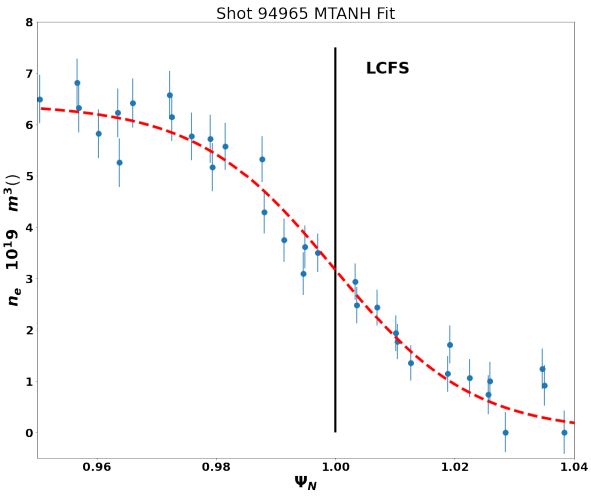


Figure 1: **Left:** A shaping figure showing r,a, triangularity. Resembles a bow **Right:** HRTS measurement profiles (blue) radially shifted to have $T_e \approx 100$ eV at the seperatrix (LCFS). The profiles are fitted in real space using the mtanh equation 1 then mapped to the normalized poloidal flux coordinate Ψ_N (red).

The key engineering quantities in the database and their units ([-] dimensionless) are listed below,

- I_P [MA], plasma current, current driven through the plasma that generates the poloidal magnetic field
- B_T [T], toroidal magnetic field
- R [m], major radius of the plasma
- a [m], minor radius of plasma
- δ [-], triangularity, normalized horizontal displacement of the top/bottom of the plasma from the main axis
- V_P [m^3], the plasma volume
- H, isotope ratio of fuel
- q_{95} [-], safety factor at the flux surface edge, where safety factor is the 'windiness' of the magnetic fields in a reactor, i.e., the number of toroidal circles the helical field line completes within one poloidal revolution, trouble arises well before q_{95} reaches 1, so devices typically operate within a range of $q_{95} \in [3, 4]$ in order to maintain stable plasmas.
- P_{NBI} [MW], neutral beam injection heating power
- P_{ICRH} [MW], ion cyclotron radio frequency heating
- P_{TOT} [MW], total power ($P_{TOT} = P_{NBI} + P_{ICRH} + P_{OHM} - P_{SH}$, where P_{OHM} is the ohmic heating due to the plasma current, and P_{SH} is the power lost due to the shine through of NBI heating)
- Γ [10^{22} electrons per second], gas fuelling rate of the main determined fueld for the shot, this changes depending on what type of fueling is used

- DC, the divertor configuration, can take on values of C/C, V/H, V/C, V/V, C/V, C/H, (see frasineeti for more information)
- TW, the type of wall, as JET was upgraded in the mid 2010s, and moved from having a Carbon wall to an 'ITER like wall' (ILW) [SOURCE]
- Γ_{SD} [10^6 nbar], the subdivertor pressure [SOURCE]

For the main engineering parameters, the uncertainties are calculated by taking the standard deviation of the values over the time period in which the measurements were taken.

The global parameters stored in the database are listed below:

- β_θ^{ped} [-], β is the ratio of plasma pressure p to the pressure exerted by the magnetic field B, $\beta = p/B^2/2\mu_0$, thus β_θ^{ped} is the pressure due to the poloidal magnetic field B_θ and plasma pressure at the pedestal p_e^{ped}
- β_N, normalized β for comparison between reactors, as β has an inherent limit based on MHD stability, and is a function of the plasma current, minor radius and magnetic field such that $\beta_N = \beta/I/aB$, is commonly known as the Troyon factor
- Z_{eff}, the effective charge state of the plasma

Eng. Param	Domain
I_P [MA]	$[0.81, 4.48]$
B_T [MW]	$[0.97, 3.68]$
a [m]	$[0.83, 0.97]$
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 [-]	$[0, 0.18]$
Γ_{SD} [10^6nbar]	$[0, 1000]$

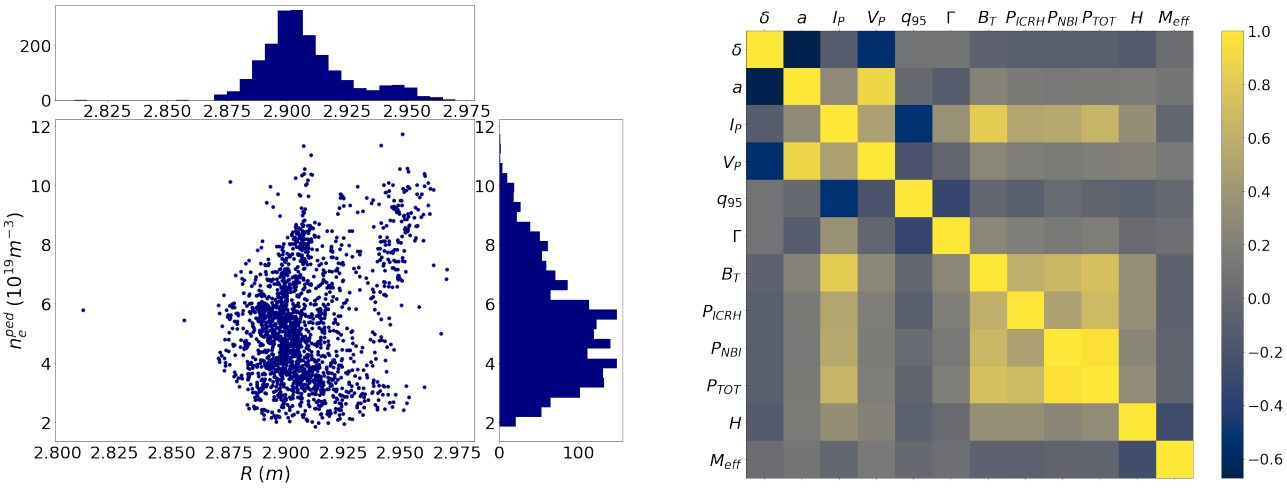
Table 1: Main engineering parameter domains of the filtered dataset.

The global parameters are certainly interesting, but within the context of this thesis are not considered to be viable inputs to a predictor, as they rely on information that is unavailable as a control knob on a reactor. A truly predictive model cannot take plasma parameters as inputs. Today, EPED takes β, n_e^{core} and Z_{eff} as inputs assuming the feedback can be used to choose the density and β. Models like EPED rely on the principle that reactor operators would 'know' these density and beta points are within reachable operational space, and that furthermore they know the recipe to get there. A model of interest would be that which uses the main engineering parameters to calculate profile parameters like height, width, or position for the pedestal quantities temperature, density, or pressure. The pedestal profile parameters stored in the database are determined using the mtanh fit (equation 1), and the uncertainties are the fit uncertainties from the use of the mtanh function [SOURCE]. The fit uncertainties are expected to be significantly smaller than the natural scatter of the data due to the fluctuation of the plasma.

Additionally, there exist so called FLAGS, which correspond to the specific setup of an experiment. For example, what element the fuel is, if resonant magnetic pulses (RMPs) [source], pellets [source] or impurity seeding [source] were used are all FLAGS contained in the database. Additionally, there is a flag corresponding to the quality of the HRTS measurement, as each entry is validated either by hand or computationally. Only entries that have been validated are used in this thesis. Shots with impurity seeding are used, as they make up about 600 entries. RMPs, pellets, and kicks are used to manipulate the pedestal for ELM control, mitigation, or suppression [Viezzer NF 2018]. To keep the dataset simple in this thesis, these entries are excluded. After filtering out the RMPs, Kicks, Pellets, non-validated HRTS, and shots that do not use deuterium, the dataset is reduced to 1888 entries. The final pedestal parameter domains are given in the table below.

	Height	Width (Ψ_N)	Position (Ψ_N)	Slope (-)
n_e^{ped}	[1.849, 11.737] (10^{19} m^3)	[0.015, 0.173]	[0.953, 1.029]	[10^{-6}, 0.188]
T_e^{ped}	[0.149, 1.894] (keV)	[0.013, 0.105]	[0.926, 1.002]	[0.026, 0.502]
p_e^{ped}	[0.808, 17.804] (kPa)	[0.014, 0.099]	[0.931, 1.002]	[0.041, 0.789]

Table 2: Domains of pedestal paramaters for deuterium shots stored in the JET pedestal database after RMPs, Kicks, Pellets are filtered out.



(a) (b)

Figure 2: Emperical data plots of the JET pedestal database. **Left:** the 'ice cream corerlation' between the major radius R, divertor configuration, and n_e^{ped}, **Right:** correlation matrix of the main engineering parameters. A grey coloring represents no correlation, whereas blue and yellow are negative and positive correlation respectively.

1.2.1 Empirical Analysis

Empirical analysis of the JET pedestal database has been done [SOURCE LORENZO DB], and has yielded the following log-linear scaling law for the pedestal density height n_e^{ped}:

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

In this thesis, we will take the pedestal prediction with ML approaches relative to the performance of the empirical scaling law as the well defined focus for the project. The choice of parameters from the log-linear regression by Lorenzo et. al., was backed by physical intuition about what drives the pedestal density height. Since log-linear regression was used, it is also assumed that the scaling law above avoided using cross-correlated variables, which can verified in Figure 2b.

In order to improve prediction quality, it may prove useful to include additional inputs from the list of main engineering quantities that were not used in the log-linear scaling. However, by plotting joint histograms between the control parameters and n_e^{ped}, some serious questions can be raised regarding which parameters can and should be given to a machine learning model. For example, looking at the dependence of the major radius R in Figure 2a, one could jump to early conclusions and say that with higher values of R, a higher pedestal height is achieved! However, this is a case of ice-cream correlation[1], and the real culprit of the causation is the

[1] ice cream correlation refers to the correlation of increasing ice cream sales and increasing number of drown-

Shafranov shift; the outward radial displacement of the magnetic axis from the geometric axis that is prominently found in MCF devices[Shafranov SOURCE]. The shift is understood to be induced by plasma pressure, and thus is normally linear in β. Since β and pressure are heavily dependent on the aspects of the machine, the Shafranov shift is also machine dependent. For this reason, R is excluded from the list of inputs to the ML models in this thesis, and only when multi-machine databases are available should it be included. The divertor configuration on the other hand, does have a real correlation, and can have a large impact on the pedestal. However, the analysis in this thesis makes use of the numerical parameters available only, and thus DC will not be used as an input parameter, as it is categorical.

Another engineering parameter that is ignored in the analysis is the sub-divertor pressure Γ_{SD}. From the filtered dataset, the values of the sub-divertor vary widly, with 130 entries having an error and value of 1000 [10^6 nbar], while having close to 500 entries that vary between [0, 0.5]. Because of this volatility, Γ_{SD} is ignored, however future research may choose to filter the dataset such that inclusion of Γ_{SD} is possible.

ings in Finland during the summer. Although the variables are indeed correlated, higher ice cream sales are not in fact the causation of higher drowning rates, nor vice-versa.

2 General Machine Learning Analysis

Within the context of this thesis, a model refers to a prediction function f that takes any combination of the main engineering parameters as inputs, $\vec{x} = (x_1, x_2, \cdots, x_p)$, and provides an estimate of the pedestal density height, $\hat{y}$, as well as the uncertainty of the estimate (when applicable). The prediction quality is quantified through both the root mean squared error (RMSE) and mean absolute error (MAE), since a robust model minimizes both of these. The RMSE penalizes predictions that are far away from the ground truth, whereas the MAE calculates a uniform distance between the prediction and the ground truth.

2.1 Model Fitting and Validation

To make a prediction, the model must first be *fitted*, which means to learn the parameters $\vec{\theta}$ of the prediction function using a model specific learning algorithm such that $f = f(\vec{x} : \vec{\theta})$. In the case of linear regression, the learning algorithm is the ordinary least squares method (OLS) [SOURCE] which minimizes the mean squared error in order to find the optimal linear coefficients $\theta(w_i)$. Not all supervised learning regression algorithms minimize the RMSE or MAE to fit model parameters. The RMSE and MAE as stated above are merely a quantization of the performance of a particular model.

The scoring of the performance of a fitted model shall be made using data that the model had previously not seen. If this were not the case, the model would simply repeat what it had been fitted with, and would fail to predict useful information on yet unseen data. This is called *overfitting*; to avoid overfitting a common practice in supervised machine learning is to hold out part of the available data as a test set. This can be done by randomly splitting the available data into training and test subsets, and to evaluate the model on the test set. Then, depending on the performance, adjust the *hyperparameters* of the model (an example of a hyperparameter is the number of trees in a random forest, or learning rate for ANN's) in order to optimize the performance on the test set. However, this runs into the problem of overfitting on the test set, since the hyperparameters can be adjusted until the model preforms best on the test set. In this sense, knowledge about the test set leaks into the model, and evaluation metrics no longer report on general performance. Additionally, in randomly splitting the data into two groups, there is a new problem of *selection bias*[SOURCE], in which the model's results are dependent on the random choice of training/test sets. To overcome these problems, *cross-validation* or CV is implemented throughout the analyses in this thesis to validate the parameters and generalization capabilities of a model. The general approach for *k-fold* CV is to split the dataset into k subsets, and apply the following procedure to each of the k folds:

- Use $k - 1$ of the folds to fit a model
- Hold out the remaining fold to validate the fitted model

Furthermore, *repeated k-fold* CV is employed, in which the above process is repeated p times. The final performance measure is then the average of the scores on the test sets left out. This method is very computationally expensive since $k * p$ models are being fit, but it is extremely efficient with the data, while additionally removing selection biases with sufficient folds and repeats.

2.2 Linear Regression

Up to now, the plasma physics community has used log-linear regression to create scaling laws like that from Lorenzo et. al. A general overview of Linear Regression can be found [SOURCE]. Additional details that are used in this thesis are as follows:

- By minimizing the MSE through OLS, the scalar linear coefficients w_i corresponding to the control parameter x_i can be determined, and from these coefficients we learn the linear correlation of an engineering parameter and n_e^{ped}, i.e., if the coefficient in front of the plasma current, w_{I_P}, is positive, then as I_P increases, so will the prediction of n_e^{ped}.

- By adding a regularization term (L^1 norm of the weights) to the MSE cost function, the coefficients will be minimized as well, resulting in some coefficients becoming 0. From this it can be determined if a engineering parameter is 'useful' in the context of predicting point estimates of n_e^{ped} using linear regressors, and can reduce dimensionality when possible. This is known as LASSO [SOURCE].

- The uncertainty in the prediction can be determined by transforming the weights from scalars into normal distributions with mean μ centered around the scalar coefficient, and spread σ^2 representing the uncertainty in the coefficient, thus transforming the point estimate into a distributional estimate. This is otherwise known as Bayesian Regression [SOURCE]. clearer explanation.

By using more input parameters than that which is used in the scaling law 2, the hope is to achieve a better RMSE while additionally maintaining interpretability, i.e., attach physical intuition behind why the determined coefficients are the way that they are. We do not expect any new revelations from the coefficients determined by a linear regressor using more engineering parameters than that listed in 2, i.e., the pedestal density will still increase as the plasma current increases. The linear models analysed come from the sklearn library.

2.3 Gaussian Processes

In contrast to linear models, Gaussian Processes (GPs) are non-parametric, in that there is no function to be minimized, but rather an optimal set of functions is to be found that best characterize predicting n_e^{ped} given the engineering parameters as inputs. Much more can be read about GPs in the following sources [SOURCES]. The details pertinent to the analysis in this thesis are as follows:

- Choice of kernel (covariance function) is normally based on the 'wigglieness' of the functions one is trying to parameterize[SOURCE], but with higher dimensional space, this means nothing. Therefore the kernels sought after are those that when optimized give best predictions of the pedestal density.

- Prediction uncertainty is built into GPs as the joint-gaussian group of functions determined through optimization of the kernel will give predictions of the pedestal density that are averaged for the point prediction, and the standard deviation is the uncertainty.

- Sensitivity analysis is used to determine the relevant engineering parameters for GPs, from which the dimensionality of the input space can be reduced if parameters are deemed irrelevant. Three different forms of sensitivity analysis used: KLD, ARD, VAR[SOURCE], which all make use of thte length-scale parameters of the kernel in a GP model. The KLD, ARD, and VAR are outlined below.

- To utilize the measurement uncertainties given in the database, there are two approaches: (a) a fixed noise kernel is added on to the base kernel[SOURCE] such that the measurement uncertainties are additive to the input space. (b) Transforming the GP model from homoscedastic to heteroscedastic, where the homoscedastic model assumes constant guassian noise and the heteroscedastic takes noise values that vary for each input entry. Furthermore, the heteroscedastic model attepmts to learn the uncertainty space given

the uncertainty inputs, i.e., not only is the laten space of n_e^{ped} mapped, but also the uncertainty latent space of n_e^{ped}.

GPs scale horribly with increasing input space size, therefore the hope of using sensitivity analysis is to remove any engineering inputs if they do not improve the prediction capability of them GP model.

Additionally, since the each variable used in the fitting procedure is numerical, there exists no urgent need to try out different combinations of kernels, i.e., multiplying or adding kernels together is not analysed in this work, but something to try in the future.

The three types of sensitivity analysis used:

- Automatic Relevance Determination (ARD): The predictive relevance of each input variable is infered from the inverse of the length-scale parameter associated with that variable within the kernel. A large length scale (infinite for example) means that no correlation between the latent space and the variable in question, and thus the relevance would be zero SOURCE paananen.

- Kullback-Leibler Divergence (KLD): The KLD is a well known measure of dissimilarity between two probability distributions, and is a function of both the latent mean and uncertainty of each distribution SOURCE Kullback Leibler 1951. In this case, the input space is 'shifted' via the pertubation of a single variable's values, and the KLD of the resulting new latent space is measured against the unperterbed case. A large change in the KLD indicates that the single variable that was perturbed has high prediction relevance. [SOURCE paananen]

- Variance of the Posterior (VAR): The same method of perturbation applies, but instead of calculating the KLD, variability in only the latent mean of the fitted GP is calculated. [SOURCE paananen]

The gaussian process models used in analysis are adapted from the GPy libarary.

2.4 Random Forests

Another popular non-linear model is the ensemble of decision trees [SOURCE] that is the Random Forest(RF). More can be read about here [SOURCE], but the details pertinent to the thesis are stated below.

- RFs are fitted using boostrap aggregation (bagging). Each decision tree within the forest is fit from a 'bag' of random samples drawn from available training entries, meaning not every tree will see every available training sample, allowing for the calculation of the average error for each sample using the predictions of trees that do not contain the sample in their bag. This allows us to approximate how many decision trees to use in the forest, as the OOB error will eventually stabalize. The bag consists of a pre-determined number of features which are also randomly sampled, which allows for the determination of the optimal number of inputs to sample, as well as which inputs are optimal. [Elements of Statistical Learning p592]

- UQ in prediction can be determined by taking the standard deviation of predictions from all of the decision trees that make up the forest.

- A variant of RFs called Extremly Randomized Trees (ERTs) will also be compared. The two main differences between RFs and ERTs are (a) decision trees in ERTs sample entries for their bags without replacing such that no decision tree contains any the same entries

and (b) nodes in decision trees are split based on different critera; RFs convert parent nodes into two homogeneuos nodes by choosing the split that minimizes the MSE, whereas ERTs convert the parent node into two child nodes via a random split. [ERTs Pierre Geurts 2006]

Random Forests and Extremely Randomized Trees offer little interprability in comparison to parametric models, but by quantifying how much the impurity of a node decreases (a pure node has no child nodes) with the probability of reaching that node, the relative importance of the feature housed in the node is determined. Using this, we can get insight into which features are driving the predictions of n_e^{ped} for RFs and ERTs. The RFs and ERTs used in this analysis are adapted from the sklearn library.

2.5 Artificial Neural Networks

Numerous previous studies have investigated the reasons to which why artificial neural networks (ANNs) work [SOURCE]. Since ANNs are very delicate, the primary goal in this thesis is to probe the hypereparameter and architecture spaces for future research to build on top of. The ANNs used in this thesis are all fully-connected feed-forward networks[SOURCE]. Work has already started on using ANNs in predicting pedestal quantities, METNION ANDREAS SOMEHOW.

- Hyperparameter & Architecture search: mini-batch size, learning rate, number and size of hidden layers, activation functions, layer types (dense, cross), length of training, regularization (batch normalization, weight decay), early stopping. The optimal hyperparameters and architectures determined are those that achieve the lowest RMSE through the cross-validation process.

- Prediction uncertainty is obtained through ensembling many ANNs of similar architectures such that the standard deviation of all the predictions in the ensemble is the uncertainty in the ensemble prediction.

The ANNs used in this analysis are built using pytorch.

2.6 Meta-Modeling

Another use of the uncertainties stored within the database is for a given entry to generate normal distributions with mean of a parameter value and spread of the uncertainty, and by sampling from the generated distribution, it is possible to create new 'synthesized' entries. As seen in the next section, many models can predict well for $n_e^{ped} \leq 10$, but struggle for densities higher than that. By including synthesized values in the fitting procedure the hope is that models would be able to predict better for the higher n_e^{ped} values.

3 Results

Each model analysed is fit using the following list of main engineering parameters as inputs:

- $I_p, B_T, a, \delta, M_{eff}, P_{NBI}, P_{ICHR}, P_{TOT}, q_{95}, \Gamma, H, V_P$.

Through the use of cross-validation, the hyperparameters of each model are tuned such that optimal performance is achieved on the average performance on each fold subset. The relevant hyperparameters, and how they were determined is discussed in each indivdual model subsection, as well as the effect of meta-modeling when applicable. Table with performance metrics for optimal hyperparameters for each model can be found in Table 4.

3.1 Linear Regression

A linear regression model without an intercept was fit using the 'as is' control parameters as inputs, and we can see in Figure 3 that by including more parameters than the original 5 used in the scaling law that the RMSE and MAE improve.

Through the use of regularization via the LASSO method, the following features were deemed 'unimportant' (their coefficients dropped to 0 or near 0): H, M_{eff}, B_T, V_P. A new model with reduced dimensionality was fit by removing the unimportant features from the input space and it was seen that the prediction quality decreased dramatically with any reduction of dimensionality; in removing H or M_{eff} from the input space, the RMSE and MAE increased to much above that of the scaling law, while when removing V_P or B_T, there was an increase of 1.5 in the RMSE and MAE. The results of LASSO can be slightly misleading, as linear regressors and their regularized extensions are prone to problems when working with many correlated variables. and many of the variables in the input space are correlated (e.g., $P_{TOT}, P_{NBI}, P_{ICRH}$), which may be why the best RMSE and MAE are achieved when all variables are included.

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

Table 3: Coefficients determined by Bayesian Linear Regression. Each coefficent is a normal distribution with mean μ and spread σ^2

Fitting a Bayesian Ridge regressor using all of the available engineering parameters yielded new coefficients (as well as the uncertainties therewith), which can be seen in Table 3. From the coefficients and their uncertainties, the general uncertainty in the point prediction of n_e^{ped} can be ascertained. The uncertainties were normally distributed and range from 1.64 to 1.8 ($10^{19}\mathrm{m}^{-3}$). Having high uncertaity is good when the prediction is far from the ground truth (high residual), but for predictions on $n_e^{ped} \geq 8.5$ the uncertainty no longer covers the residual. Just like the scaling law, the predictions from the Bayesian linear model taper off at around $n_e^{ped} \geq 8.5$, which suggests that parametric models like linear regressors are generally unable to capture the hidden secrets of higher pedestal density heights from the given set of input parameters.

3.2 Gaussian Process

To determine the results of sensitivity analysis, a GP model with a RBF kernel with added constant bias term is setup and fitted using all of the available parameters. The model is optimized to the maximum of the marginal likelihood, and the relevance of each variable is calculated using ARD, KLD, and VAR. This process of optimization is repeated 5 times and

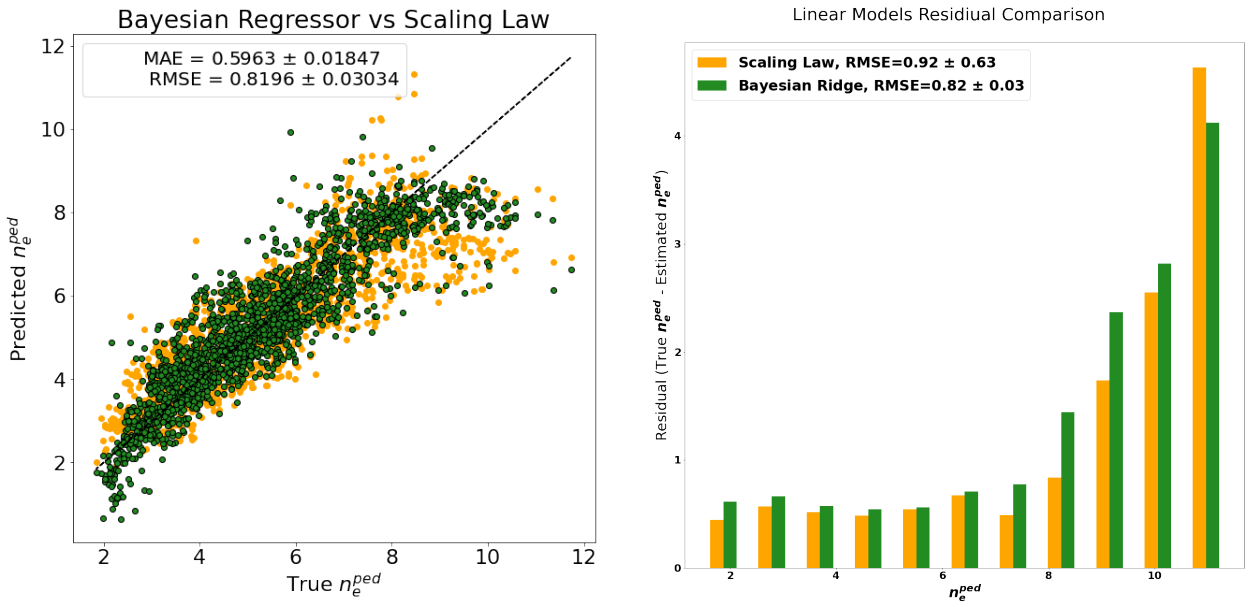


Figure 3: Comparison of a Bayesian Ridge Regressor fit using all available engineering parameters against the log-linear scaling law give in 2. **Left:** The predictions of a Bayesian Regressor (green) vs the scaling law (orange) with the ground truth being the black dotted line. **Right:** The Residual comparison of the bayesian regressor (green) and the scaling law (orange)

the results over all 5 are averaged. We can see in Figure 4 that each method gives relatively similar results, and the order of importance of the three methods averaged is relisted below.

- $\delta, a, I_p, V_P, P_{NBI}, \Gamma, P_{TOT}, q_{95}, P_{ICRH}, B_T, H, M_{eff}$

The sensitivity analysis for GPs suggests that H and M_{eff} do not aid in predicting the pedestal density, which may be due to the filtered dataset used for fitting being exclusive in its values for H and M_{eff} (only deuterium experiments were considered), I would expect this to change if a wider range of fuelling elements were included in the dataset, and for future work do not expect M_{eff} to rank as low as it does. Each sensitivty analysis also ranks B_T low, which is most likely due to the inherent correlation between $B_T, q_{95}, \text{and} I_P$, as most of the information of B_T is contained within q_{95}.

Using the results from the sensitivity analysis, the performance vs dimensionality was measured for four homoscedastic GPs of varying kernels and is plotted in 4. It can be seen that the top performing kernels were the MLP and Rational Quadratic (RQ). We observe also that the MLP and RQ do not improve *after* B_T is added to the input space, i.e., confirming the results of the sensitivity analysis. This means that in regards to the current dataset, it is unnessecary to supply H and M_{eff} to a GP, which will reduce the computation time when fitting. However, this reduction can be subject to change, since, as stated before, the dataset used was exclusive in its choice of H and M_{eff}, thus for multi machine or multi element datasets, the importance of these two variables may be more than what was found here. Nevertheless, the remainder of the GP models that are analysed in this section do not use H or M_{eff} during the fitting procedure.

To determine the effect of uncertainty propagation, the two approaches described in Section 2 are applied; (a) a fixed kernel with n_e^{ped} measurement uncertainties along the diagonal is added to a base kernel, (b) a heteroscedastic model is fixes the built in noise variance component of GPs to be the measurement uncertainty of n_e^{ped} such to learn the latent space of the used uncertainty. This process is done for both the RQ and MLP kernel and compared to the homoscedastic models. We see from Figure 5 that the homoscedastic slightly outpreforms the heteroscedastic model, yet their uncertainties could not be more different. As the

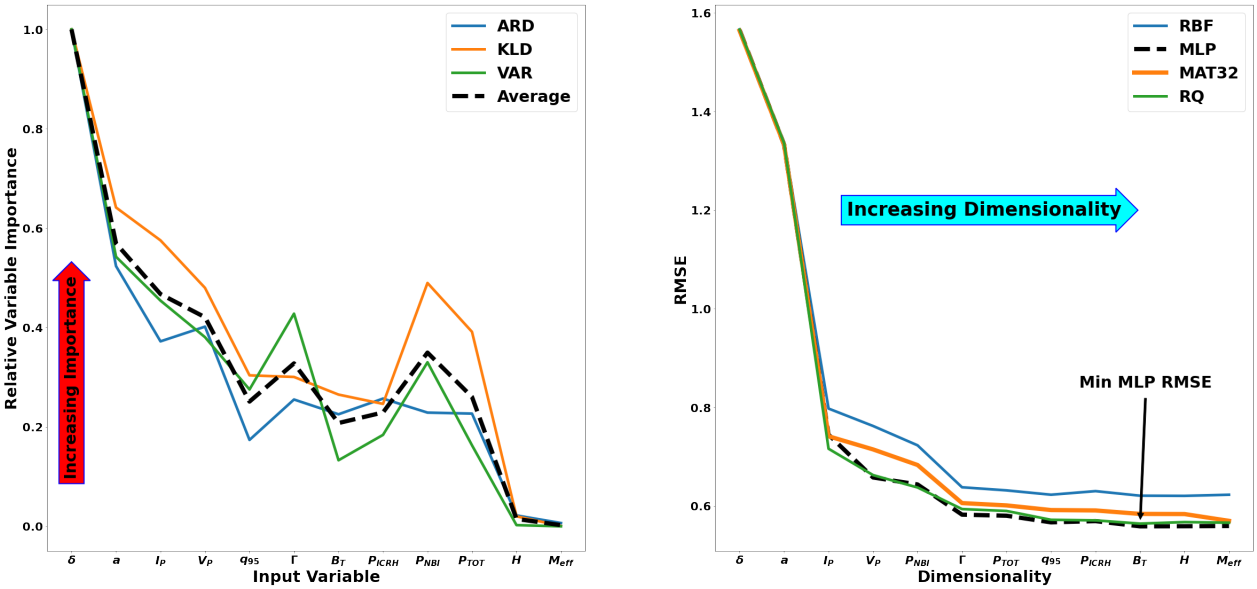


Figure 4: Steps towards dimensionality reduction through the use of Gaussian Processes. **Left:** The result of sensitivity analysis as well as the average of the three types used plotted in the dashed black line. **Right:** The dimensionality order of input variables comes from their ranking via the average of the three sensitivity analyses (dashed black line in diagram to the left). For each kernel, a GP model is fit using cross-validation (5 folds, 5 repeats) for each additional dimension of data, starting with 1d input of δ, followed by 2d input of δ, a and so on. Then the RMSE is calculated on folds left out and averaged across all folds.

heteroscedastic MLP model attempts to learn the uncertainty space, we can see that although the predictions for $n_e^{ped} > 10$ are furthest, its uncertainty is much lower. Since heteroscedastic GP learns the propogated uncertainties, the generated prediction functions end up being much closer to eachother, resulting in a lower variance in prediciton. This is very different from the homoscedastic and fixed models, which although perform very well, suffer from having their prediction generating functions far apart (a variance $\geq \pm 1.2$), regardless of prediction accuracy.

We also compare how the GPs predict on each type of impurity as seen in Figure 6. It is well known that Nitrogen (7.0) and Carbon (6.0) play a large roll in the pedestal, and therefore models could in general have the most trouble in predicting these quantities [SOURCE]. On the other hand, Neon (10.0) and Argon (18.0) also play a roll on the pedestal (although for different reasons than Nitrogen and Carbon), yet the GPs are able to predict more accurately on those. Nearly three-quarters of the entries in the filtered dataset are unseeded (0.0), and of the remaining third, about 80 percent have Neon seeding, and even fewer of the rest (for example only 3 entries have Argon or Oxygen (8.0) seeding). Even with the equally few amount of entries for both Carbon, Nitrogen, Argon and Neon, the GP models are still able extrapolate the relations of Argon and Neon better than the entries with Carbon and Nitrogen. This suggests that still there exist pedestal dependences on Carbon and Nitrogen seeding that current GP models can not extrapolate.

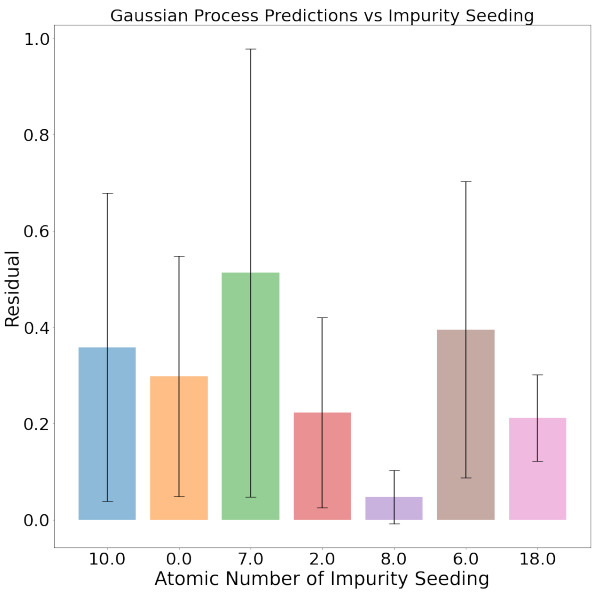


Figure 6

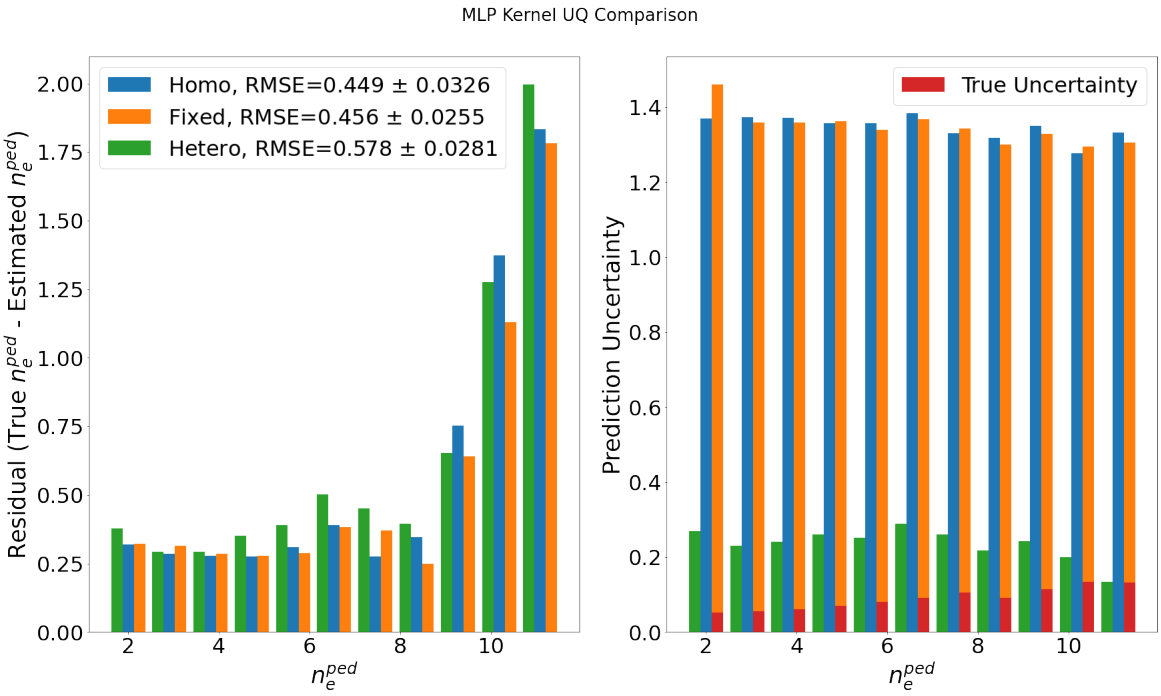


Figure 5: Three different methods of uncertainty quantification are compared for a GP with an MLP kernel. **Left:** The true values of n_e^{ped} are grouped into 11 equaly sized bins and the distance between the predictions of those values and the true values are caluclated. The residuals are averaged across each bin and compared between the methods. **Right:** The same binning procedure, but with averaging the prediction uncertainties of the same three methods are compared.

3.3 Random Forests

To deterimine the optimal size (number of decision trees) of an RF and ERT, the out-of-bag error is measured. Additionaly, the optimal number of features to sample in creating each node within each tree can be determined using the OOB, and the results are plotted in Figure 7. We see that after 254 decision trees in the RF, the OOB error does not improve, i.e., the forest reaches its maximum generalization capability at 254 decision trees. Furthermore, the optimal number of features to sample is 5. This is completely different for what occurs with the ERTs, where the optimal number of features to sample is 12 (all), and the number of trees is 142. The larger number of features sampled by the ERT compared to the RF is most likely because of the random splitting of nodes that ERTs make use of in creating their trees, such that they need to make use of all the features in order to generalize better, whereas the RF aims to minimize the MSE with their splits, thus not requiring all the inputs for an optimal split. For both models, overfitting begins to occur as more trees are added past the minimum OOB, and although the RMSE may improve, the generalizibility does not.

Plotted in Figure 9 are the resulting predictions from the optimal RFs and ERTs (determined from OOB above), and the uncertainties in the predictions ascertained from the standard deviation between each of the decision tree within the ensemble. It is clear that the individual predictors in the random forest do not vary as much as that of the homoscedastic/fixed Gaussian Process models. Additionally, the uncertainties are able to generally be around equal to that of the residual for the corresponding bin. This is good, as the model is able to more or less provide an uncertainty that covers its residual. This does not hold for $n_e^{ped} \geq 10.0$, but it is certainly within the ballpark.

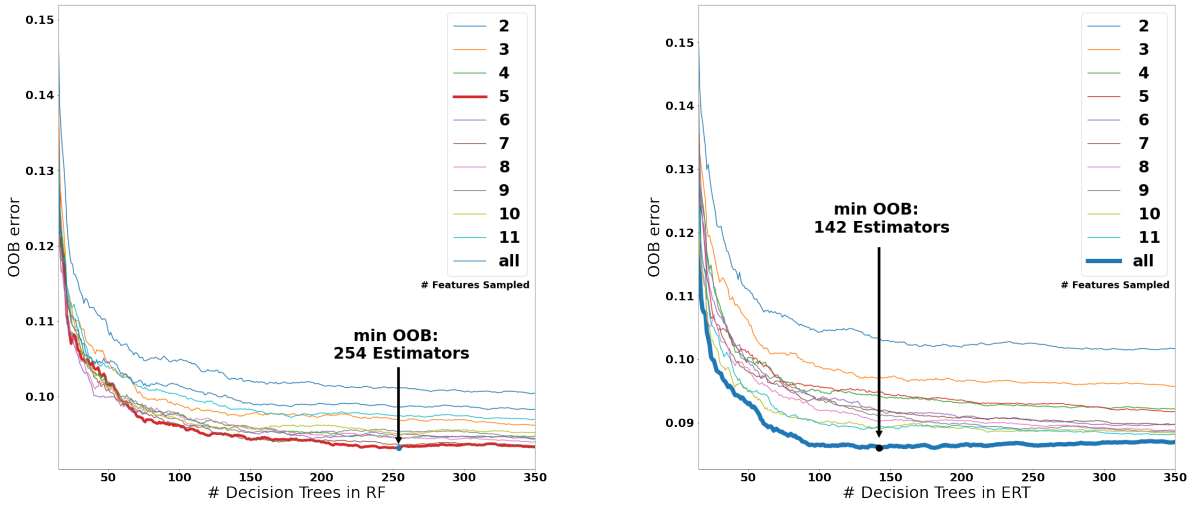


Figure 7: OOB error vs number of decision trees of **Left:** Random Forests **Right:** Extreme Random Trees. No cross validation is necessary for this procedure, since the process of bagging inherently prevents all trees from seeing the same data. The colored lines correspond to different RFs and ERTs which vary by the amount of features to sample when splitting a node.

Metamodeling had generally no effect on RFs and ERTs, as seen in Figure 8. For the RF, splits in the nodes of the decision trees are made only when the RMSE of the prediction decreases, whereas for the ERT the split is randomly selected from the distribution of parameters within input space. Although only up to 500 meta model samples are visualised in Figure 8, the sporatic bouncing of the RMSE between 0.57 and 0.58 repeats for when even 2000 synthesized entries are added! This suggests that in order to minimize the MSE of predictions across the entire dataset, RFs and ERTs ignore the additional entries and in conclusion that meta-modeling has little to no effect on ERTs and RFs.

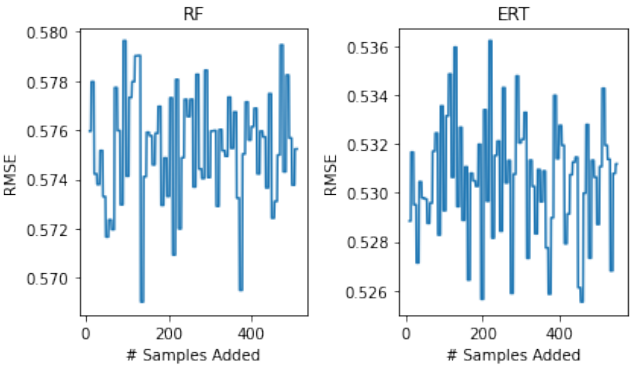


Figure 8: Effect of Meta Modeling in RF and ERTs. The # of synthesized samples added into dataset is plotted against the resulting RMSE of an RF/ERT that is fit with the additional number of synthesized samples added into its training dataset.

3.4 Artificial Neural Networks

From Figure 10, we see that shallow ANNs ($3-5$ hidden layers) perform the best, therefore the hyperparameter optimization process is focused on these networks. Since the main engineering parameters vary in magnitude (scalar value of P_{TOT} is much greater than that of q_{95}), each parameter is scaled such that it has a mean of 0 and standard deviation of 1. Throughout the hyperparameter optimization process, each model was trained and tested using the repeated cross-fold validation method, with 5 folds and 5 repeats,

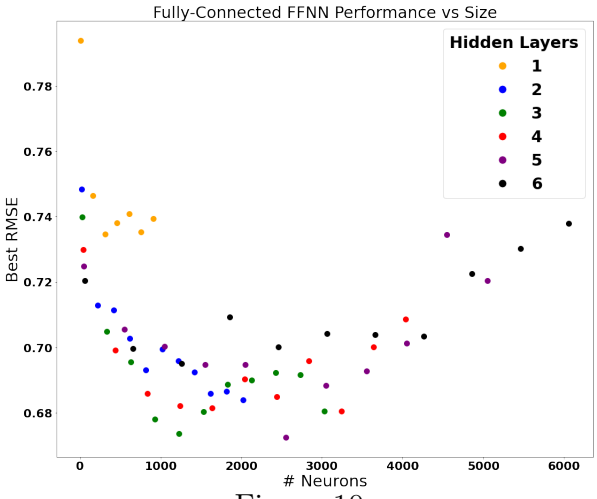


Figure 10

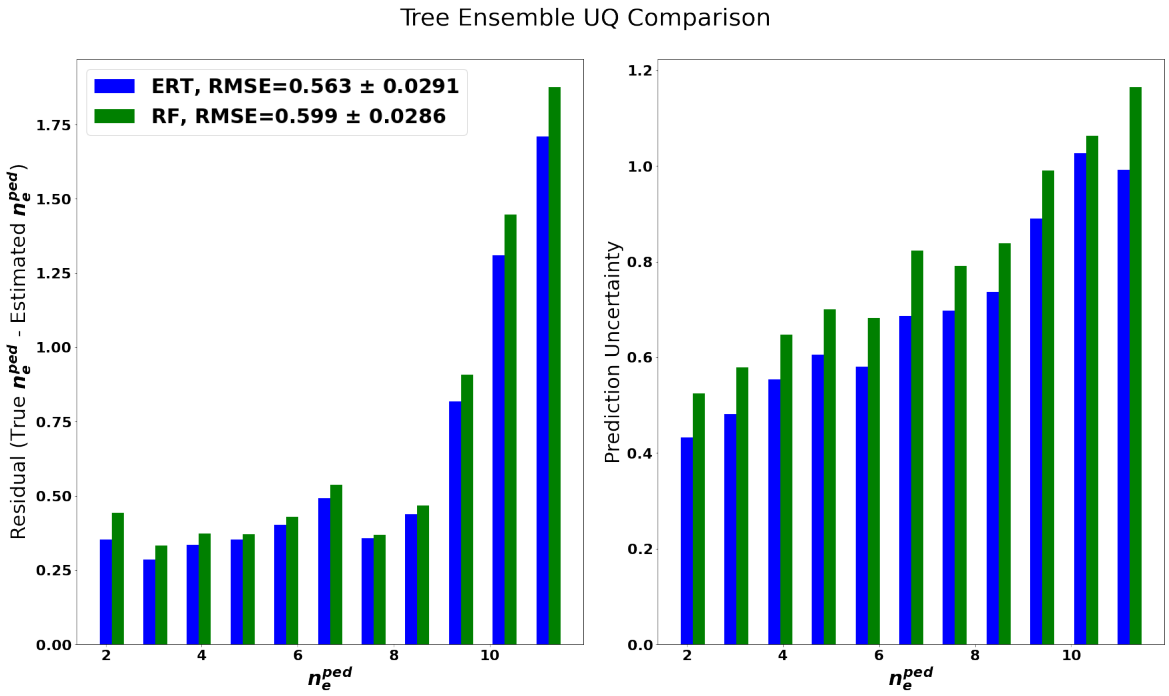


Figure 9: Comparison of predictions and uncertainties between Random Forests (green) and Extreme Random Trees (blue). **Left:** The residual between predictions and the true values,**Right:** The prediction uncertainties on the same bins of the residuals.

and the average RMSE of predictions on the
left out sets is the overall performance of the ANN. The first hyperparameters to be optimized are the learning rate and mini-batch size, and using random search, the optimal MBS and LR were found to be 396 and 0.004 respectively. Considering the dataset is around 2000 entries, the mini-batch size is relatively large, and the learning rate could be considered very small for such a large mini-batch size. However, since each training epoch, the data is randomly shuffled, it is possible that the training samples in each batch 'compete' with eachothers gradient. For example, the training samples of $n_e^{ped} \geq 9.5$ pull the model weights in an opposite direction than that $n_e^{ped} \leq 9.5$. The gradient updates applied by the learning rate for this mini-batch size seemed to balance the effect of the training samples, thus resulting in the best training/testing performance.

Then, via grid search (across all available activation layers offered by Pytorch), the optimal activation function was deterimend to be ELU (Exponential Linear Unit), a close cousin to the well known ReLU (Rectefied Linear Unit) [SOURCE]. Both are ridge functions that act on a linear combination of the input variables, but since they are applied element-wise (for each node in each layer), they are non-linear. Since the above tools like GPs and RFs are non-linear models, it makes sense that a non-linear activation function performs the best.

Using these hyperparameters, models of varying sizes of hidden layers were fit, and as seen in Figure 10, the initial ansatz of optimal hidden layers was determined to be either 3, 4 or 5, with between 1000-2000 total nodes (split between each of the hidden layers). This criteria was used as a space for further architecture search via random search. The optimal sizes of each layer for 3, 4, and 5 hidden layer networks is listed below.

- **3 Hidden Layers**: 483, 415, 254
- **4 Hidden Layers**: 636, 537, 295, 261
- **5 Hidden Layers**: 390, 484, 678, 290, 284

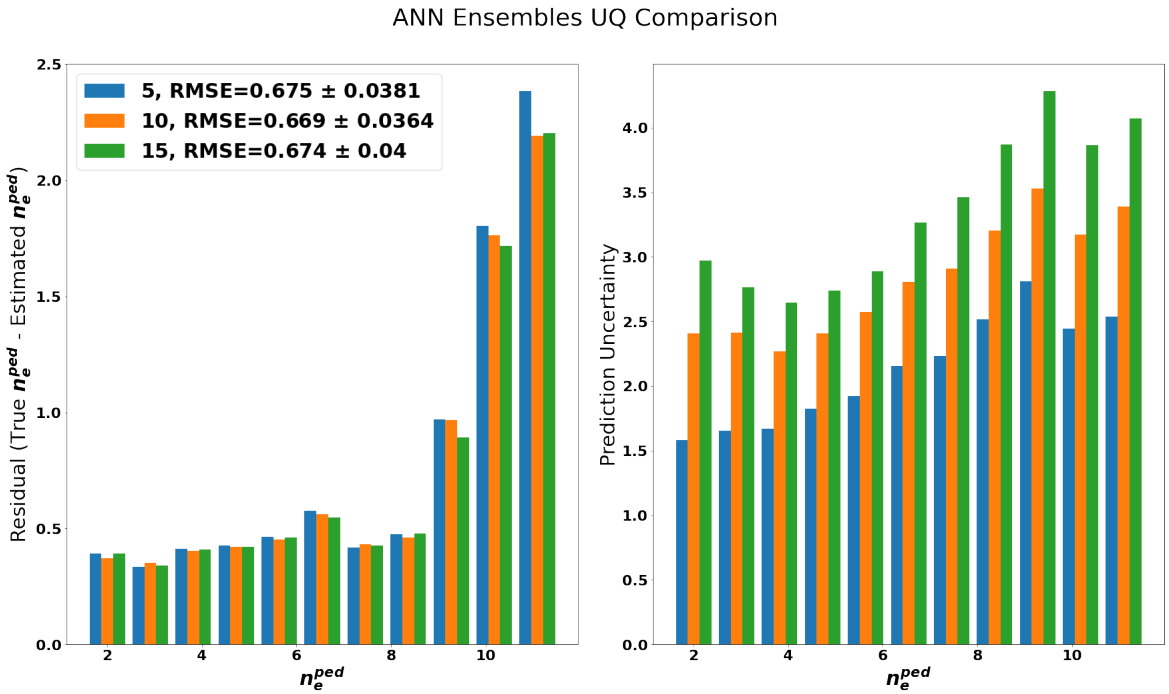


Figure 11: A three layer feed foward fully connected ANN with layer sizes 483, 415, 254 was transformed into ensembles of sizes 5 (blue), 10 (orange, 15 (green).

Of the networks listed above, the one with 4 hidden layers performed the best, with an optimal RMSE of 0.6596 ± 0.023. It was generally seen that shallow networks (≤ 4 hidden layers) with steadily decreasing layer size performed the best, whereas the larger networks performed best with this 'bell' shaped sizes, as seen in the 5 hidden layer network above. There could very well be better sizes out there, but the general idea of large first hidden layer followed by steadily decreasing layer sizes performs the best.

At the end, the best performing 3 layer ANN was used to make an ensemble, with each ANN within having different initial weights, and the prediction and uncertainty was compared across varying ensemble sizes. The ensemble was fitted in the same cross-validation format, with 5 folds and 5 repeats. We see in Figure 11 that as the ensemble size grows, there is a slight decrease in RMSE and thus improved prediction quality. However, this comes with the cost of higher prediction uncertainty, where the uncertainty in the ensemble with 15 ANNs has nearly double the uncertainty for that of 5 ANNs. Although the prediction uncertainty covers the spread, it does so by a margin that resembles the GPs, however they do increase as we reach higher pedestal densities unlike that of the GPs.

Meta modeling as an additional form of UQ did seem to improve predictions for high n_e^{ped}, while sacrificing the overall performance. This was unique for the ANNs, as all other usages of meta-modeling for other ML tools either did not affect the model or had only adverse effects to performance.

3.5 What didn't work

Many different machine learning models not listed in the above analyses were initially tested, but due to their underperformance compared to RFs/GPs/ANNs, they were abandoned before delving deeper into the others. For example, Nearest neighbours methods like K-nn and R-nn were tested and abandoned. Radius neighbours especially so, which although they could overfit extremely well with sufficiently small radius, they can not generalize for data entries which they are not fitted with, most often providing no prediction. Support vector machines

(SVMs) were tested, but due to the difficulties of deriving the prediction uncertainty and general underperformace, they were dropped. Additionally, SVMs work well when there is a clear seperation between regression points, but as we have seen, for $n_e^{ped} \geq 9$ there is no clear cut discrepncies, so SVMs offer little to no utility. Additionally, it is difficult to obtain uncertainties in predictions when using neighborhood methods and SVMs, and one of the goals of this thesis was to analyse models that could provide uncertainties. Other ensemble methods like AdaBoost showed to perform equally or slightly worse than RFs and ERTs, but some scope was needed in this thesis, so they were ultimately dropped from analysis. The combination of multiple model types into an ensemble (e.g. voting ensemble of ERT, RF and GPs) proved only slightly beneficial, and may be looked into futher in future research.

Model	RMSE	MAE
Scaling Law	0.9203 ± 0.63	TBD
Linear	0.8166 ± 0.0605	0.5956 ± 0.0379
GP	0.4566 ± 0.0217	0.3395 ± 0.01383
RF	0.5938 ± 0.0352	0.4225 ± 0.0191
ERT	0.5623 ± 0.0368	0.3927 ± 0.0199
ANN	0.6126 ± 0.0694	TBD

Table 4: The optimal relevant hyperparameters are chose for each model type, and the best RMSE and MAE are caulculated by averaging the results across each fold and repeat of the repeated cross-validation method. Uncertainty in the calculated RMSE and MAE is derived from the standard deviation of the RMSE across each fold.

4 Conclusion and Outlook

It is clear that through the use of non-linear machine learning models, there can be major improvements towards accurately predicting the pedestal density height while only using main engineering parameters.

The models analysed are ranked via their respective performance on unseen data through the use of cross validation, which is given in Table 4. It is hard to point to a clear winner among the non-parameteric/linear models, as they all score relatively close to eachother, but it is easy to see that all ML tool analysed outpreform the scaling law. Strictly speaking, the homoscedastic Gaussian Process model with an MLP kernel achieved the lowest RMSE and MAE. All models could predict well on pedestal densities less than 9.5, however they also all struggled on densities higher than that. Future work would include analysis on how splitting the data into different subsets (e.g., two subsets, one with low density, one with high density) and fitting models on the independent subsets affects predictions. This could be extended not just for splitting high and low density, but also for example low and high q_{95}. The non-linear models performed worse on Carbon and Nitrogen seeded entries than they did for other impurity seedings, therefore future work would include further independent analysis into why that happens.

For each model, the prediction uncertainty was ascertained. The prediction uncertainty for Random Forests, Extreme Random Trees, aptly covered the residuals of their predictions without being 'overly cautious', while the homoscedastic and fixed kernel GPs generally held relative constant and high uncertainties for each prediction. We were able to utelize the measurement uncertainties of n_e^{ped} in the JET pedestal database by propogating them into a heteroscedastic Gaussian Process model, and were ultimately able to (roughly) map the latent uncertainty space. This could prove useful should these machine learning models every be used as surragate models in simulations. Future work on uncertainty quantification would likely include using the main engineering parameters, for example using a fixed kernel like was done for n_e^{ped}.

The effect of meta-modeling was limited to RFs and ERTs, but from this we were able to ascertain that there is no effect on these types of models. Future work would include using meta-modeling on GPs and ANNs.