# Machine Learning Analysis on the JET Pedestal Database

**Adam Kit**

**Advisors:** Cichos, F. and Groth, M and Järvinen, A.

A thesis presented for the degree of
B.Sc. Physics

Universität Leipzig
Germany
17.06.21

# Contents

# 1 Introduction

Harnessing 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 [1]. The field of MCF research is entering the era of superconducting, reactor-scale, long pulse devices, such as ITER and DEMO [2, 3]. 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 from experimental approaches to theory based predict-first and plasma flight simulator methods [4, 5]. To bridge the gap between computational and experimental efforts, and to rapidly design reactors, there exists a need for data-driven approaches to produce models through the use of machine learning (ML). The topic of this thesis is to analyze and compare predictive ML tools in estimating plasma parameters based on experimental data. The focus is on the plasma density in the edge of tokamaks, where high performance scenarios typically establish an edge transport barrier and plasma pedestal.

## 1.1 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 (Fig. 1a), 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, the edge plasma is magnetically diverted to a separate divertor area, such that the LCFS is not in direct 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, $nT\tau_E$, must be high enough [6]. A deuterium-tritium plasma is considered ignited when the heating provided by the resulting 3.5 MeV alpha particles is sufficient to overcome the energy losses. The highest achieved triple product in MCF devices thus far is $1.53 \times 10^{21}$ keVs/m$^3$ in the JT-60 U tokamak operating with deuterium plasmas [7]. ITER and DEMO plan to achieve triple product values on the order of $10^{22}$ keVs/m$^3$ operating with deuterium-tritium plasmas [1].

The maximum pressures achievable in MCF devices are limited by magnetic field strength and magnetohydrodynamic (MHD) instabilities. The field strength is limited by the capabilities of the available superconductor technology such that, in conventional tokamak reactor designs, the fields in the center of the plasmas are around 5-6 T [8]. A useful figure of merit for an MCF device is the ratio of the confined fuel pressure to that of the magnetic field pressure, $\beta = p/(B/2\mu_0)$, which in tokamaks is always significantly less than unity for stable plasmas. Higher $\beta$ plasmas are desired for high fusion performance and power density, but the maximum $\beta$ is limited by MHD stability. Often $\beta$ is further normalized with the Troyon factor ($B_T\, a/I_P$), originating from the Troyon $\beta$ limit [9]. This is called $\beta_N$. Typically in tokamaks, the outcome of these limits is that the confined fuel pressure is a few times the atmospheric pressure, and the energy confinement time is around seconds [10].

The energy confinement time is limited by plasma turbulence, which leads to radial transport across the flux surfaces significantly faster than would be expected based on classical or neoclassical transport [2]. Typically, the turbulence modes show critical gradient behavior [2],

such that the radial gradients are limited near their critical value, and the effective transport increases with more heating power. This enhanced transport results in reduction of $\tau_E$ with heating power, as can be seen in common confinement time scalings, for low (L-mode) and high confinement mode (H-mode) [2]. As a result, reaching $nT\tau_E$ is 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, H-mode, was discovered in plasmas operating with the divertor configurations and neutral beam heating [11]. 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 plasma pressure 'pedestal'. To achieve H-mode confinement, a minimum amount of power flow through the edge is required [12]. H-mode ultimately provides about a factor of two increase of the energy confinement time. Therefore, H-mode is the standard operational mode expected in future reactor-scale 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 MHD instabilities, called edge localized modes (ELMs) [13, 14]. The current understanding is that high performance pedestal plasmas are limited by ideal MHD peeling-ballooning instabilities that trigger type-I ELMs [13]. 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 [15], do not reach the ideal MHD peeling-ballooning stability threshold [16]. Therefore, determining which transport or instability phenomena limit the pedestal is a very active topic of research [17, 18, 19].

Since edge transport barriers are key ingredients of future ITER and DEMO burning plasma scenarios, predictive capability is necessary for the pedestal region to confidently design the future reactors and scenarios. Due to complexity of the pedestal plasmas and the non-linear interaction of the pedestal with the core and SOL plasmas, the simulation codes for the pedestal make simplifying assumptions. The most widely used model for pedestal pressure predictions is EPED [20, 21]. EPED has been very successful for predicting pedestal pressure width and height for many present-day tokamaks, by assuming that the pressure gradient in the pedestal is limited by local kinetic balooning modes (KBM) and the total pressure by ideal-MHD peeling-balooning modes. However, these assumptions are not justified for a large fraction of the JET pedestal database [16]. EPED also takes as input certain plasma characteristics, including information about pedestal densities and density profiles, confined normalized pressure, and dilution of the plasmas by impurities. Therefore, EPED cannot be considered a fully predictive model. EUROPED aims to bridge some of these shortcomings of EPED by using other models for core transport and pedestal density [22, 23].

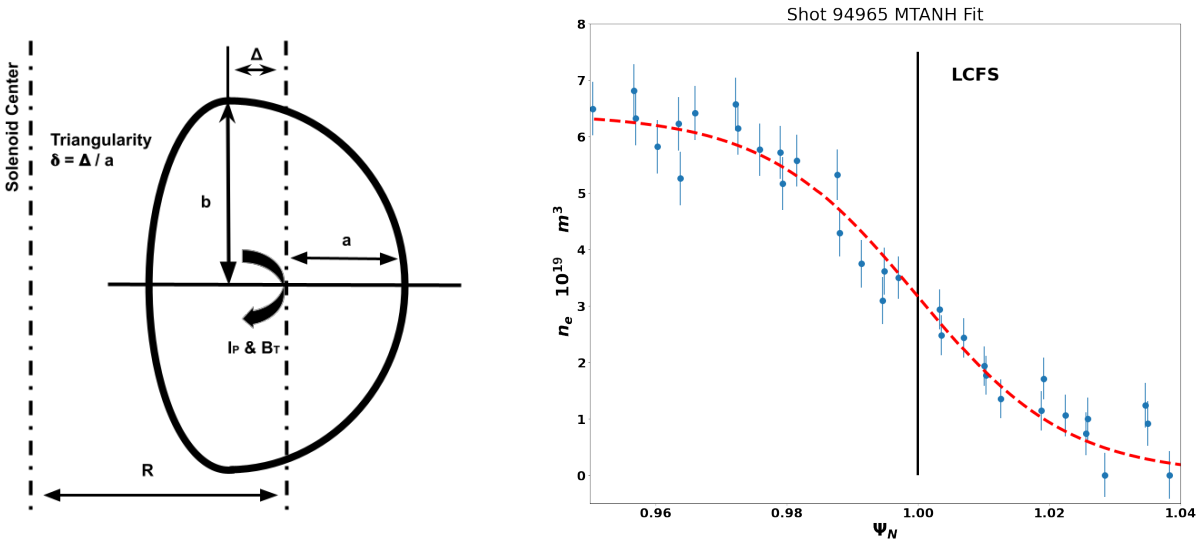
In this study, the focus is predicting the pedestal density, and ML tools analyzed are compared to an experimental log-linear fit published in [16].

## 1.2 JET Pedestal Database

The JET pedestal database contains over 3000 entries, with each entry corresponding to time averaged measurements of various plasma parameters over the course of 70-95% of an ELM cycle, representing the conditions of the highest pedestal plasma pressure prior to the next ELM. The measurements are done using high resolution Thomson scattering (HRTS)[24], and are then fitted using the modified tangent hyperbolic function (mtanh):

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

where the pedestal height, position, width, and slope inside the pedestal top are $h_1$, $p$, $w$, and $s$ respectively, and $r$ the normalized radius $\Psi_N$ (Fig 1b). Since the measurements are taken near




(a) (b)

Figure 1: **(a)** Plasma shape parameters from the cross-section of a tokamak. **(b)** HRTS measurement profiles (blue), which are fitted in real space using the mtanh equation (eq. 1) then mapped to the normalized poloidal flux coordinate $\Psi_N$ (red).

the end of the ELM cycle, the pedestal parameters should be saturated near their maximum, right before the ELM.

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, (Fig. 1a)
- $B_T$ [T], toroidal magnetic field, (Fig. 1a)
- $R$ [m], major radius of the plasma, (Fig. 1a)
- $a$ [m], minor radius of plasma, (Fig. 1a)
- $\delta$ [-], triangularity, normalized horizontal displacement of the top/bottom of the plasma from the main axis, (Fig. 1a)
- $V_P$ [$m^3$], the plasma volume,
- $H$, isotope ratio of fuel, hydrogen / deuterium, with
- $M_{eff}$, effective mass of the fueling isotope,
- $q_{95}$ [-], safety factor at the 95% flux surface. 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. $q$ is called the safety factor as low $q$ plasmas are susceptible to MHD instabilities. Typically, the baseline scenarios operate $q_{95}$ around 3-4,
- $P_{NBI}$ [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),
- $\Gamma$ [ $10^{22}$ electrons per second], gas fueling rate of the deuterium or hydrogen,
- $DC$, the divertor configuration, can take on values of C/C, V/H, V/C, V/V, C/V, C/H,

(see [16] 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),
- $P_{SD}$ [$10^6$ nbar], the subdivertor pressure.

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.

There are also global parameters stored in the database, such as $\beta_\theta^{ped}$, $\beta_N$, $Z_{eff}$. However, in this study, we are first considering a model that only utilized machine engineering parameters as input. This assumption is in contrast to EPED that also takes plasma parameters as input, including pedestal density and core $\beta$ for example. These type of models assume that these plasma parameters can be achieved via actuation of the engineering inputs.

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

| Eng. Param | Domain |
|---|---|
| $I_P$ [MA] | $[0.81, 4.48]$ |
| $B_T$ [MW] | $[0.97, 3.68]$ |
| $a$ [m] | $[0.83, 0.97]$ |
| $\delta$ [-] | $[0.16, 0.48]$ |
| $M_{eff}$ [-] | $[1.88, 2.18]$ |
| $P_{NBI}$ [MW] | $[10^{-3}, 32.34]$ |
| $P_{ICRH}$ [MW] | $[0, 7.96]$ |
| $P_{TOT}$ [MW] | $[3.4, 38.22]$ |
| $V_P$ [$m^3$] | $[58.3, 82.19]$ |
| $q_{95}$ [-] | $[2.42, 6.04]$ |
| $\Gamma$ [$10^{22}$ e/s] | $[0, 15.5]$ |
| $H$ [-] | $[0, 0.18]$ |
| $P_{SD}$ [$10^6$nbar] | $[0, 1000]$ |

A model of interest is one that uses the main engineering parameters to calculate pedestal 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 eq. (1), and the uncertainties are the fit uncertainties from the use of the mtanh function [16]. The fit uncertainties are expected to be significantly smaller than the natural scatter of the data due to the fluctuation of the plasma.

Additionally, the database entries contain the parameter FLAGS, which correspond to the specific setup of an experiment. Examples of FLAGS contained in the database: main fueling element, usage of RMPs, pellets, or impurity seeding, divertor configuration. Each entry in the database is validated either by hand or computationally, and there is a FLAG corresponding to the quality of the HRTS measurement determined by the validation [16]. Only entries that have been validated are used in this thesis. Shots with impurity seeding are used, as they make up about 600 entries. To keep the dataset simple, entries with RMPs, pellets, and kicks are excluded, as these are used to manipulate the pedestal for ELM control, mitigation, or suppression [14].

After filtering out the shots with RMPs, kicks, pellets, non-validated HRTS, and shots that do not use deuterium, the dataset is reduced to 1888 entries. The final main engineering and pedestal parameter domains are given in Tables 1 and 2 respectively.

Table 2: Domains of pedestal parameters for deuterium shots stored in the JET pedestal database after RMPs, kicks, pellets, and non-validated HRTS are filtered out.

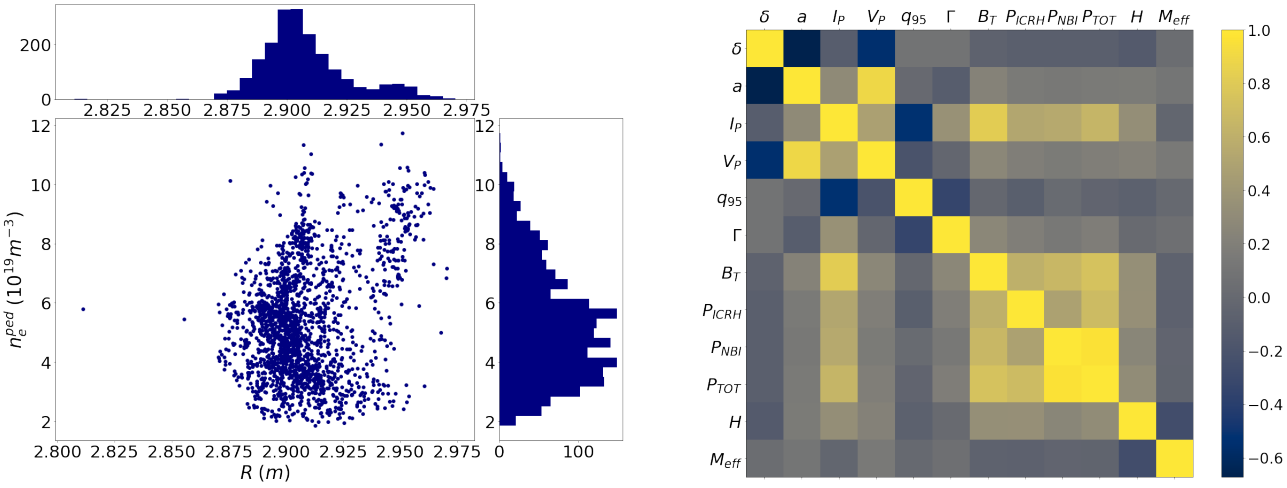
| | Height | Width [$\Psi_N$] | Position [$\Psi_N$] | Slope [-] |
|---|---|---|---|---|
| $n_e^{ped}$ | [1.849, 11.737] ($10^{19}$ $m^3$) | [0.015, 0.173] | [0.953, 1.029] | [$10^{-6}$, 0.188] |
| $T_e^{ped}$ | [0.149, 1.894] (keV) | [0.013, 0.105] | [0.926, 1.002] | [0.026, 0.502] |
| $p_e^{ped}$ |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

### 1.2.1 Empirical Analysis

Empirical analysis of the JET pedestal database was carried out [16], and has yielded a 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 focus on the pedestal prediction with ML approaches relative to the performance of the empirical scaling law. The choice of parameters in the log-linear regression by Lorenzo Frassinetti et. al. [16], was backed by physical intuition on the pedestal density height. Since log-linear regression was used, the scaling law above avoided using cross-correlated variables, which can verified in Figure 2.




(a) (b)

Figure 2: Empirical data plots of the JET pedestal database: **(a)** correlation without causation between the major radius $R$ and $n_e^{ped}$, **(b)** Correlation matrix of the main engineering parameters. A grey coloring represents no correlation, whereas blue and yellow are negative and positive correlation, respectively.

To improve prediction quality, it is 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 $n_e^{ped}$, serious questions can be raised regarding which parameters can and should be given to a machine learning model. For example, the dependence of the pedestal density on the major radius could lead to an early conclusion that with higher values of $R$, a higher pedestal height is achieved (Fig. 2a). However, this is a case of causation without correlation, or ice-cream correlation[1], and the actual culprit of the causation is the Shafranov shift; the outward radial displacement of the magnetic axis from the geometric axis that is prominently found in MCF devices [25, 26]. The shift increases with normalized pressure, which increases $R$. As normalized pressure increases with pedestal pressure, there is a cross-correlation between pedestal pressure and $R$. For this reason, $R$ is excluded from the list of inputs to the ML models in this thesis, and only when multi-machine databases with significantly larger variation of $R$ 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 only makes use of the numerical parameters available, and thus divertor configuration will not be used as an input parameter, as it is categorical.

[1]Ice-cream correlation refers to the correlation of increasing ice cream sales and increasing number of drownings in Finland during the summer. Although the variables are indeed correlated, higher ice cream sales are not in fact the cause of higher drowning rates, nor vice-versa.

Another engineering parameter that is ignored in the analysis is the sub-divertor pressure $P_{SD}$, which is correlated with the gas fueling rate, but also depends on other main engineering parameters, such as divertor configuration, input power, and wall conditions. From the filtered dataset, the values of $P_{SD}$ vary widely, 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, $P_{SD}$ is ignored, however future work may choose to filter the dataset such that inclusion of $P_{SD}$ is possible.

# 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 score penalizes predictions that are far away from the ground truth, whereas the MAE uniformly calculates the distance between predictions 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 via 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), which minimizes the mean squared error in order to find the optimal linear coefficients $\theta(w_i)$ [27]. Not all supervised learning regression algorithms minimize the RMSE or MAE to fit model parameters.

The performance of the fitted model is scored using data that the model has not observed previously. If the same data used in the fitting was used to score the model, the model would simply repeat predictions that it had been fitted with, and would fail to predict useful information on unseen data. This is called *overfitting*. To avoid overfitting a common practice in supervised machine learning is to hold 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 by evaluating the model on the test set. Depending on the performance on the test set, the *hyperparameters* of the model are adjusted to optimize the performance on the test set (an example of a hyperparameter is the number of trees in a random forest, or learning rate for artificial neural networks). However, this method can lead to overfitting on the test set, since the hyperparameters can be adjusted until the model performs 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*, in which the results are dependent on the random choice of entries contained in the training and test sets [28]. 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

A commonly used approach in the plasma physics community is log-linear regression to create scaling laws like that from Lorenzo Frassinetti et. al [16]. A general overview of linear regression can be found in [29]. 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 zero. From this procedure it is determined whether an 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. [30, 31, 32]
- By transforming the point prediction of $n_e^{ped}$ into a normal distribution we can obtain the uncertainty in the prediction. This is done by determining the optimal linear coefficients via maximum likelihood estimation instead of OLS, and thus the coefficients transform from scalars into normal distributions with mean $\mu$ centered around the scalar coefficient and spread $\sigma^2$ representing the uncertainty in the coefficient. This is otherwise known as Bayesian Regression [33].

By using more input parameters than those which are used in the scaling law, eq. (2), the intent is to achieve a lower RMSE while additionally maintaining interpretability, i.e., attach physical intuition to the coefficients determined by the linear model. However, by including more parameters, we also introduce cross-correlation to the system, and expect that the coefficients determined may vary from those in the scaling law. We do not expect drastic changes from the coefficients determined by a linear regressor using more engineering parameters than those listed in Eq. (2), i.e., the pedestal density will still increase as the plasma current increases. The linear models analyzed come from the sklearn and PyMC libraries [34, 35].

## 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 are found that best characterize predicting $n_e^{ped}$ given the engineering parameters as inputs. More in-depth analysis of GPs can be found in the following sources [36, 37, 38]. 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 [39], but with higher dimensional space, this means nothing. Therefore, the kernels desired are those that best predict the pedestal density when optimized.
- 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 [37].

- 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 are used and are described below [40].

- There are two approaches to utilize the measurement uncertainties given in the database: (a) a fixed noise kernel is added on to the base kernel 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 Gaussian noise and the heteroscedastic takes noise values that vary for each input entry. Furthermore, the heteroscedastic model attempts to learn the uncertainty space given uncertainty of the input, i.e., both the mean and *local* variance of $n_e^{ped}$ are estimated [41].

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

Three types of sensitivity analysis are used:

- Automatic Relevance Determination (ARD): The predictive relevance of each input variable is inferred from the inverse of the length-scale parameter associated with that variable within the kernel. A large length scale (infinite, for example) means that there is no correlation between the latent space and the variable in question, and thus the relevance would be zero [40].

- 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 [42]. In this case, the input space is 'shifted' via perturbing values of individual variables, and the KLD of the resulting new latent space is measured against the unperturbed case. A large change in the KLD indicates that the single variable that was perturbed has high prediction relevance [40].

- 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 [40].

The Gaussian Process models used in this analysis are adapted from the GPy library [43].

## 2.4 Random Forests

Another popular non-linear model is the ensemble of decision trees [44] that is the Random Forest(RF). General information on Random Forests can be found in the following sources [45, 46], but the details pertinent to the thesis are stated below:

- RFs are fitted using bootstrap 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 stabilize. The bag consists of a predetermined 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 [29].

- 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 Extremely 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 them such that no decision tree contains any of the same entries and (b) nodes in decision trees are split based on different criteria; RFs convert parent nodes into two homogeneous nodes by choosing the split that minimizes the MSE, whereas ERTs convert the parent node into two child nodes via a random split [47].

Random Forests and Extremely Randomized Trees offer little interpretability 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 sci-kit learn library [34].

## 2.5 Artificial Neural Networks

Numerous previous studies have investigated the reasons why artificial neural networks (ANNs) work [48, 49, 50]. The ANNs used in this thesis are all fully-connected feed-forward networks [51, 52]. Work has already started on using feed-forward ANNs in predicting pedestal quantities [53]. Since ANNs are very delicate, the primary goal in this thesis is to probe the hyperparameter and architecture spaces for future research upon which to build. We optimize the following hyperparameters: mini-batch size, learning rate, number and size of hidden layers, activation functions, layer types, length of training, and regularization. The optimal hyperparameters are those that achieve the lowest RMSE through the cross-validation process.

In this thesis, the prediction uncertainty of ANNs is determined through grouping many ANNs of similar hyperparameters into an ensemble such that the standard deviation of the predictions of each network in the ensemble is the uncertainty in the ensemble prediction.

The ANNs used in this analysis are built using PyTorch [54].

## 2.6 Meta-Modeling

Another use of the uncertainties stored within the database is for a given entry to generate normal distributions with the mean of a parameter value and spread of the uncertainty. 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 better predict higher $n_e^{ped}$ values.

# 3 Results

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

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

There are cross-correlations among these parameters (Fig. 2), and, as will be observed, some models do not perform very well due to these correlations. 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 individual model subsection, as well as the effect of meta-modeling when applicable. Table 3 presents the final performance metrics for each optimized model.

Table 3: The best RMSE and MAE of the predictions from each optimal model are calculated 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.

| Model | RMSE | MAE |
|---|---|---|
| Scaling Law | $0.9203 \pm 0.63$ | $0.7189 \pm 0.63$ |
| Linear | $0.8166 \pm 0.0605$ | $0.5956 \pm 0.0379$ |
| **GP** | $\mathbf{0.4566 \pm 0.0217}$ | $\mathbf{0.3395 \pm 0.01383}$ |
| RF | $0.5938 \pm 0.0352$ | $0.4225 \pm 0.0191$ |
| ERT | $0.5623 \pm 0.0368$ | $0.3927 \pm 0.0199$ |
| ANN | $0.6126 \pm 0.0694$ | $0.4418 \pm 0.0421$ |

## 3.1 Linear Regression

Using a Bayesian linear regression model without an intercept using the control parameters as inputs, the RMSE and MAE improve (for $n_e^{ped} < 11 \times 10^{19}\mathrm{m}^{-3}$) by including more parameters than in the reference scaling law, eq. (2) (Fig. 3). Due to cross-correlated variables, we

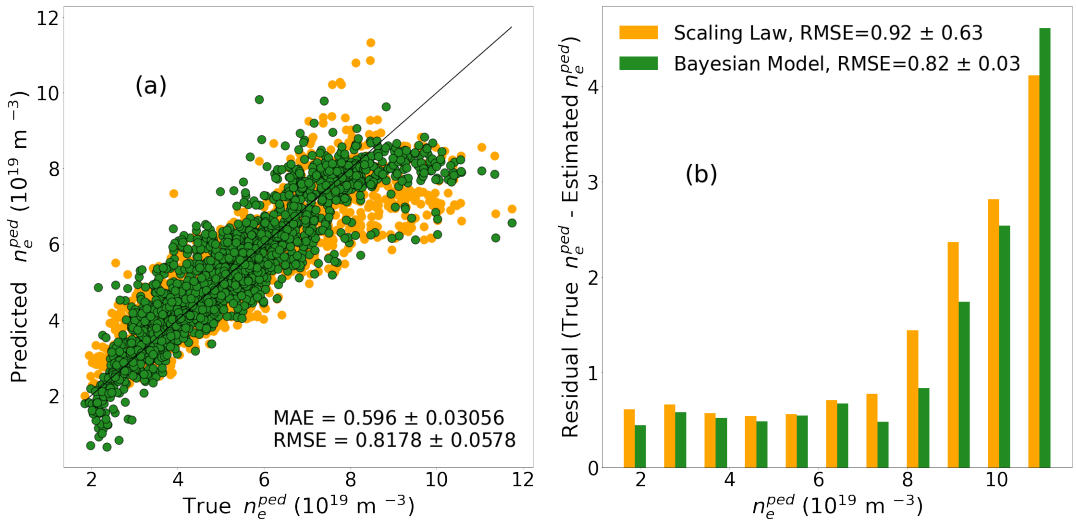



Figure 3: Comparison of a Bayesian Ridge regressor fit using all available engineering parameters against the log-linear scaling law eq. ( 2). **(a)** The predictions of a Bayesian regressor (green) vs the scaling law (orange) with the ground truth being the black dotted line. **(b)** The Residual comparison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

observed that LASSO did not perform well in reducing the dimensionality of the input space. The LASSO method deemed the following features as 'unimportant' (their coefficients dropped to zero or near zero): $H, M_{eff}, B_T, V_P$. A new model with reduced dimensionality was fit by removing the unimportant features from the input space and 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 by 0.3, while when removing $V_P$ or $B_T$, there was an increase of 1.5 in the RMSE and MAE.

Fitting a Bayesian Ridge regressor using all of the available engineering parameters yielded new coefficients (as well as the uncertainties) (Table 4). It is observed that the coefficients for $P_{TOT}$ compete with $P_{NBI}$ and $P_{ICRH}$. From Table 1 we know that there are plasmas with $P_{NBI}$ 100% of $P_{TOT}$, so the regressor balances the two out by choosing $P_{TOT}$ and $P_{NBI}$ to have opposite signs. 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 $\times 10^{19}\text{m}^{-3}$. Having high uncertainty is good when the prediction is far from the ground truth (high residual), but for predictions on $n_e^{ped} \geq 8.5 \times 10^{19}\text{m}^{-3}$ the uncertainty no longer covers the residual. Just like the scaling law, the predictions from the Bayesian linear model taper off at around $n_e^{ped} \geq 8.5 \times 10^{19}\text{m}^{-3}$, which suggests that parametric models like linear regressors are generally unable to capture the non-linear complexity of higher pedestal density heights from the given set of input parameters.

Table 4: Coefficients determined by Bayesian Linear Regression. Each coefficient is a normal distribution with mean $\mu$ and spread $\sigma^2$

| Feature | $\mu$ | $\sigma^2$ |
|---|---|---|
| $I_p$ | 0.15 | 0.06 |
| $B_T$ | 0.956 | 0.072 |
| $a$ | 2.966 | 0.479 |
| $\delta$ | 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 |
| $\Gamma$ | 0.125 | 0.007 |
| $H$ | -4.016 | 0.374 |
| $M_{eff}$ | 1.369 | 0.053 |

## 3.2 Gaussian Process

Sensitivity analysis of GPs yielded the following variable importance ranking (Fig. 4):

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

To determine these results, a GP model with a radial basis function (RBF) kernel [55] with added constant bias term was fit using all main engineering parameters, and the relevance of each variable is calculated using ARD, KLD, and VAR. This process is repeated 5 times and the results over all 5 are averaged. Each method gives relatively similar results and suggest that the correlated variables $H$ and $M_{eff}$ do not aid in predicting the pedestal density (Fig. 4a). This may be due to the filtered dataset, which was exclusive in its values for $H$ and $M_{eff}$ (only deuterium experiments were considered). This is expected to change if a wider range of fueling elements was included in the dataset, and in future work we do not expect $M_{eff}$ to rank as low as it does. Each sensitivity analysis also ranks $B_T$ low, which is most likely due to the inherent correlation between $B_T$, $q_{95}$, and $I_P$, as most of the information of $B_T$ is contained within $q_{95}$ and $I_P$.

Using the results from the sensitivity analysis, it was found that $H$ and $M_{eff}$ do not aid in improving predictive quality of GPs and that the Multi-Layer Perceptron (MLP) [56] and Rational Quadratic (RQ) [55] kernels perform the best (Fig. 4b). We observe that the MLP and RQ do not improve after $B_T$ is added to the input space, thus confirming the results of the sensitivity analysis (Fig. 4b). This means that regarding the current dataset, it is unnecessary to supply $H$ and $M_{eff}$ to a GP, which will reduce the computation time when fitting. However, this reduction may be subject to change, since, as stated above, the dataset used was exclusive

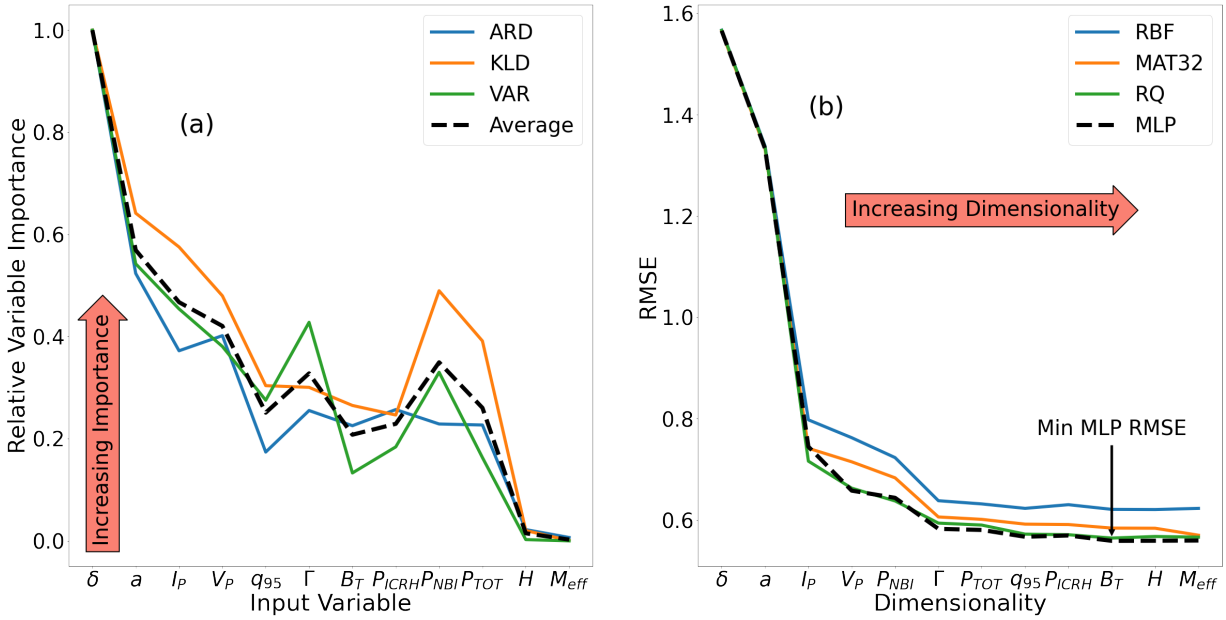



Figure 4: Steps towards dimensionality reduction through the use of Gaussian Processes. **(a)** The result of sensitivity analysis as well as the average of the three types used plotted in the dashed black line. **(b)** The dimensionality order of input variables comes from their ranking via the average 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 $\delta$, followed by 2D input of $\delta, a$ and so on. The RMSE is calculated on folds left out and averaged across all folds.

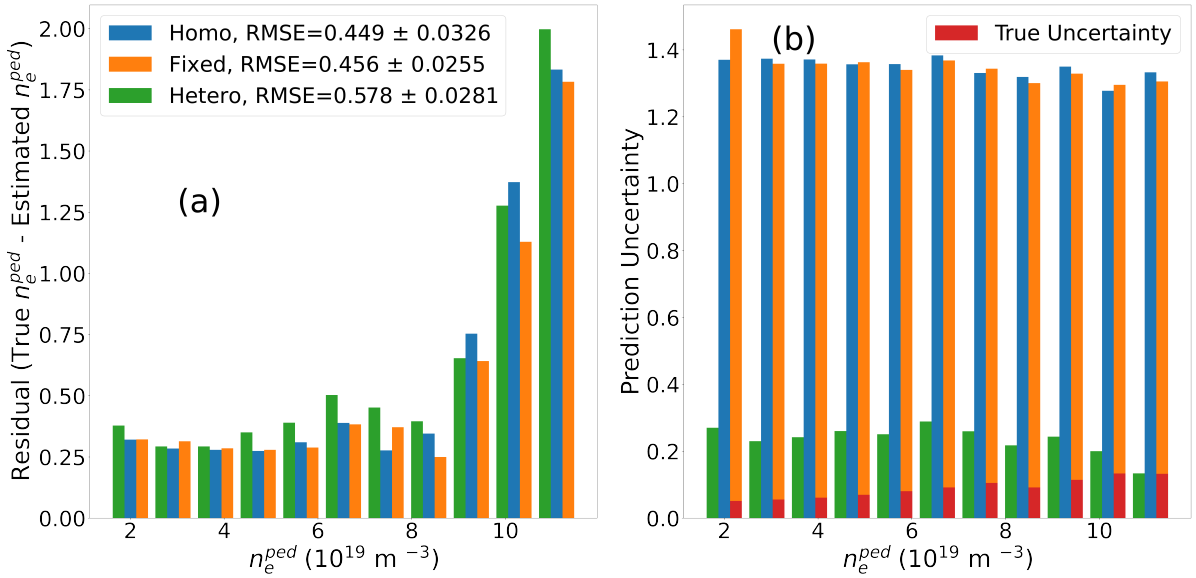



Figure 5: Three different methods of uncertainty quantification are compared for a GP with an MLP kernel. **(a)** The true values of $n_e^{ped}$ are grouped into 11 equal-sized bins and the distance between the predictions of those values and the true values are calculated. The residuals are averaged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

in its choice of $H$ and $M_{eff}$; thus for databases with multiple hydrogen isotopes included, the importance of these two variables may be more than what was found here. For example, if shots with hydrogen fueling was used, the domains of $M_{eff}$ and $H$ would expand to $[1.0, 2.0]$ and $[0.0, 1.0]$, respectively, which could potentially change the results of the sensitivity analysis. Nevertheless, the remainder of the GP models that are analyzed in this section do not use $H$ or $M_{eff}$ during the fitting procedure.

Through uncertainty propagation it was observed that a heteroscedastic model was able to map the relative measurement uncertainties into its latent posterior at the cost of decreased prediction quality (Fig. 5). 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 fixes the built-in noise variance component of GPs for each input entry to be the entries respective 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. The homoscedastic slightly outperforms the heteroscedastic model (Fig. 5). However, the uncertainty of the homoscedastic model is on average 600% higher than that of the heteroscedastic model. As the heteroscedastic MLP model attempts to learn the uncertainty space, the predictions for $n_e^{ped} > 10 \times 10^{19}\mathrm{m}^{-3}$ are further away from the ground truth than the other models, however, its uncertainty is much lower and similar to that of the propagated uncertainty. Since heteroscedastic GP learns the propagated uncertainties, the generated prediction functions end up being much closer to each other, resulting in a lower variance in prediction. This is very different from the homoscedastic and fixed models, which although performing very well, suffer from having their prediction-generating functions be far apart (variance $\geq \pm 1.2$), regardless of prediction accuracy. The heteroscedastic model tends to be overconfident at higher densities. However, the uncertainties closely match that of the actual local uncertainty, which is the goal of a heteroscedastic model (Fig. 5).

## 3.3 Random Forests

The optimal number of decision trees and features to sample to utilize in RFs and ERTs was determined using the out-of-bag error (Fig. 6).

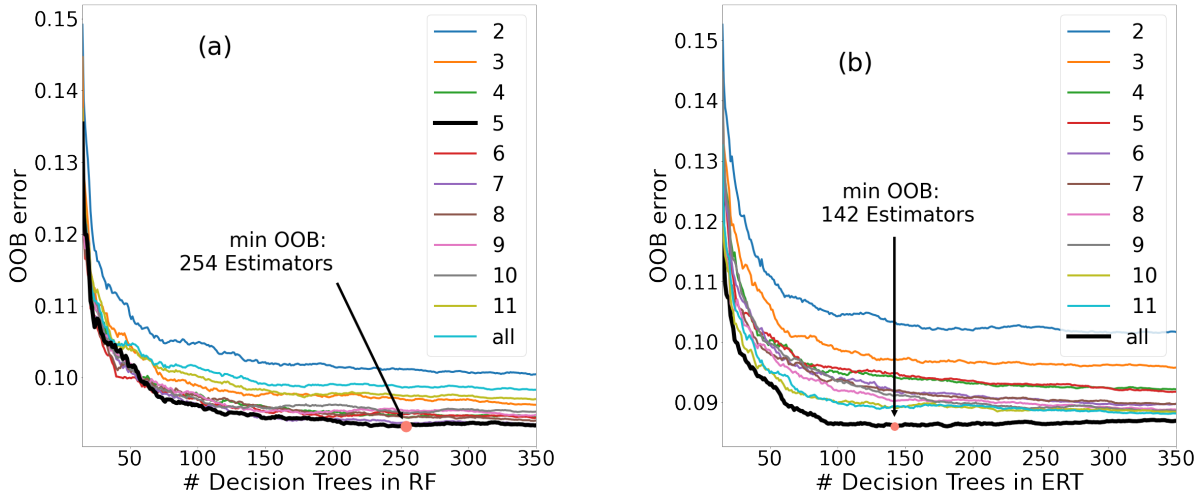



Figure 6: OOB error vs number of decision trees of **(a)** Random Forests **(b)** Extreme Random Trees. No cross validation is necessary for this procedure, since the process of bagging inherently prevents all trees from seeing the same data. The colored lines correspond to different RFs and ERTs which vary by the amount of features to sample when splitting a node.

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 from 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 due to 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 generalizability does not.

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 (Figs. 6 and 8). Additionally, the uncertainties are generally around equal to that of the residual for the corresponding bin. The model is able to more or less provide an uncertainty that covers its residual, as desired. This does not hold for $n_e^{ped} \geq 10.0 \times 10^{19} \mathrm{m}^{-3}$, but it is certainly within the ballpark.

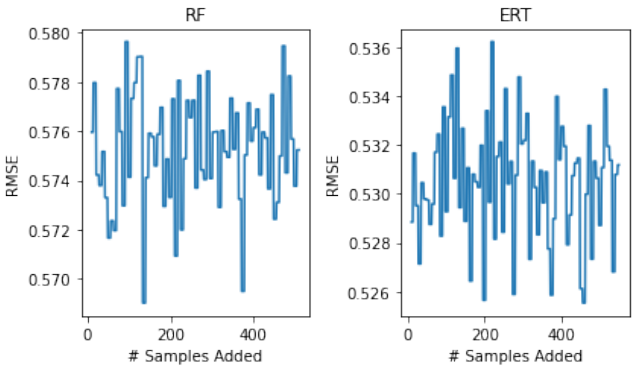



Figure 7: Effect of meat-modeling in RF and ERTs. The # of synthesized samples added into dataset is plotted against the resulting RMSE of a model fitted using the synthesized samples.

Meta-modeling had generally no effect on RFs and ERTs (Fig. 7). Although only up to 500 meta-model samples are visualized in Figure 7, the sporadic 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 we conclude that meta-modeling has little to no effect on ERTs and RFs.

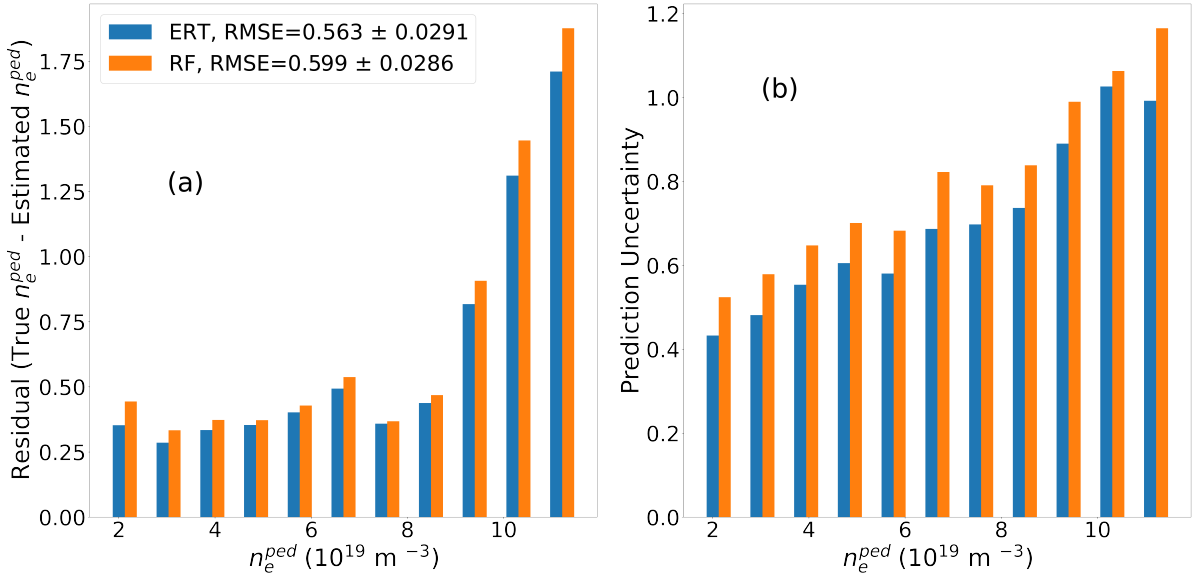



Figure 8: Comparison of predictions and uncertainties between Random Forests (green) and Extreme Random Trees (blue). **(a)** The residual between predictions and the true values,**(b)** The prediction uncertainties on the same bins of the residuals.

### 3.4 Artificial Neural Networks

Shallow ANNs ($3-5$ hidden layers) outperform larger ANNs (Fig. 9). 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, 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 (LR) and mini-batch size (MBS), 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 MBS is relatively large, while the LR could be considered very small for such a large MBS. However, since each training epoch, the data is randomly shuffled, it is possible that the training samples in each batch 'compete' with each other's gradient. For example, the training samples of $n_e^{ped} \geq 9.5 \times 10^{19}\mathrm{m}^{-3}$ pull the model weights in an opposite direction than that $n_e^{ped} \leq 9.5 \times 10^{19}\mathrm{m}^{-3}$. 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.

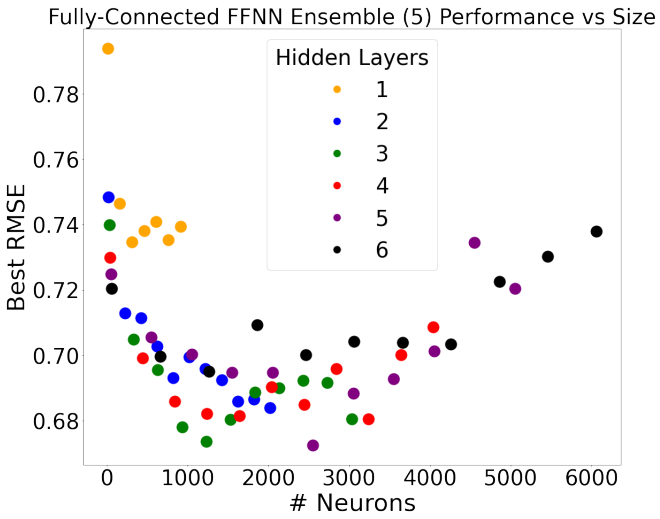



Figure 9: Comparison of the size of an ANN and its predictive capability. Each point represents an ANN which hidden layers are equal in size.

Then, via a grid search (across all available activation layers offered by PyTorch), the optimal activation function was determined to be ELU (Exponential Linear Unit), a close cousin to the well known ReLU (Rectified Linear Unit) [57]. 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.

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) (Fig. 9). This criteria was used as a space for further architecture search via random search, while using the optimal MBS, LR and activation function during the 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

The network with 4 hidden layers performed the best, with an optimal RMSE of $0.6596 \pm 0.023$. It was generally seen that shallow networks ($\leq 4$ hidden layers) with steadily decreasing layer size performed the best, whereas the larger networks performed best with this 'bell' shaped size, as seen in the 5 hidden layer network above.

In ensembles it was observed that the prediction uncertainty grows with increasing ensemble size, while the prediction quality generally did not change (Fig. 10). The ensembles were

comprised of the top performing 3 layer ANN, each network in the ensemble having different initial weights, and the prediction and uncertainty were compared across varying ensemble sizes. As the ensemble size grows, there is a slight decrease in RMSE and thus improved prediction quality (Fig. 10). 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. Similar to GPs, the prediction uncertainty of the ANN ensembles is over 300% the prediction itself. However, unlike GPs, the prediction uncertainty increases with higher pedestal densities.

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.

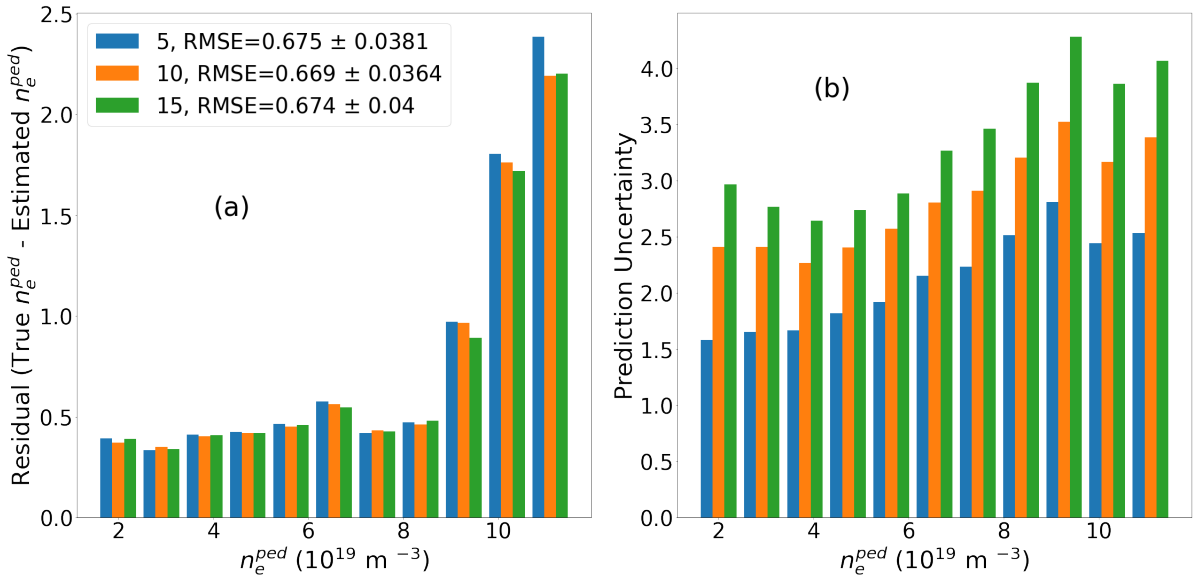



Figure 10: A three layer feed forward fully connected ANN with layer sizes 483, 415, 254 was transformed into ensembles of sizes 5 (blue), 10 (orange, 15 (green) and compared: **(a)** The residual between predictions and the true values, **(b)** The prediction uncertainties on the same bins of the residuals.

## 3.5 Under-performing ML Methods

Many different machine learning models not listed in the above analyses were initially tested, but due to their under-performance compared to RFs/GPs/ANNs, they were abandoned before delving deeper into them. For example, Nearest Neighbor methods like K-nn and R-nn [58] were tested and abandoned. R-nn 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) [29] were tested, but due to the difficulties of deriving the prediction uncertainty and general under-performance, they were dropped. Additionally, SVMs work well when there is a clear separation between regression points, but as we have seen, for $n_e^{ped} \geq 9 \times 10^{19} \mathrm{m}^{-3}$ there is no clear cut discrepancies, 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 analyze models that could provide uncertainties. Other ensemble methods like AdaBoost [59] performed 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 further in future research.

# 4 Conclusion and Outlook

It is clear that major improvements towards predicting the pedestal density height is achieved through the use of non-linear machine learning models while only using main engineering parameters.[2]

The models analyzed are ranked via their respective performance on unseen data through the use of cross validation (Table 3). Each ML tool analyzed outperformed the scaling law in predicting $n_e^{ped}$. The root mean squared error and mean absolute error scores for Gaussian Processes, Random Forests, Extreme Random Trees, and Artificial Neural Networks are quite similar, ranging between 20% of each-other. Strictly speaking, the homoscedastic Gaussian Process model with an MLP kernel achieved the lowest root mean squared and mean absolute error. All models could predict well on pedestal densities less than $9.5 \times 10^{19}\mathrm{m}^{-3}$, however they each failed to extrapolate well on densities greater than $9.5 \times 10^{19}\mathrm{m}^{-3}$, with each model performing 300% worse on high density points.

For each model, the prediction uncertainty was ascertained. The prediction uncertainty for Random Forests and 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 with high uncertainties for each prediction. We were able to utilize the measurement uncertainties of $n_e^{ped}$ in the JET pedestal database by propagating 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 be used as surrogate models in simulations.

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

The pedestal is key to high performance H-mode operation. Machine learning, in the future, could provide fast predictions for plasma scenarios. In this work, application of ML tools for $n_e^{ped}$ predictions using the JET pedestal database was done. Future work should expand these studies to other pedestal parameters, such as $T_e^{ped}$ or $p_e^{ped}$ as well as the other features of the pedestal profile (slope at pedestal top, position, width). Future work could also consider combining numerical and experimental data through Bayesian methods such as described in [60].

[2]The code developed for the experiments, figures, and scorings used in this thesis, can be found in (and reproduced via) `https://github.com/fusionby2030/bsc_thesis`

# References


[1] A. J. H. Donne. The European roadmap towards fusion electricity. *Philosophical Transactions of the Royal Society A: Mathematical, Physical, and Engineering Sciences*, 377(2141), 2019.

[2] K. Ikeda. Progress in the ITER physics basis. *Nuclear Fusion*, 47(6), 2007.

[3] H. Zohm et al. A stepladder approach to a tokamak fusion power plant. *Nuclear Fusion*, 2017.

[4] O. Meneghini et al. Self-consistent core-pedestal transport simulations with neural network accelerated models. *Nuclear Fusion*, 57(8):086034, 2017.

[5] P. Moreau et al. Development of a generic multipurpose tokamak plasma discharge flight simulator. *Fusion Engineering and Design*, 86(6):535–538, 2011. Proceedings of the 26th Symposium of Fusion Technology (SOFT-26).

[6] J. D. Lawson. Some criteria for a power producing thermonuclear reactor. *Proceedings of the Physical Society. Section B*, 70(1):6–10, 1957.

[7] S. Ishida et al. High performance experiments in JT-60U high current divertor discharges. In *Proc. 16th Inter. Conf. on Fusion Energy*, 1996.

[8] H. Zohm et al. On the Use of High Magnetic Field in Reactor Grade Tokamaks. *Journal of Fusion Energy*, 38(1):3–10, 2019.

[9] F. Troyon et al. Beta limit in tokamaks. experimental and computational status. *Plasma Physics and Controlled Fusion*, 30(11):1597–1609, 1988.

[10] R. Wenninger et al. Overview of eu demo design and r& d activities. *Fusion Engineering and Design*, 89(7):882–889, 2014. Proceedings of the 11th International Symposium on Fusion Nuclear Technology-11 (ISFNT-11) Barcelona, Spain, 15-20 September, 2013.

[11] F. Wagner et al. Regime of improved confinement and high beta in neutral-beam-heated divertor discharges of the asdex tokamak. *Phys. Rev. Lett.*, 49:1408–1412, 1982.

[12] Y. R. Martin et al. Power requirement for accessing the h-mode in ITER. *Journal of Physics: Conference Series*, 123, 2008.

[13] H. Zohm et al. Edge localized modes (ELMs). *Plasma Physics and Controlled Fusion*, 38(2), 1996.

[14] E. Viezzer et al. Ion heat transport dynamics during edge localized mode cycles at ASDEX Upgrade. *Nuclear Fusion*, 58(2):026031, 2018.

[15] V. Philipps et al. Overview of the jet iter-like wall project. *Fusion Engineering and Design*, 85(7):1581–1586, 2010. Proceedings of the Ninth International Symposium on Fusion Nuclear Technology.

[16] L. Frassinetti et al. Pedestal structure, stability and scalings in JET-ILW: the EUROfusion JET-ILW pedestal database. *Nuclear Fusion*, 61(1):016001, 2020.

[17] M. Hamed. *Electron heat transport in tokamak H-mode pedestals*. PhD thesis, Aix Marseille Universite, 2019.

[18] P. J. Catto et al. Kinetic effects on a tokamak pedestal ion flow, ion heat transport and bootstrap current. *Plasma Physics and Controlled Fusion*, 55(4):045009, 2013.

[19] M. Kotschenreuther et al. Gyrokinetic analysis and simulation of pedestals to identify the culprits for energy losses using 'fingerprints'. *Nuclear Fusion*, 59(9):096001, 2019.

[20] P. B. Snyder et al. The eped pedestal model and edge localized mode-suppressed regimes: Studies of quiescent h-mode and development of a model for edge localized mode suppression via resonant magnetic perturbations. *Physics of Plasmas*, 19(5):056115, 2012.

[21] P.B. Snyder et al. A first-principles predictive model of the pedestal height and width: development, testing and ITER optimization with the EPED model. *Nuclear Fusion*, 51(10):103016, 2011.

[22] S. Saarelma et al. Integrated modelling of h-mode pedestal and confinement in JET-ILW. *Plasma Physics and Controlled Fusion*, 60(1):014042, 2017.

[23] S. Saarelma, L. Frassinetti, et al. Self-consistent pedestal prediction for JET-ILW in preparation of the DT campaign. *Physics of Plasmas*, 26(7):072501, 2019.

[24] R. Pasqualotto et al. High resolution thomson scattering for joint european torus (jet). *Review of Scientific Instruments*, 75(10):3891–3893, 2004.

[25] V. D. Shafranov. Equilibrium of a toroidal pinch in a magnetic field. *Soviet Atomic Energy*, 13(6):1149–1158, 1963.

[26] J. P. Freidberg. *Plasma Physics and Fusion Energy*. Cambridge University Press, Cambridge, 2007.

[27] B. Mahaboob et al. A treatise on ordinary least squares estimation of parameters of linear model. *International Journal of Engineering and Technology(UAE)*, 7, 2018.

[28] D. Allen. The relationship between variable selection and data agumentation and a method for prediction. *Technometrics*, 16(1):125–127, 1974.

[29] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[30] E. Bisong. *Regularization for Linear Models*, pages 251–254. Apress, Berkeley, CA, 2019.

[31] F. Santosa et al. Linear inversion of band-limited reflection seismograms. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1307–1330, 1986.

[32] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[33] *Nature of Bayesian Inference*, chapter 1, pages 1–75. John Wiley and Sons, Ltd, 1992.

[34] F. Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[35] J. Salvatier et al. Probabilistic programming in python using PyMC3. *PeerJ Computer Science*, 2:e55, 2016.

[36] J. Görtler et al. A visual exploration of gaussian processes. *Distill*, 2019. https://distill.pub/2019/visual-exploration-gaussian-processes.

[37] C. Rasmussen. *Gaussian Processes in Machine Learning*, pages 63–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[38] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995.

[39] European Mathematical Society. Covariance Matrix. In *Encyclopedia of Mathematics*. EMS Press, 2021.

[40] T. Paananen et al. Variable selection for gaussian processes via sensitivity analysis of the posterior predictive distribution. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1743–1752. PMLR, 2019.

[41] Q. V. Le et al. Heteroscedastic gaussian process regression. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 489–496, 2005.

[42] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951.

[43] GPy. GPy: A gaussian process framework in python. `http://github.com/SheffieldML/GPy`, since 2012.

[44] W. L. Buntine. Decision tree induction systems: A bayesian analysis. *CoRR*, abs/1304.2732, 2013.

[45] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[46] T. Ho. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1, 1995.

[47] P. Geurts et al. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006.

[48] D. Silver et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[49] D. Maitra et al. Cnn based common approach to handwritten character recognition of multiple scripts. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1021–1025, 2015.

[50] L. Zongyi et al. Fourier neural operator for parametric partial differential equations. *CoRR*, abs/2010.08895, 2020.

[51] I. Goodfellow et al. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[52] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[53] A. Gillgren. ANN surrogate modeling of pedestal parameters, 2021. Manuscript in preparation.

[54] A. Paszke et al. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[55] D. Duvenaud. *Automatic Model Construction with Gaussian Processes*. PhD thesis, University of Cambridge, 2014.

[56] GPy. Gpy docs. `https://gpy.readthedocs.io/en/deploy/_modules/GPy/kern/src/mlp.html#MLP`.

[57] A. Paszke et al. Pytorch activation functions. `https://pytorch.org/docs/stable/nn.html#non-linear-activations-weighted-sum-nonlinearity`.

[58] A. Mucherino et al. *k-Nearest Neighbor Classification*, pages 83–106. Springer New York, New York, NY, 2009.

[59] R. Schapire. Explaining adaboost. In *Empirical inference*, pages 37–52. Springer, 2013.

[60] W. Xu et al. Inverse uncertainty quantification using the modular bayesian approach based on gaussian process, part 2: Application to trace. *Nuclear Engineering and Design*, 335:417–431, 2018.