Supervised learning approaches to modeling pedestal density

A Kit1,∗, A E Järvinen1,2, L Frassinetti3, S Wiesen4 and JET Contributors5

1 University of Helsinki, FI-00014 Helsinki, Finland
2 VTT Technical Research Centre of Finland, FI-02044 VTT, Finland
3 KTH Royal Institute of Technology, SE-11428 Stockholm, Sweden
4 Forschungszentrum Jülich GmbH, Institut für Energie—und Klimaforschung—Plasmaphysik, DE-52425 Jülich, Germany

E-mail: adamkit@helsinki.fi

Received 2 August 2022, revised 21 December 2022
Accepted for publication 17 January 2023
Published 22 February 2023

Abstract

Pedestals are the key to conventional high performance plasma scenarios in tokamaks. However, high fidelity simulations of pedestal plasmas are extremely challenging due to the multiple physical processes and scales that are encompassed by tokamak pedestals. The leading paradigm for predicting the pedestal top pressure is encompassed by EPED-like models. However, EPED does not predict the pedestal top density, \( n_{e,\text{ped}} \), but requires it as an input. EUROPED (Saarelma et al 2019 Phys. Plasmas 26 072501) employs simplified models, such as log-linear regression, to constrain \( n_{e,\text{ped}} \) with tokamak machine control parameters in an EPED-like model. However, these simplified models for \( n_{e,\text{ped}} \) often show disagreements with experimental observations and do not use all of the available numerical and categorical machine control information. In this work it is observed that using the same input parameters, decision tree ensembles and deep learning models improves the predictive quality of \( n_{e,\text{ped}} \) by about 23% relative to that obtained with log-linear scaling laws, measured by root mean square error. Including all of the available tokamak machine control parameters, both numerical and categorical, leads to further improvement of about 13%. Finally, predictive quality was tested when including global normalized plasma pressure and effective charge state as inputs, as these parameters are known to impact pedestals. Surprisingly, these parameters lead to only a few percent further improvement of the predictive quality. The corresponding code for this analysis can be found at github.com/fusionby2030/supervised_learning_jetpdb.

5 See the author list of ‘Overview of JET results for optimising ITER operation’ by J Mailloux et al Nucl. Fusion 62 042026.

∗ Author to whom any correspondence should be addressed.

Original Content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Supplementary material for this article is available online

Keywords: machine learning, fusion, pedestal

(Some figures may appear in colour only in the online journal)

1. Introduction

One of the primary challenges faced by future fusion reactors is to integrate high performance fusion plasmas with tolerable conditions at the reactor components [6, 7]. The conventional approach towards a high performance fusion core in tokamak plasmas is to operate the plasma in a high confinement mode (H-mode), such that a transport barrier and pedestal are formed at the edge of the plasma [8]. The tension of core-edge integration balances at the pedestal, with mutually competing requirements of high pedestal pressure for fusion performance and mitigation of plasma heat fluxes in the scrape-off layer to avoid component overheating. Therefore, predictive capability for the pedestal region of the tokamak plasma is essential for predicting performance and core-edge integration in future fusion reactors.

First principles simulations for pedestal plasmas are extremely challenging, and the ongoing effort of the scientific community has led to models such as EPED [1–3], EUROPED [4, 9], and iPED [5] that predict the pedestal height and width based on certain simplifying assumptions. The EPED-like models encompass the leading paradigm for predicting pedestal top pressures, assuming that the pedestal gradient is turbulence limited by the kinetic ballooning mode and the pedestal top pressure is magnetohydrodynamic stability limited by the peeling-ballooning mode [1–3]. However, EPED cannot be considered a fully predictive model as, in addition to machine control parameters, EPED applies information about the confined plasma state, such as total normalized plasma pressure relative to the magnetic pressure, \( \beta_N \), and pedestal top density, \( n_{e, \text{ped}} \), as inputs. EUROPED [4] can work exactly as EPED or can additionally employ simple core transport models to constrain \( \beta_N \) and models for \( n_{e, \text{ped}} \). With these additional models, EUROPED is able to predict pedestal plasmas with an EPED-like model using only engineering control parameters. However, both models for \( n_{e, \text{ped}} \) that are employed by EUROPED, (i) the log-linear regression given by Urano et al [10] based on experimental data and (ii) the neutral penetration model, show disagreements with experimental observations for the JET database, especially for \( n_{e, \text{ped}} \) exceeding \( 8 \times 10^{19} \text{ m}^{-3} \) [4].

Supervised machine learning (ML) offers potential for regression beyond the standard log-linear approach to predict \( n_{e, \text{ped}} \) used by EUROPED. In this work, various ML models are applied for predicting \( n_{e, \text{ped}} \) using the JET pedestal database described in [11], and the performance of these models is compared to the log-linear scaling laws for \( n_{e, \text{ped}} \) produced by Urano et al [10] and Frassinetti et al [11]. A related study by Gillgren et al [12] investigated the application of shallow artificial neural networks (ANNs), with promising results for pedestal electron density and temperature predictions to provide boundary conditions for core transport simulations.

With this work we aim to answer the following questions:

(a) **What is the hierarchy of input parameters in terms of providing information to improve the pedestal density prediction quality?** The current state-of-the-art log-linear scaling laws from Urano et al [10] and Frassinetti et al [11] use reduced subsets of all available machine control parameters as inputs. These subsets were chosen as the leading order input parameters using domain knowledge. Machine learning algorithms can utilize all possible input data, even if there are strong cross-correlations among the inputs. Therefore, it is important to compare the predictive performance of the ML algorithms when the input space is expanded to include 1. all the available numerical machine control parameters, 2. categorical machine control parameters, such as divertor configuration, and 3. global plasma information through dimensionless normalized plasma pressure, \( \beta_N \), and effective charge, \( Z_{\text{eff}} \), that are known to impact the performance of the pedestal.

(b) **Can advanced machine learning methods like tree-based ensembles and deep learning improve prediction quality for pedestal density predictions?** ML can encompass a wide variety of approaches, ranging from traditional linear regression to more complex deep learning networks or decision tree ensemble methods. It is not always a priori obvious which of these approaches is best suited for the given application. Additionally, with increased complexity, the time for training and application (inference) increases. Even though it would intuitively seem that more complex models would increase the risk of overfitting by having the capacity to memorize samples, over-parameterized deep learning networks tend to still maintain generalization capabilities, if sufficient training data is available [13]. This work aims to document the advantages and disadvantages of these approaches when developing a regression-based predictor for \( n_{e, \text{ped}} \).

2. JET pedestal database

The data used in this work is obtained from the EUROfusion JET pedestal database [11]. A full analysis of the database can be found in [11]. The data is in tabular form with rows (entries) of shots and time windows that have columns (features) corresponding to time averaged machine control parameters and measurements of various plasma parameters. From 2625 unique JET shots, the dataset consists of 3557 entries.

The subset of the JET pedestal database used in this analysis includes techniques to control edge localized modes, such as
kicks [14] or resonant magnetic perturbations (RMPs) [15], as well as plasmas with impurity seeding and pellets [16, 17]. These are included in the database as binary inputs on whether the scheme is applied or not. In addition, for impurity seeding the type of impurity is given as a categorical input. We can feed this information to machine learning models in a numerically encoded format, such that shots with kicks are given a column value 1 and shots without a column value 0. This approach extends to an arbitrary number of categories as the list can be extended to as many numbers as are needed to describe the category. While the included dataset contains over 1500 shots with the JET ITER-like tungsten beryllium wall (JET-ILW) [18], there are also 422 shots with the previous carbon wall (JET-C). While a dedicated wall material flag is not given to the algorithms, the information about JET-C is mostly coded in the categorical variable containing the charge number of the impurity species in the plasma, $Z_{\text{imp}}^{\text{atomic}}$. However, there are 21 shots with JET-ILW and $Z_{\text{imp}}^{\text{atomic}} = 6$, due to seeding of methane. Future studies will investigate the impact of including a dedicated wall material flag. Furthermore, a dedicated impurity injection rate value is not given in the input set, which is expected to reduce predictive performance as the database contains plasmas at varying impurity injection rates. The role of the impurity injection rate will be addressed in future studies. Even though impurity injection rate is not included explicitly, this information is partially coded into $Z_{\text{eff}}$ and, therefore, accessible to model experiments containing $Z_{\text{eff}}$ information as input. The major radius, $R$, is notably missing from the list of main engineering parameters found in table 1. For single machine analysis, $R$ does not vary in a meaningful way, and through the Shafranov shift of the magnetic configuration [19], encodes information about the pedestal pressure.

3. Machine learning for tabular datasets

Decision tree ensembles and deep learning methods can capture many nuances of tabular datasets. A decision tree ensemble can exploit multiple decision trees fitted on unique subsets of the datasets, while deep learning models make use of specialized layers designed for tabular data. A brief description of each model used and why they are chosen in this analysis is given in section 4.1. In contrast to decision tree ensembles and deep learning methods, log-linear regression seeks to find a single equation to parameterize the entirety of a dataset. Deep learning has had particular success with structured data in tabular form, where decision trees still dominate [23].

However, tree-based learning approaches do not make use of differentiable gradients during their construction, meaning they cannot be used in component pipelines combining different models and individual modules. Here, deep learning has the potential to create end-to-end pipelines for problems, where some of the inputs could come from tabular data and others from raw diagnostic measurements or images, and a full deep learning model could be trained in one computational graph. This, as well as the potential for improved performance, has led to many proposed deep learning solutions specific for tabular data. In this work, we compare the state-of-the-art deep learning architectures designed for tabular data with popular decision tree ensemble techniques.

Unlike log-linear regression, other ML models can efficiently utilize categorical variables. In the case of pedestal databases, these variables can be specific experimental setups of a shot, e.g. the divertor configuration, which can take on one of 6 values written in table 1. The deep networks analyzed in this paper make use of varying approaches to extract meaningful relationships within the categorical variables. These approaches include embedding or attention layers [24]. Some decision tree based methods, such as CATBoost [25], are designed specifically to efficiently process categorical inputs.

4. Machine learning experiments

4.1. Comparing models

We investigate which tree-based ensemble and deep learning methods have advantages over the log-linear regression used for the problem of regression of $n_{\text{eff,ped}}$ using the JET pedestal database. We focus on the relative performance of different models. Model-agnostic deep learning methodologies such as data augmentation, learning rate warmup or decay [26], or pretraining [27], are not used, except in TabNet [28], which employs these in the recommended settings. While these practices have the potential to improve performance, our goal is to evaluate the intrinsic model.

The Urano and Frassinetti log-linear regression models serve as a performance baseline. We then compare that performance with the following decision tree models, which have been chosen for their generally good performance on a wide variety of datasets as seen in [23, 29, 30]:

(a) Random forests (RF) [31]. An ensemble of decision trees, where each tree is fitted in parallel using a unique subset of the dataset (a process known as bagging) and the total forest prediction is the averaged prediction between all decision trees in the ensemble. As opposed to a log-linear regression tool trying to fit all points in the dataset at once, the RF ensemble leverages the fact that each tree fits a small subset of the total dataset, which could lead to a better generalization of the whole dataset.

(b) Extreme randomized trees (ERT) [32]. An implementation of RF that leads to more diversified trees. This is achieved by changing the algorithm of how a node is split in a decision tree. In the RF a split is made based on an information criterion, and in ERT the split is made randomly.

(c) XGBoost [33]. A decision tree ensemble that can build problem specific trees to aid in the ensemble prediction. Instead of fitting decision trees in parallel like RF and ERT, gradient boosted trees (GBT) conducts tree fitting sequentially: each new decision tree is fitted by applying
Table 1. Relevant parameter domains and univariate statistics of JET pedestal database used in this analysis. Individual parameters are detailed in [11]. The F-statistic (F-score), mutual information (MI) [20, 21], and linear correlation with respect to $n_{e,ped}$ are given. Larger F- and MI-scores imply larger statistical dependence of $n_{e,ped}$ on the feature. Features with †, ‡ correspond to those used in the Urano and Frassinetti subsets of input space, respectively. The triangularity, $\delta$, is defined as the average between upper and lower triangularities, i.e. $\delta = \frac{\delta_u + \delta_l}{2}$. The divertor configuration categorical variable represents the specific divertor plate of the strike points. A plasma with a ‘V/H’ divertor configuration has an inner strike point on the vertical target and an outer strike point on the horizontal target, and is detailed in figure 5 of [11]. $P_{NBI}$ refers to the applied heating power from neutral beam injection. The linear correlation matrix for each variable is visualized in figure 1.

| Engineering | Corr. w/ $n_{e,ped}$ | F-score | MI  | Min | Max  |
|-------------|----------------------|---------|-----|-----|------|
| $l_F$ (MA)  | 0.610                | 1945    | 0.60| 0.813, | 4.481 |
| $B_T$ (T)   | 0.381                | 557     | 0.53| 0.962, | 3.856 |
| a (m)       | -0.218               | 163     | 0.21| 0.828, | 0.979 |
| $q_{95}$    | -0.484               | 1004    | 0.41| 2.267, | 6.095 |
| $V_F$ (m$^3$)| -0.022               | 1.6     | 0.25| 58.30, | 82.500 |
| $\delta$ (→)| 0.371                | 523     | 0.31| 0.130, | 0.482 |
| $\kappa$ (→)| 0.470                | 931     | 0.36| 1.470, | 1.819 |
| $P_{NBI}$ (MW)| 0.157               | 83      | 0.15| 0.000, | 32.267 |
| $P_{CRH}$ (MW)| 0.223              | 171     | 0.09| 0.000, | 7.963 |
| $P_{TOT}$ (MW)| 0.238               | 198     | 0.15| 3.402, | 38.220 |
| $\Gamma$ ($10^2$ e/s) | 0.445 | 810 | 0.31 | 0.000, | 22.307 |

| Global | Corr. w/ $n_{e,ped}$ | F-score | MI  | Min | Max  |
|--------|----------------------|---------|-----|-----|------|
| $\beta_{\text{MHD}}$ (→)| -0.397 | 614 | 0.14| 0.567, | 3.625 |
| $Z_{\text{eff}}$ (→)| -0.212 | 154 | 0.08| 1.001, | 12.632 |

| Categorical | Corr. w/ $n_{e,ped}$ | F-score | MI  | Domain |
|-------------|----------------------|---------|-----|--------|
| Kicks       | -0.071               | 16.5    | 0.02| [0, 1] |
| RMP         | -0.112               | 41      | 0.01| [0, 1] |
| Pellets     | 0.009                | 0.28    | 0.01| [0, 1] |
| Divertor    | 0.413                | 674     | 0.17| [C/C, V/H, V/C, V/N, C/V, C/H] |
| $Z_{\text{atomic}}$| 0.084 | 23 | 0.06| [0, 2, 6, 7, 8, 10, 18, 36] |

Figure 1. Correlation matrix between control, global, and categorical parameters.

the gradient of the error in the prediction. In XGBoost, the algorithm follows the Newton–Raphson method in function space, while a generic GBDT uses gradient descent. This could enable the model to build problem specific trees, e.g. trees dedicated to predicting large values of $n_{e,ped}$. |
Table 2. A list of parameters included in each input parameter space from the categories in table 1. The parameter sets are named by which variables they include: E: main engineering, C: categorical, B: βN, Zeff. U and F refer to the Urano and Frassinetti subspaces respectively.

| Parameter sets | U | F | E | EC | ECB | ECBZ | EB | EBZ |
|----------------|---|---|---|----|-----|------|----|-----|
| Main engineering | † | ‡ | * | * | * | * | * | * |
| Categorical | * | * | * | * | * | * | * | * |
| βN | * | * | * | * | * | * | * | * |
| Zeff | * | * | * | * | * | * | * | * |

(d) CATBoost [25]. A GBDT implementation that leads to more balanced trees than XGBoost. In each level of the decision tree, the node split combination that leads to the best prediction is selected and used for all the nodes on that level. This could enable the model to reduce overfitting in comparison to XGBoost, while still utilizing the benefits of gradient boosting.

To compare against the tree-based ensembles, we investigate the following deep learning architectures designed specifically for tabular data:

(a) Multi-layer perceptron/feed-forward neural network (MLP/FFNN). The simplest ANN to establish the basic $n_{\text{ped}}$ regression capabilities of deep learning.

(b) Feed-forward nets with category embedding (FFCAT). A simple feed forward network with additional layer types that are specifically designed for categorical input parameters. With this model we can leverage the MLP capabilities while adding functionality to the model in regard to using categorical variables.

(c) TabNet [28]. Once fit, this model selects which features to be used in prediction on a per-instance basis (attention layers). This could be useful when predicting $n_{\text{ped}}$, as the scaling seems to be different for low and high $n_{\text{ped}}$ ranges, as observed in the struggle of the log-linear regression in fitting both.

(d) AutoInt [34]. This model can learn correlations between parameters by mapping both numerical and categorical inputs into a low-dimensional space. It also utilizes attention layers like TabNet.

(e) NODE [35]. An ensemble of decision trees that are fitted instead using the gradient descent methods of ANNs. Like CATBoost, this model uses symmetric splitting of each tree, but once again in a differential form. This model leverages the benefits of decision tree ensemble methods, but has the ability to be paired with other ANN models in the case of a multi-modal pipeline.

4.2. Input space

We investigate whether the inclusion of both categorical and more numerical input parameters leads to an increase in the performance of ML models in comparison to the inputs defined in traditional log-linear scalings. Additionally, we investigate whether the inclusion of global parameters, βN, Zeff, also aid in prediction. A fully predictive model can function without global parameters, but it is important to determine to what extent they can improve regression performance. All possible input parameters and their corresponding sub-domains are listed in table 1.

The set of control and global parameters were chosen as they coincide with the standard set of input variables for EPED/EUROPED-like models. For completeness, the $F$-statistic and MI scores for each variable with respect to $n_{\text{ped}}$ are also given in table 1. As expected, the scores reflect that of previous physics basis development by Frassinetti et al [11]. For example, the plasma current, $I_P$, is positively correlated and has high association strength with pedestal density.

In the context of feature selection for machine learning problems, one might consider removing weakly correlated variables from the feature space. In this case, the power variables, $P_{\text{NBI}}$, $P_{\text{ICRH}}$, $P_{\text{TOT}}$, are a possible example of such, as they show low correlation and information scores. However, as discussed in [11], the dependence of $n_{\text{ped}}$ on power is likely highly non-linear, and, when related additionally with other plasma parameters, such as the upstream separatrix density, show positive with $n_{\text{ped}}$. The $F$-statistic and MI scores are certainly useful when little previous domain knowledge is known, and future work would include statistical approaches to removing ‘unnecessary’ features from the feature space.

Using these input parameters, we create six unique sets of possible inputs to a model, which are listed in table 2. For input spaces U and F, we use the input parameters from the established scaling laws from Urano and Frassinetti, respectively.

4.3. Experiment details

For each input space, the testing of each model is done using the following systematic approach.

4.3.1. Preprocessing. The main engineering parameters and target variable, $n_{\text{ped}}$, are standard scaled. This transforms each feature to a domain with a mean of zero and standard deviation of one, which reduces the impact of the relative magnitude of each feature. The categorical features are transformed into integer representations, e.g. binary features like kicks are transformed into a one or zero for a shot with or without kicks, respectively.

The dataset is split into training and test subsets, so that all model types are trained and tested on the same set of data. Since the target variable, $n_{\text{ped}}$, is continuous and not uniformly distributed within the JET PDB, a pseudo stratified binning procedure is implemented such that the training and test splits are evenly in their representation of $n_{\text{ped}}$ (figure 2). For exact details of this procedure see section 6.1 of the supplementary material.
4.3.2. Hyperparameter tuning. For each model, we use the Hyperopt library [36] to find optimal hyperparameters from a model specific search space (see section 6.2 of the supplementary material). The full training set is split into a reduced training and validation set, and the optimal hyperparameters are chosen to be those that perform best on the validation split.

4.3.3. Evaluation and metrics. We run 15 numerical experiments with different random seeds. The training set was further split into reduced training and validation sets, and the mean and standard deviation of the root-mean-squared-error (RMSE) on the unseen test set was reported across the 15 numerical experiments.

5. Results

5.1. Model comparison

5.1.1. Model performance. If sufficient experimental data is available to build an interpolating regression model for \( n_{e,\text{ped}} \), as is the case for many present-day tokamaks, the results indicate that using GBDT-type regression approaches is likely to outperform the conventional log-linear regression approach (figure 3). Decision tree ensembles and deep learning based regression models outperform the log-linear scaling laws by about 27% and 20%, respectively, when fitted using the Urano input space (figure 3 and table 3). This observation clearly shows the utility of these regression tools in building experimental models for physical observables that encompass highly non-linear dependencies in the plasmas. Decision tree based models perform systematically better than deep learning models, with XGBoost performing the best. Of the deep learning architectures, NODE achieves the lowest RMSE and has very similar performance as the decision tree techniques when input spaces do not contain categorical features. This comes as no surprise, as NODE is essentially a neural network wrapper for GBDTs.

5.2. Input spaces

Increasing input parameters to all available engineering control parameters, the prediction quality is further improved by about 13% for both decision tree ensembles and deep learning models with respect to models fitted using the Urano input space (figure 3). This indicates that, in addition to the better generalization capability, the ML models can also benefit from the capability to extract further information from the parameters that are not considered as leading terms in predicting \( n_{e,\text{ped}} \), such as residual shaping information contained in \( a, \kappa, q_{95}, \) and \( V_P \).

When categorical variables are introduced, the change in prediction quality is negligible for decision tree ensembles and NODE, but leads to an average degradation by 3% for other deep learning models. The lack of improvement suggests that the additional categorical variables have less impact on \( n_{e,\text{ped}} \) than the numerical variables, which is surprising as diverter configuration and impurity seeding, for example, should impact \( n_{e,\text{ped}} \) prediction. The decrease in quality of predictions from deep learning models may come from the categorical variables competing for relevance in the attention layers.
of the model, which are normally utilized with more leading order numerical parameters.

Including global plasma quantities $\beta_N$ and $Z_{\text{eff}}$ leads to a further 5% increase in prediction quality for both decision tree ensembles and deep learning models. An improvement in prediction quality is expected, since $\beta_N$ plays a role in pedestal performance and encodes density information via the global plasma pressure. However, the small improvement relative to only using machine parameters as inputs suggests that most of the relevant information about $n_{e,\text{ped}}$ is contained in the machine parameters, and would be a small sacrifice in order to maintain a fully predictive model.

5.3. Model limits and outlier detection

The mapping between the machine control configuration and $n_{e,\text{ped}}$ is not bijective. For a given set of machine control parameters, $n_{e,\text{ped}}$ can obtain a range of values, depending on the preceding time-evolution of the plasma, and also different machine control configurations can lead to the same $n_{e,\text{ped}}$ due to the same reason. In this section, three examples of non-bijective entries of the dataset are investigated: (a) multiple flat-top entries from the same shot (figure 5), (b) entries with similar machine control parameter sequences and values that lead to different pedestals (figure 6), and (c) entries that land outside the flat-top (figure 4). Since the database is supposed to contain only flat-top entries, the few entries outside flat-top are actually caused by human error in preparing the database. In this work, these entries show systematically a large prediction error in the ML models, highlighting the utility of the approach to pinpoint outliers that are not representative of the statistical distribution of the database. The other examples show that without additional information about the plasma state or time-evolution history, the model is unlikely to be able to resolve the multiple possible $n_{e,\text{ped}}$ solutions. Since these observations are inherent to the data, they would equally apply to any data-driven models, including log-linear regression, DT and DL approaches.

5.3.1. Multiple flat-top entries from the same shot. When two entries both fall within the flat-top window, all models tend to predict $n_{e,\text{ped}}$ near the training set value (figure 5). Fluctuations in the $n_{e,\text{ped}}$ value within a flat-top can be caused by transients due to, for example, MHD-events or impurity accumulation. While these effects can encode information in the global plasma parameters, such as $\beta_N$, without sufficient statistics of such events in the training dataset, the algorithm is not able to appropriately predict the observed $n_{e,\text{ped}}$ value with such events. In the example in figure 5, a factor-of-two increase in core radiated power is observed between the two time slices, indicating impurity accumulation in the plasma. Providing information about the radiated power or radiated power fraction (which is included in the database but is not considered a machine parameter in this analysis) for the regression algorithm would probably improve the prediction quality for these types of plasmas.

5.3.2. Entries from different shots and pedestals but with similar static control parameters. When two entries have similar machine parameters but come from different shots, the models tend to predict near the entry in the training set (figure 6). Two plasmas with similar machine control parameters do not necessarily exhibit exactly the same $n_{e,\text{ped}}$, since the time-evolution of the plasmas might differ from each other and the wall conditions might not be exactly the same, for example, due to impurity seeding in the previous plasma. These may result in a different trajectory of the pedestal conditions within the time-evolution of the plasma, which is difficult for the data-driven model to learn, unless sufficient additional information from the plasma state or history is provided for the algorithm.
5.3.3. Outside of flat-top entries. The JET pedestal database is intended to contain only H-mode flat-top entries. In this work, the data-driven models showed systematically a large prediction discrepancy for certain entries in the database. Further investigation revealed that some of these entries were actually outside flat-top time windows (figure 4). This is caused by, practically unavoidable, human error when manually creating large databases with thousands of entries. This indicates the utility of the data-driven models in identifying outliers that do not follow the statistical distribution of the investigated database. As expected, the model predicts an \( n_{e,\text{ped}} \) value representative of equivalent flat-top conditions with the given machine control parameters, and this value differs from the transient value observed outside flat-top windows. In future versions of the database, these entries outside the flat-top will be removed.

Future studies will investigate methods to include time-dependency of the pedestal density in the learning process. These studies will then also look into methods to extend the predictions beyond flat-top windows (e.g. during ramp-down), and shots with similar control sequences.

5.3.4. Outlier detection. Additionally, outliers that likely come from manual fitting procedures of the HRTS data in creating the JET PDB dataset were detected by both the ANN and decision-tree ensemble models (figure 7). These outliers were detected by investigating the test set entries that all models systematically misevaluated and did not fall into the previous three categories. The outliers identified were subsequently removed from the dataset and all models were retrained, and although the performance metrics of each model showed a 1%–2% improvement, the overall rankings from table 3 did not
change, therefore the subsequent performance is not doubly reported.

5.4. Computational efficiency

We observed that the decision tree based techniques required 99% less time to train and optimize than the more complex deep learning methods (table 4). Additionally, it was seen that hyperparameter tuning played a minor role in improving the performance metrics of each model, which is likely due to the relatively small size of the dataset. All models are capable of producing more than 50 predictions per second and thus could be components of a fast surrogate model pipeline.
Figure 7. An example of an outlier that is likely not a misprediction by the models (red and orange lines). The JET PDB estimation of high \( n_e \) (black dashed line) likely comes from the lone ‘ear’-like profile which does not appear to be a generalization of the rest of the profiles within 75%–99% of the ELM percentage found in the window (grey circles).

Table 4. For selected models, the following statistics are collected: the training time (Training) and the amount of predictions per second when performing inference for each model (Throughput) and the wall time taken to find optimal hyperparameters (Optimization). These experiments use input space EC from table 2, which has 17 total features. Notation: ↓ lower is better, ↑ higher is better. Models with GPU compatibility are trained and tuned on one or more Nvidia V100s and others used 40 Intel Xeon processors. Inference timings were done using the Intel processors to eliminate differences in final performance. The optimization time is the wall time needed to find the optimal set of hyperparameters for the given model. *The optimization search for NODE consisted of a grid search with 24 total iterations.

| Model     | Random forest | XGBoost | MLP   | TabNet | NODE |
|-----------|---------------|---------|-------|--------|------|
| Training (s) ↓ | <1           | 5       | 10    | 465    | 792  |
| Throughput (pred/s) ↑ | 50          | 850    | 7400  | 230    | 150  |
| Optimization (s) ↓ | 105         | 732    | 4038  | 25 176 | 148 826* |

6. Conclusion

In this work, decision tree ensembles and deep learning algorithms were explored for building regression models for \( n_e \) at JET. Based on the model experiments conducted on the EUROfusion JET pedestal database [11], decision tree ensembles and deep learning methods are viable alternatives for interpolative regression for \( n_e \) using the JET pedestal database. Both decision tree ensembles and deep learning architectures outperform log-linear regression by 23% when applying the same set of input parameters as used in log-linear regressions by Frassinetti et al [11] and Urano et al [10]. Extending the input space to more available machine control parameters, some of which are strongly cross-correlated, we achieve an additional 13% improvement. Including global parameters, such as \( \beta_N \) and \( Z_{\text{eff}} \), to the input space, improves prediction quality further by 5% but also renders the regression model not fully predictive. Unexpectedly, the categorical parameters, such as divertor configuration or atomic number of seeded impurity, did not seem to impact the prediction quality significantly, even though divertor configuration is known to impact pedestal density in tokamaks [17]. Detailed investigations of the reasons for this observation are a subject for future studies. This work can serve as a basis for further development of decision tree based or deep learning models as alternatives for \( n_e \) regression in pedestal models, such as EUROPED.

Data availability statement

The authors do not have permission to share the data analysed during the current study. Data access can be obtained by establishing a user account for the JET computer system.

Acknowledgment

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No. 101052200—EUROfusion). Views and opinions
expressed are however those of the author(s) only and do
not necessarily reflect those of the European Union or
the European Commission. Neither the European Union nor
the European Commission can be held responsible for them.

The authors wish to acknowledge CSC—IT Center for Sci-
ence, Finland, for computational resources.

The foundations of this paper were done as part of the BSc
thesis of Adam Kit, which was supervised by Prof. Mathias
Groth of Aalto University and Prof. Frank Cichos of Leipzig
Universität, and therefore the authors extend their thanks to
these two for their help in that process.

The authors wish to thank Yoeri Poels and Andreas Gillgren
for the useful conversations and contributions to this project.

ORCID iDs

A Kit https://orcid.org/0000-0003-2621-6291
A E Järvinen https://orcid.org/0000-0002-4525-8158
L Frassinetti https://orcid.org/0000-0002-9546-4494
S Wiesen https://orcid.org/0000-0002-3696-5475

References

[1] Snyder P B, Groebner R J, Leonard A W, Osborne T H and
Wilson H R 2009 Phys. Plasmas 16 056118
[2] Snyder P B, Groebner R J, Hughes J W, Osborne T H,
Beurskens M, Leonard A W, Wilson H R and Xu X Q 2011
Nucl. Fusion 51 103016
[3] Snyder P B, Osborne T H, Burrell K H, Groebner R J,
Leonard A W, Nazikian R, Orlov D M, Schmitz O,
Wade M R and Wilson H R 2012 Phys. Plasmas 19 056115
[4] Saarelma S, Frassinetti L, Bilkova P, Challis C D, Chankin A,
Fridström R, Garzotti L, Horvath L and Maggi C F 2019
Phys. Plasmas 26 072501
[5] Dunne M G et al 2016 Plasma Phys. Control. Fusion
59 025010
[6] Ikeda K 2007 Nucl. Fusion 47 E01
[7] Zohm H, Trauble F, Biel W, Fable E, Kemp R, Lux H,
Siccinio M and Wenninger R 2017 Nucl. Fusion 57 086002
[8] Zohm H 1996 Plasma Phys. Control. Fusion 38 105–28
[9] Saarelma S, Challis C D, Garzotti L, Frassinetti L, Maggi C F,
Romanelli M and Stokes C 2017 Plasma Phys. Control.
Fusion 60 014042
[10] Urano J H, Maggi C F and Joffrin E 2016 Characterization of
electron density based on operational parameters in JET
H-mode plasmas with C and ILW Proc. 43rd EPS Conf. on
Plasma Physics (Leuven, Belgium) p O4.121
[11] Frassinetti L et al 2020 Nucl. Fusion 61 016001
[12] Gillgren A, Fransson E, Yadykin D, Frassinetti L,
Strand P and Contributors J 2022 Nucl. Fusion
62 096006
[13] Zhang C, Bengio S, Hardt M, Recht B and Vinyals O 2021
Commun. ACM 64 107–15
[14] de la Luna E et al 2015 Nucl. Fusion 56 026001
[15] Evans T et al 2006 Nat. Phys. 2 419–23
[16] Lang P T, Büchler K, Kaufmann M, Lang R S, Mertens V,
Müller H W and Neuhäuser J (ASDEX Upgrade and NI
Teams) 1997 Phys. Rev. Lett. 79 1487–90
[17] Giroud C et al 2015 Plasma Phys. Control. Fusion
57 035004
[18] Matthews G F et al 2011 Physica Scripta T 145 014001
[19] Gutkin T and Lozovsky S 1974 Phys. Rev. Lett. 31 923–5
[20] Kraskov A, Stögbauer H and Grassberger P 2004 Phys. Rev. E
69 066138
[21] Ross B C 2014 PLoS One 9 1–5
[22] Goodfellow I J, Bengio Y and Courville A 2016 Deep
Learning (Cambridge, MA: MIT Press)
[23] Shwartz-Ziv R and Armon A 2021 CoRR (arXiv:2106.03253)
[24] Mikolov T, Sutskever I, Chen K, Corrado G and Dean J 2013
Distributed representations of words and phrases and their
compositionality Proc. 26th Int. Conf. on Neural
Information Processing Systems—Volume 2 (NIPS’13) (Red
Hook, NY: Curran Associates Inc.) pp 3111–9
[25] Prokhorenkova L, Gusev G, Vorobev A, Dorogush A V and
Gulin A 2019 Catboost: unbiased boosting with categorical
features (arXiv:1706.09516)
[26] Goyal P, Dollár P, Girshick R B, Noordhuis P, Wesolowski L,
Kyrola A, Tulloch A, Jia Y and He K 2017 CoRR
(arXiv:1706.02677)
[27] Erhan D, Bengio Y, Courville A, Manzagol P A, Vincent P and
Bengio S 2010 J. Mach. Learn. Res. 11 625–60
[28] Arik S O and Pfister T 2019 CoRR (arXiv:1908.07442)
[29] Katzir L, Elidan G and El-Yaniv R 2021 Net-dnf: effective
deep modeling of tabular data ICLR
[30] Gorishny Y V, Rubachev I, Khruilkov V and Babenko A 2021
arXiv:2106.11959
[31] Breiman L 2001 Random forests Mach. Learn.: Sci. Technol.
45 5–32
[32] Geurts P, Ernst D and Wehenkel L 2006 Extremely
randomized trees Mach. Learn.: Sci. Technol.
63 3–42
[33] Chen T and Guestrin C 2016 XGBoost: a scalable tree
boosting system Proc. 22nd ACM SIGKDD Int. Conf. on
Knowledge Discovery and Data Mining (New York: ACM)
pp 785–94
[34] Song W, Shi C, Xiao Z, Duan Z, Xu Y, Zhang M and Tang J
2018 CoRR (arXiv:1810.11921)
[35] Popov S, Morozov S and Babenko A 2019 CoRR
(arXiv:1909.06312)
[36] Bergstra J and Cox D 2013 Making a science of model search:
hyperparameter optimization in hundreds of dimensions for
vision architectures ICML