Developing Deep Learning Algorithms for Inferring Upstream Separatrix Density at JET

A. Kit[a,*], A.E. Järvinen[a,1], S. Wiesen[c], Y. Poels[d], L. Frassinetti[e], JET Contributors[f]

[a] *University of Helsinki, Helsinki, Finland*
[b] *VTT Technical Research Centre of Finland, FI-02044 VTT, Finland*
[c] *Forschungszentrum Jülich GmbH, Institut für Energie- und Klimaforschung– Plasmaphysik, 52425 Jülich, Germany*
[d] *Eindhoven University of Technology, Eindhoven, Netherlands*
[e] *KTH Royal Institute of Technology, Stockholm, Sweden*
[f] *See the author list of 'Overview of JET results for optimising ITER operation' by J. Mailloux et al. to be published in Nuclear Fusion Special issue: Overview and Summary Papers from the 28th Fusion Energy Conference (Nice, France, 10-15 May 2021)*

Abstract

Some nice abstract

Keywords: Separatrix, machine learning, JET, generative modeling

1. Introduction

The electron density at the upstream separatrix is a key parameter for core-edge integration in tokamaks. The power exhaust properties and scrape-off layer (SOL) are strongly regulated by the upstream SOL density and collisionality [**?**]. High confinement mode (H-mode) pedestal performance on the other hand depends strongly on the $n_{sep}/n_{e,ped}$ ratio. Need accurate scaling or modeling to determine power exhaust parameters in SOL as well as general pedestal performance.

1. Why is $n_{e,sep}$ important and why is there a need for predictive modeling of $n_{e,sep}$?
2. How is $n_{e,sep}$ determined experimentally vs modeling (using HRTS vs using some simluation)
3. How is $n_{e,sep}$ currently calculated using HRTS diagnostic? (MTANH fit)
4. How do we go about calculating it differently?

Determining $n_{e,sep}$ is difficult in existing devices, as length scales of temperature and density gradients in H-mode pedestals are typically less than the uncertainty given by equilibrium reconstruction for the location of such gradients, which has led to the development of analysis using HRTS diagnostics to better determine $n_{e,sep}$ [SOURCES]. Traditionally, a modified hyperbolic tangent function is fit to time-/elm- averaged density and temperature profile(s) and $n_{e,sep}$ is considered the point on the fit where $T_e = 100$ eV. The choice of the mtanh curve is purely based on the idea that the profile looks generally like a MTANH. [SOURCE from 15 years ago]. The choice of $T_e = 100$eV comes from applying JET parameters to the power-balance two-point model as proposed by Stangeby [SOURCE].

*Corresponding author: adam.kit@helsinki.fi

2. Machine Learning for Predictive Modeling

1. Overview of Supervised vs Weakly-Supervised machine learning
2. Direct mapping (XGBoost)
3. Representation learning (VAE/DIVA)
4. DIVA context and the changes we make to our (conditional prior for fully-predictiveness and aux regressor, featurewise-extraction for elm perc, physics losses)

Machine learning offers the potential to move beyond log-linear 'scaling law' regression analysis. To get more accurate predictions of nesep we can leveraging a higher quantity of data, than those used in traditional scaling law methodologies. To create scaling laws, a quantity of anywhere between 1000-3000 time-averaged slices of selected relevant pulses are used, often or always ignoring ELM characteristics. The choice of log-linear regression helps to understand what physical mechanisms are driving the predictions as quantified by the regression coefficients, but often do not hold for fractions of the data, and often do not hold at all when taken outside the context of the subset of data used in the analysis, e.g., comparing nesep scaling law with inter-ELM nesep. Advanced techniques in supervised and unsupervised learning can help overcome these issues.

Supervised learning depends on a 'labeled' dataset, which is used to find a function that best maps a set of input parameters to the 'labeled' target variable. In this analysis, the supervised learning dataset has time-slice entries with machine control parameters and ELM percentage as inputs and $n_{e,sep}$ as the target variable is. In order to find the mapping function a model is used, which in this analysis the is XGBoost, a decision tree ensemble that makes use of differentiable trees.

Weakly supervised learning behaves differently, as the function is instead geared to mapping a representation of the input. In the unsupervised learning setting, the dataset has pedestal profiles (psi -¿ 0.85-1.1), from which a model tries to find the function that best reproduces the profiles. The model in this analysis is a varational autoencoder.

A variational autoencoder (VAE) was trained on HRTS density and temperature profiles from the time windows found within the JET pedestal database. A VAE is a probabilistic generative model in the sense that it attempts to model a dataset (with samples $x \in X$) as a conditional distribution given some latent variable, $z \in Z$: $p(x, z) = p(x|z)p(z)$. This idea is implemented using an encoder $q(z|x)$, and decoder $p(x|z)$ distribution, parameterized by artificial neural networks. A VAE with this parameterization would take profiles as inputs, x, encode the information to lower dimensional latent variables, z, then decode from the latent variables to profiles again, x.

We add additional features to the VAE in order to incorporate machine parameters, ELMs, and additional physics constraints.

We include a conditional prior $q(z|y)$, parameterized by neural networks that takes machine control parameters y, and provides the priors for the conditional distributions for the latent variables z, giving space for the model to be fully predictive.

The model is further enforced to predict machine parameters from the latent space via an auxiliary regressor $q(y|z)$. PHYSICS LOSSES BLAH BLAH

3. Results

1. XGBoost on JET Pedestal database
2. XGBoost on created dataset
3. DIVA on created dataset

4. Discussion/Conclusion

1. Benefits of VAE: i) get the whole profile ii) probabilstic, and get a sense of how it produces its profile representation
2. Benefits of XGBoost: i) get a direct answer for $n_{e,sep}$, ii) Faster to train/optimize
3. Want to include radial decay length from HRTS signal to get self-consistant nesep prediction.