DEVELOPMENT OF SURROGATE MODELS FOR DIVERTOR POWER LOAD PREDICTION BASED ON MACHINE LEARNING TECHNIQUES

Inaugural-Dissertation

zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

MARTIN BRENZKE

aus Tönisvorst

Düsseldorf, Juli 2021

aus dem Institut für Theoretische Physik I
der Heinrich-Heine-Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Berichterstatter:

1. Prof. Dr. Alexander Pukhov

2. Prof. Dr. Yunfeng Liang

Tag der mündlichen Prüfung: 16.03.2022

ABSTRACT

The power loads at the divertor targets are one of the main challenges in the sustained operation of tokamak fusion reactors. These power loads define critical operational limits for current and future fusion devices, such as ITER. In order to account for these limits in the planning of fusion reactors and to mitigate critical divertor power loads during the operation, reliable predictions for the divertor power loads are needed. However, modeling divertor power loads prior to performing experiments is an involved task. Simplified analytical models do not necessarily take into account the complex atomic and molecular physics close to the divertor targets. On the other hand, fully fledged simulation codes, taking into account the atomic and molecular processes, are computationally very expensive and can suffer from numerical instability.
To alleviate these challenges machine learning based models for divertor power load predictions were tested in this analysis. Since the models can describe highly non-linear functions, the learned dependencies might describe the problem at hand more closely than simplified models based on first principles. On top of that, the machine learning models, once set up, are computationally less expensive than established simulation codes.
To set up and evaluate the models a data base utilizing almost 6 years of data from the *ASDEX Upgrade* (AUG) experiment was established from the experimental data. Based on this data, several machine learning approaches were investigated with regard to their ability to infer divertor power loads from generally accessible plasma parameters. All of the tested models show that it is possible to obtain predictions of the divertor power loads as a function of the selected parameters.
Furthermore, the learned dependencies of some of the models were analyzed. While there are some effects of spurious correlations that can be observed, some main dependencies of the models match the expectations from other physics analyses.
Moreover, approaches to obtaining adequate predictions of model uncertainties were examined. For this, *Mixture Density Network*s (MDNs) and *Gaussian Process Regression* (GPR) were utilized. However, the analysis shows that further work with regard to the uncertainty estimation is needed.

ZUSAMMENFASSUNG

Der Transport von Wärme und Teilchen auf die Divertorplatten in Tokamaks stellt eine der größten Herausforderungen für den langfristigen Betrieb von Fusionsreaktoren dar. Die deponierte Leistung darf materialspezifische Grenzwerte nicht überschreiten, woraus sich Einschränkungen für den Betrieb aktueller und zukünftiger Fusionsanlagen, wie etwa ITER, ergeben. Um diese Einschränkungen bei der Planung von Fusionsanlagen berücksichtigen und die Wärmelast auf den Divertorplatten während des Betriebs kontrollieren zu können, sind Vorhersagen für die zu erwartenden Wärmelasten nötig. Allerdings ist die Vorhersage dieser Wärmelasten vor einem Experiment ein komplizierter Prozess. Vereinfachte analytische Modelle berücksichtigen nicht alle atomistischen und molekularen Prozesse, die nahe an den Divertorplatten auftreten. Im Gegensatz dazu haben Simulationscodes, die diese Prozesse berücksichtigen, sehr lange Laufzeiten und können unter numerischer Instabilität leiden.
Um diese Probleme zu umgehen, wurden in der vorliegenden Arbeit Methoden des maschinellen Lernens zur Vorhersage der Wärmelasten an den Divertorplatten getestet. Da diese Modelle hochgradig nicht lineare Funktionen beschreiben können, könnten die gelernten Abhängigkeiten das Problem besser beschreiben als vereinfachte analytische Modelle. Außerdem sind die präsentierten Methoden des maschinellen Lernens rechnerisch günstiger als etablierte Simulationscodes.
Um die Methoden des maschinellen Lernens zu optimieren und zu testen wurde eine Datenbank aus beinahe 6 Jahren experimenteller Daten des *ASDEX Upgrade* (AUG) Experiments erstellt. Auf Basis dieser Daten wurden mehrere Methoden des maschinellen Lernens im Bezug auf ihre Fähigkeit, die Wärmelast an den Divertorplatten als Funktion zugänglicher Plasmaparameter vorherzusagen, untersucht. Alle untersuchten Modelle zeigen, dass es möglich ist, Vorhersagen der Wärmelast an den Divertorplatten als Funktion der gewählten Parameter zu erhalten.
Darüber hinaus wurden die erlernten Abhängigkeiten einiger Modelle untersucht. Obwohl sich Effekte störender Korrelationen zeigen, bestätigen einige der Abhängigkeiten die Erwartungen anderer physikalischer Analysen.
Schließlich wurden Ansätze getestet, um adäquate Vorhersagen für die Modellunsicherheiten zu erhalten. Zu diesem Zwecke wurden *Mixture Density Network*s (MDNs) und *Gaussian Process Regression* (GPR) genutzt. Allerdings zeigt die Analyse, dass weitere Arbeit nötig ist, um die Modellunsicherheiten vorherzusagen.

EIDESSTATTLICHE ERKLÄRUNG

Ich versichere an Eides Statt, dass die Dissertation von mir selbständig und ohne unzulässige fremde Hilfe unter Beachtung der „Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Heinrich-Heine-Universität Düsseldorf“ erstellt worden ist.

_____ Aachen, den _____

Martin Brenzke

PUBLICATIONS

Sections 4.2, 4.3 and 4.4 as well as parts of sections 5.1.1, 5.1.2, 5.2.1 and 5.2.2 have been published in [MBo].
The following figures have been published in the mentioned publication: 4.2, 4.3, 4.4, 4.5, 5.4, 5.5(a), 5.6(a), 5.6(c), 5.8(a), 5.9, 5.10(a), 5.10(d), 5.10(e), 5.11(c) 5.12(a), 5.13, 5.14, 5.15(a), 5.16(a), 5.16(c), 5.16(e), 5.17(b), 5.17(e), 5.18(a), 5.22, 5.23, 5.25(a), 5.25(e), 5.26(a), 5.26(c), 5.26(f), 5.24(a).

The author of this thesis implemented the data selection and the analysis of the data as well as the optimization and evaluation of the machine learning models presented in the publication.

[MBo] M. Brenzke et al., *Divertor power load predictions based on machine learning*, 2021 046023, ISSN: 0029-5515, DOI: 10.1088/1741-4326/abdb94.

CONTENTS

Part I

DEVELOPMENT OF SURROGATE MODELS FOR DIVERTOR POWER LOAD PREDICTION BASED ON MACHINE LEARNING TECHNIQUES

1 INTRODUCTION

The generation of power through nuclear fusion is inspired by the internal processes of the Sun. For example, in the proton-proton cycle two protons of the Sun's plasma interact via the weak interaction to form a deuteron (^{2}H), D [1]:

$$\mathrm{p} + \mathrm{p} \rightarrow \mathrm{D} + \mathrm{e}^+ + \nu_\mathrm{e} + 0.42\,\mathrm{MeV} \tag{1.1}$$

In this reaction an electron neutrino, ν_e, with an average energy of 0.267 MeV is released, as well as a positron which is immediately annihilated in the reaction with an electron to two photons, γ:

$$\mathrm{e}^+ + \mathrm{e}^- \rightarrow 2\gamma + 1.022\,\mathrm{MeV}\,. \tag{1.2}$$

In a subsequent step the deuteron can react with another proton via the strong interaction to form the isotope ^{3}He. The released energy stems from the mass difference between the inital protons and the final fusion product. This energy difference is referred to as binding energy.
Generating power through a similar process on Earth would be desirable. However, due to the weak interaction involved in the first step of the aforementioned reaction, the reaction rates are very small and require a vast active volume to substantiate relevant fusion rates. Furthermore, the driving mechanism of the Sun's stability is the interaction of the Sun's gravitation, which drives the fusion processes, and the thermal pressure caused by the fusion processes which counteracts the gravitational collapse. Since the accumulation of Sun-like masses on Earth is not possible the gravitational confinement of the plasma is not an option for sustained terrestrial fusion.
Thus, a different confinement mechanism as well as a different reaction channel are required for fusion on Earth.
Figure 1.1 shows the binding energy per nucleon against the mass number of isotopes. The arrows labeled 'Fusion' and 'Fission' indicate which process is beneficial in terms of achieving a larger binding energy per nucleon. Since the difference in binding energy between two elements is released during the fusion/fission process, as exemplified by the proton-proton cycle above, a large difference in binding energy per nucleon is beneficial for power generation. As can be seen in the figure, a large difference in binding energy per nucleon is achieved by the fusion of a deuteron and a triton (^{3}H) to ^{4}He. Thus, this fusion process yields a large energy gain per constituent.
Another factor to be taken into account for the selection of a desirable fusion process is the cross section of the process. The reaction rate of fusion processes between species 1 and 2 per unit volume can be calculated by

$$r = n_1 n_2 \langle \sigma v \rangle \ , \tag{1.3}$$

where n_1 and n_2 are the densities of the corresponding species and $\langle \sigma v \rangle$ is the reactivity, or reaction coefficient, where the product of the cross section σ and the relative velocity of the particles v is averaged over the distribution of velocities.

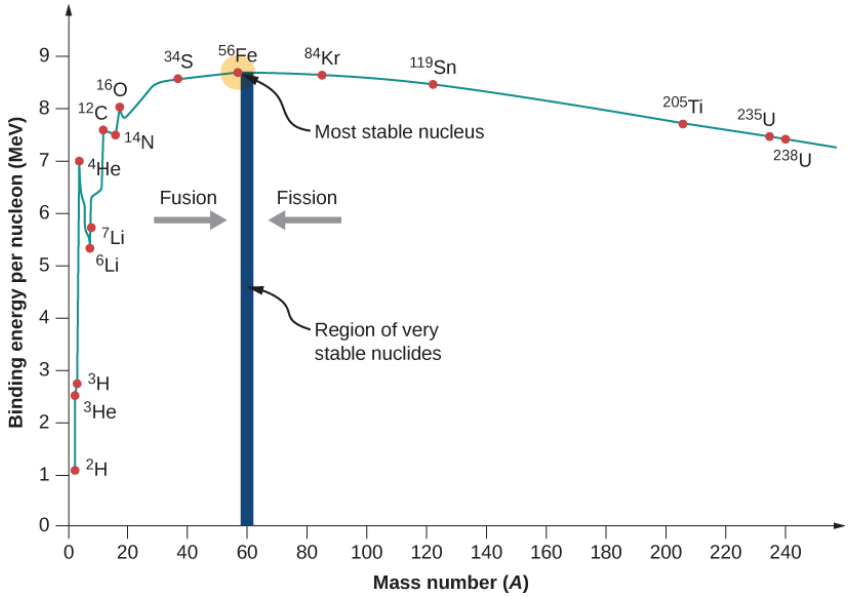


Figure 1.1: Binding energy per nucleon against atomic mass. The arrows labeled 'Fusion' and 'Fission' indicate which process is beneficial in terms of achieving a larger binding energy per nucleon. The region of maxium binding energy per nucleon (around ^{56}Fe) is marked. Image from [2].

Figure 1.2 shows the cross sections of several fusion processes of light isotopes as a function of the center-of-mass kinetic energy, i.e. the temperature of the plasma. It is obvious that the cross section for the fusion of deuteron and triton is largest at most temperatures. Hence, this process is the preferred process for fusion on Earth. In this reaction a deuteron and a triton form a ^{4}He and a neutron:

$$\mathrm{D} + \mathrm{T} \rightarrow {}^{4}\mathrm{He} + \mathrm{n} + 17.6\,\mathrm{MeV}\,. \tag{1.4}$$

The 17.6 MeV are split among the ^{4}He and the neutron according to their masses. This results in the neutron carrying an energy of 14.1 MeV. This energy is envisioned to be extracted from the plasma for power generation. The cross section for this reaction peaks around 64 keV which corresponds to a plasma temperature of about $7 \cdot 10^{8}$ K [3]. A sufficient plasma temperature for such fusion processes would be reached at about 15 keV since then ions with energies corresponding to the tail of the velocity distribution could already cause fusion reactions. However, despite the smaller cross section mostly the process of deuteron-deuteron fusion is studied in experiments since the acquisition of tritium is expensive and the envisioned in-vessel breeding of tritium has not yet been established.

Figure 1.2 and equation 1.3 show that a sufficiently large temperature and density

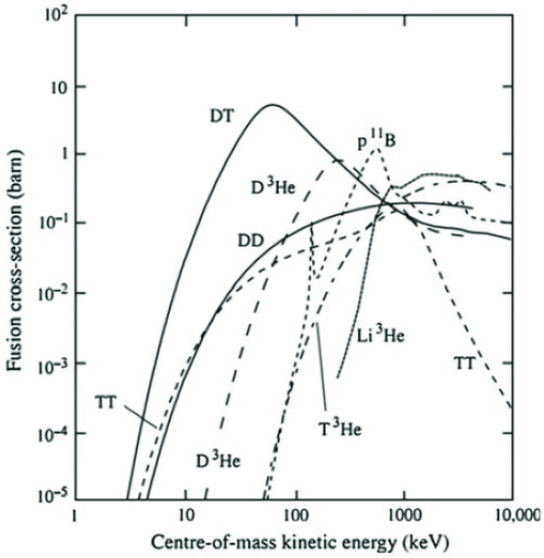


Figure 1.2: Cross sections of different fusion processes as function of the center-of-mass kinetic energy. Image from [4].

are prerequisites to enable frequent fusion processes. Furthermore, to achieve sustained fusion these conditions must be upheld over a longer time frame. To keep the temperature on the necessary level the plasma needs to be heated externally (e.g. via the injection of energetic particles). Since this heating costs energy it would be beneficial to utilize the energy of the fusion products, namely the ^{4}He, to keep the plasma heated. For this the plasma particles as well as the fusion products and the energy have to be contained for extended time periods.
Thus, the triple product of density, temperature and energy confinement time is a beneficial measure to quantify the performance of a fusion plasma. The Lawson criterion gives a lower limit for this triple product in order to obtain a self-heated deuterium-tritium fusion plasma with a net power gain [5]:

$$nT\tau_{\mathrm{E}} > 3 \cdot 10^{21} \frac{keVs}{m^3} \,. \tag{1.5}$$

Here n is the density of the fusion plasma, T is its temperature and τ_{E} is the energy confinement time.

As mentioned before, gravitational confinement of the plasma, as it appears in the Sun, is not feasible on Earth. Furthermore, simply encapsulating the plasma in a fixed volume is not feasible due to several reasons, one of which lies in the extreme temperatures of the plasma that would cause any material in direct contact with it to melt.
One way of confining the plasma and, thus, controlling the energy confinement time, is by containing the particles in a magnetic field. An established way to achieve this is the tokamak principle schematically depicted in figure 1.3.

The magnetic field is generated by the toroidal and poloidal field coils as well as the plasma current. The plasma current is caused by the transformator action between the primary winding, i.e. the central solenoid, and the plasma, which functions as secondary winding. This limits the operation time per plasma discharge as the current through the central soleniod needs to be changed monotonously.

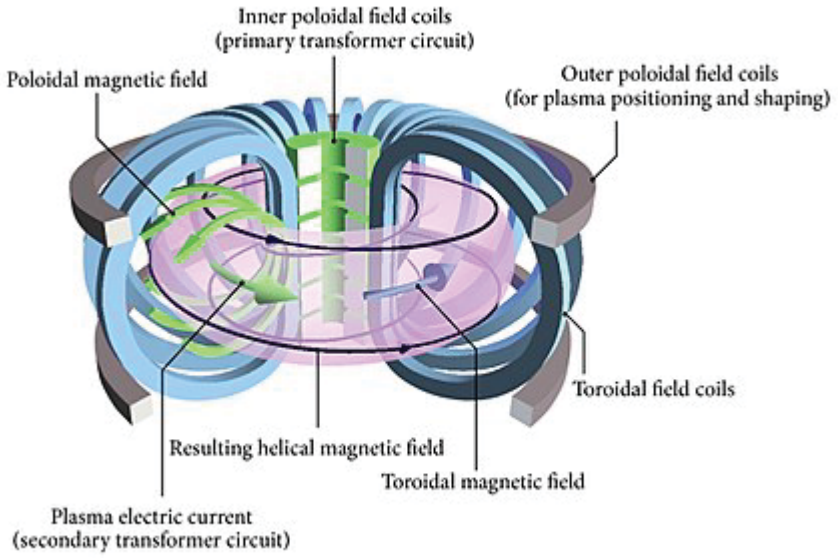


Figure 1.3: Schematic of a tokamak configuration. Image from [6].

In the conventional case of straight field lines, the Lorentz force limits the radial motion of charged particles in a magnetic field to the Larmor radius given by

$$r_{\mathrm{L}} = \frac{m v_{\perp}}{qB} \tag{1.6}$$

where m is the particle species' mass, $v_{\perp}$ the particle's velocity perpendicular to the field, q the absolute value of its electrical charge and B is the magnetic induction. Furthermore, since the magnetic moment, $\mu = \frac{m v_{\perp}^2}{2B}$, of the charged particles is an adiabatic constant, an increase of the magnetic induction can be utilized to increase the particles' velocity perpendicular to the magnetic field. Due to the conservation of energy, this results in a decrease of the particles' velocity parallel to the magnetic field and allows to invert the direction of this velocity component, assuming that the parallel kinetic energy of the particles is not too large. This principle is used to trap charged particles, e.g. in a magnetic mirror. To eliminate particle losses due to large parallel kinetic energies, the field lines can be closed to give the toroidal configuration of a tokamak.
However, in this case, the gradient of the magnetic field along the torus' major radius

causes additional drift effects. The particles' drift velocity caused by the gradient of the magnetic field and the geometric curvature is given by

$$\mathbf{v}_\mathrm{D} = \frac{m}{qB^3}(v_\parallel^2 + v_\perp^2)\mathbf{B} \times \nabla B\,. \tag{1.7}$$

As the direction of the drift depends on the particle's charge, this would lead to charge separation in the absence of the poloidal magnetic field resulting in an electric field and a corresponding additional outward drift of the particles with drift velocity

$$\mathbf{v}_\mathrm{D,E} = \frac{\mathbf{E} \times \mathbf{B}}{B^2}\,. \tag{1.8}$$

Since this $\mathbf{E} \times \mathbf{B}$ drift velocity does not depend on the charge, electrons and ions would be accelerated towards the outside surface of the torus. This outward drift would result in a so called kink instability of the plasma. This instability can be prevented by the introduction of the poloidal magnetic field component [5]. The combination of the toroidal and poloidal fields results in helical field lines along which the particles mainly move. These field lines lie on nested surfaces each of which has a constant magnetic flux.

The electrical power generation of a tokamak power plant would be mainly based on extracting the heat from the vessel surface caused by the impinging neutrons from the deuterium-tritium fusion. As the neutrons are not bound to the magnetic field, their energy will be deposited across the whole area of the vessel surface and can be extracted, e.g. via a heat exchanger.

Even though heating the plasma with the energy of the He ions is beneficial in terms of a net power gain, the ions themselves need to be extracted from the main plasma in order not to dilute the fusion fuel to a point at which the fusion reaction might stop. The same holds for all other impurities that would be sputtered from the walls by impinging plasma particles. One possible way to achieve this separation is the installation of a divertor in the tokamak.
The principle idea and the resulting poloidal cross-section of a plasma in a tokamak with a poloidal divertor are depicted in figure 1.4. The basic principle is that the magnetic field caused by an additional current causes a discontinuity in the flux surfaces. By interrupting the poloidal magnetic field the flux surfaces are opened from a certain radial point on outwards. These flux surfaces are intersected by the target plates of the divertor and lead particles towards these targets (divertor targets). The last of the closed flux surfaces is called the separatrix and marks the transition from the confined core of the plasma to the edge region, named *Scrape-Off Layer* (SOL). The point of vanishing poloidal magnetic field on the separatrix is called X-point.
The figure shows that in this configuration the plasma core is not in direct contact with any material. Thus, the influx of impurities can be diminished and controlled.

The plasma particles as well as other ions that are bound to the magnetic field lines follow the helical field lines in the SOL and are eventually led to the divertor targets. Since the field lines have a poloidal and a toroidal component the actual length the

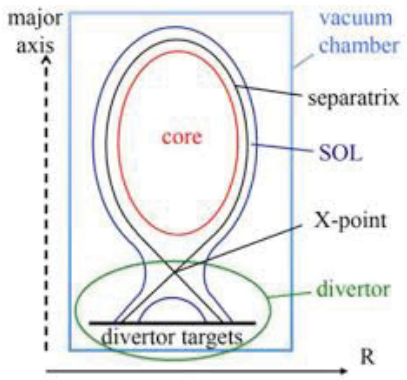


Figure 1.4: Schematic of a poloidally diverted plasma cross section. Marked are the last closed flux surface, called separatrix, the SOL outside the separatrix and the X-point where the poloidal magnetic field vanishes. Image from [7].

particles have to cover is larger than the immediate poloidal connection to the divertor targets. The actual travel length of the particles is given by the connection length, L, as given in [5]:

$$L \approx \pi R q \tag{1.9}$$

with the major radius of the torus R measured from the central axis, and the safety factor $q \approx \frac{rB_\phi}{RB_\theta}$. B_ϕ and B_θ refer to the toroidal and poloidal field strengths, respectively and r is the torus' minor radius measured across its cross section. At the divertor targets the ions interact with the target material and are neutralized. The resulting neutral gas can then be removed from the system by pumps.
However, due to the impinging hot ions from the plasma the divertor targets need to be able to withstand significant heat loads. This criterion will become even more crucial in future fusion devices of larger plasma volume and power, such as ITER for which the steady state power flux to the divertor targets needs to be limited to $\sim 10\,\frac{\mathrm{MW}}{\mathrm{m}^2}$ (see e.g. [8]). To assess this material limit and the challenges it imposes on the operation of future fusion devices, one can calculate the fraction of power that needs to be radiated in the SOL in order to stay within this limit. The deposited plasma power on the divertor targets can be estimated as

$$q_{\mathrm{target}} = (1 - f_{\mathrm{diss}}) \frac{P_{\mathrm{SOL}}}{4\pi R \lambda_{\mathrm{q}}} \frac{B}{B_\theta} sin(\alpha)\,. \tag{1.10}$$

Here, f_{diss} is the fraction of power dissipated in the SOL, P_{SOL} is the power crossing the separatrix into the SOL, α is the incidence angle of the field line with respect to the surface area of the divertor target and λ_{q} is the power fall-off width. An even splitting of the power between the inner and outer divertors is assumed. Assuming a maximum acceptable steady state power load of $q_{\mathrm{target,max}} = 10\,\frac{\mathrm{MW}}{\mathrm{m}^2}$ and ITER-like parameters (see [9] and [10]) $P_{\mathrm{SOL}} = 100\,\mathrm{MW}$, $R = 6.2\,\mathrm{m}$, $\lambda_{\mathrm{q}} \sim 1\,\mathrm{mm}$, $\frac{B}{B_\theta} \sim 3$ and $\alpha = 3^\circ$ results in $f_{\mathrm{diss}} \sim 0.95$. Thus, about 95% of the power entering the SOL need to be lost by line radiation or plasma-neutral interaction before the plasma reaches the divertor targets.

With a connection length of tens of meters for, e.g. the AUG tokamak, the plasma needs to be cooled from temperatures on the order of keV to several eV within this distance in order for the divertor targets to be able to withstand the power loads for extended time periods. On top of this the power loads are localized on scales of millimeters, see e.g. [10]. This sets the steady state power loads on the divertor targets as one of the main challenges in sustained operation of fusion reactors. Moreover, the approximation above does not include power deposited by radiation or neutral particles on the divertor target which aggrevates the challenge of controlling the power loads at the divertor targets.
A way of reducing the power loads at the divertor targets during the experiment is cooling of the plasma by controlled injection of impurity species, such as nitrogen or neon, into the divertor region [11]. These impurities will cool the plasma via line radiation which allows to spread the energy of the plasma over larger areas since the emitted photons are not bound to the magnetic field lines. The effective volumetric power loss depends on the impurity species and the electron temperature (see e.g. [12]). Thus, impurity species of larger atomic numbers, such as argon and krypton tend to mainly cause energy losses at high electron temperature (i.e. close to the plasma core), whereas lighter species, such as nitrogen and neon are more suited to mostly cause radiative losses in the divertor and SOL region.
The aforementioned requirement of significant power dissipation within the SOL and the effects of impurities on the performance of the core plasma (i.e. fuel dilution and cooling of the plasma) conflict with one another and limit the operational scenarios for any (future) fusion devices. Hence, a careful balancing of operational parameters with regard to particle and power exhaust and plasma core performance is a crucial challenge in fusion science.
A further important process in reducing divertor power loads is the achievement of plasma detachment. In the regime of plasma detachment a reduction of power and particle fluxes to the target surface is achieved by inducing significant power and momentum losses in the SOL, thus, effectively reducing the power loads the divertor targets experience [13]. Future fusion devices will have to utilize this operational regime in order to fulfill the divertor power load restrictions.
The regime of plasma detachment can be accessed by e.g. increasing the plasma density at the separatrix. With increasing separatrix density the density at the divertor targets shows a roll-over and decreases (see e.g. [14]). Low plasma temperatures are a necessary criterion for achieving divertor detachment. At these temperatures (a few eV) atomic and molecular processes become an important aspect of the divertor physics. Due to the significance of the interaction between plasma particles and neutrals, a combination of fluid dynamics and kinetic approaches is necessary to model and investigate this regime. On top of this, the interplay of plasma physics, nuclear physics and geometry effects hamper (analytical) modeling approaches.

Nevertheless, predictions of steady state divertor power loads prior to performing discharges are an essential part of safe operation of fusion reactors and studying the dependencies of such power loads is integral to the design of experiments and machines. For the purpose of large scale dependency studies simplified models within coupled fluid Monte Carlo simulation codes are usually used (see e.g. [15]). Applying simplified

models for these purposes is necessary since the fully fledged codes usually take weeks to months until convergence for a single plasma discharge which commonly leads to prohibitve costs for large scale studies. Simplified analytical models for the prediction of divertor power loads do exist and will be discussed in a later chapter. However, these models do not apply in the regime of detached plasmas.
Both, simplified analytical models and fully-fledged simulation codes, also require parameters that are usually not well known prior to performing a discharge, such as the power crossing the separatrix, loss terms describing the power and momentum losses in the SOL (as e.g. in equation 1.10) or parameters to describe the anomalous (turbulent) transport in the plasma edge which requires additional approximations.

The aim of this thesis is to develop and test models based on machine learning techniques to alleviate the problems of divertor power load predictions. Machine learning based approaches have been tested and established for a range of tasks in fusion research. Most commonly these models are applied to predict disruptions of the plasma (see e.g. [16], [17]). Another exemplary application is the speed-up of turbulent transport simulations [18].
Machine learning based models for divertor power load predictions could potentially be more accurate than simplified analytical models and will have a significant advantage over simulations in terms of computing time, thus, ameliorating the potential for large scale studies of divertor power loads. One of the main benefits of the models under investigation is the fact that they can describe very complex nonlinear dependencies. In the present application this is necessary due to the multitude of physics processes and scales involved. Another advantage of these models is that they do not require a first principles based approach and could thus (implicitly) describe dependencies not well described by conventional analytical models, such as the plasma-neutral interaction. Hence, the resulting models could be more suitable for use in the regime of plasma detachment than analytical models, since the participating processes get more involved as the importance of molecular and atomic interactions increases when approaching detachment.

By using generally accessible plasma parameters as the input quantities to the models the applicability of the models is extended in comparison to established models that require knowledge of less accessible parameters (e.g. the power flux in the SOL). A broad selection of data for the analysis ensures the models can make predictions over a wide range of operational scenarios. The analysis of the models' dependencies shows agreement with dedicated physics analyses but also reveals dependencies that require further investigation.

The following chapter will introduce the physics background focusing on SOL transport and divertor physics. Simplified models will be elaborated and general approaches and problems of simulation codes will be discussed. The basics of the machine learning approaches used in this thesis will be presented in chapter 3. The construction and analysis of the data base used for this thesis are described in chapter 4. Chapter 5 contains the results obtained in this analysis and a discussion of these results. Chapter 6

consists of this thesis' conclusion and gives an outlook on potential further research in the area of machine learning based approaches to divertor power load predictions.

2

SOL AND DIVERTOR PHYSICS AND MODELING

This chapter introduces physics concepts of and (one dimensional) analytical modeling approaches to the *Scrape-Off Layer* (SOL) and divertor plasma with a focus on particle and heat fluxes towards the divertor target surface. The basic concepts, such as the fluid equations, will also be relevant for explanations of established simulation codes at the end of this chapter.

2.1 THE KINETIC AND FLUID EQUATIONS

The following section gives an introduction to the kinetic and fluid equations often used to model the SOL plasma and follows [19], [5] and [20].

The large number of particles in a plasma necessitates a statistical approach to modeling the plasma. The particle distribution function $f_\alpha(\mathbf{x}, \mathbf{v}, t)$ gives the probability density of every phase space state $(\mathbf{x}, \mathbf{v})$ at a given time t for the given particle species α. The evolution of this function is given by the kinetic equation and can be described at various levels of rigor. Considering particles in external electric and magnetic fields, $\mathbf{E}$ and $\mathbf{B}$, and including a term for collisions one such description results in the Boltzmann equation

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{v} \cdot \nabla_\mathbf{x} f_\alpha + \frac{q_\alpha}{m_\alpha}(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_\mathbf{v} f_\alpha = \left(\frac{\partial f_\alpha}{\partial t}\right)_\mathrm{c} . \tag{2.1}$$

Here, $\nabla_\mathbf{x}$ denotes the spatial derivative, $\nabla_\mathbf{v}$ the derivative in velocity space, q_α and m_α are the particle species' electric charge and mass and $\left(\frac{\partial f_\alpha}{\partial t}\right)_\mathrm{c}$ describes the changes in the distribution function due to collisions. This approach does not take into account microscopic fluctuations but gives a representation of a large number of particles in a given volume.

Solving this equation approximately is possible in some situations but is costly for large numbers of particles. A simplified approach is valid in the case of a collisional plasma where fluid equations suffice to describe the system to a satisfactory level. The fluid equations can be derived from the kinetic equation by calculating moments of the particle distribution function, for example:

$$n_\alpha = \int f_\alpha(\mathbf{x}, \mathbf{v}', t) d\mathbf{v}' \tag{2.2}$$

$$\mathbf{v}_\alpha = \frac{1}{n_\alpha} \int \mathbf{v}' f_\alpha(\mathbf{x}, \mathbf{v}', t) d\mathbf{v}' \tag{2.3}$$

$$\mathbf{P}_\alpha = m_\alpha \int (\mathbf{v}' - \mathbf{v})^2 f_\alpha(\mathbf{x}, \mathbf{v}', t) d\mathbf{v}' . \tag{2.4}$$

Here, $\mathbf{v}'$ denotes the single particle velocity whereas $\mathbf{v}$ denotes the fluid velocity, n_α is the fluid's density and $\mathbf{P}_\alpha$ is the pressure tensor.
An approach based on fluid equations is mostly valid for the SOL since the plasma in the SOL is highly collisional due to its low temperature (in comparison to the plasma core). The mean free path length of electron-electron and ion-ion self collisions, λ_{ii} and λ_{ee}, is given by $\lambda_{ii} \approx \lambda_{ee} \approx \frac{10^{16} T^2}{n_e}$ [20]. With electron temperatures of $\lesssim 100\,\mathrm{eV}$ and densities of $\sim 10^{19} \frac{1}{\mathrm{m}^3}$ this results in $\lambda_{ii} \approx \lambda_{ee} \lessapprox 10\,\mathrm{m}$. For a plasma to be considered collisional $\lambda < L$ must hold, where in the case of the SOL L is the connection length. Thus, the SOL plasma can usually be considered collisional and fluid approximations are often sufficient to describe the main properties of the SOL.
From equation 2.1 and equations 2.2 to 2.4 follow the fluid equations describing the evolution and transport of density, momentum and energy in the plasma by calculating moments of the kinetic equation [19]:

$$\frac{\partial n_\alpha}{\partial t} + \nabla \cdot (n_\alpha \mathbf{v}_\alpha) = S_\alpha \tag{2.5}$$

$$m_\alpha n_\alpha \frac{\partial \mathbf{v}_\alpha}{\partial t} + m_\alpha n_\alpha (\mathbf{v}_\alpha \cdot \nabla)\mathbf{v}_\alpha = -\nabla \cdot \mathbf{P}_\alpha - \nabla \cdot \Pi_\alpha + q_\alpha n_\alpha (\mathbf{E} + \mathbf{v}_\alpha \times \mathbf{B}) + \mathbf{R}_\alpha + \mathbf{S}_{\mathrm{mom},\alpha} \tag{2.6}$$

$$\begin{aligned} \frac{\partial}{\partial t}\left(\frac{n_\alpha m_\alpha}{2} \mathrm{v}_\alpha^2 + \frac{3}{2} n_\alpha T_\alpha\right) + \nabla \cdot \left[\left(\frac{n_\alpha m_\alpha}{2} \mathrm{v}_\alpha^2 + \frac{5}{2} n_\alpha T_\alpha\right) \mathbf{v}_\alpha + \Pi_\alpha \mathbf{v}_\alpha + \mathbf{q}_{\mathrm{heat},\alpha}\right] \\ = q_\alpha n_\alpha \mathbf{E} \cdot \mathbf{v}_\alpha + \mathbf{R} \cdot \mathbf{v}_\alpha + Q + S_{\mathrm{en},\alpha}. \end{aligned} \tag{2.7}$$

These are the conservation equations for particles, momentum and energy, called Braginskii equations. Herein, S_α, $\mathbf{S}_{\mathrm{mom},\alpha}$ and $S_{\mathrm{en},\alpha}$ are source terms for particles, momentum and energy that have been added to describe an additional influx or loss of the given quantity for species α, e.g. via ionization of neutrals. Π_α denotes the stress tensor with components $\Pi_\alpha^{ij} = n_\alpha m_\alpha \langle (\mathbf{v}' - \mathbf{v})_i (\mathbf{v}' - \mathbf{v})_j - \frac{(\mathbf{v}' - \mathbf{v})^2}{3} \delta_{ij} \rangle$ with the Kronecker delta δ_{ij}. $\mathbf{R}_\alpha$ denotes the rate of momentum transfer caused by collisions with other species. T_α is the species' temperature. $\mathbf{q}_{\mathrm{heat},\alpha}$ is the heat flux density remaining after a transformation to the fluid's rest frame, i.e. heat conduction. Finally, Q describes the heat caused by collisions of particles of species α with particles of other species. The term $q_\alpha n_\alpha \mathbf{E} \cdot \mathbf{v}_\alpha$ describes the Ohmic heating of species α.
Thus, the fluid equations, when calculated up to the given order, result in 3 equations for 4 unknowns, i.e. n_α, $\mathbf{v}_\alpha$, T_α and $\mathbf{q}_{\mathrm{heat},\alpha}$.

As can be seen in equations 2.5 to 2.7, the fluid equations for each moment of the kinetic equation depend on the moment of the next order, e.g. the equation for n_α depends on $\mathbf{v}_\alpha$. This leads to a closure problem and requires additional assumptions. The closure problem becomes easy when heat conduction can be ignored, as is the case in e.g. the sheath-limited regime (see section 2.3.1). Otherwise a common assumption about the contribution of conductive parallel-to-$\mathbf{B}$ heat transport is given by

$$q_{||,\mathrm{cond},\alpha} = -\kappa_{0,\alpha} T_\alpha^{\frac{5}{2}} \nabla_{||,x} T_\alpha \tag{2.8}$$

with $\kappa_{0,e} = \frac{30692}{Z_i ln(\Lambda)}$ and $\kappa_{0,i} = \frac{1249}{Z_i^4 \sqrt{m_i} ln(\Lambda)}$ for electrons and ions, respectively. Here, only one ion species with charge Z_i is considered and $ln(\Lambda)$ is the Coulomb logarithm, describing the ratio of the Debye length (see the next section) and the impact parameter, i.e. vertical distance between the path of an incident particle and a stationary target particle, for a 90° scatter, which depends on the particle velocity [5]. By making an assumption as in equation 2.8, it is possible to determine a closed form of the fluid equations. Since the transport of particles and heat is dominated by parallel-to-**B** transport, it is a common approach to decouple the perpendicular and parallel directions and to solve the fluid equations for each direction separately.

2.2 THE PLASMA SHEATH

The plasma sheath forming in front of the plasma facing components (PFC) is an essential (and limiting) part of particle and energy transport in the SOL. This section will introduce basic concepts of the plasma sheath and some key properties linked to transport in the SOL. This section follows [5] and [20].

The surfaces in contact with the plasma, i.e. the divertor targets, function as a sink and a source for plasma particles. This results in a plasma flow towards the surfaces caused by the gradient of the dynamic pressure. During the start-up phase of a plasma discharge (in the first few μs), the electrons reach the surfaces of the divertor targets before the ions due to the fact that their thermal velocity is larger by a factor $\sqrt{\frac{m_i}{m_e}}$, assuming equal temperatures of electrons and ions, $T_e \approx T_i$. This results in a net negative charge of the PFC which counteracts the electron flux and increases the ion flux towards the surface. The potential difference between the plasma and the surface adjusts until an ambipolar flux of ions and electrons is reached, resulting in a wall potential of $V_{wall} \sim -3\frac{T_e}{e}$ with respect to the plasma potential, where e denotes the elementary charge. This results in a region of net positive charge in front of the surface, called the plasma sheath. This sheath almost perfectly shields the plasma from the accumulated charge on the surface. The spatial extent of this sheath is on the order of a Debye length given by

$$\lambda_{Debye} = \sqrt{\frac{\epsilon_0 T_e}{n_e e^2}}\,. \tag{2.9}$$

Due to this, the fluid approximations discussed in section 2.1, do not hold in this area of the SOL since the condition $\lambda < L$ is violated with $L \approx \lambda_{Debye}$ and $\lambda_{Debye} \approx 10^{-5}$ m for, e.g. $T_e = 20$ eV and $n_e = 10^{19} m^{-3}$.

Since the plasma sheath does not fully shield the bulk plasma from the wall potential, the resulting electric field, penetrating through the sheath into the plasma, accelerates the ions such that an ambipolar flux of electrons and ions is achieved. The remaining

pre-sheath potential can be approximated to be roughly given by $V_{\text{pre-sheath}} \sim -0.7\frac{T_e}{e}$. The requirement of an ambipolar flux results in the Bohm criterion [21]

$$v_{i,se} \geq c_s = \sqrt{\frac{T_e + \gamma T_i}{m_i}} \tag{2.10}$$

with the adiabatic coefficient γ, the ion flow velocity at the sheath entrance $v_{i,se}$ and the ion sound speed c_s. Usual values for γ are $\gamma = 1$ for isothermal flow, $\gamma = \frac{5}{3}$ for adiabatic flow with isotropic pressure and $\gamma = 3$ for 1D adiabatic flow.
Given the Bohm criterion it is possible to estimate the one dimensional parallel-to-**B** ion flux towards the divertor target's surface. Making the simplifying assumption of a drifting Maxwellian distribution for the ions with drift velocity $v_{i,drift}$, the ion flux can be estimated as $\Gamma_i = n_i v_{i,drift}$. Using the conservation of ion flux one obtains $\Gamma_{i,target} = n_{i,target} v_{i,target} = \Gamma_{i,se} = n_{i,se} c_s$. With this, it is possible to estimate the power flux density towards the target transmitted by the ions as a function of plasma temperature and density:

$$\begin{aligned} q_{i,se} &= \left(\frac{5}{2}T_i + \frac{1}{2}m_i v_{i,drift}^2\right)\Gamma_{se} \\ &= \left(\frac{5}{2}T_i + \frac{1}{2}m_i c_s^2\right)\Gamma_{se} \\ &= \gamma_i T_i \Gamma_{se}\,. \end{aligned} \tag{2.11}$$

For the last equation, we assumed $T_e = T_i$ and an adiabatic coefficient $\gamma = 1$. The coefficient γ_i is the ion sheath heat transmission coefficient. The first summand in the brackets describes the thermal energy carried by the ions, whereas the second summand is the ion kinetic energy.

An analysis for the electron power flux density yields:

$$q_{e,se} = \gamma_e T_e \Gamma_{se} \tag{2.12}$$

with the electron sheath heat transmission coefficient $\gamma_e = 2 + \frac{|eV_{wall}|}{T_e} + \frac{|eV_{\text{pre-sheath}}|}{T_e}$. Due to the ambipolarity of ion and electron flux, it is $\Gamma_{i,se} = \Gamma_{e,se} = \Gamma_{se}$. The two terms including the potential describe the effects of the pre-sheath potential and the potential drop along the sheath. In both cases, the potential acts in such a way as to reduce the electron flux and increase the ion flux. Thus, effectively transferring energy from electrons to ions, see figure 2.1. The total power flux density to the divertor target is therefore given by

$$\begin{aligned} q_{ss} &= q_{i,ss} + q_{e,ss} \\ &= q_{i,se} + q_{e,se} \\ &= (\gamma_i T_i + \gamma_e T_e)\Gamma_{se} \end{aligned} \tag{2.13}$$

since, in this simplified analysis, the only process effectively changing the power flux densities at the solid surface (ss) in comparison to those at the sheath edge (se) is the

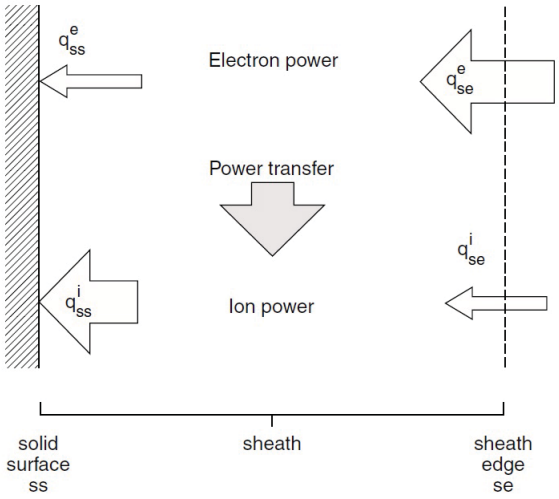


Figure 2.1: Schematic of the plasma sheath in front of the solid surface and the transfer of energy from the electrons to the ions due to the potential drop across the sheath. Image from [20].

transfer of energy from the electrons to the ions, so that $q_{\mathrm{i,ss}} + q_{\mathrm{e,ss}} = q_{\mathrm{i,se}} + q_{\mathrm{e,se}}$.

The results above do not take into account a potential obliqueness of the divertor target surface with respect to the magnetic field lines. However, an oblique target surface is usually beneficial in terms of deposited power flux density, as the deposited power flux density can be reduced by a factor of $\cos(\Psi)$ when the target surface is inclined, where Ψ denotes the angle between the magnetic field lines and the surface normal. An analysis concerning oblique target surfaces, carried out by Chodura [22], first introduced an additional region in front of the sheath, the so-called magnetic pre-sheath. This magnetic pre-sheath has an extension of a few ion Larmor radii and is electrically quasi-neutral. As Chodura's analysis showed, the Bohm criterion must already be fulfilled for the parallel-to-**B** component of the ion velocity at the entrance to the magnetic pre-sheath. A schematic of the sheath region is given in figure 2.2. As can be seen, the electrons pass the magnetic pre-sheath unaffected due to their high mobility. However, the trajectory of the ions is changed so that the Bohm criterion is fulfilled for the ion's velocity component perpendicular to the target surface. Hence, the obliqueness of the target surface has little influence on the results given above except for geometrically reducing the deposited power flux density.

In total the sheath sets the boundary conditions for the SOL plasma and is an integral part for modeling the plasma and the plasma-wall interactions. However, simplified, i.e. fluid, approaches do not fully suffice to model this region of the SOL plasma.

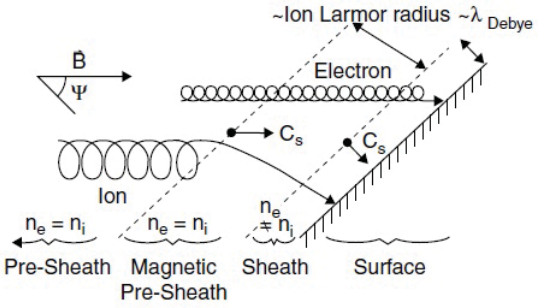


Figure 2.2: Schematic of the plasma sheath region in front of an oblique surface including the quasi-neutral magnetic pre-sheath with a thickness of a few ion Larmor radii. As shown by Chodura [22], the ions must fulfill the Bohm criterion at the entrance to the magnetic pre-sheath, so that the obliqueness of the surface has little influence on the results given above. Image from [20].

2.3 REGIMES OF DIVERTOR PLASMAS

As indicated in chapter 1, the state of the divertor can pass through different physical regimes, one of which is the aforementioned regime of plasma detachment. This further complicates finding an overarching model, since dependencies and boundary conditions change with the physical regime of the divertor. This section gives an introductory overview of the different regimes and typical characteristics of each. Basic analytical models describing the plasma conditions close to the divertor targets are introduced. The section follows [20].

2.3.1 *The Sheath-Limited Regime*

The physical regime of the divertor and SOL plasma is strongly dependent on the plasma temperature near the divertor target, see [20], which, in turn, strongly depends on the plasma density close to the separatrix (see section 2.3.2). However, the actual parameter determining the physical regime of the SOL and divertor plasma is the plasma's collisionality which is a function of the plasma density and strongly depends on the plasma thermal velocity, i.e. temperature (see [23]).

As the plasma density near the separatrix increases, different regimes are realized in the divertor and SOL plasma. Figure 2.3 shows results from the ASDEX experiment visualizing the dependence of the plasma density (i.e. particle flux) and the plasma temperature near the divertor targets on the core density, $\bar{n}_e$. For the smallest $\bar{n}_e$ the divertor is in the low recycling or sheath-limited regime. Here, the target density n_t increases approximately linear with the core density $\bar{n}_e$ and the electron temperature close to the target is rather large with $T_{ed} \gtrapprox 20\,\text{eV}$. The sheath-limited regime is characterized by negligible temperature gradients along the magnetic flux tubes and thus also

heat conduction is negligible as it scales with the temperature gradient (cf. equation 2.8). Hence, heat transport is dominated by convective transport. Furthermore, as the name low recycling regime implies, recycling, i.e. the emission of neutralized plasma particles from the solid surface, is weak in this regime, so that the divertor target can be neglected as a particle source and only the influx of particles and power from the confined plasma core is relevant. In the sheath-limited regime the heat transmission of the sheath determines the transport properties of the SOL as described in section 2.2.

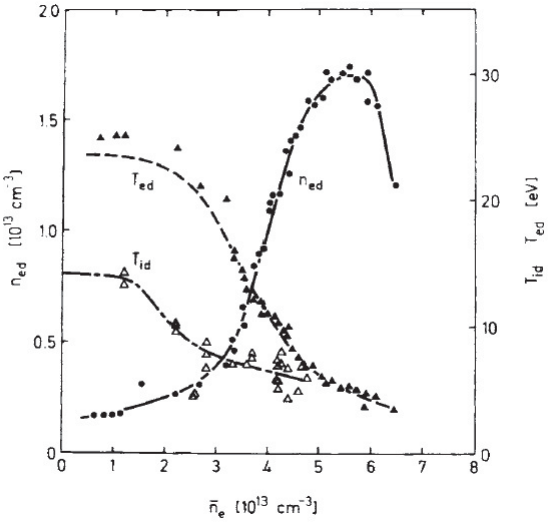


Figure 2.3: Dependency of electron density, n_{ed}, and electron and ion temperature, T_{ed} and T_{id}, at the divertor target on core plasma density $\bar{n}_e$. Image from [20].

A low collisionality of the plasma is characteristic for the sheath-limited regime which is in contrast to the approach of a model based on the fluid equations of section 2.1. Nevertheless, a comparative evaluation of different models shows that a fluid approach still yields the same main results as other models, e.g. kinetic models, and is, thus, still valid [20]. With the approach of an isothermal fluid model and the assumptions that all particles enter the SOL by crossing the separatrix with zero velocity and that power also only enters the SOL via cross-field transport from the core, it can be found:

$$\begin{aligned} n_{se} &= \frac{1}{2} n_0 \\ \Gamma_{se} &= \frac{1}{2} n_0 c_s \end{aligned} \tag{2.14}$$

where n_0 denotes the density at the point of the particles' entrance into the SOL. From this and equation 2.13 one obtains

$$\begin{aligned} T &\propto \left(\frac{q_{ss}}{\gamma' n_0}\right)^{\frac{2}{3}} \\ \Gamma_{se} &\propto \left(\frac{q_{ss} n_0^2}{\gamma'}\right)^{\frac{1}{3}} \end{aligned} \tag{2.15}$$

where $\gamma' = \gamma_e + \gamma_i$. In this form both, temperature and particle flux towards the divertor target, can be expressed as functions of the plasma density close to the separatrix which can be considered a controllable parameter, as well as the power flux density towards the divertor target, which is equal to the power flux density close to the separatrix since the parallel-to-**B** power flux density throughout the SOL is constant under the given conditions, i.e. $q_{ss} = q_{se} = q_{||,SOL}$.

2.3.2 *The Conduction-Limited Regime and the Basic Two Point Model*

As $\bar{n}_e$ in figure 2.3 increases, the density at the divertor target starts to increase more than quadratically with the core plasma density. This indicates the onset of the conduction-limited regime which is also termed high recycling regime for the largest $\bar{n}_e$ still in this regime. As the term high recycling indicates, the effect of the divertor target surfaces acting as particle sources can no longer be neglected in the high density part of this regime as the influx of neutralized plasma particles from the wall is significant. Furthermore, there occur significant temperature gradients along the SOL in the conduction-limited regime caused by the finite conductivity of the plasma and the dominance of conductive heat transport over convective transport at the increased collisionality of the plasma. These temperature gradients allow for small plasma temperatures near the divertor targets (see figure 2.3), and thus reduced sputtering of the target material, while maintaining large temperatures close to the separatrix.
The cooling effect of radiation processes becomes stronger as the electron temperature close to the divertor targets decreases leading to a significant effect of radiative energy losses close to the targets.

Under these conditions, a simple zero dimensional model connecting the plasma properties close to the separatrix and at the divertor targets can be derived [20]. The basic *Two Point Model* (TPM) does not describe the plasma properties as a function of the position along the magnetic flux tube, but directly relates the separatrix conditions to the target conditions.
Assuming that the plasma particles are only accelerated in the ionization zone close to the divertor targets, that their velocity at the sheath entrance is equal to the sound speed (see equation 2.10), that total pressure along a magnetic flux tube is constant and that

the full power flux density, $q_{||,\mathrm{SOL}}$, enters from the plasma core via the separatrix and is only transported via conduction one obtains:

$$\begin{aligned} n_\mathrm{t} T_\mathrm{t} &= \frac{1}{2} n_\mathrm{u} T_\mathrm{u} \\ T_\mathrm{u}^{\frac{7}{2}} &= T_\mathrm{t}^{\frac{7}{2}} + \frac{7}{2} \frac{q_{||,\mathrm{SOL}} L}{\kappa_{0,\mathrm{e}}} \\ q_{||,\mathrm{SOL}} &= \gamma' n_\mathrm{t} T_\mathrm{t} c_\mathrm{s} \,. \end{aligned} \tag{2.16}$$

With the indices t and u indicating target and upstream, i.e. close to the separatrix, positions, respectively and the connection length L (see equation 1.9). So n_u is identical to n_0 in 2.15. Here, it is assumed that volumetric power losses can be neglected since they only happen in the thin ionization region and, thus, the temperature change therein was neglected resulting in $q_{||,\mathrm{SOL}} = q_\mathrm{t} = \gamma' n_\mathrm{t} T_\mathrm{t} c_\mathrm{s}$.
Solving these equations under the assumption $T_\mathrm{t} \ll T_\mathrm{u}$ then results in the main relations of the basic TPM:

$$T_\mathrm{t} \propto q_{||,\mathrm{SOL}}^{10/7} L^{-4/7} n_\mathrm{u}^{-2} \tag{2.17}$$

$$n_\mathrm{t} \propto n_\mathrm{u}^{3} L^{6/7} q_{||,\mathrm{SOL}}^{-8/7} \tag{2.18}$$

$$\Gamma_\mathrm{t} \propto n_\mathrm{u}^{2} L^{4/7} q_{||,\mathrm{SOL}}^{-3/7} \,. \tag{2.19}$$

It is evident that the plasma temperature at the target depends strongly on the upstream density, so that large upstream densities are preferred in order to reduce the plasma temperature at the target and thus reduce sputtering of the target surface. Furthermore, the temperature at the target increases with the parallel power flux density. With regard to the desired parameters of the core plasma, the density dependency of T_t is beneficial as large core densities, tentatively also resulting in large n_u, result in a larger output of fusion power (cf. equation 1.3 and [5]).
On the other hand the particle flux toward the target, Γ_t, increases with the upstream density. Thus, the net effect of increasing the upstream density on the sputtering of the target surface is not immediately evident. However, since the physical sputtering flux is given by $\Gamma_\mathrm{sputtering} = Y(T_\mathrm{t})\Gamma_\mathrm{t}$ where $Y(T_\mathrm{t})$ is the sputtering yield, which is usually strongly increasing with T_t under the given conditions, an increase of n_u should result in a reduced physical sputtering of the target surface[1].
The dependency of the target temperature and the particle flux towards the divertor target on the power flux density, $q_{||,\mathrm{SOL}}$, shows an inverse behavior in comparison to n_u. Increasing the power flux density in the SOL causes an increase of the plasma temperature at the targets but a decrease of the particle flux. However, the particle flux depends only weakly on the power flux density in comparison to the target temperature. Thus, a reduction of the power flux density in the SOL would also result in reduced sputtering of the target surface. Hence, stronger radiation losses along the SOL would reduce the physical sputtering caused by the plasma particles impinging on the target.

1 Chemical sputtering, i.e. the formation and desorption of weakly bound compounds of impinging ions and surface atoms of the divertor target material, plays a negligible role for metallic surfaces, such as tungsten, which is going to be the main material of the divertor targets of ITER. However, chemically assisted physical sputtering can still play an important role and is under further investigation [24].

2.3.3 *Extensions of the Basic Two Point Model*

Processes that are not being accounted for in the basic TPM can be approximated by additional factors. The power loss due to volumetric radiation and charge exchange processes of neutrals and plasma particles in the SOL can be summarized by a power loss factor, f_{power}, so that $q_{\text{t}} = (1 - f_{\text{power}})q_{||,\text{SOL}}$. This approximation holds as long as the zone of ionization is located in a thin layer in front of the divertor targets, as was the assumption in the previous section. Additionally, a factor describing momentum losses caused by collisions of plasma particles with neutrals, viscosity of the plasma and volumetric recombination of plasma particles can be introduced. Using the relation of upstream and target pressure, 2.16, it is $p_{\text{t}} = f_{\text{mom}} \frac{1}{2} p_{\text{u}}$ with the momentum loss factor f_{mom}. Furthermore, convection could cause a reduction of temperature gradients in comparison to the purely conduction based TPM above. Hence, a conduction factor, f_{cond}, is introduced, so that $q_{||,\text{cond}} = f_{\text{cond}} q_{||,\text{SOL}}$.
For these additional factors one finds:

$$T_{\text{t}} \propto \frac{(1 - f_{\text{power}})^2}{f_{\text{mom}}^2 f_{\text{cond}}^{4/7}} \tag{2.20}$$

$$n_{\text{t}} \propto \frac{f_{\text{mom}}^3 f_{\text{cond}}^{6/7}}{(1 - f_{\text{power}})^2} \tag{2.21}$$

$$\Gamma_{\text{t}} \propto \frac{f_{\text{mom}}^2 f_{\text{cond}}^{4/7}}{1 - f_{\text{power}}} \,. \tag{2.22}$$

Thus, volumetric power loss processes tend to strongly decrease the plasma temperature at the divertor target but cause a weak increase in the particle flux towards the target. Momentum loss processes tend to have the inverse effect since they strongly increase the temperature of the plasma at the divertor target but decrease the particle flux. Hence, a balancing of power and momentum losses is necessary in order to achieve the aim of small temperatures and fluxes at the divertor target. The plasma density at the target strongly depends on both, power and momentum loss processes.

It has to be noted that the loss factors introduced here encode complex and strongly non-linear processes and that analytically modeling these factors proves to be difficult. Therefore, these factors have to be estimated from experiments or simulation codes (cf. [25]).
Moreover, all of the derivations and results above only apply under the assumption of a plasma in a steady state. However, in certain regimes, which are envisioned for the operation of future fusion devices, the plasma exhibits periodic transient events in the edge region. So called *Edge Localized Mode*s (ELMs) cause a temporary increase in the particle and power transport on a time scale of ∼ms and, thus, further exacerbate the challenge of modeling the SOL plasma [26].

2.3.4 *The Regime of Divertor Detachment*

With a further increase in core density, the plasma density at the target starts to stagnate and finally rolls over (see figure 2.3). As was shown in the TPM, an increasing upstream density results in a decreasing plasma temperature at the divertor target. The decreasing plasma temperature enhances the power and momentum losses due to more available excitation states of the neutrals as well as the onset of volumetric recombination as dominating loss process at very low temperatures of $\lessapprox 1\,\mathrm{eV}$ (cf. figure 2.4). Hence, atomic and molecular processes become more important in the regime of plasma detachment which hampers analytical modeling approaches.

As observations show, an important feature of the regime of detachment is a substantial drop in the measured ion flow towards the divertor target, also indicated by the drop in n_{ed} in figure 2.3.
To induce both, the drop in ion temperature and flow, power and momentum losses are needed along the SOL, as indicated by the extended TPM. From equations 2.20 and 2.22 it is evident that power losses cause a reduction of the plasma temperature at the divertor target. However, if only power losses were present the flux towards the target would actually increase. Hence, momentum losses are also needed to reduce the particle flux towards the divertor target. The latter is in accordance with observations of significant drops in the plasma pressure along the SOL, e.g. [27]. This observation, however, is in contrast to the aforementioned regimes where no such pressure gradient is present.

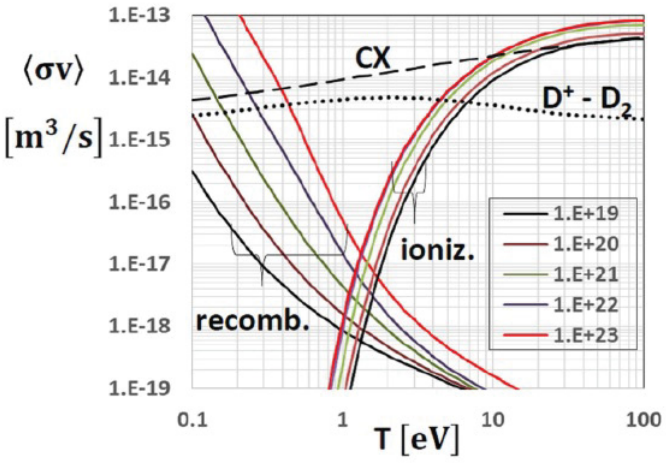


Figure 2.4: Reactivity of ionization, recombination, charge exchange and D^+-D_2 elastic collisions for deuterium as a function of plasma temperature; different colors indicate different electron densities as given in the table. Image from [25].

Figure 2.4 shows the reactivity of ionization, recombination, charge exchange and D^+-D_2 elastic collision, i.e. plasma-neutral friction, processes as a function of plasma

temperature, with $T = T_i = T_e$, for a deuterium plasma. It shows that mainly the rates of ionization and recombination strongly depend on the plasma temperature. As the plasma temperature decreases towards the target, the dominating process varies with the distance to the divertor target. At low plasma temperatures, i.e. close to the target, recombination processes might dominate whereas further away from the target line radiation and ionization will be the most significant processes. It is the increase of the reactivity of recombination processes in combination with charge exchange events that enables the divertor detachment. The observed drop in plasma pressure along the SOL in the regime of detachment can be brought about by charge exchange and recombination processes. As the reactivity of the former shows only a comparatively weak dependence on the plasma temperature, it is the latter that shows a significant change at very low target temperatures, which accompany the plasma detachment. In both processes ions from the plasma are converted to neutral particles and can thus freely leave the flux tube to dissipate power and reduce the plasma pressure which results in a detachment of the plasma from the divertor target.
To achieve a significant effect of the two processes, power losses are necessary in order to reduce the plasma temperature. Hence, the combination of power and momentum losses is needed to achieve plasma detachment. Thus, besides raising the density in the plasma core, another way of inducing detachment is the injection of impurity species, such as nitrogen, into the SOL plasma, see e.g. [28].

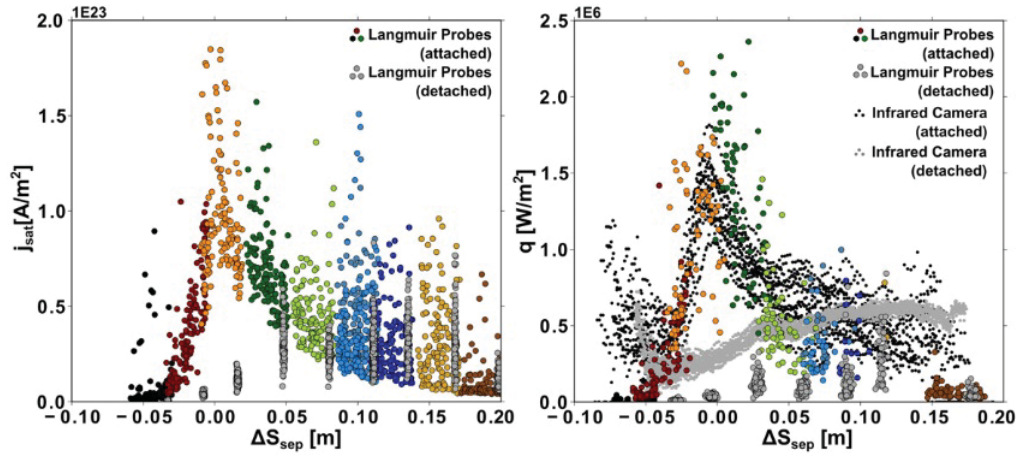


Figure 2.5: Comparison of ion flux density (left) and power flux density (right) profiles on the outer target of AUG in attached conditions (colored points) and with a completely detached plasma with nitrogen injection (gray points). Image from [28].

Figure 2.5 shows the target profiles of the ion flux density and the power flux density at the outer target of *ASDEX Upgrade* (AUG) in attached conditions (colored points) and in completely detached conditions with nitrogen injection (gray points) in an exemplary discharge. The abscissa gives the distance from the strikepoint, i.e. where the separatrix intercepts the target surface. Obviously the ion flux and the power flux are significantly

reduced under detached conditions in the region around the strikepoint. However, the fluxes at further distances do not show such a large reduction. This is caused by the effect that the plasma flux and, thus, the influx of neutralized ions from the divertor target is smaller further away from the strikepoint. This results in lower levels of plasma-neutral interaction and therefore in lower levels of power and momentum losses than the ones close to the strikepoint.

In summary plasma detachment can be characterized by three main categories, cf. [13]:

ENERGY DETACHMENT Increasing radiation causes significant energy losses and reduces the power flux towards as well as the plasma temperature close to the divertor target.

MOMENTUM DETACHMENT The reduction of ion fluxes towards the divertor target, which is interwoven with momentum losses in the SOL mainly caused by charge exchange reactions and elastic plasma-neutral collisions, causes a reduction of momentum transport towards the target.

PARTICLE DETACHMENT The increase of the rate of recombination reactions of the plasma ions at very low temperatures causes a reduction of the ion content close to the divertor target.

Sometimes detachment refers to the occurrence of only one or two of these criteria, for example, the occurrence of momentum detachment without energy detachment. However, in this work detachment refers to the combination of all three phenomena.

2.4 SHORTCOMINGS OF THE TWO POINT MODEL

The TPM, even in its extended form, is a simplified model for parallel-to-**B** transport of power and particles. The model does not take into account perpendicular transport caused by, e.g. external forces acting on the plasma particles which would result in a drift of the center of the particle's gyration with the velocity

$$v_{\text{drift}} = \frac{\mathbf{F}_\perp \times \mathbf{B}}{qB^2} \tag{2.23}$$

where $\mathbf{F}_\perp$ is the component of the external force perpendicular to the magnetic field. Moreover, the effect of perpendicular-to-**B** turbulent (anomalous) transport is neglected. Anomalous transport refers to transport processes that are not caused by collisions or the system's toroidicity, which also leads to additional transport via drifts. The effects of classical, i.e. collisional, transport and toroidicity-induced transport are described by the neoclassical transport theory. However, experiments show that transport coefficients can be orders of magnitude larger than the expected values from neoclassical theory [29]. The additional transport is summarised under the term anomalous transport.
Furthermore, the model assumes a tight coupling between the upstream density n_u,

which is actually the density at the separatrix, and the core density. Since transport and radiation processes in the plasma core can cause differences between the two, an additional model linking the densities is required.
In addition to this, the power fall-off length, λ_q, i.e. the radial extent of the power flux, is required to calculate $q_{||,SOL}$ from the actual control parameter that is the heating power applied to the plasma. Scalings for λ_q, even across tokamaks, have been found under attached conditions, e.g. [30] and [31], but not for partially detached scenarios.
Finally, as mentioned before, the loss factors in the extended TPM encode a range of complex highly non-linear atomic and molecular processes. Thus, in order to obtain predictions of power loads at the divertor targets prior to performing an experiment, several additional and not easily accessible paramters need to be known.

2.5 SIMULATING THE SOL - SOLPS-ITER

To incorporate the two dimensional transport of particles and power as well as volumetric processes neglected by the TPM and effects of divertor geometry, sophisticated simulation codes exist. As an example, the SOLPS-ITER code ([32], [33]), which is a commonly used code and is heavily used for design studies and decisions, will be discussed briefly to exemplify simulation approaches and potential issues.

The main constituents of the SOLPS-ITER code package are the B2.5 plasma transport solver [34] and the EIRENE Monte Carlo neutral transport code [35]. The B2.5 code solves the particle, momentum and energy fluid equations (see section 2.1) in two dimensions, i.e. parallel and perpendicular to the magnetic field, for each plasma species, i.e. main ions (D, T) and impurities (N, Ne, ...) as well as for the electrons via an implicit Euler method. Moreover, an equation for the plasma potential is solved (Ohm's law). From this, the electron temperature, T_e, the ion temperature, T_i, which is shared among all ion species in this code, the plasma potential V_{plasma}, the parallel velocity $v_{||,\alpha}$ and the density n_α are obtained for each ion species α. In addition to this the EIRENE code employs Monte-Carlo methods to generate solutions of the kinetic equations describing the evolution of the neutral particles. For the latter, the local plasma conditions are required to evaluate the cross sections of interaction processes encoded in the collision term on the right-hand side of equation 2.1. Furthermore, as shown by equations 2.5 to 2.7, the plasma conditions depend on the properties of the neutral gas. For example, neutral deuterium might be ionized and, thus, contribute to the plasma fluid equations via the source terms. Hence, a coupling of both codes is needed to self-consistently describe the system of plasma and neutral particles. With the combination of both codes the spatio-temporal evolution of the plasma and neutral particles can be computed iteratively for a given magnetic equilibrium, i.e. fixed flux surface geometries. However, the temporal resolution is limited by the minimum typical time scale of evolution of the aforementioned plasma quantities, so that the number of iterative calculations is given by the ratio of largest to smallest time scale. With a time scale of seconds for the confinement time in the plasma core and transport times on the order of ms in the SOL,

this results in ∼1000 iterative simulation steps.

As was the case in the TPM, SOLPS-ITER does also not include a parametrization for the turbulent anomalous transport. Instead, the SOLPS-ITER code models perpendicular transport with a diffusive ansatz with effective diffusion coefficient $D_{\text{effective}}$ so that in one dimension

$$\Gamma_{\perp} = -D_{\text{effective}} \frac{dn}{dx} \,. \tag{2.24}$$

A similar approach is used for the anomalous transport of heat with the transport coefficient $\chi_{\text{effective}}$:

$$q_{\perp} = -n\chi_{\text{effective}} \frac{dT}{dx} \,. \tag{2.25}$$

The values for these transport coefficients have to be taken from experiments and constitute additional parameters of the model.

Furthermore, inclusion of drifts of the center of gyration of the plasma particles, caused by additional forces, into the code used to cause problems with regard to numerical stability (for first results for ITER relevant parameters with inclusion of drifts see [36]) and significantly increased the time to convergence for simulations. Recent developments ameliorated this problem [37], but simulations of the full system of plasma and neutral particles are still very time consuming. $\sim 10^{-6}\,s$ to $10^{-8}\,s$ are common orders of magnitude for the time step of integration in the simulation code resulting in a total time of ~ 1 month for simulations of ITER with inclusion of drift effects, see e.g. [38]. A further critical point in the simulation approach is the difference between the two coupled codes. Where B2.5 calculates the solutions of the fluid equations implicitly and, thus, only yields results at the end of each time step of the simulation, EIRENE produces explicit solutions for the source and sink terms of the plasma fluid equations at the beginning of each time step. To consolidate these two schemes, additional calculations are needed [39] which also slow down the simulation.

Finally, to obtain fully verified results from a simulation run, the simulation must be matched to actual experiments by scanning the parameters of the simulation, e.g. transport parameters, until the results match observed profiles, see e.g. [40] and [41]. For this it is necessary to manually scan a highly dimensional space of parameters and to mimic the properties of actual diagnostics in order to translate the simulation results into data that is comparable with experimental observations.

Thus, due to intolerable convergence times and the lack of knowledge about parameters such as the transport parameters for anomalous transport, machine learning based approaches, utilizing experimental data, are an attractive option as they might yield adequate surrogate models with lesser computational demands and an implicit description of the (unknown) physics at hand.

3 MACHINE LEARNING METHODS

This chapter introduces the machine learning concepts and methods used to construct surrogate models for divertor power load predictions. The final aim of the application of machine learning techniques in this thesis is to ameliorate the challenges of analytically modeling the highly non-linear atomic and molecular processes involved from first principles and to circumvent the prohibitive time requirements of fully fledged simulation codes. However, in this chapter the general methods are introduced without the explicit application to fusion. This thesis is focused on approaches based on *Random Forest* (RF) regression, artificial *Neural Network*s (NNs) and *Gaussian Process Regression* (GPR). These methods were chosen as RFs constitute a simple machine learning approach and seemed to be a good candidate for a baseline model, but suffer from a lack of smoothness as they do not deliver analytical models. Hence, NNs were selected as a second approach to obtain smooth analytical models and GPR was chosen to test a fully Bayesian and interpretable approach. The first and last approaches have the property of being non-parametric, whereas the NNs are parametric. However, none of these methods require an analytical model based on first principles to begin with and physical effects, not included by simplified analytical models, could potentially be modeled implicitly. Thus, it could be possible to obtain working models for the prediction of expected power loads at the divertor targets without explicitly modeling the intricate particle interactions close to the divertor surface which usually hamper modeling approaches based on first principles.

3.1 RANDOM FOREST REGRESSION

This section gives an introduction to the method of *Random Forest* (RF) regression. Since a RF consists of several decision trees, the basic decision tree will be introduced first. Then the advantages and features of an ensemble of such decision trees will be discussed. The section is based on [42], [43] and the documentation on decision trees of scikit-learn [44].

3.1.1 *Decision Tree*

A decision tree is a tree model with split criteria based on a given feature at every node. The final nodes in the tree are called leaves and give the result of parsing the decision tree for a given data point.
Assuming a multidimensional input vector $\mathbf{x}$ of independent variables (features) and a corresponding target variable (label) y the tree makes a prediction $\hat{y}$ for $\mathbf{x}$. This prediction is obtained by traversing the tree according to whether $\mathbf{x}$ fulfills the split criterion at a given node or not.

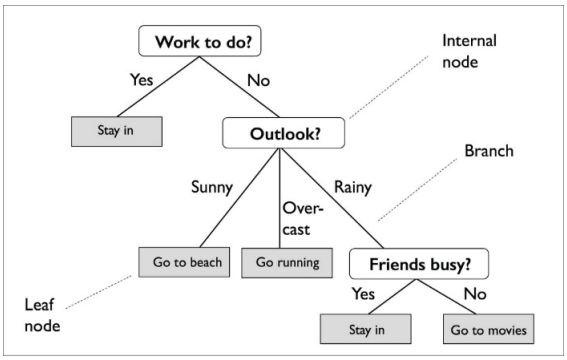


Figure 3.1: Schematic of a decision tree. The value of a feature at a given node determines the way the tree is traversed and leads to a final prediction. Image from [45].

Figure 3.1 shows an example of a decision tree for a classification task, i.e. the target variable y does not take continuous values but categorical values [45]. In the given example, the input vector $\mathbf{x}$ needs to consist of variables that contain information on whether or not there is work to do, on the weather forecast and on the current occupation of friends. Assuming that there is no work to be done, the outlook is rainy and friends are available, the model sugests to go to the movies.
In case of a regression task, i.e. continuous values of the target variable y and the features $\mathbf{x}$, the splits at the tree nodes are given by boundary values.

In the following a binary tree structure is assumed, i.e. every node that is not a leaf has two child nodes.

The optimization (also called training) of the model consists of selecting the ideal feature as well as the values for the split at each node in the tree. A standard criterion for this selection in a regression problem is the variance reduction [46]:

$$\begin{aligned} H(Q_m) &= \frac{1}{N_m} \sum_{y \in Q_m} (y - \langle y \rangle)^2 \\ L(Q_m, \theta) &= \frac{N_m^{left}}{N_m} H(Q_m^{left}(\theta)) + \frac{N_m^{right}}{N_m} H(Q_m^{right}(\theta)) \,. \end{aligned} \tag{3.1}$$

Here, Q_m is the set of data of N_m samples in the parent node whereas $Q_m^{left}(\theta)$ and $Q_m^{right}(\theta)$ refer to the set of data in the child nodes after applying the split, θ, consisting of selecting a feature and a threshold value. The child nodes then contain N_m^{left} and N_m^{right} data samples, respectively. $\langle y \rangle$ denotes the arithmetic mean calculated from the data remaining in the given set. Thus, $H(Q_m^{left}(\theta)) = \frac{1}{N_m^{left}} \sum_{y \in Q_m^{left}} (y - \langle y \rangle)^2$ with $\langle y \rangle = \frac{1}{N_m^{left}} \sum_{y \in Q_m^{left}} y$.
Rewriting $H(Q_m) = \frac{1}{N_m} \sum_{y \in Q_m} (y - \langle y \rangle)^2 = \frac{1}{N_m} \sum_{y \in Q_m} y^2 - 2y\langle y \rangle + \langle y \rangle^2 = \langle y^2 \rangle - \langle y \rangle^2$ shows that this indeed corresponds to the variance among the data in the given node. Thus, the intuition behind this optimization scheme is to group similar data into one

leaf node and describe this data with one value, the arithmetic mean.
The optimization process then minimizes $L(Q_m, \theta)$ with respect to θ at every node individually while traversing the tree. This process is iterated until an external criterion is met, for example a limiting value for the tree's depth might be reached.
Finally, the predicted value of the optimized model, $\hat{y}$, for a given input $\mathbf{x}$ is given by the arithmetic mean of all values that were assigned to the corresponding leaf node of $\mathbf{x}$ during the optimization process.

Figure 3.2 shows an example of applying a decision tree to a modeling problem. Here, noisy data was generated based on $y = x^2 + \epsilon$ where ϵ corresponds to Gaussian noise. Then two decision trees were trained, i.e. their splits were optimized, on the given data. The figure depicts the resulting predictions of both trees on the given data and intermediate data points based on the optimized splits. For the first tree, depicted by the blue line, no limiting criterion was set. This resulted in a tree with only one sample in every leaf node as this is the resulting limiting criterion for trees with no other limiting criteria. The resulting tree structure is shown in figure 3.3. The depth of the second tree was limited to two which also shows in the tree structure, see figure 3.4. Concerning the problem of fitting the data, the tree without limitation in depth describes the data points in figure 3.2 better than the tree with limited depth. However, since the data are noisy, this might result in *overfitting*, which is a general problem of large models. The more parameters a model has, in this case the more splits the decision tree can have, the more the model can adapt to noise in the observed data which might result in a deteriorating model performance when evaluating the model on unseen data. Hence, generally some form of regularization of the model complexity is used in most machine learning models.
Another property of decision trees that shows in the example is the step-like structure caused by the hard splits in the model. This effect can be mitigated by increasing the model complexity, i.e. the number of splits in the tree. Thus, finding the right complexity of a model is a seperate task on top of the actual optimization of the model parameters. Such additional parameters describing the structure of a model, like the maximum depth of a decision tree, are commonly called *hyperparameters*.
Furthermore, the example shows that the decision tree can only predict values for $\hat{y}$ based on the y values observed during the optimization. In figure 3.2 this is especially evident for the first four data points and the decision tree with a limited depth, which predicts the mean of the four values throughout the range of these values.

The foremost advantages of a decision tree based modeling approach are the non-linearity and the interpretability of the model. For example, the decision tree automatically selects important features to be at the top of the tree structure. Furthermore, the model does not rely on statistical assumptions about the data, such as the generating distribution or the distribution of residuals. On top of this, decision trees are resistant to the inclusion of input paramters that are uncorrelated with the target variable [43].
Disadvantages of these models are the aforementioned problem of overfitting, usually requiring some form of model regularization, the lack of smoothness of the fitting function and the inherently hampered ability of extrapolation. Further disadvantages

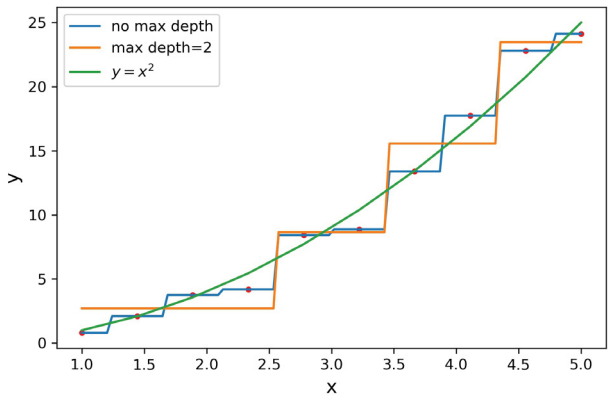


Figure 3.2: Example of fitting noisy data (red dots) with a decision tree without limiting its maximum depth (blue line) and with a maximum depth of 2 (orange line). The actual model is given in green.

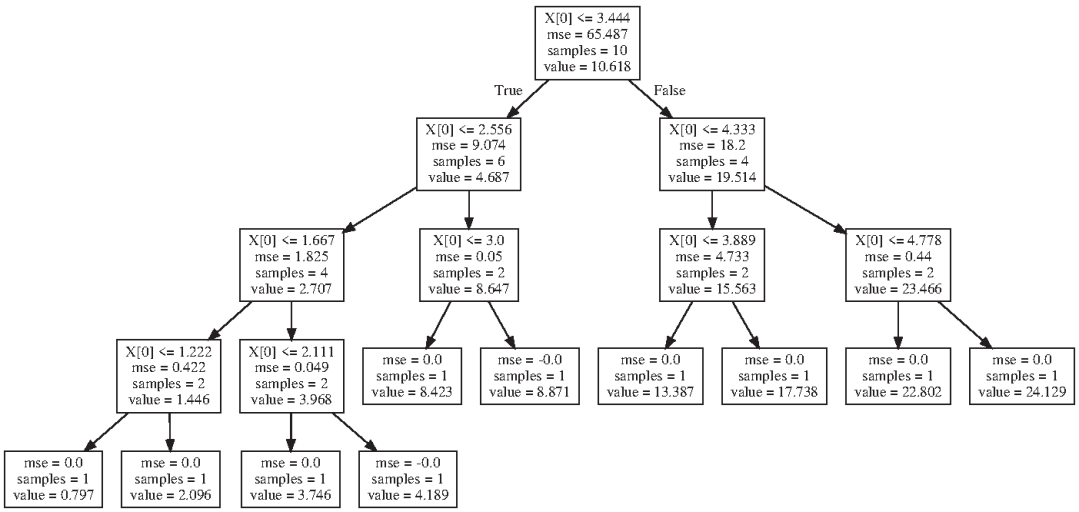


Figure 3.3: Decision tree without limitation of depth from figure 3.2. The value of the split, the mean squared error (H in equation 3.1), the number of samples and the assigned value are given for every node.

are that the usual optimization algorithms do not guarantee to find the global optimum of the fit and that decision trees can be unstable with regard to variation of the data, so that small variations of the data can lead to significantly different trees. The last two points can be addressed by using an ensemble of decision trees, a RF. For an extensive survey of decision tree properties and methods see, e.g. [47].

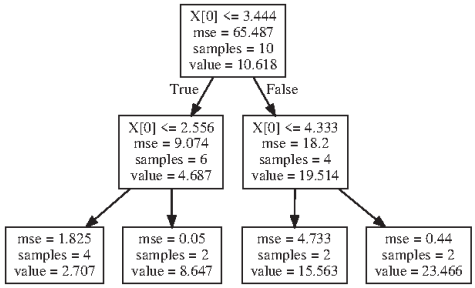


Figure 3.4: Decision tree with a maximum depth of 2 from figure 3.2. The value of the split, the mean squared error (H in equation 3.1), the number of samples and the assigned value are given for every node.

3.1.2 *Random Forest*

A *Random Forest* (RF) model consists of several decision trees. In contrast to the training procedure mentioned above, each tree is trained with a randomly chosen set of data from all the available data points, rather than using the full data set. The random data points are selected from the complete data set with replacement for every decision tree in the ensemble. This procedure is called *bootstrap aggregating* or *bagging*. It is used to reduce the variance[1] of the model and counteract overfitting, i.e. two of the disadvantages of decision trees mentioned above.
The overall model prediction for a given input $\mathbf{x}$ is given by the average of all predictions of the single decision trees.

Figures 3.5 and 3.6 show the same exemplary problem as before, fitting a model to noisy data generated from $y = x^2 + \epsilon$ with Gaussian noise ϵ, but this time with RF based modeling approaches. The RF models used for figure 3.5 consist of 10 decision trees and the ones used for figure 3.6 consist of 100 decision trees. In both figures, the results depicted in blue come from a RF with no limitation in tree depth, whereas the results shown in orange come from a RF with a maximum tree depth of two.
In comparison to figure 3.2 it can be seen that all models show more steps in their predictions than a single decision tree. This is due to the averaging of several trees optimized with randomly selected data points from the whole set. Furthermore, the RF without limitation in tree depth does not describe the given data points perfectly, in contrast to the single decision tree. This and the results observed for data points in between the given data, indicate the ameliorating effect of the ensemble approach on the overfitting problem. When comparing figures 3.5 and 3.6 it can be seen that a larger number of decision trees also smoothens the resulting function.

Another characteristic of RF models is the random selection of features available for a given split during the training. Given a training input $\mathbf{x}$ containing m variables,

1 For more details on model bias and variance see [48] and appendix A.

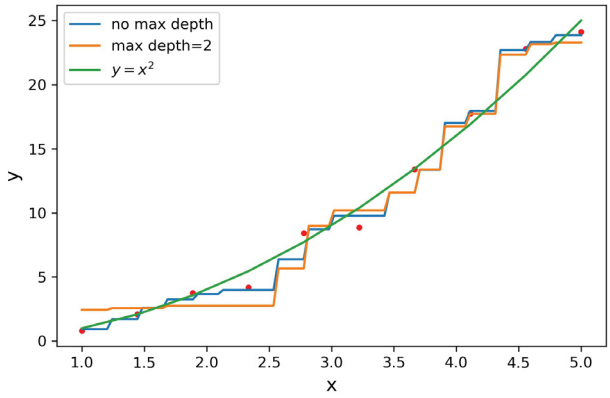


Figure 3.5: Example of fitting noisy data (red dots) with a random forest of 10 decision trees without limiting the trees' maximum depth (blue line) and with a maximum depth of 2 (orange line). The actual model is given in green.

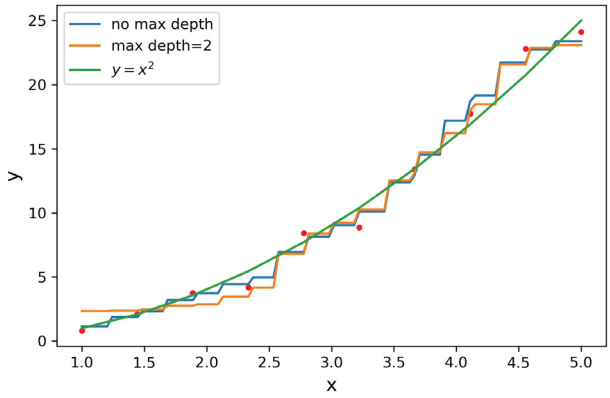


Figure 3.6: Example of fitting noisy data (red dots) with a random forest of 100 decision trees without limiting the trees' maximum depth (blue line) and with a maximum depth of 2 (orange line). The actual model is given in green.

only a subset of p $\leq$m of these variables is considered when determining the optimal parameters for each split. This results in an ensemble of weakly correlated decision trees and further reduces model variance. According to [43] the variance of an ensemble average calculated from B decision trees with pairwise correlation ρ is given by

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2 \tag{3.2}$$

where σ^2 is the variance of a single decision tree's prediction. Thus, the ensemble has a smaller variance than a single decision tree if the correlation among the trees is small. However, this random selection of features was not used for the exemplary case presented above, since the input x is only one-dimensional.

3.2 ARTIFICIAL NEURAL NETWORKS

This section introduces the fundamental principles of artificial *Neural Network*s (NNs) with a focus on fully connected NNs. Further extensions, such as *Mixture Density Network*s (MDNs) will also be discussed. The section is based on material from [48], [49] and [50].

3.2.1 *Fully Connected Neural Networks*

An artificial NN is a structured model consisting of single neurons. Each neuron constitutes a unit applying mathematical operations to the given input.

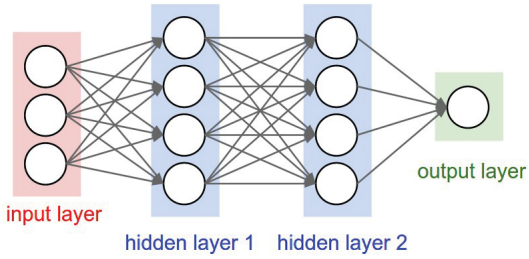


Figure 3.7: Schematic of a fully connected neural network consisting of four neurons in both hidden layers and one output neuron. Image from [51].

Figure 3.7 shows a schematic of a fully connected NN. The input variables of the model, x_i, lie in the input layer. From the input layer the input values are fed to every neuron in the next layer, which is the first hidden layer of the model. Each edge between neurons in the model corresponds to a parameter or weight of the model, w_{ij}. At every neuron a weighted sum of all inputs to that neuron is computed, so that for a given neuron j

$$a_j = \sum_i w_{ij} x_i + w_{0j} \tag{3.3}$$

where i indicates the input variable and w_{0j} is the bias[2] of the given neuron. The neurons also apply a non-linear *activation function* to this weighted sum. So the output of a neuron in the first hidden layer is given by

$$z_j = \sigma(a_j) = \sigma\left(\sum_i w_{ij} x_i + w_{0j}\right) \tag{3.4}$$

where σ denotes the nonlinearity. This process is repeated throughout the layers of the model until it reaches the output layer. So, for the second hidden layer x_i is replaced by z_j and new weights, corresponding to the edges between first and second hidden layer, are introduced into equation 3.3, resulting in $z_k = \sigma(a_k) = \sigma\left(\sum_j w_{jk} z_j + w_{0k}\right)$, with k

2 Not to be confused with model bias in the bias-variance decomposition.

indicating the given neuron in the second hidden layer.
The output layer in the example above consists of only one neuron as only one value is to be predicted by the model. In principle, however, the output layer could contain several neurons for several output values and the model could contain more or fewer than two hidden layers.

The activation functions applied in the layers of the NN do not have to be the same in every layer. For example, it is common to simply choose the identity as activation function in the output layer for regression tasks. However, in order to model functions beyond linear dependencies, non-linear functions have to be included in the hidden layers. Otherwise, the model would result in a stacking of linear functions and the overall function would be linear as well.
A crucial criterion for the selection of activation functions is that the function needs to be differentiable with respect to the weights of the NN as this is a key property for the optimization process of the model.
Some common activation functions are the *Exponential Linear Unit* (ELU) [52] and the *Rectified Linear Unit* (ReLU) [53]. The ELU is given by

$$\sigma_{\text{ELU}}(a) = \begin{cases} \alpha(e^a - 1) & a < 0 \\ a & a \geq 0 \end{cases} \tag{3.5}$$

with $\alpha > 0$, whereas the ReLU is the same for $a \geq 0$ but is simply cut to zero for $a < 0$. Here a denotes the weighted sum of inputs to any given neuron. Historically, functions with upper boundaries, such as tanh, were common choices for the activation function of a NN. However, such functions face the problem of a vanishing gradient for large and small values of the activation a, which can lead to a halt of the training process. This problem is exacerbated further by the nature of the optimization process (see below) which results in the multiplication of many small values of the gradients for these bounded activation functions due to the application of the chain rule.

As is the case with more conventional models, e.g. linear models or decision trees, the model parameters have to be optimized in order to describe the problem at hand. To this end a loss function needs to be defined that describes the mismatch of the model and the data used to optimize the model. A commonly used loss function for regression tasks is the *Mean Squared Error* (MSE) given by

$$\mathcal{L}_{\text{MSE}} = \frac{1}{\text{N}} \sum_{l=1}^{\text{N}} (\text{y}(\mathbf{x}_l, \mathbf{w}) - \text{y}_{\text{target},l})^2 \tag{3.6}$$

where N is the number of data points, $\text{y}(\mathbf{x}_l, \mathbf{w})$ is the neural network prediction for a given input $\mathbf{x}_l$ and a set of parameters, i.e. weights and biases, $\mathbf{w}$ and $\text{y}_{\text{target},l}$ is the corresponding observed value of the target quantity.
To optimize the weights of the NN, an iterative optimization process updates the weigths' values according to

$$\mathbf{w}' = \mathbf{w} - \eta \nabla_{\mathbf{w}} \mathcal{L}_{\text{MSE}} \tag{3.7}$$

where $\nabla_{\mathbf{w}}$ denotes the derivative with respect to the network's weights and η is the *learning rate*, which is an adjustable parameter and has to be set manually, as it constitutes a hyperparameter of this type of model. The derivatives of the loss function with respect to weights in early layers of the NN can be computed by repeatedly applying the chain rule in order to pass backwards through the network's layers (*backpropagation* [54]).
The fundamental iterative optimization approach then consists of splitting the training data, i.e. data used for the model optimization process, into random *mini batches* without replacement, calculating the model's loss for every data point in the mini batch and then updating the weights according to equation 3.7 using the mini batch averaged gradients of the loss function [55]. Performing the updates once with all mini batches constitutes a so called *epoch*. After each epoch the mini batches are chosen anew and the process repeats. This process leads to step-wise approaching the minimum of the loss function with the step size determined by the learning rate η. This approach is termed *Stochastic Gradient Descent* (SGD).
The introduction of mini batches has some advantages over the use of the full data set for every update. First, it is computationally less expensive to calculate the gradients on a mini batch basis. More importantly, however, is the fact that by introducing stochasticity into the optimization process, the optimizer is able to escape local minima which might cause an optimization based on the whole data set to get stuck [55].
The initial values of the network's weights are set at random, usually according to a modified uniform or normal distribution [56].

A typical extension to this optimization scheme is the application of an adaptive learning rate. With a fixed learning rate the optimization process might overshoot the minimum of the loss function if the learning rate is too large, resulting in a less than optimal model. If the learning rate is too small, the optimization process will take significantly longer. Figure 3.8 shows a schematic of this problem. Depicted is a loss function, $\mathcal{L}(\beta)$, of a model with one parameter, β, and the optimizer's path along that loss function. The red arrows indicate a potential trajectory with a fixed large learning rate, whereas the blue arrows show that of an algorithm with an adaptive learning rate. Obviously, the fixed step size of the algorithm depicted in red causes the optimizer to miss the minimum of the loss function and hinders its convergence. On the other hand, an adaptive learning rate that reduces the step size close to the minimum can achieve better and faster convergence.

As artificial NNs tend to have a lot of free parameters (tens of thousands or even millions of weights are not uncommon), it is usually necessary to regularize the model complexity. In the case of the RF this could be achieved e.g. via limitation of the tree depth of the model. For NNs the most common approach is to introduce an additional regularization term into the loss function. Usually L1 and L2 regularization are used for this purpose. The corresponding additional terms are

$$\mathcal{L}_1 = \lambda \sum |\mathrm{w}| \tag{3.8}$$

and

$$\mathcal{L}_2 = \lambda \sum \mathrm{w}^2 \tag{3.9}$$

where the sum is over all weights of the model and λ is a constant that determines the tradeoff between the actual loss and the additional regularization term. Especially the L1 regularization term tends to shrink some weights towards zero and could, thus, help eliminate meaningless input quantities from the model. Other means of regularization also include (temporary) changes in the structure of the NN, e.g. *dropout* [57] which will be discussed later as well as *early stopping* of the training process.

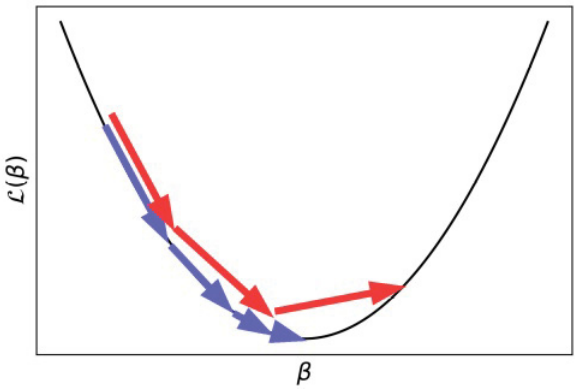
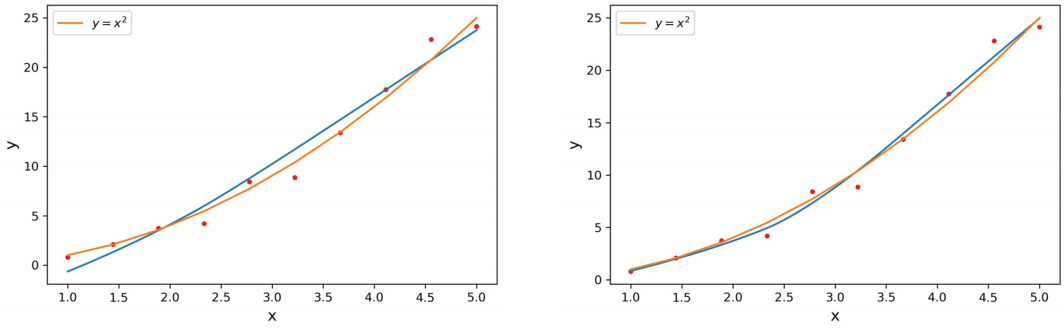


Figure 3.8: Schematic example of an optimization process with fixed learning rate (red) and adaptive learning rate (blue) for a model with one parameter, β. $\mathcal{L}(\beta)$ denotes the loss function.



(a) Result of fitting a neural network to noisy data (red) after 500 epochs of training. The true model is shown in orange.

(b) Result of fitting a neural network to noisy data (red) after 1000 epochs of training. The true model is shown in orange.

Figure 3.9: Exemplary results of fitting a neural network to data with results after different epochs.

Figures 3.9(a) and 3.9(b) show predictions of a NN after fitting it to some exemplary data (red dots in the figures). The same data as for the example of the RF regression was used. The data comes from $y=x^2+\epsilon$, where ϵ denotes Gaussian noise. The results shown are predictions of the model on data points in the range $x \in [1, 5]$ after training

the network for 500 and 1000 epochs, respectively. The network predictions are shown in blue and the actual underlying model is depicted in orange. It can be seen that the fit quality improves from epoch 500 to 1000 and that the network describes the true model rather well. In contrast to the decision tree regression and the RF regression, the function described by the NN is inherently smooth. The NN used for this example consisted of two hidden layers with 20 and 30 neurons. An ELU activation function was chosen for the neurons in the hidden layers. The output layer only consisted of one neuron and applied no (i.e. a linear) activation function. The MSE was chosen as loss function and an L2 regularization term was included. The structure of the network is depicted in figure 3.10. Here, the activation functions are denoted as separate layers and each layer's shape is given, where a questionmark denotes the size of the mini batch used for the training process. This size was set to 3 in this very small example.
Figure 3.11 shows the evolution of the loss during the training of the NN. It can be seen that, after significant improvements during the first $\sim$ 200 epochs, the reduction of the loss per epoch decreases. The depicted loss was calculated based on the training data. Usually, it is necessary, to split the data into at least two sets, one for training the model and one for testing its final performance on unseen data. On top of this, a third data set can be used for validation, i.e. monitoring of the model performance on unseen data during training. However, for this very basic example, there are only the training set (red dots in figures 3.9(a) and 3.9(b)) and the test set, which includes data points in between the training data.

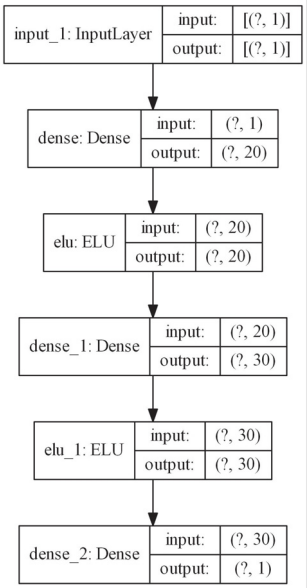


Figure 3.10: Structure of the neural network used for the example task above. The ELU activations are depicted as separate layers and layer shapes are given, where a questionmark indicates the mini batch size during the training, which was set to 3 here. Dense layers are fully connected layers as described above.

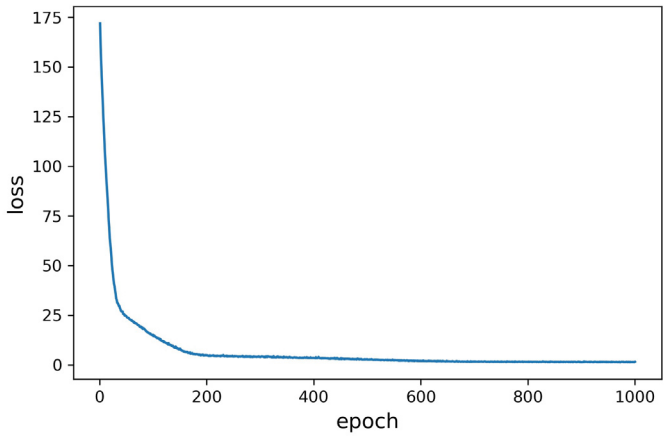
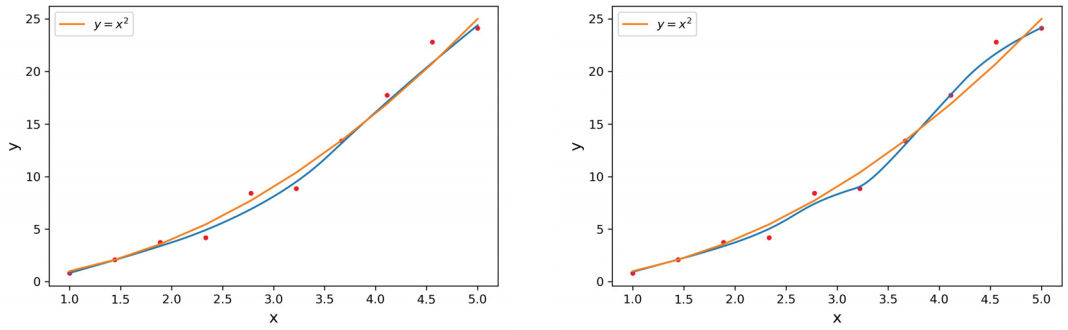


Figure 3.11: Evolution of the loss of the neural network during training calculated on the training data for the exemplary problem shown above.

The training data set in this example is, of course, very small for a NN based approach. As NNs tend to have a very large number of model parameters (the exemplary network above has about $20 + 20 \cdot 30 + 30 = 650$ parameters, which is comparatively small), the need for large amounts of data is commonly crucial for this type of model. Otherwise, the model is prone to encounter the aforementioned problem of overfitting. For the exemplary problem this is visualized in figures 3.12(a) and 3.12(b). The first image shows the results for the network predictions after training for 10^4 epochs with an additional L2 regularization term in the loss function. The second image shows the corresponding results without the regularization term. It is obvious that the model overadapts to the training data when the regularization term is dropped. Hence, regularization and/or very large data sets tend to be essential to mitigate the overfitting problem.

Another way of regularizing the model complexity is given by *dropout* [57]. In this approach, neurons are randomly removed from the training process to temporarily reduce the model complexity and to prevent co-adaptation of the neurons. Which neurons are removed from the model is reevaluated after every training step. Given a fixed probability p of neurons to be dropped from the training, the weights of the model have to be scaled by $\frac{1}{1-p}$ in order to obtain the same expectation values of the activations. During training, the selected neurons and their corresponding connections are removed from the model, so that they do not contribute to the weight update process. For the evaluation, the model is then used without dropout and with scaled weights. This can be seen as subsampling the more complex full model with an ensemble of smaller models of reduced complexity. Figure 3.13 shows a schematic of the dropout principle.



(a) Result of fitting a neural network to noisy data (red) after 10000 epochs of training with L2 regularization. The true model is shown in orange.

(b) Result of fitting a neural network to noisy data (red) after 10000 epochs of training without L2 regularization. The true model is shown in orange.

Figure 3.12: Exemplary results of fitting a neural network with and without regularization terms to noisy data.

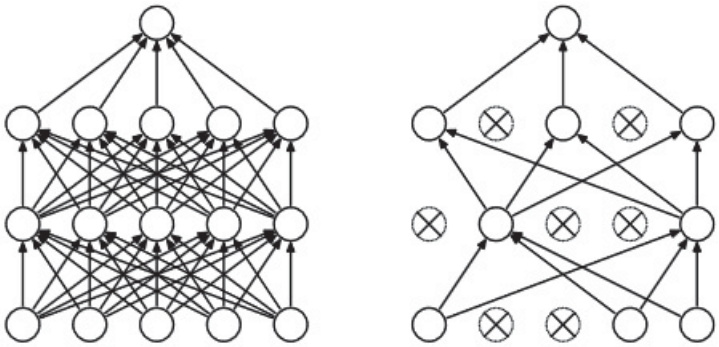


Figure 3.13: Schematic of the dropout technique with the full neural network on the left and a reduced model with dropped neurons on the right. Image from [57].

3.2.2 *Mixture Density Networks*

As discussed by Bishop in [48] and [50], a common potential shortcoming of data-based modeling is that the optimal solution found is given solely by the conditional average of the distribution $p(y|\mathbf{x})$ with observed values of the target quantity y and values of the independent variables $\mathbf{x}$ [3]. However, this conditional average might not be an

3 This holds for conventional sum-of-squares optimization schemes, see chapter 1.5.5 of [48], which will also be used for the regression tasks later in this thesis. It might be intuitive to consider a data set $\mathcal{D} = \{\mathbf{x}_i, y_i\}$. The sum-of-squares approach seeks to minimize $\mathcal{L} = \sum_i (y_i - f(\mathbf{x}_i))^2$ where $f(\mathbf{x})$ is a function that is supposed to model the data. Splitting the sum over all data points into a sum over all unique $\mathbf{x}$ and a sum over all y corresponding to any given $\mathbf{x}$ gives $\mathcal{L} = \sum_{\text{unique } \mathbf{x}} \sum_{y(\mathbf{x})} (y - f(\mathbf{x}))^2$. The term in the inner sum describes the variance of the observed y at a given $\mathbf{x}$ around the model's value. Since the expectation value of a distribution minimizes the variance, the best estimate the model can yield is the average of the distribution of y conditioned on the given $\mathbf{x}$.

adequate prediction, since the average could be an unphysical solution. To circumvent this problem, it would be beneficial to obtain a predictive distribution for every input to the model. In this way a potential multimodality of the target quantity could be modeled. A possible solution to this problem, introduced in [50], is the application of *Mixture Density Networks* (MDNs).

The inclusion of an MDN based approach in this thesis is motivated by the fact that the input quantities to the models will not include every parameter relevant to the operation of a fusion reactor. Hence, for a given input vector, $\mathbf{x}$, the output quantity might still vary due to dependencies not covered by the selected quantities. This could induce multimodality in the data.

The main difference between MDNs and the aforementioned NNs lies in the output layer. Where the NNs have one output neuron per predicted quantity, MDNs have several output neurons per predicted quantity. For simplicity, and since this will be the relevant case in the main part of this thesis, assume only one quantity that is to be predicted. This would result in one output neuron for a simple fully connected NN as discussed above. However, a MDN predicts the expectation values, μ_i, standard deviations, σ_i, and normalizing constants, α_i, for a mixture of density functions, usually Gaussians, where i indicates the mixture component. Hence, the function predicted by an MDN will be a probability density of the form

$$p(y|\mathbf{x},\mathbf{w}) = \sum_i \alpha_i(\mathbf{x},\mathbf{w})\phi_i(y|\mathbf{x},\mathbf{w}) \tag{3.10}$$

with

$$\phi_i(y|\mathbf{x},\mathbf{w}) \propto \exp\left(\frac{(y-\mu_i(\mathbf{x},\mathbf{w}))^2}{2\sigma_i^2(\mathbf{x},\mathbf{w})}\right). \tag{3.11}$$

Here $\mathbf{w}$ denotes the network's weights that determine the predicted parameters α, μ and σ, $\phi_i(y|\mathbf{x},\mathbf{w})$ denotes the components of the mixture model and exp is the exponential function. Note that $\sum_i \alpha_i = 1$ and $0 \leq \alpha_i \leq 1$ have to be fulfilled. The conditional notation indicates that the distribution of values y of the quantity to be predicted has to be conditioned on the values of the input vector $\mathbf{x}$ and on the network weights $\mathbf{w}$. This is to say, that the approach approximates a probability distribution of the target quantity for every given input vector as a sum of Gaussians, where the expectation value, the standard deviation and the contribution of each Gaussian component are a function of the input vector and the network's weights. Thus, if the model consists of a number of m Gaussian components, the MDN would have 3m output neurons as α, μ and σ have to be predicted for every component of the mixture model.

It is possible to determine the predicted value of the target quantity from this probability density, e.g. by using the expectation value of the mixture component with the strongest contribution α_i. This assures that the predicted value does not lie in a physically unreasonable region. Furthermore, this approach allows to determine the variance of the predicted probability density, which can serve as a measure for the model uncertainty.

The optimization of this type of model also differs from that of the aforementioned NN, as a different loss function has to be employed. An adequate loss function for this approach is given by the negative logarithm of the likelihood described by

$$\mathcal{L}(\mathbf{w}) = -\sum_{n=1}^{N} \ln\left(\sum_{i=1}^{m} \alpha_i(\mathbf{x}_n, \mathbf{w})\mathcal{N}(y_n|\mu_i(\mathbf{x}_n, \mathbf{w}), \sigma_i^2(\mathbf{x}_n, \mathbf{w}))\right) \tag{3.12}$$

under the assumption of independent data points. Here N is the total number of data points used for the optimization, m is the number of mixture components and $\mathcal{N}$ denotes a Gaussian distribution with expectation value μ and variance σ^2. Intuitively, this loss forces the probability distribution predicted by the MDN to follow the distribution of the observed data points by adjusting the weights of the network. The rest of the optimization procedure remains the same as for the fully connected NN from above.

Different activation functions might also have to be used for the MDN than for the NN. The activation functions of the hidden layers can still be tanh or others of the activation functions mentioned in 3.2. However, the activation functions in the output layer need to be changed. For the output neurons predicting the expectation values μ_i, a linear activation function is still applicable, but in order to assure that the conditions for the α_i are met, it is advisable to use a softmax activation function for the output neurons predicting these paramters. The result for a single α_i of the softmax activation function is then given by

$$\alpha_i = \frac{\exp(a_i)}{\sum_i \exp(a_i)} \tag{3.13}$$

where a_i denotes the weighted sum of the inputs to one output neuron calculating one α_i and exp is the exponential function. This assures that $\sum_i \alpha_i = 1$ and $0 \leq \alpha_i \leq 1$. For the output neurons calculating the standard deviations σ_i, it is beneficial to use an activation function of the form

$$\sigma_i = \exp(a_i)\,. \tag{3.14}$$

Here a_i indicates the weighted sum of inputs to a single output neuron calculating one σ_i. This assures that the calculated standard deviations will be non-negative.

3.3 GAUSSIAN PROCESS REGRESSION

This section introduces *Gaussian Process Regression* (GPR) and its principle mechanics. The contents of this section summarize some of the very thorough explanations given by Rasmussen and Williams in [58]. The definitions of the covariance functions are taken from the documentation on GPR of scikit-learn[44]. In contrast to RFs and NNs, GPR offers a fully probabilistic approach and yields the complete predictive distribution, more comparable to MDN.

According to definition 2.1 of [58], a *Gaussian Process* (GP) is defined as follows:
A Gaussian process is a collection of random variables, any finite number of which have a joint

Gaussian distribution.

In the GPR approach, the distribution of all possible function values $f(\mathbf{x})$ at all inputs $\mathbf{x}$ is this collection of random variables. An illuminating example from [58] is the case of Bayesian regression of a linear model. Assuming a model $f(\mathbf{x}) = \mathbf{w} \cdot \phi(\mathbf{x})$, with some non-linear transformation ϕ [4] and model parameters $\mathbf{w}$, and a Gaussian prior over the model parameters $p(\mathbf{w}) = \mathcal{N}(0, \Sigma_p)$, where Σ_p denotes the covariance matrix of this prior distribution[5], it can be seen that the expectation value over the unknown parameters $\mathbf{w}$ at any given input $\mathbf{x}$ is $\mathbb{E}[f(\mathbf{x})] = \phi(\mathbf{x})\mathbb{E}[\mathbf{w}] = 0$. Furthermore, for the covariance of any two function values at two different input values $\mathbf{x}$ and $\mathbf{x}'$ it can be found that $\mathbb{E}[f(\mathbf{x})f(\mathbf{x}')] = \phi(\mathbf{x})\Sigma_p\phi(\mathbf{x}')$. Thus, Rasmussen and Williams conclude that under these assumptions any pair of function values has a joint Gaussian distribution, which indicates that the realization of function values is given by a GP. Hence, the GPR approach can be seen as based on distributions over functions where the realizations of function values come from a GP with some mean function, $m(\mathbf{x})$, and a covariance function, $k(\mathbf{x}, \mathbf{x}')$, with

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \,. \end{aligned} \tag{3.15}$$

In other words, every potential function $f(\mathbf{x})$ and, thus, every set of parameters corresponds to a certain realization of this GP. The main goal of GPR is then to condition the properties of this GP on the observed data and to obtain a predictive distribution for new data points $\mathbf{x}_*$.

As the GP governing the realizations of the functions $f(\mathbf{x})$ is completely described by the mean function (which will be set to zero in the following[6]) and the covariance function (also called kernel), it is necessary to determine such a covariance function. A common choice is the *Radial Basis Function* (RBF) (also called squared exponential) given by

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2l^2}|\mathbf{x} - \mathbf{x}'|^2\right) \tag{3.16}$$

with the exponential function exp and the length-scale parameter l. Other common choices include the rational quadratic kernel given by

$$k(\mathbf{x}, \mathbf{x}') = \left(1 + \frac{|\mathbf{x} - \mathbf{x}'|^2}{2\alpha l^2}\right)^{-\alpha} \tag{3.17}$$

4 Even though the model inputs are transformed non-linearly, the model is still linear with respect to the model parameters. Thus, this is still an example of linear regression.

5 This Bayesian interpretation with a Gaussian prior distribution over the model parameters is comparable to the aforementioned L2 regularization scheme. For a very brief reminder on Bayesian regression see appendix B.

6 Note that this does not mean that the average of any given funtion needs to be zero but rather that for any given input $\mathbf{x}$ the average of infinitely many realizations of $f(\mathbf{x})$ drawn from the GP will amount to zero.

with the length-scale parameter l and the mixture parameter α and the Matérn kernel defined as

$$k(\mathbf{x}, \mathbf{x}') = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{\sqrt{2\nu}}{l} |\mathbf{x} - \mathbf{x}'| \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu}}{l} |\mathbf{x} - \mathbf{x}'| \right) \tag{3.18}$$

with the length-scale paramter l, a modified Bessel function K_{ν}, the gamma function Γ and the adjustable parameter ν which controls the smoothness of the resulting functions[7]. Some of the parameters in these covariance functions, such as the length-scale parameter, are inferred from the data, whereas others, such as ν, need to be set manually.

With a given covariance function the prior distribution of the GP is set and for a matrix $\mathbf{X}_*$, composed of input vectors $\mathbf{x}_*$, for which predictions are to be obtained, it is $\mathbf{f}_* \sim \mathcal{N}(0, \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*))$, with the vector of predicted values $\mathbf{f}_*$ and the matrix of covariances of the new input vectors $\mathbf{K}(\mathbf{X}_*, \mathbf{X}_*)$. In principle, realizations of the function f could already be drawn from this prior distribution and certain properties, such as the smoothness, would be determined by the chosen covariance function. However, these realizations would not be conditioned on the observed data. Conditioning the resulting distribution of functions on the data in the given data set results in the predictive distribution (also called posterior distribution)

$$\mathbf{f}_* | \mathbf{X}, \mathbf{y}, \mathbf{X}_* \sim \mathcal{N}(\bar{\mathbf{f}}_*, \mathrm{cov}(\mathbf{f}_*)) \tag{3.19}$$

where the notation from [58] was used, so that $\mathbf{f}_*$ is a vector of values to be predicted for unseen inputs $\mathbf{X}_*$, $\mathbf{X}$ and $\mathbf{X}_*$ denote the matrices constructed of all input vectors of the training data used to optimize the model and the new data points, respectively, and $\mathbf{y}$ is the vector of all noisy values of the quantity to be predicted corresponding to the training inputs. Furthermore, it is

$$\begin{aligned} \bar{\mathbf{f}}_* &= \mathbb{E}[\mathbf{f}_* | \mathbf{X}, \mathbf{y}, \mathbf{X}_*] = \mathbf{K}(\mathbf{X}_*, \mathbf{X}) \left(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_{\mathrm{n}}^2 \mathbb{I} \right)^{-1} \mathbf{y} \\ \mathrm{cov}(\mathbf{f}_*) &= \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*, \mathbf{X}) \left(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_{\mathrm{n}}^2 \mathbb{I} \right)^{-1} \mathbf{K}(\mathbf{X}, \mathbf{X}_*) \,. \end{aligned} \tag{3.20}$$

Here $\mathbf{K}(\cdot, \cdot)$ is the matrix of all covariances calculated from the data, given as arguments of the function $k(\cdot, \cdot)$, with the chosen covariance function. So $\mathbf{K}(\mathbf{X}, \mathbf{X})$ is the symmetric matrix obtained from calculating $k(\mathbf{x}, \mathbf{x})$ for all $\mathbf{x}$ in the training data, whereas $\mathbf{K}(\mathbf{X}_*, \mathbf{X}_*)$ is the covariance matrix calculated from all input vectors in the set of new data for which the predictions are to be obtained. The σ_{n}^2 was added to account for additive Gaussian noise in the data. σ_{n} constitutes another parameter of the kernel that needs to be inferred from the data. $\mathbb{I}$ denotes the identity matrix.
Hence, the expectation value of this distribution of function values, $\bar{\mathbf{f}}_*$, gives a prediction for new data points and the covariance of the distribution $\mathrm{cov}(\mathbf{f}_*)$ immediately provides a measure of confidence for these predictions. It might be instructive

7 To keep the discussion of kernels and their properties rather short, it shall only be noted that the RBF results in functions that are infinitely differentiable, whereas the order up to which functions resulting from a GP with Matérn kernel are differentiable is determined by ν. For more details see chapter 4 of [58].

to consider the example of $\mathbf{X}_* = \mathbf{X}$, so the case in which one wants to obtain predictions for the training data on which the distribution has been conditioned. For $\mathbf{X}_* = \mathbf{X}$ it is $\bar{\mathbf{f}}_* = \mathbf{K}(\mathbf{X},\mathbf{X})\left(\mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_n^2\mathbb{I}\right)^{-1}\mathbf{y}$. Neglecting the noise term, this results in $\bar{\mathbf{f}}_* = \mathbf{K}(\mathbf{X},\mathbf{X})\left(\mathbf{K}(\mathbf{X},\mathbf{X})\right)^{-1}\mathbf{y} = \mathbf{y}$ and $\mathrm{cov}(\mathbf{f}_*)$ becomes zero. Thus, under the assumption that the model does not consider noise in the data, the predictions obtained from the predictive distribution will perfectly pass through data points on which the distribution has been conditioned, independent of the actual covariance function and its parameters. Reintroducing the noise term would lead to potential deviations from the given training data points caused by the model's additional flexibility due to σ_n.

The optimization of the kernel parameters, such as the length-scale parameter, can be done via maximization of the log marginal[8] likelihood given by

$$\ln(p(\mathbf{y}|\mathbf{X},\theta)) = -\frac{1}{2}\mathbf{y}^T\mathbf{K}_y^{-1}\mathbf{y} - \frac{1}{2}\ln(|\mathbf{K}_y|) - \frac{n}{2}\ln(2\pi) \tag{3.21}$$

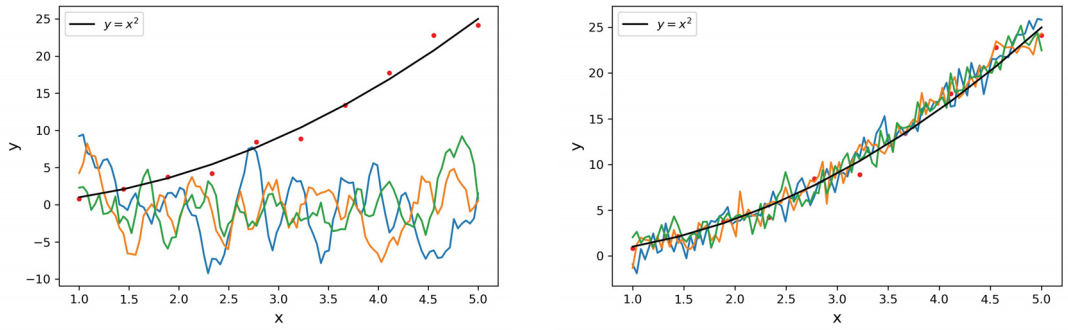
where ln is the natural logarithm, $\mathbf{K}_y$ denotes the covariance matrix (including the noise term) of the target values $\mathbf{y}$, θ denotes the set of the kernel's parameters (also including the noise term σ_n), $|\mathbf{K}_y|$ is the determinant of the matrix $\mathbf{K}_y$ and n is the number of data points used in the optimization process. By maximizing this term with respect to θ, e.g. via gradient ascent similar to SGD, the optimal parameters of the covariance function can be found for a given set of datapoints. [9]

Figures 3.14(a) and 3.14(b) show an example of realizations of functions drawn from the prior and posterior distributions, respectively, of a GP. While the functions drawn from the prior have nothing to do with the actual data given by the red points (the same data that was used for the examples of RF and NN), as was to be expected, it can be seen that the realizations drawn from the posterior lie very close to the actual function underlying the red data points. The kernel used in this example consisted of a RBF covariance function with a multiplicative prefactor, C, and an additional term for the noise. The initial values of the parameters used in the prior distribution are $C = 15$, $l = 0.1$ and $\sigma_n^2 = 0.5$ with the length-scale parameter l of the RBF kernel. The parameters of the posterior distribution after optimization are $C = 357.21$, $l = 3.2$ and $\sigma_n^2 = 1.15$.

Figure 3.15 shows $\bar{\mathbf{f}}_*$ obtained from the posterior distribution conditioned on the red data points. The blue line corresponds to the realization of $\bar{\mathbf{f}}_*$, the black line shows the function from which the noisy data was generated and the gray area indicates one standard deviation around $\bar{\mathbf{f}}_*$ obtained from $\mathrm{cov}(\mathbf{f}_*)$. In contrast to the realizations of

8 This is a marginal likelihood as it is marginalized over all realizations of functions $\mathbf{f}$, i.e. $p(\mathbf{y}|\mathbf{X},\theta) = \int p(\mathbf{y}|\mathbf{f},\mathbf{X},\theta)p(\mathbf{f}|\mathbf{X},\theta)d\mathbf{f}$.

9 It might be helpful to consider this as an optimization of hyperparameters of the model, like Rasmussen and Williams do in chapter 5 of [58]. The optimization of the GP's 'parameters', which would correspond to the training of a NN, is done via the conditioning of the predictive distribution on the training data. This conditioning limits the possible realizations of function values that can be drawn from the distribution at the training data points, these function values could be interpreted as 'parameters' of this modeling approach.



(a) Three realizations of the function $f(x)$ drawn from the prior distribution of a Gaussian process. Obviously the function realizations have nothing to do with the data indicated by the red dots as the distribution has not been conditioned on the data.

(b) Three realizations of the function $f(x)$ drawn from the posterior distribution of a Gaussian process. The example realizations describe the real function (indicated by the black line) underlying the noisy red data points rather well even if they are quite noisy still.

Figure 3.14: Exemplary realizations of f(**x**) from the prior and posterior distributions of a GP. The posterior is conditioned on the noisy red data points coming from the model indicated by the black line.

$\mathbf{f}(x)$ shown in figure 3.14(b) the expectation value of the posterior, $\bar{\mathbf{f}}_*$, is less noisy, which could be expected, and describes the actual function even better.

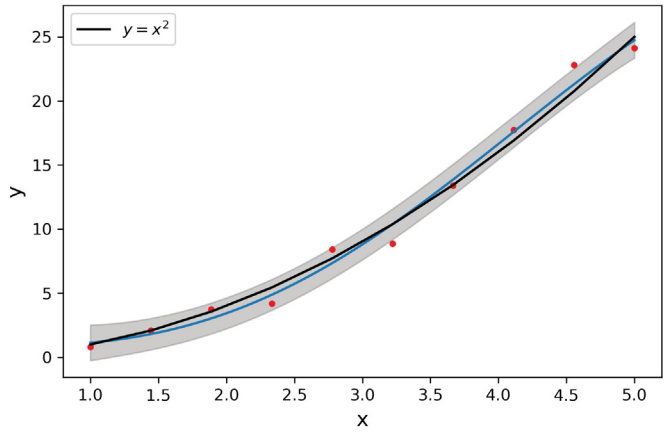


Figure 3.15: Prediction $\bar{\mathbf{f}}_*$ of a GP conditioned on the noisy red data points. The black line indicates the true function from which the red data was generated. The gray area indicates $\bar{\mathbf{f}}_* \pm \sigma$ where σ is the point-wise standard deviation obtained from $cov(\mathbf{f}_*)$.

4 FROM EXPERIMENT TO DATA

This chapter presents the *ASDEX Upgrade* (AUG) tokamak experiment which provided the data analyzed in this thesis and introduces the quantities and data selection of this analysis. The selection of parameters, the compilation and preprocessing of the data as well as first analyses of the data extracted from the experiment constitute original contributions. The contents of sections 4.2, 4.3 and 4.4 have been published in [59] and the corresponding sections reiterate the contents of this publication with the addition of information on the data selection for the *Gaussian Process Regression* (GPR) based approach.

4.1 THE ASDEX UPGRADE EXPERIMENT

The analysis presented in this thesis is based on experimental data from the AUG tokamak located at the Max Planck Institute for Plasma Physics in Garching, Germany. Some of the tokamak's general technical data as well as typical parameters of the plasma are given in table 4.1.

Technical data	
Maximum toroidal magnetic field	3.9 T
Plasma current	0.4 MA-1.6 MA
Pulse duration	< 10 s
Plasma heating power	up to 27 MW
Typical plasma parameters	
Major plasma radius	1.65 m
Minor horizontal plasma radius	0.5 m
Minor vertical plasma radius	0.8 m
Triangularity (top/bottom)	0.1/0.3
Electron density	10^{20} m^{-3}

Table 4.1: List of technical parameters of the AUG experiment and some typical plasma parameters.

The AUG experiment is an important steppingstone on the way to realizing large scale fusion reactors such as ITER. AUG was specifically designed to probe the parameter space relevant for the operation of future fusion devices. For example, AUG incorporates tungsten (W) as material of the plasma vessel walls and the divertor since W is also envisioned to be the primary material for the divertor plasma facing components of

ITER [8].

Figure 4.1 shows a poloidal cross section of an exemplary plasma discharge in AUG. The nested surfaces of constant magnetic flux are indicated and the separatrix is marked with a broad line. Furthermore, the inner and outer divertor targets are marked in green and red, respectively. The coordinate along the major radius of the torus is denoted on the abscissa whereas the position with regard to the center of the poloidal plane is shown on the ordinate.

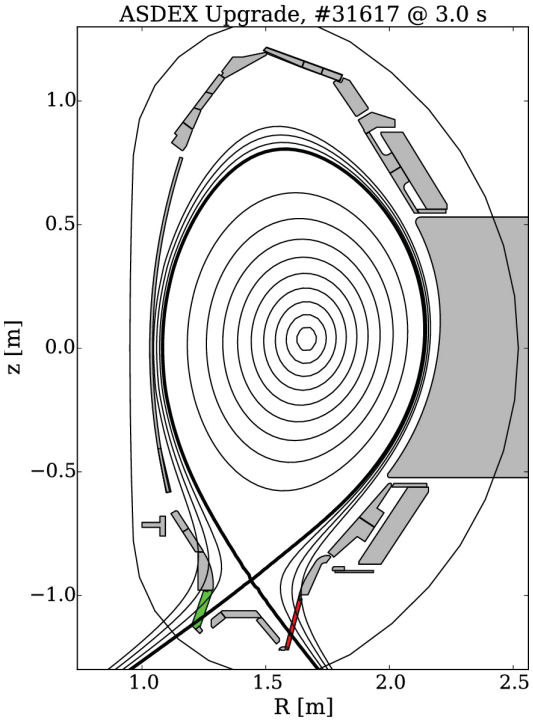


Figure 4.1: Poloidal cross section of ASDEX Upgrade during an exemplary plasma discharge. The inner and outer divertor targets are marked in green and red, respectively. Image from [31].

4.2 PARAMETERS OF THE ANALYSIS

The main parameter of this analysis is an AUG specific[1] quantity, T_{div} [60]. The quantity that is actually measured is the thermo-electric current into a tile of the divertor measured via a shunt. It was found that this current could be scaled to give a good real-time estimate of the electron temperature (measured in eV) in the outer divertor [60]. However, since this current is determined by the difference of the temperatures of inner and outer divertor [61], obtaining an estimate for the electron temperature of the outer divertor would strictly require the inner divertor to be in a detached state or at least

1 The quantity could also be measured for other tokamaks, but this measurement is not always implemented.

at very low plasma temperatures. This is usually the case for experiments at AUG, at least in high power and high denstiy scenarios relevant for future reactors. However, this analysis aims at spanning a wide range of parameters, therefore, this criterion might not always be fulfilled. Nevertheless, this quantity is used as a proxy variable for the power loads at the outer divertor target and constitutes the target variable for the regression analysis in this thesis[2]. Thus, the aim of this analysis is to obtain models for predictions of T_{div} from other, more general, plasma parameters which will be introduced in this section. Note that T_{div} might reach negative values as it is only a proxy for and not actually a measurement of the electron temperature. Values close to and below 0 eV can be indicative of a detached outer divertor as the power flux (and the current) towards the divertor target drops. Small values of $T_{div} \lesssim 10\,\text{eV}$ are indicative of partial detachment of the divertor [62]. As T_{div} is an *Edge Localized Mode* (ELM)-filtered quantity, it is an appropriate parameter to quantify steady state thermal loads at the outer divertor target. More importantly, T_{div} is a standardly available signal in experiments on AUG allowing for the construction of a large data base.

Table 4.2 gives an overview of the input parameters to the models, i.e. the quantities used to predict T_{div}, and shows what symbols will be used to refer to the given quantities. The indicated value range as well as the criteria for the data selection, the use of a reduced data set and the split of the data into a training and a test set will be explained later. First every quantity will be briefly introduced.

The plasma current, I_p, is the current induced by the transformer action. Its key property is that it sets the strength of the poloidal magnetic field which is an important quantity in the problem of particle and power transport in the *Scrape-Off Layer* (SOL) (see e.g. equation 1.10 or the equations of the *Two Point Model* (TPM) where the poloidal magnetic field enters via the connection length L).
The next quantity is the toroidal magnetic field imposed on the plasma, B_t.
P_{tot} is the heating power deposited in the plasma. Obviously the heating power is closely linked to the plasma temperature and therefore to its transport. Furthermore, the heating power is a key factor in setting the physical regime of the experiment, e.g. an increased confinement can be reached when the heating power surpasses a threshold [63]. In this regime of increased confinement periodic ELMs cause an increased transport of particles and power in the SOL. However, these ELMs are not considered in this analysis and their inclusion into the models would need further considerations.

The radiated power, $P_{rad,tot}$, much like the heating power, is a key quantity for the particle and power transport as shown e.g. by the TPM. In this analysis the radiated power only considers power losses occurring above the X-point. This quantity is measured by bolometry.
The core and edge electron densities are used as inputs to the models. The TPM showed

2 The interpretation of T_{div} as a temperature in regimes where the inner divertor is not in a detached state might be faulty but is also not necessarily needed for the presented analysis. The main goal is to simply predict T_{div} over a wide range of operational parameters.

Quantity	Symbol	Value Range
plasma current	$\mathbf{I_p}$	[0.2, 1.2] MA
toroidal magnetic field	$\mathbf{B_t}$	[-3.2, -0.99] T
heating power	$\mathbf{P_{tot}}$	[0.096, 20] MW
radiated power	$P_{rad,tot}$	(0, 17.9] MW
core electron density (line-integrated)	**H-1**	$[2.21{\cdot}10^{16}, 2.86{\cdot}10^{20}]\ \frac{1}{m^2}$
edge electron density (line-integrated)	H-5	$[6.98{\cdot}10^{13}, 2.86{\cdot}10^{20}]\ \frac{1}{m^2}$
neutral density in the divertor	$n_{div,ist}$	$[0, 2.19{\cdot}10^{22}]\ \frac{1}{m^3}$
stored energy	W_{MHD}	$[0.49, 1.25{\cdot}10^{6}]$ J
lower triangularity	$\boldsymbol{\delta_{untn}}$	[0.021, 0.54]
upper triangularity	$\boldsymbol{\delta_{oben}}$	[-0.085, 0.49]
elongation	$\boldsymbol{\kappa}$	[1.05, 1.96]
strike line position	Suna2b	[0, 1.26] m
hydrogen throughput	$\mathbf{H_{tot}}$	$[0, 3.54{\cdot}10^{22}]\ \frac{el.}{s}$
deuterium throughput	$\mathbf{D_{tot}}$	$[0, 8.13{\cdot}10^{22}]\ \frac{el.}{s}$
helium throughput	$\mathbf{He_{tot}}$	$[0, 8.83{\cdot}10^{21}]\ \frac{el.}{s}$
neon throughput	$\mathbf{Ne_{tot}}$	$[0, 5.03{\cdot}10^{21}]\ \frac{el.}{s}$
nitrogen throughput	$\mathbf{N_{tot}}$	$[0, 4.41{\cdot}10^{22}]\ \frac{el.}{s}$

Table 4.2: Input parameters of the models; bold-faced parameters were used for models with reduced set of inputs. The value range gives the minimum and maximum value in the training data after applying the selection criteria to the data.

that the density at the separatrix has a major influence on the power loads at the divertor targets. As an approximation for this, the edge density is used since the location of the separatrix can be difficult to determine. Furthermore, it is the core density that is the main handle on the separatrix density as well as the divertor regime (cf. figure 2.3). As the actual measurement via interferometry (see e.g. [64]) with a DCN laser probes the line-integrated densities, this is the quantity used in this analysis.
Another very important quantity determining the power loads at the divertor targets is the density of neutral particles in the divertor. Neutral particles cause both, energy and momentum losses, e.g. via plasma-neutral friction and charge exchange events. Again, the TPM indicates that these loss processes have a significant influence on the transport of particles and power towards the divertor targets.
The energy stored in the plasma is also included as an input to the models.
The upper and lower triangularity as well as the elongation of the plasma describe the shape of the plasma cross-section. As previous studies have shown, the plasma shape can have an important influence on the transport and confinement properties of the plasma (see e.g. [65] and further references therein). Therefore, these parameters were included in this analysis.
The strike line position describes the position at which the separatrix intersects the divertor target surface, measured along the target. As the structure of the divertor and its targets does also have an influence on the power loads on the target (directly via

geometric effects but also indirectly, e.g. due to confinement of neutral particles close to the target), the position at which the plasma primarily hits the divertor target was included in the analysis.
The geometric quantities of the plasma as well as the energy stored in the plasma are determined via magnetic equilibrium reconstruction based on the Grad-Shafranov equation, see e.g. [66] and references therein.
The final quantities included in the analysis are gas fluxes (also called throughputs) of different species, measured in electrons per second. Here, hydrogen, deuterium, helium, neon and nitrogen are considered. Hydrogen, deuterium and helium are used as primary fuel for fusion reactions in AUG, whereas neon and nitrogen are used as impurity species to cool the plasma in the SOL. The quantity that is more commonly used to determine the influence of impurity species in modeling is the impurity concentration. However, since the actual gas fluxes are more easily accessible, especially prior to performing an experiment, these signals were included in this work.

These quantities have been selected for this analysis due to their influence on the general plasma conditions as well as on the problem of particle and power transport towards the divertor targets. Furthermore, these signals were readily available and are commonly recorded for experiments at AUG and other tokamaks. Thus, this selection of parameters could allow for a comparatively easy inclusion of data from further reactors. Even though this is not an exhaustive list of all parameters that influence the transport of particles and power towards the divertor targets, the list of input quantities to the models was limited to these parameters to conduct this first approach of predicting power loads at the divertor targets via machine learning based techniques.

4.3 DATA SELECTION AND PREPARATION

To establish a large data base for the application of various machine learning methods the experimental data of AUG was used for this analysis. The experiments from which the data for this analysis was extracted were conducted between 2012 and 2018. A total of $\sim$ 7000 experiments was considered and scanned for appropriate data (discharge numbers 28000 to 35489). The use of experimental data, rather than data obtained from simulations, was motivated by the idea that the resulting models could implicitly model dependencies not captured by simulation codes. Furthermore, restricting the analysis to simulation data would have limited the best performance achievable by the machine learning models to that of the simulation.

From the experiments, those with a so called lower single null configuration, i.e. the only active divertor being on the bottom of the reactor, were selected. Furthermore, experiments with an inverted magnetic field were removed from the data base as there were very few such experiments. Hence, the value range of the field strength of the toroidal magnetic field in table 4.2 is limited to negative values, indicating that the

magnetic field direction leads to a gradient induced drift of the ions towards the bottom.

To extract data points for the analysis the signals of all quantities mentioned above were averaged over time frames of $0.2\,s$. This approach is based on the assumption that the divertor evolves through a series of (pseudo) equilibria and that, therefore, the data points can be considered to be independent of each other. With particle transport times on the order of $\frac{L}{c_s} \approx$ms and energy transport times on the scale of tens of ms in the SOL of AUG, changes of the plasma properties in the core, that happen on time scales of $\sim 0.1\,\mathrm{s}$ to $\sim 1\,\mathrm{s}$, would result in an adapted divertor state within tens to a few hundreds of ms. Thus, taking the average over $0.2\,\mathrm{s}$ should suffice to capture the equilibrium state corresponding to current plasma conditions.
Transient events, such as ELMs, are neglected in this analysis. Hence, these transient events might still have some influnce on the quantities used in the analysis, e.g. the measured radiated power might be larger than it would be in a truly stationary scenario due to the short outbursts of plasma and energy from the core.

To limit the analysis to the actual regime of operation, a stability criterion was imposed on the plasma current. For all $0.2\,\mathrm{s}$ time frames the plasma current was required to vary by at most 10% of the mean value from the mean within this time frame. If any data point within the current or the following $0.2\,\mathrm{s}$ time frame varied more strongly, the data point corresponding to the current time frame was removed from the analysis. The aim of this was to remove the current ramp-up and ramp-down as well as potential plasma disruptions, i.e. the abrupt termination of the plasma discharge caused by instabilities and indicated by a sudden reduction of the plasma current, from the analysis.
An example for the results of this algorithm is depicted in figure 4.2. The black line shows the time trace of the plasma current during an exemplary experiment and the red dots indicate data points where the plasma current fulfills the stability criterion. It can be seen that faulty data points have still been extracted from the ramp-up and ramp-down processes (the very first data point and the last one before the 8 s mark) and that some data points also come from the time after the end of this plasma discharge. However, the latter could be removed by introducing further criteria on the data.

Obviously erroneous measurements were removed by applying additional cuts to the extracted data. All data points with negative density measurements in the plasma core or the edge region, a confined energy of $\leq 0\,\mathrm{J}$ or radiated power of $\leq 0\,\mathrm{W}$ were removed from the data base. On top of this the fraction of total radiated power over heating power was required to be smaller than one, as the plasma would be unstable otherwise, and the total deposited heating power was limited to the range $(0, 20]$ MW. T_{div} was limited to $[-5, 30]$ eV, motivated by the reliability of T_{div} as an estimate of the electron temperature in the outer divertor which only holds for a detached inner divertor as discussed before. The result of these additional criteria is depicted in figure 4.2 by red squares. In the example, the averaged plasma current was extracted at every red point, but only the data points corresponding to the red squares remain after these additional data cuts. Hence, the obviously faulty data points from the time after the end of the experiment

could be removed. However, a few data points from the ramp-up and ramp-down of the current still remain.
With these additional data selection criteria data points from about 4500 seperate experiments remained in the data base.

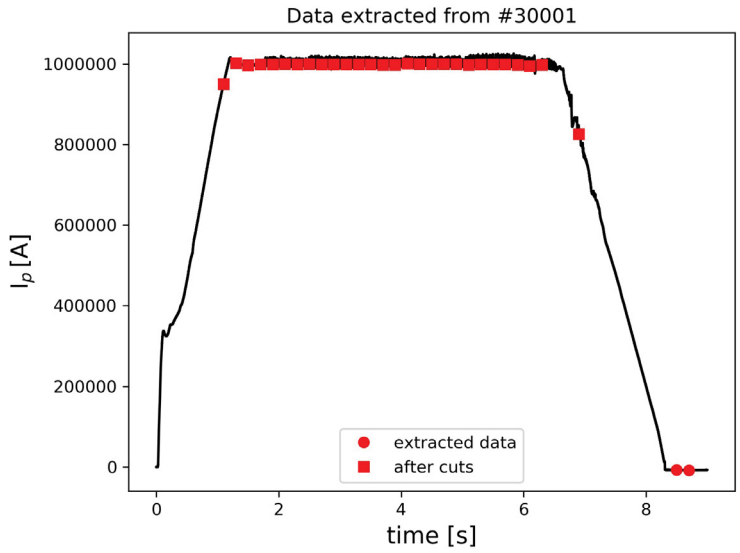


Figure 4.2: Exemplary data selection from a time trace of one discharge. The red dots mark data points fulfilling the stability criterion imposed on the plasma current, squares indicate data points remaining after additional cuts to the data. [59]

As the ranges of the parameter values in table 4.2 also indicate, there are still unusual data points in the data base. For example, the very small lowest value of the heating power is the result of a drop in heating power during an experiment probing the parameter space at a very small plasma current and an overall small heating power (discharge number 35135).

On top of using all the quantities in table 4.2 as inputs to the models, a reduced set of inputs was tested as well. This selection only contains the directly and indirectly controllable engineering parameters accessible prior to a given experiment marked by bold-faced letters in table 4.2. Thus, with this reduced input set the models are close to the actual planning phase of experiments.

In order to test the models on different data than the one used for the optimization, the whole data base was split in a training set and a test set. This was achieved by randomly selecting experiments for either set and assigning all data points extracted from the corresponding experiment to that data set. In this way 70% of the analyzed experiments were randomly assigned to the training set while the remaining data was used for the test set to evaluate the models' performance after the optimization. Since the

computational cost of GPR scales roughly as the number of training data points squared and all training data points are needed for the predictions of the model (see equations 3.20), the fraction of data used for the GPR approach was limited to 7% of all discharges, so that the optimization of the GPR is based on roughly a tenth of the data used for the *Random Forest* (RF) and *Neural Network* (NN) optimization. Again, the remaining data was assigned to the test set.

Finally all inputs of training and test set were standardized to zero mean and unit variance by subtracting the mean of the given quantity over the training data from every data point and dividing by the standard deviation of the quantity within the training set.

The extracted time averages will be denoted by the symbol $< \cdot >$ or by the prefix mean.

4.4 DATA ANALYSIS

As a first step of analyzing the extracted data, the distributions of the target quantity, T_{div}, in the training and the test set were analyzed. The resulting distributions are shown in figure 4.3. The figure shows the distribution of T_{div} values in the training data in blue and in the test data in yellow. Both distributions show similar shapes and are peaked around 10 eV, indicating a reasonable split of the data. The peak in the distribution also obviously indicates preferred operational scenarios in AUG.

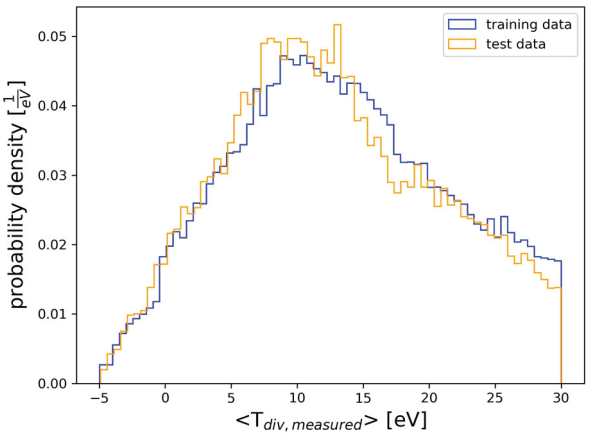


Figure 4.3: Distribution of $<T_{div}>$ extracted from the experiments for training and test set used for the random forest and neural network based regression. [59]

Figure 4.4 shows the matrix of pairwise Pearson correlation coefficients among the input quantities and the target quantity calculated from the data in the training set. The correlation matrix shows that T_{div} is only weakly correlated with most of the other

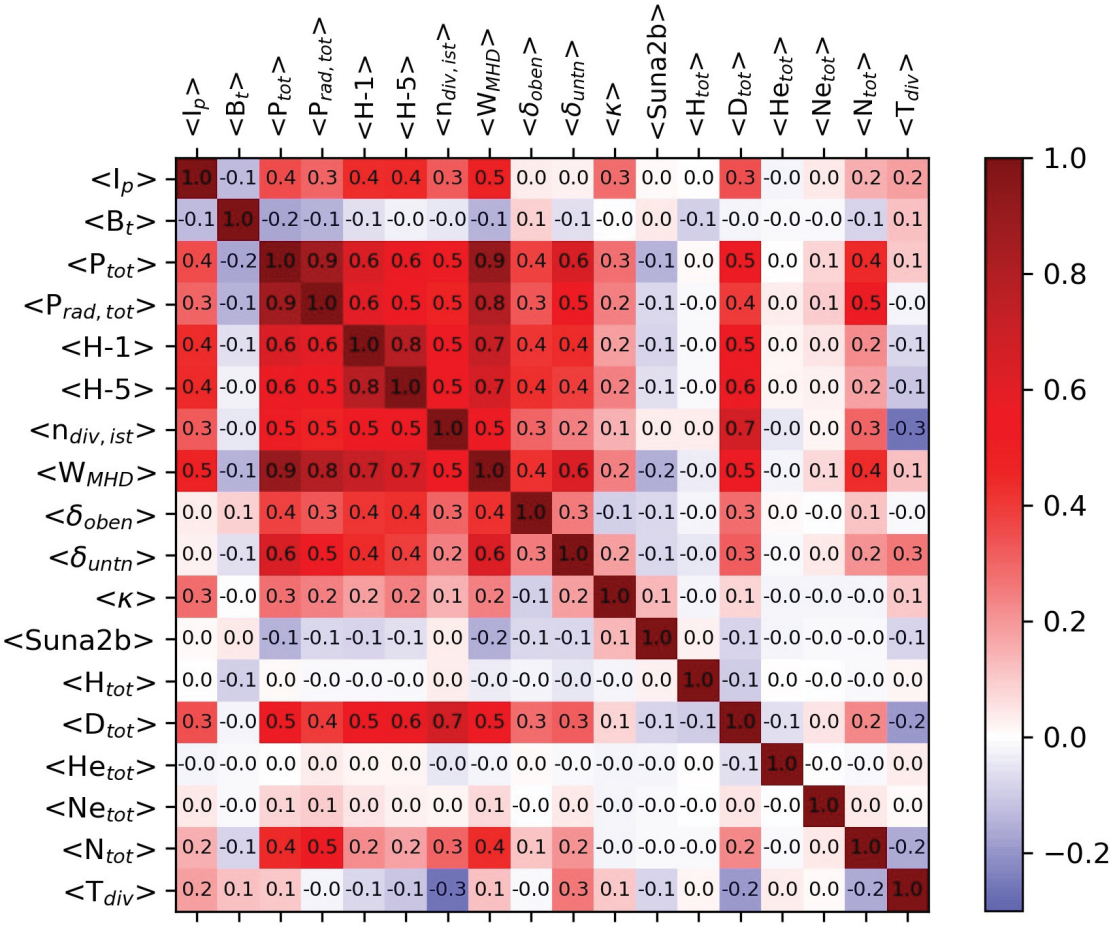


Figure 4.4: Matrix of pairwise Pearson correlation coefficients of all input quantities and the target quantity, $<T_{div}>$, calculated from the training data used for the random forest and neural network based approaches. [59]

quantities. The strongest correlations are a correlation of -0.3 with the density of neutral particles in the divertor and a correlation of 0.3 with the lower triangularity. A (strong) correlation of T_{div} with the neutral density in the divertor could be expected as the neutral particles in the divertor are crucial for the plasma-neutral interactions which cause energy and momentum losses, reducing the power load at the targets, as indicated by the TPM. Furthermore, a drop in T_{div} with increasing neutral density is also indicated by dedicated physics analyses with a more rigorous data selection such as [67], where the authors found that T_{div} decreases with an increase in the concentration of nitrogen in the divertor.

As these correlation coefficients only indicate pairwise linear dependencies, T_{div} might still be a non-linear function of any combination of the given quantities.

Further correlations in the analysis reinforce the viability of the presented data selection. Expected correlations such as a strong correlation between core and edge densities and the heating power and the radiated power are evident in figure 4.4. Another expected correlation can be observed between the deuterium and nitrogen throughput and the neutral density in the divertor. As could be expected, the density of neutral particles in the divertor tends to increase with the injection of nitrogen and deuterium. Missing correlations of the hydrogen, helium and neon throughputs are caused by a lack of variation in these quantities as the data base contained only few data points with a significant throughput of these species.

For the results obtained from the data split used for the GPR based approach, see appendix C.

It has to be noted that the presented analysis contains physics-induced correlations such as the one between core and edge density and the neutral density in the divertor but also includes correlations set by the operational boundary conditions of the experiment. One such example would be the correlation between the heating power and the core and edge density which is imposed by the operational restrictions of the experiment. These two types of correlations are not uncoupled in this analysis.

To further test the viability of the data selection, results from [68] were reproduced. In the referenced publication the authors found that the electron density at the separatrix, $n_{e,sep}$, depends on the pressure of neutral particles in the divertor, P_0, as

$$n_{e,sep} \propto P_0^{0.31} . \tag{4.1}$$

As these two signals are not directly part of this analysis, the line-integrated electron density in the edge, <H-5 >, was divided by the experimentally determined geometric chord lengths of the interferometric measurement to obtain an estimate of the line averaged density at the separatrix. Furthermore, the neutral density in the divertor is used as a proxy for the pressure of neutral particles in the divertor. The regression resulting from the data selection in this thesis is depicted in figure 4.5. The figure shows the decadic logarithm of the line averaged edge density as a proxy for the separatrix density versus the decadic logarithm of the neutral density in the divertor. The red data points are data from the training set. Due to the coarse data selection, the data contains some outliers with densities in the edge region of less than $10^{18}\,m^{-3}$. Due to these outliers, two regression approaches were tested. The result of a conventional least squares approach is shown by the dashed blue line and yields $n_{e,sep} \propto n_{div,ist}^{0.325}$ with an R^2 value of 0.59. The orange line indicates the result obtained from a repeated median regression [69], which could be expected to be more robust to the outliers in the data. This regression yields $n_{e,sep} \propto n_{div,ist}^{0.277}$ with an R^2 value of 0.57. In comparison to the original result from [68], with $n_{e,sep} \propto n_{div,ist}^{0.31}$ and an R^2 of 0.7, the results obtained in this thesis deviate from the reference but show similar trends despite the less rigorous data selection.

For additional results of the analysis of the collected data see appendix D.

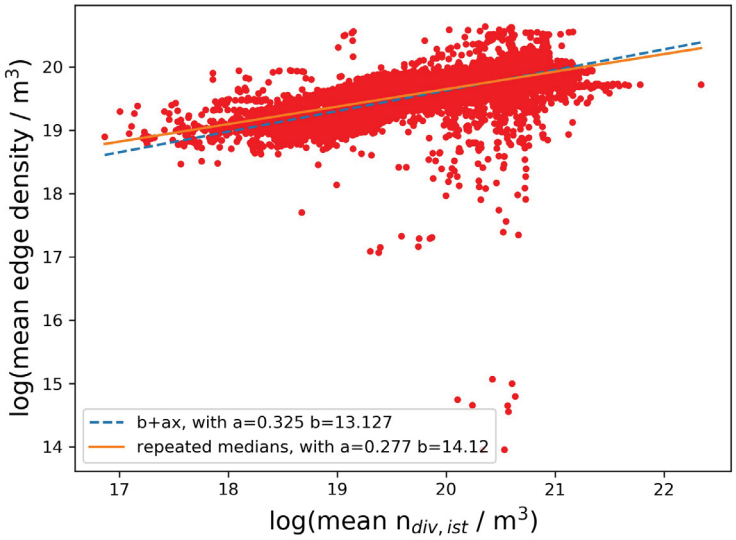


Figure 4.5: Dependency of the line averaged edge density on the neutral density in the divertor to reproduce results reported in [68]. The dashed blue line shows the result of a least squares fit. The continuous orange line shows the result of repeated median regression. [59]

5 MACHINE LEARNING BASED SURROGATE MODELS FOR POWER EXHAUST PREDICTION

This chapter contains the main results of the presented analysis. First, the results obtained with a *Random Forest* (RF) based approach to modeling divertor power loads will be presented. After that the results of the *Neural Network* (NN), *Mixture Density Network* (MDN) and the *Gaussian Process Regression* (GPR) based approaches will be presented. The analysis focuses on the model performance and the investigation of the feasibility of machine learning based approaches to modeling divertor power loads in a tokamak. However, besides the analysis of the models' performances, the extracted dependencies of some of the models will also be discussed as these give a handle on the interpretability of the models and allow for comparisons to other studies.

5.1 RANDOM FORESTS FOR DIVERTOR POWER LOAD PREDICTION

In this section the details of the RF based approach to modeling divertor power loads will be discussed and results obtained with this model will be presented. Parts of this section have been published in [59].

The RF models presented here have been set up and evaluated using scikit-learn [44] version 0.20.3.

5.1.1 *Random Forest based on all input quantities*

The RF tested in this analysis consists of 10 single decision trees with no limitation in tree depth. The depth of the trees was not limited as an analysis of the effect of the maximum tree depth on the model's performance did not show a deterioration of the model's predictiveness on the test data. For this test, the maximum tree depth was varied between 5 and 50. The resulting distributions of model predictions versus T_{div} values from the test data are shown in figures 5.1 and 5.2. The figures show the model predictions obtained on the test data and the blue line indicates a 1:1 relation, i.e. a perfect match of model predictions and the values actually extracted from the experiments.

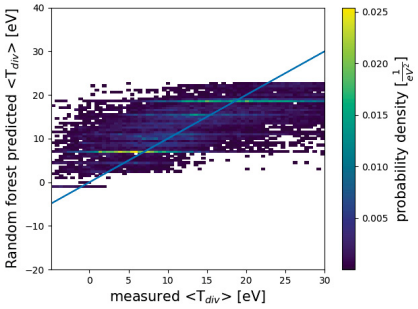
As the results show, the model performance does not tend to deteriorate with large maximum tree depths but actually improves up to tree depths of at least 20 or 25. Hence, the maximum tree depth in this analysis was not limited. Furthermore, as the distributions for a maximum depth of 45 and 50 appear to be the same, this is where the model's dependency on the maximum tree depth seems to saturate. This is also confirmed by investigating the number of nodes in both RFs. Both models show that the

ten trees have the same number of nodes with both a maximum depth of 45 and one of 50. With these maximum depths all ten trees have around 63000 nodes in total. The large number of nodes in the models compared to the total amount of training data points ($\sim$ 50000) could imply overfitting issues. However, these do not show in the evaluation of the models on the test data shown in figure 5.1 and 5.2 as the model predictions for very large maximum tree depths still fit the values from the experiments.
The evolution of the model's performance as a function of maximum tree depth also shows that a smaller maximum depth might suffice to allow for adequate predictions. This could be caused by the lack of correlation of some of the features with the target quantity, T_{div}. This might reduce the number of necessary splits in the tree. A further analysis of the final model's dependencies will be shown later in this thesis and seems to support this view.

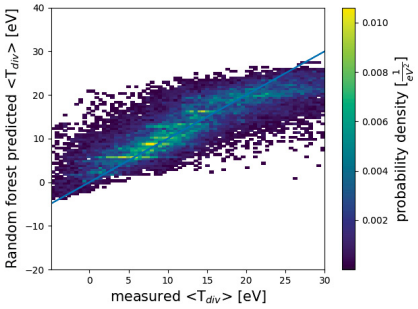
As mentioned in section 3.1.2, usually not all trees in the decision tree have access to all input quantities during the training process to assess the best split. However, this was not the case in this analysis. An approach based on using only 70% of the available input quantities at every split for the optimization of the model was tested with another RF with ten decision trees and no limitation in depth. However, this test did not show significant differences to the approach utilizing all the input quantities. The resulting predictions on the test data for this approach with a limited number of inputs considered for each split is shown in figure 5.3. The figure shows the predictions of the trained RF model on the test data versus the values obtained from the experimental data. Again, the blue line indicates a 1:1 relation of predictions and measured values. In comparison to the results obtained in the study of the maximum depth, there is no significant difference visible. Hence, the number of input quantities considered at each split for the optimization of the RF models was not limited in this analysis.

A varying number of decision trees within the RF was also tested. The results of this parameter scan for 20, 50, 100, 200, 500 and 1000 trees are depicted in figures E.1 and E.2 in appendix E. In this test the maximum depth of the trees was not limited. Again, there is no obvious difference between the results obtained with only 10 trees in the RF and the larger numbers of trees tested here. Thus, the baseline model was set to contain 10 trees.

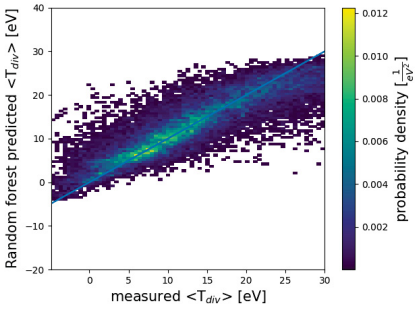
The tests presented above were carried out with a RF model utilizing the full set of all inputs of table 4.2. The results of the final model with ten decision trees and no limitation in tree depth are shown in figure 5.4.
As can be seen the model manages to capture the trend of a 1:1 relation as well and reproduces the distributions observed for the two largest maximum tree depths. The latter is due to the fact that this model, unlimited in tree depth, also grew to around 63000 nodes in all ten trees and shows the same number of nodes in each tree as the model with a maximum depth of 50. Another aspect in these observed similarities is the fixed random seed that was passed to all models in order to guarantee reproducible results. Furthermore, the distribution shows a peak around 5 to 10 eV caused by two



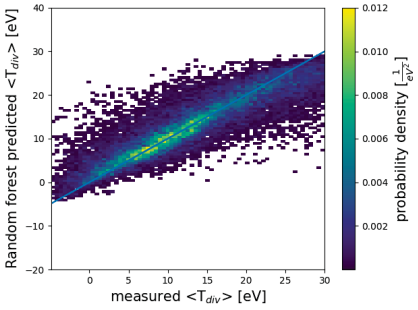
(a) Distribution of random forest predictions versus T_{div} values from the test data for a maximum tree depth of 5.



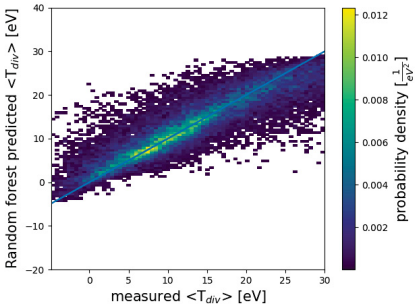
(b) Distribution of random forest predictions versus T_{div} values from the test data for a maximum tree depth of 10.



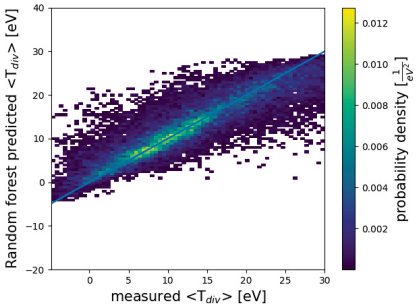
(c) Distribution of random forest predictions versus T_{div} values from the test data for a maximum tree depth of 15.



(d) Distribution of random forest predictions versus T_{div} values from the test data for a maximum tree depth of 20.

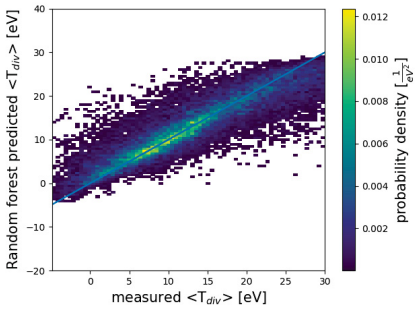


(e) Distribution of random forest predictions versus T_{div} values from the test data for a maximum tree depth of 25.

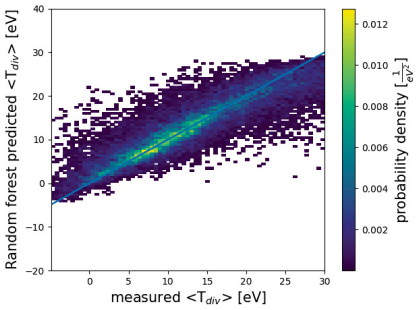


(f) Distribution of random forest predictions versus T_{div} values from the test data for a maximum tree depth of 30.

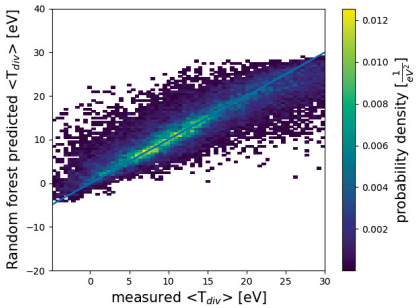
Figure 5.1: Predictions of random forest models with different maximum tree depths versus T_{div} values from the test data. The blue line indicates where the model predictions would perfectly fit the data obtained from the experiments.



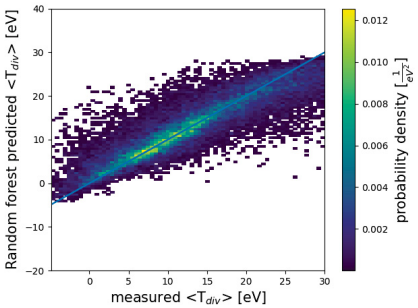
(a) Distribution of random forest predictions versus T_{div} values from the test data for a maximum tree depth of 35.



(b) Distribution of random forest predictions versus T_{div} values from the test data for a maximum tree depth of 40.



(c) Distribution of random forest predictions versus T_{div} values from the test data for a maximum tree depth of 45.



(d) Distribution of random forest predictions versus T_{div} values from the test data for a maximum tree depth of 50.

Figure 5.2: Predictions of random forest models with different maximum tree depths versus T_{div} values from the test data. The blue line indicates where the model predictions would perfectly fit the data obtained from the experiments.

effects. First, this is where the distribution of T_{div} values from the experiment showed a peak (see figure 4.3) and second, this is where the model tends to perform best.
As a figure of merit of the model's performance, the median and the central 68^{th} percentile of the distribution of absolute differences between the model predictions and the measured values from the experiment was determined. For this model this results in a median absolute difference of $1.8^{+3.1}_{-1.3}$ eV.

The aforementioned assumption that a lot of the input quantities, that are mostly weakly correlated with T_{div}, have little influence on the model is reinforced when investigating the feature importances of the RF model. By analyzing the normalized (across all input quantities) averaged (across the different trees) variance reduction of every feature in the optimized RF, it was found that the relative importances of the features are dominated by the contributions of the plasma current, the density of neutral particles in the divertor and the lower triangularity. These features have importances of 0.153, 0.239 and 0.145, respectively and, thus, make up for more than half of the overall reduction in

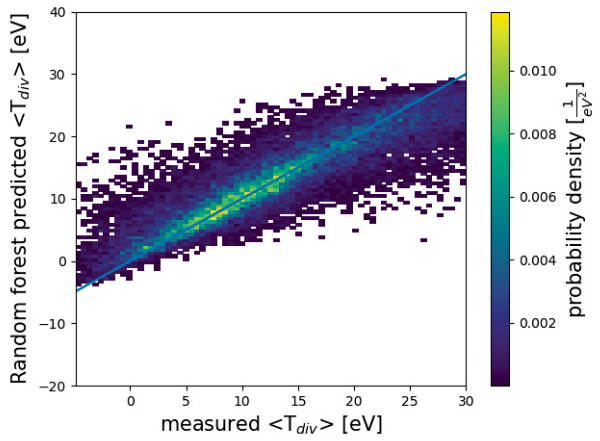


Figure 5.3: Predictions of the random forest model using 70% of the available input quantities to determine the optimal split versus T_{div} values from the test data. The blue line indicates where the model predictions would perfectly fit the data obtained from the experiments.

variance of the model.

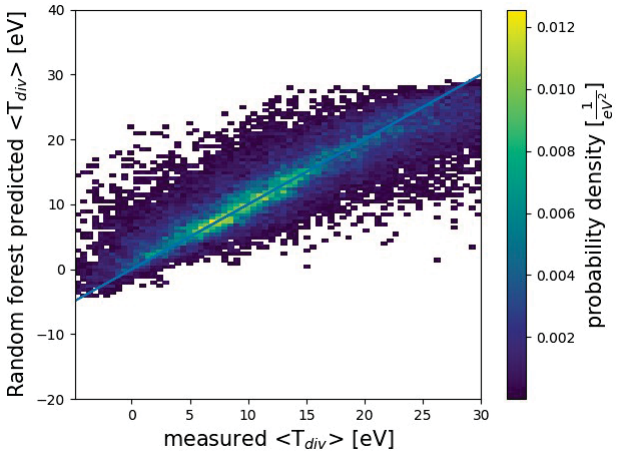


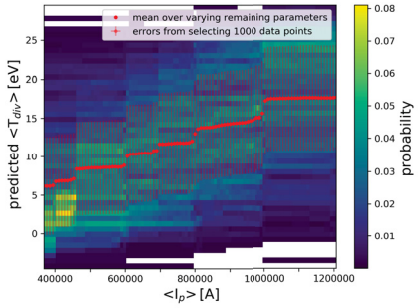
Figure 5.4: Predictions of the random forest model using the full set of inputs of table 4.2 versus T_{div} values from the test data. The blue line indicates where the model predictions would perfectly fit the data obtained from the experiments. [59]

To further analyze the learned dependencies of the model, partial dependency plots are shown in figures 5.5, 5.6 and 5.7. In these plots the dependencies of the optimized RF model on the different input quantities are shown. To achieve this, the relevant quantity (on the abscissa in the plots) was varied between $q_1 - 0.1|q_1|$ and $q_{98} + 0.1|q_{98}|$, where q_1 and q_{98} are the 1^{st} and 98^{th} percentile of the distribution of the values of the given

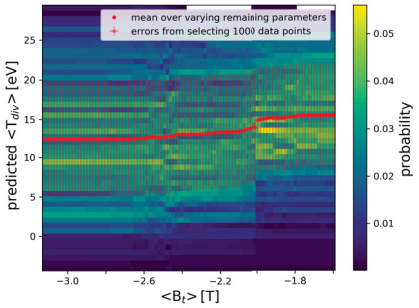
input quantity in the training data. To marginalize over the dependencies on all other input quantities, all remaining input values were determined by randomly selecting 1000 data points from the training data. Thus, for every value on the abscissae in the plots, a distribution of 1000 model predictions was obtained. The red dots indicate the mean of this distribution and the error bars represent the distribution's standard deviation. This procedure approximates a proper marginalization over the other input quantities (cf. [70]).

As can be seen in the plots, the model predictions depend linearly on the plasma current, figure 5.5(a). The step-like structure is caused by the inherent tree structure of the model. The roughly linear dependency on the plasma current matches the expectation from other physics analyses, as the power load at the divertor target is expected to be proportional to $\frac{1}{\lambda_q}$ (see equation 1.10) and it is roughly $\lambda_q \propto \frac{1}{I_p}$ [10]. Thus, from the *Two Point Model* (TPM) it is to be expected that T_{div} scales (almost) linearly with I_p.
Most of the other dependencies of the model are very weak or non-existent. The latter is the case for the neon throughput, figure 5.7(d). Here, the variation within the limits given by q_1 and q_{98} did not result in any significant value for the throughput. This is caused by the fact that the data base only contains 395 data points with non-zero neon throughput (cf. appendix D).
Furthermore, due to the RF's capability to effectively reduce the influence of non helpful input quantities by placing splits based on such quantities very deep in the tree structure, the model only shows very weak dependencies on a variety of the inputs. This finding is in accordance with the aforementioned feature importances that indicated significant dependencies of the model on the plasma current, the density of neutral particles in the divertor and the lower triangularity.
Another result in accordance with physics based expectations is the dependency on the neutral particle density in the divertor. As described before, the neutral particles in the divertor tend to reduce the power load on the divertor target. This result is recovered by the model.
The model's dependency on the lower triangularity was not as clearly expected as the dependencies on the plasma current and neutral particle density. A tentative improvement of the confinement in the plasma core with increasing triangularity might be expected, see e.g. [71]. However, this is not in line with the increasing model predictions of T_{div}. This dependency might be caused by cross-correlations of the lower triangularity with other input quantities. As the matrix of correlation coefficients, figure 4.4, indicates, the lower triangularity is primarily correlated with the deposited heating power, $<P_{tot}>$, and the stored energy in the plasma, $<W_{MHD}>$. A further study of these cross-correlations could yield insights into the cause of the model's dependencies.

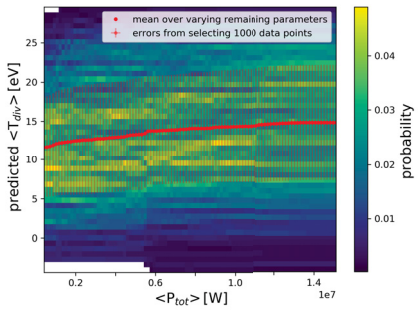
To further evaluate the model's performance on unseen data, the model was tested on all time averages over 0.2 s from some exemplary plasma discharges from the test data. In this test, the selected data points were neither required to fulfill the stability criterion imposed on the plasma current, nor the additional criteria of the data selection



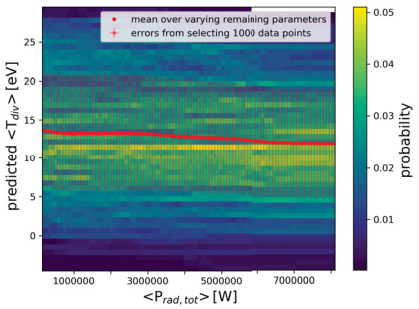
(a) Dependency of random forest predictions using all inputs on the plasma current. The red dots indicate mean and standard deviation of the column-wise normalized distributions. [59]



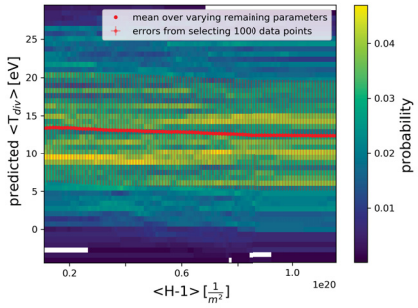
(b) Dependency of random forest predictions using all inputs on the strength of the toroidal magnetic field. The red dots indicate mean and standard deviation of the column-wise normalized distributions.



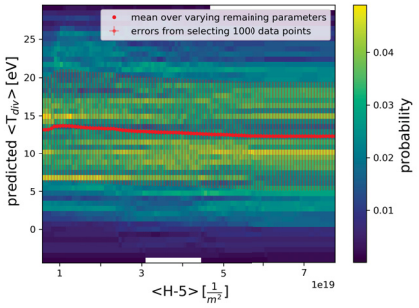
(c) Dependency of random forest predictions using all inputs on the deposited heating power. The red dots indicate mean and standard deviation of the column-wise normalized distributions.



(d) Dependency of random forest predictions using all inputs on the radiated power above the x-point. The red dots indicate mean and standard deviation of the column-wise normalized distributions.

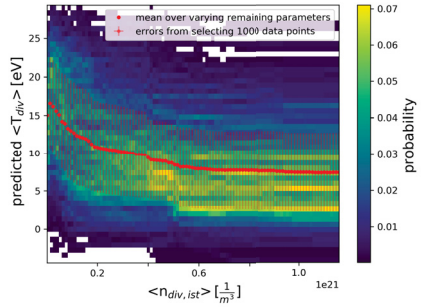


(e) Dependency of random forest predictions using all inputs on the line-integrated electron density in the core. The red dots indicate mean and standard deviation of the column-wise normalized distributions.

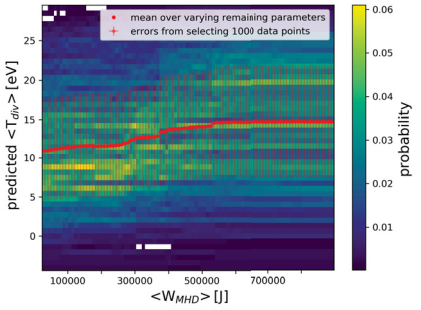


(f) Dependency of random forest predictions using all inputs on the line-integrated electron density in the edge region. The red dots indicate mean and standard deviation of the column-wise normalized distributions.

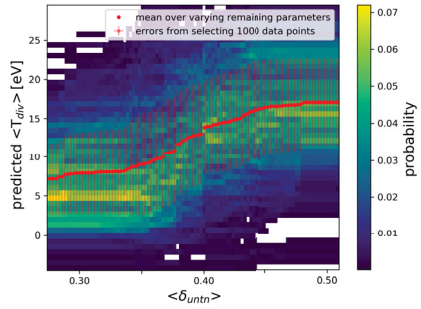
Figure 5.5: Dependencies of the random forest model using all input quantities on the various input quantities. The depicted distributions were obtained by inserting the given value of the quantity on the abscissa into 1000 randomly selected data points from the training set.



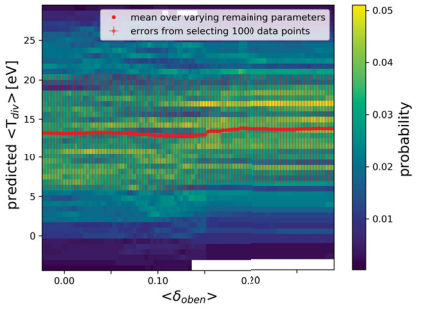
(a) Dependency of random forest predictions using all inputs on the neutral particle density in the divertor. The red dots indicate mean and standard deviation of the column-wise normalized distributions. [59]



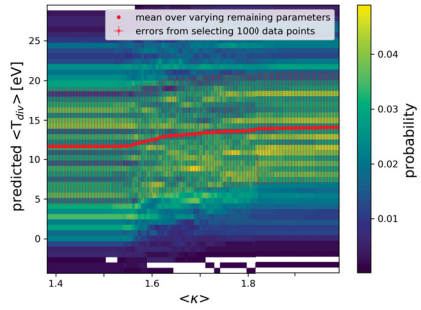
(b) Dependency of random forest predictions using all inputs on the stored energy in the plasma. The red dots indicate mean and standard deviation of the column-wise normalized distributions.



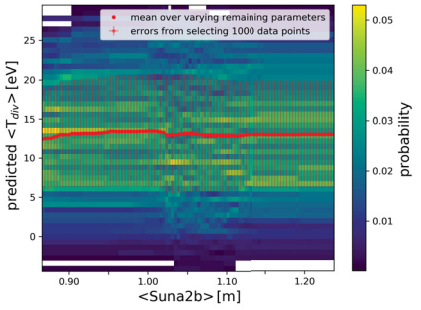
(c) Dependency of random forest predictions using all inputs on the lower triangularity. The red dots indicate mean and standard deviation of the column-wise normalized distributions. [59]



(d) Dependency of random forest predictions using all inputs on the upper triangularity. The red dots indicate mean and standard deviation of the column-wise normalized distributions.

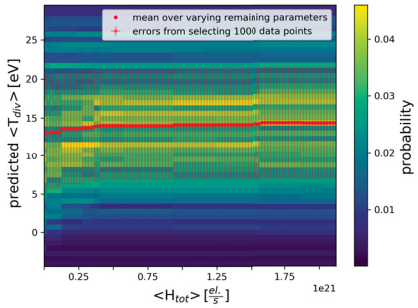


(e) Dependency of random forest predictions using all inputs on the elongation of the plasma cross section. The red dots indicate mean and standard deviation of the column-wise normalized distributions.

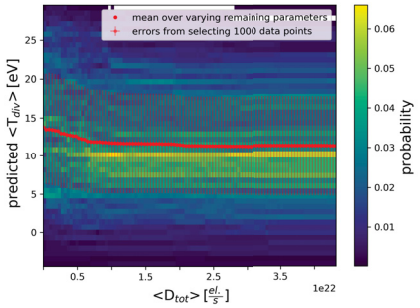


(f) Dependency of random forest predictions using all inputs on the position of the strike line. The red dots indicate mean and standard deviation of the column-wise normalized distributions.

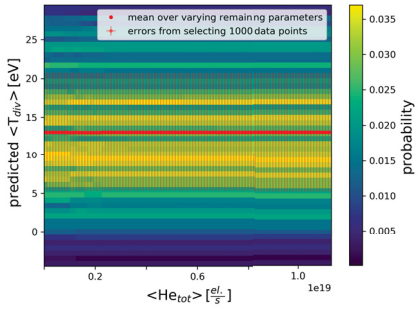
Figure 5.6: Dependencies of the random forest model using all input quantities on the various input quantities. The depicted distributions were obtained by inserting the given value of the quantity on the abscissa into 1000 randomly selected data points from the training set.



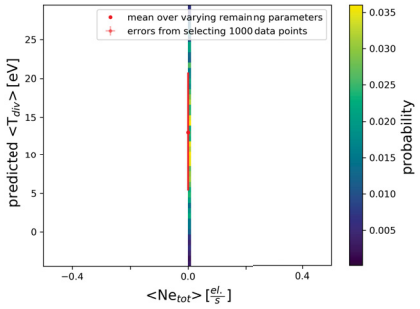
(a) Dependency of random forest predictions using all inputs on the hydrogen throughput. The red dots indicate mean and standard deviation of the column-wise normalized distributions.



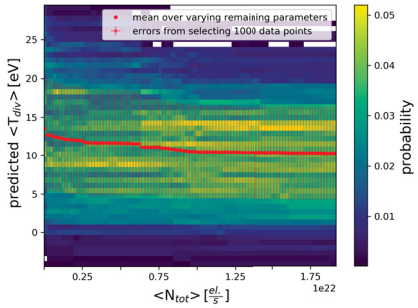
(b) Dependency of random forest predictions using all inputs on the deuterium throughput. The red dots indicate mean and standard deviation of the column-wise normalized distributions.



(c) Dependency of random forest predictions using all inputs on the helium throughput. The red dots indicate mean and standard deviation of the column-wise normalized distributions.

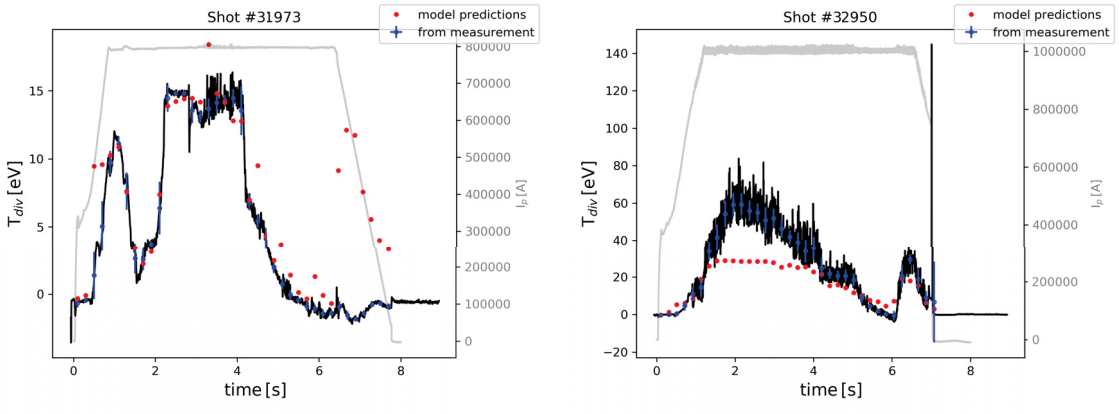


(d) Dependency of random forest predictions using all inputs on the neon throughput. The red dots indicate mean and standard deviation of the column-wise normalized distributions.



(e) Dependency of random forest predictions using all inputs on the nitogen throughput The red dots indicate mean and standard deviation of the column-wise normalized distributions.

Figure 5.7: Dependencies of the random forest model using all input quantities on the various input quantities. The depicted distributions were obtained by inserting the given value of the quantity on the abscissa into 1000 randomly selected data points from the training set.



(a) [59] (b)

Figure 5.8: RF predictions (red dots) for some exemplary discharges from the test data. The black line shows the measured T_{div} curve and the blue points indicate the extracted averages over 0.2 s (without applying any cuts to the data). The standard deviations stem from the variation within 0.2 s. The gray curve shows the plasma current.

presented in section 4.3. This allows for a test of the model in a worst case scenario. The results obtained from some of the exemplary discharges are depicted in figure 5.8. The plots show the time traces of the measured signal for T_{div} (black line), the time averaged values extracted from this signal (blue dots) with a standard deviation determined by the standard deviation of the signal within the 0.2 s, the model predictions (red dots) and the time trace of the plasma current (gray line).

In figure 5.8(a) it can be seen that the model predictions match the values extracted from the measured signal rather well within the operational regime of a constant plasma current. Above this, the model also captures the falling trend in T_{div} from the 4 s mark onwards. At this point in the discharge the nitrogen throughput was increased which results in this reduction in T_{div}. Despite only using the instantaneous throughput as input to the model, the RF accurately predicts the falling tendency in T_{div}. Since the model shows hardly any dependency on the nitrogen throughput (cf. figure 5.7(e)), this effect might, again, be caused by correlations among input quantities. However, this test also shows that the model performance deteriorates during the current ramp-up and ramp-down. Of course, these regimes are not the main focus of this analysis and were mostly removed from the training data, explaining this effect.
Another important result and inherent property of RFs shows in figure 5.8(b). Between the $\sim$ 2 s and 4 s mark the model predictions systematically underestimate the actual values of T_{div}. This is a result of the limit imposed on T_{div} in the training data and its effect on the RF. Since the RF predictions correspond to averages of values assigned to leaf nodes of the decision trees and these values are taken from the training data, the model predictions can not lie outside the range of values observed during the training process.

5.1.2 *Random Forest based on a reduced set of input quantities*

The same RF model as above, with 10 decision trees and no limitation in tree depth, was used to obtain predictions of T_{div} based on the reduced set of input quantities marked by bold faced characters in table 4.2. To this end, the model was trained and tested on the same data as before, but with fewer input quantities. With this reduced selection of input quantities the model is closer to conditions actually available prior to performing a plasma discharge. Most of the selected quantities (with exception of the line-integrated core electron density) are considered engineering parameters suited for planning an experiment.

Figure 5.9 shows the results obtained on the test data. Depicted are the model predictions versus the target values obtained from the experiments, the blue line indicates a 1:1 relation of model predictions and experimental values. In comparison to the results of the RF model with all inputs, see figure 5.4, the distribution in figure 5.9 is slightly less focused. However, the reduced model also manages to capture the trend of a 1:1 relation between model predictions and values extracted from the experiments. To quantify the model's performance, again the median and the central 68^{th} percentile of the distribution of absolute differences between model predictions and target values were determined. The result is a median absolute difference of $2.1^{+3.6}_{-1.5}$ eV. Thus, the model shows a tentatively worse performance than the RF model with all inputs. Nevertheless, the model performances are comparable within the uncertainty on the median absolute difference of model predictions and experimental values. This indicates that the reduced set of input quantities could suffice to make predictions of the divertor power loads prior to performing a discharge.

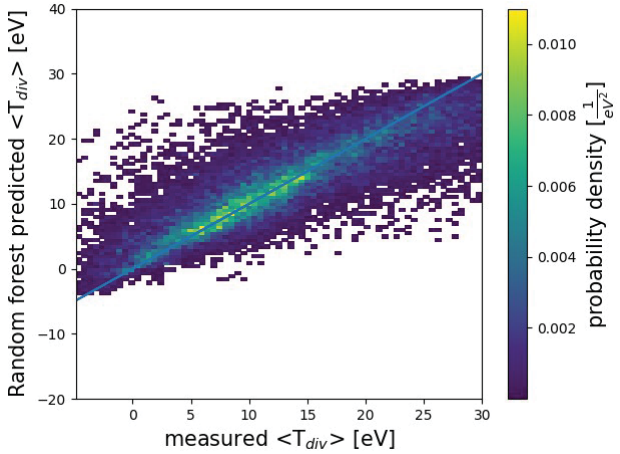
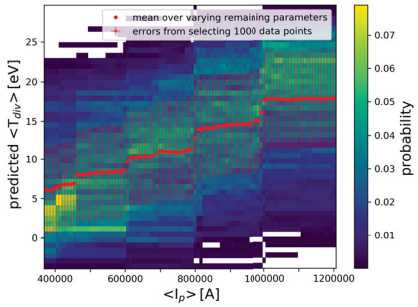


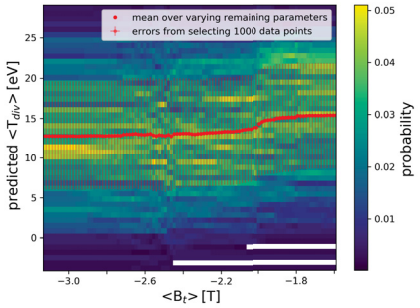
Figure 5.9: Predictions of the random forest model using the reduced set of inputs of table 4.2 versus T_{div} values from the test data. The blue line indicates where the model predictions would perfectly fit the data obtained from the experiments. [59]

A further analysis of the model's learned dependencies is shown in figures 5.10 and 5.11. The partial dependencies were obtained as described before.
As for the RF model with all inputs two of the strongest dependencies are those on the plasma current and the lower triangularity, also confirmed by the feature importances of 0.161 and 0.182, respectively. The shape of these dependencies is also comparable to that of the RF with all inputs. Again, a linear dependency on the plasma current is found that is in accordance with the expectation.
The first difference can be observed for the deuterium throughput. With the reduced set of inputs the deuterium throughput has a feature importance of 0.212 and the dependency of the model predictions on the deuterium throughput is much more evident in figure 5.11(c) than is the case for the corresponding result of the RF with all inputs. Another interesting observation is that the dependency on the deuterium throughput now shows a similar shape to that of the dependency on the neutral particle density in the divertor observed before, see figure 5.6(a). This result is also in accordance with other analyses where a tight coupling of deuterium throughput and neutral particle density was found [72]. Furthermore, this is also a result of the strong correlation of deuterium throughput and the neutral particle density in the divertor observed in figure 4.4.
Other changes with respect to the observations from the model with all inputs include the dependency on the line-integrated core density, H-1, and on the nitrogen throughput. A tendency of decreasing model predictions with increasing core density and nitrogen throughput can be seen. This would also be in accordance with expectations from e.g. the TPM.
As was the case before, the data does not contain a significant amount of data points with non-zero neon throughput, which is why the variation in the plots, depending on the distribution of observed values for each quantity, does not show any variation in the neon throughput.
The model predictions show little to no dependency on the other parameters. Again, this is due to the ability of the RF model to reduce the effects of non helpful inputs. For example, the dependency on the elongation, in combination with the distribution shown in figure D.2(f) in appendix D, shows that the RF predictions only depend on the plasma elongation in the region of the bulk of the distribution of values of the elongation in the training data. As could be expected, predictions corresponding to sparsely populated regions in the training data yield constant results.

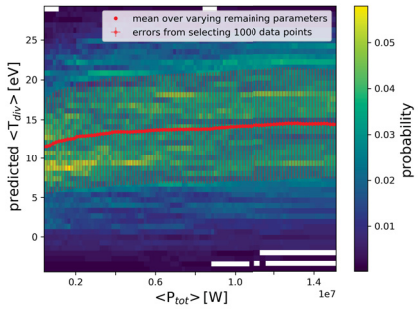
This model was also tested on some exemplary discharges disregarding the data selection criteria, as described for the RF model with all inputs. The results for two of these discharges are shown in figures 5.12(a) and 5.12(b). Again, the black line is the measured signal of T_{div}, the blue dots are averages over 0.2 s of this signal and the error bars on the blue dots represent the standard deviation of the signal within these 0.2 s. The red dots indicate the model predictions and the gray curve shows the plasma current.
Comparing figures 5.12(a) and 5.8(a), slight differences of the model performances can be observed. In general the RF model with a reduced set of inputs still manages to predict T_{div} and also captures the trend of falling T_{div} from 4 s onwards, even more



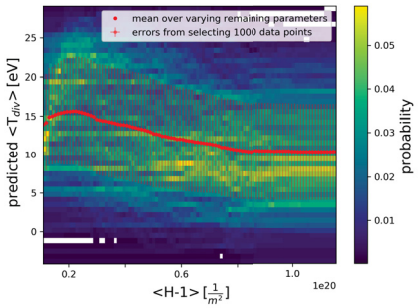
(a) Dependency of random forest predictions using the reduced set of inputs on the plasma current. The red dots indicate mean and standard deviation of the column-wise normalized distributions. [59]



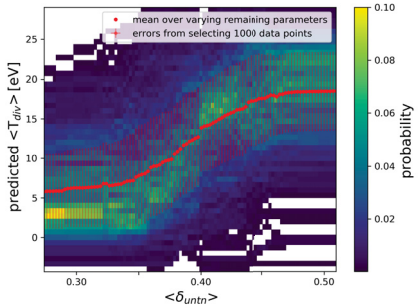
(b) Dependency of random forest predictions using the reduced set of inputs on the strength of the toroidal magnetic field. The red dots indicate mean and standard deviation of the column-wise normalized distributions.



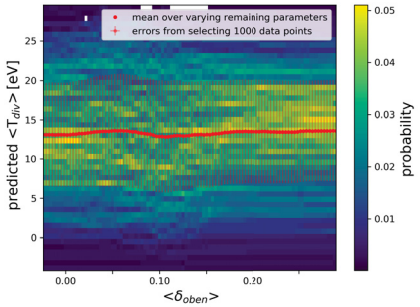
(c) Dependency of random forest predictions using the reduced set of inputs on the deposited heating power. The red dots indicate mean and standard deviation of the column-wise normalized distributions.



(d) Dependency of random forest predictions using the reduced set of inputs on the line-integrated electron density in the core. The red dots indicate mean and standard deviation of the column-wise normalized distributions. [59]

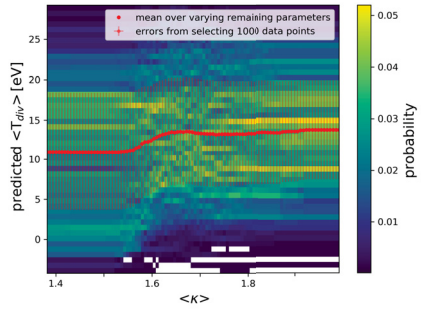


(e) Dependency of random forest predictions using the reduced set of inputs on the lower triangularity. The red dots indicate mean and standard deviation of the column-wise normalized distributions. [59]

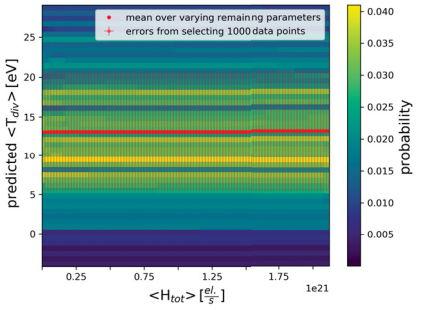


(f) Dependency of random forest predictions using the reduced set of inputs on the upper triangularity. The red dots indicate mean and standard deviation of the column-wise normalized distributions.

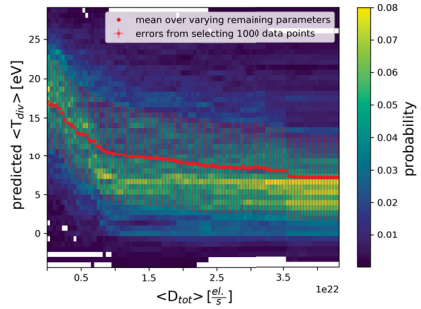
Figure 5.10: Dependencies of the random forest model using the reduced set of input quantities on the various input quantities. The depicted distributions were obtained by inserting the given value of the quantity on the abscissa into 1000 randomly selected data points from the training set.



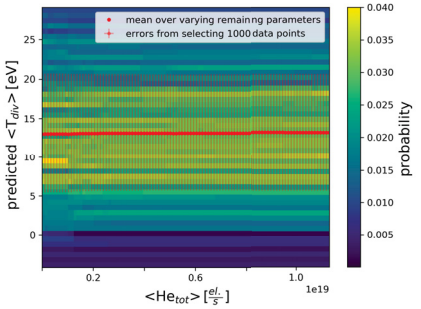
(a) Dependency of random forest predictions using the reduced set of inputs on the elongation of the plasma cros section. The red dots indicate mean and standard deviation of the column-wise normalized distributions.



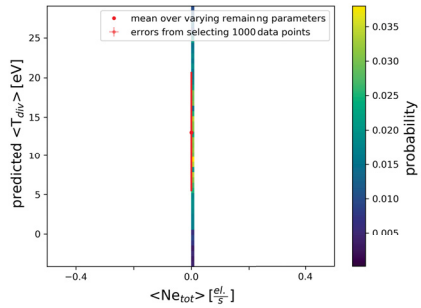
(b) Dependency of random forest predictions using the reduced set of inputs on the hydrogen throughput. The red dots indicate mean and standard deviation of the column-wise normalized distributions.



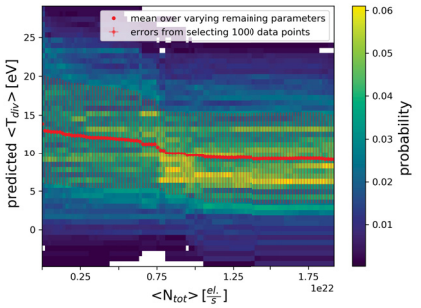
(c) Dependency of random forest predictions using the reduced set of inputs on the deuterium throughput. The red dots indicate mean and standard deviation of the column-wise normalized distributions. [59]



(d) Dependency of random forest predictions using the reduced set of inputs on the helium throughput. The red dots indicate mean and standard deviation of the column-wise normalized distributions.

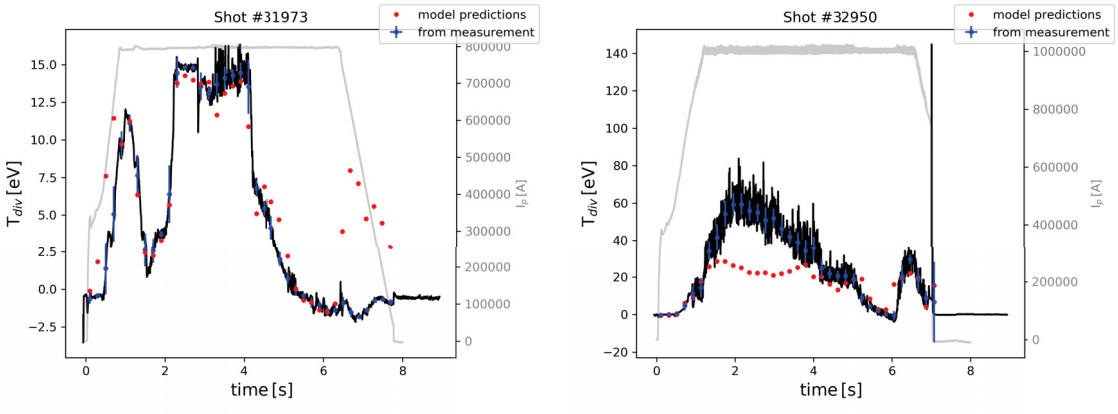


(e) Dependency of random forest predictions using the reduced set of inputs on the neon throughput. The red dots indicate mean and standard deviation of the column-wise normalized distributions.



(f) Dependency of random forest predictions using the reduced set of inputs on the nitrogen throughput. The red dots indicate mean and standard deviation of the column-wise normalized distributions.

Figure 5.11: Dependencies of the random forest model using the reduced set of input quantities on the various input quantities. The depicted distributions were obtained by inserting the given value of the quantity on the abscissa into 1000 randomly selected data points from the training set.



(a) [59] (b)

Figure 5.12: Predictions of the random forest model (red dots) for some exemplary discharges selected from the test data. The black line shows the measured T_{div} curve and the blue points indicate the extracted averages over 0.2 s (without applying any cuts to the data). The standard deviations stem from the variation within 0.2 s. The gray curve shows the plasma current.

accurately than the RF with all inputs despite removing the radiated power, which could be expected to be of major importance in this scenario of impurity injection, from the set of inputs.
The results for the second exemplary discharge show again that the model predictions are limited and can not exceed the limit of 30 eV imposed on T_{div} in this analysis. Except for this limitation, the model predictions are rather accurate and mostly lie within one to two standard deviations from the expected value, comparable to the results obtained for the RF with all inputs.

5.1.3 *Conclusion*

The results presented above show that both approaches, using RFs and the full set of input quantities and using a reduced set of inputs, yield a reasonable performance of the resulting model. However, the model utilizing all input quantities of table 4.2 showed a slightly better performance in terms of absolute differences between model predictions and expected target values.
It was found that the model predictions in both cases primarily depend on the plasma current, the lower triangularity and either the density of neutral particles in the divertor or the deuterium throughput. When removing the neutral particle density in the divertor from the set of input quantities, it was observed that the deuterium throughput takes on a similar role for the model as the neutral density. This and the result that the model predictions depend linearly on the plasma current are in accordance with other analyses. Furthermore, the RF based approach is computationally very cheap. Training and evaluating the models could all be done within a few minutes. Moreover, the optimized

model only requires about 50 MB to be stored. Thus, the RF based approach results in a very inexpensive, yet quite accurate, model for divertor power load predictions.

5.2 NEURAL NETWORKS FOR DIVERTOR POWER LOAD PREDICTION

This section presents the details and results of training *Neural Network* (NN) based models on the task of predicting divertor power loads. Parts of this section have been published in [59].

The NN models presented here have been set up and evaluated using TensorFlow [73] version 1.13.1 and Keras [74] version 2.2.4.

5.2.1 *Neural Networks based on all input quantities*

The baseline NN presented in this thesis is a fully connected NN as introduced in section 3.2.1. The network consists of 3 hidden layers with 100 neurons in each layer and employs *Exponential Linear Unit* (ELU) activation functions in the hidden layers and a linear activation function in the single output neuron. With this network structure the NN has about 22000 parameters. An L1 regularization term with a constant prefactor of 0.01 was added to the *Mean Squared Error* (MSE) loss term for the model optimization. This was based on the observation of cross correlations among input quantities and the result that the RF models only show dependencies on few inputs. Thus, the L1 regularization term was used to reduce the complexity of the model by inducing sparsity in the model parameters. The weights of the model were randomly initialized with a He uniform initialization scheme [75] in the hidden layers and a Glorot uniform initialization scheme [76] in the output layer. The Adagrad [77] optimizer was used to train the NN on the training data and the size of mini batches was set to 30.
To assess the evolution of the model performance during training, 30% of the training data points were held out as a validation set. This limits the number of data points used for optimizing the model to $\sim$ 35000 as compared to $\sim$ 50000 used for the RF models.
The neural networks were trained for 2000 epochs. To estimate the influence of the random initialization of the model's weights on the model performance, the NN was trained three times. From these three training processes the minimum loss on the validation data was obtained. This resulted in values of 12.39 eV^2, 12.35 eV^2 and 12.45 eV^2. These values are only the contribution of the MSE term. From these three iterations, the one with the smallest minimum validation loss was chosen for the further analysis.
The evolution of the selected model's loss as a function of training epochs is depicted in figure 5.13. The figure shows the MSE on the training data in red and on the validation data in blue, i.e. the contribution of the regularization term has been subtracted. The figure indicates that the model's performance on both, the training data and the validation data, improved significantly in the first $\sim$ 250 epochs of training and kept improving slightly after that. The difference between training and validation loss shows

that the model performs better on the training data, as should be expected.

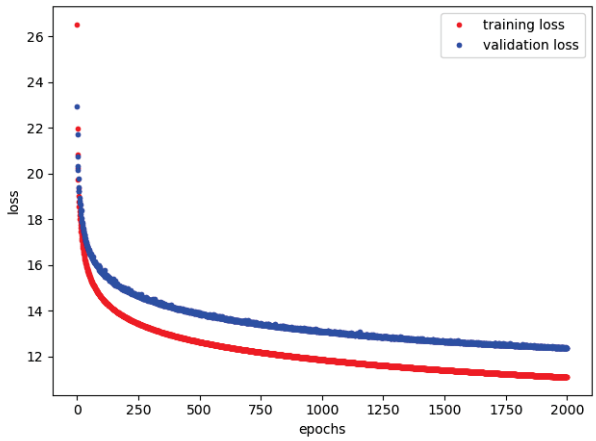


Figure 5.13: Evolution of the neural network loss during the training process. The red dots indicate the loss on the training data and the blue dots that on the validation data. Depicted is only the MSE term of the loss without the regularization term. [59]

Evaluating this model in its state of minimum validation loss on the test data results in the distribution of predictions shown in figure 5.14. Again, the figure shows the model predictions versus the T_{div} values obtained from the experiment and the blue line indicates where the model predictions would perfectly fit the experimental values. As is the case for the RF models, the NN model manages to achieve a trend of matching the expected values. However, in comparison to the results of the RF with all input quantities, figure 5.4, the distribution obtained from the NN is less focused around the line of a 1:1 relation. This also shows in the median absolute difference between model predictions and target values. For the RF with all inputs this value was $1.8^{+3.1}_{-1.3}$ eV. In contrast to this, the NN yields $2.3^{+3.1}_{-1.6}$ eV. This indicates that the baseline NN presented here tends to perform worse than actually both RF models presented before. However, the model performances, with regard to the median absolute difference between model predictions and expected target value, are still comparable within the uncertainties.

As for the RF models, the learned dependencies of the NN model were analyzed as well. To this end, the partial dependencies were determined as presented in section 5.1.1. The resulting distributions are shown in figures 5.15, 5.16 and 5.17.
The NN, like the RF models, shows a linear dependency on the plasma current, again matching the expectation from other analyses. Also a similar dependency on the neutral particle density in the divertor to that of the corresponding RF model can be observed. For the NN this dependency is even more pronounced than for the RF and also matches the expectation of a reduction in T_{div} with increasing neutral particle density.
A first noteworthy difference between the dependencies of the NN model and the RF models is that the NN shows dependencies on quantities that have little to no influence

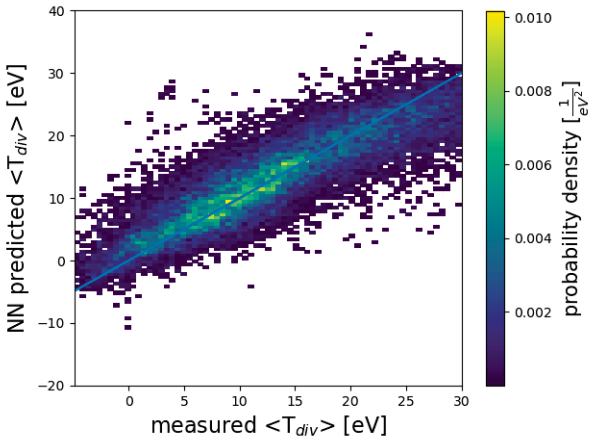
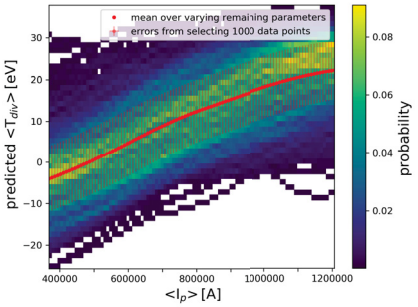
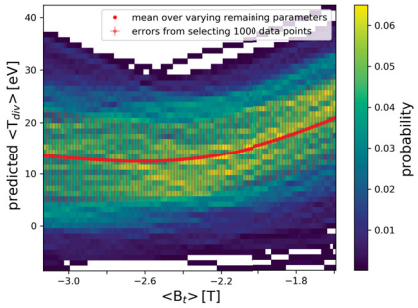


Figure 5.14: Predictions of the NN model versus T_{div} values from the test data. The blue line indicates where the model predictions would perfectly fit the data obtained from the experiments. [59]

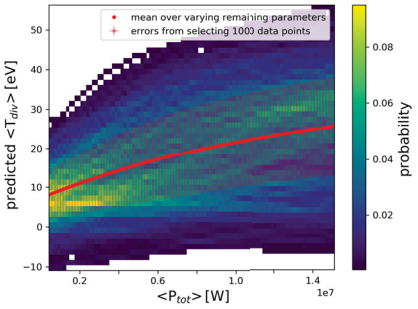
on the predictions of the RF models. For example, the NN shows a clear dependency on the energy stored in the plasma, $<W_{MHD}>$, and on the nitrogen throughput. Both of these quantities seem to have no significant influence on the predictions of the RF model (see figures 5.6(b) and 5.7(e)). The NN's dependency on the nitrogen throughput tentatively matches the expectation of a reduction in T_{div} with increasing nitrogen throughput, that would lead to increased radiative losses in the plasma and, thus, reduce the plasma temperature. However, the dependency on the stored energy in the plasma is less intuitive. The change in the dependency at $<W_{MHD}> \approx 350000\,J$ could indicate a change in the physical regime that necessitates a change in the model's dependency. A potential change in the confinement of the core plasma was investigated but does not seem to be the cause of this behavior, see appendix F. A further analysis of this dependency might be justified. The tendency of reduced T_{div} predictions of the NN at larger values of the electron density in the plasma edge, $<H-5>$, is also in accordance with expectations from the TPM, see equation 2.17, and was not found for the RF model with all input quantities. Furthermore, the NN shows a different dependency on the lower triangularity than the RF model. These different dependencies are possibly caused by the fact that the NN does not allow to reduce the influence of input quantities on the model as easily and efficiently as the RF. Therefore, the NN might capture spurious dependencies omitted by the RF. The dependency of the model on the neon throughput could, again, not be investigated since the data did not contain a significant amount of data points with non-zero neon throughput. However, considering that the models investigated do not show a significant dependency on the hydrogen and helium throughputs, which were non-zero in only 1207 and 1023 data points, respectively, a similar result could be expected for the helium throughput. In all cases the lack of significant data hampers the models' ability to extract significant dependencies from the data.



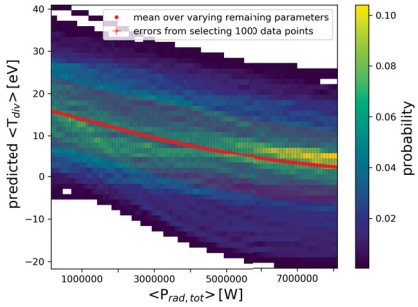
(a) Dependency of neural network predictions using all inputs on the plasma current. The red dots indicate mean and standard deviation of the column-wise normalized distributions. [59]



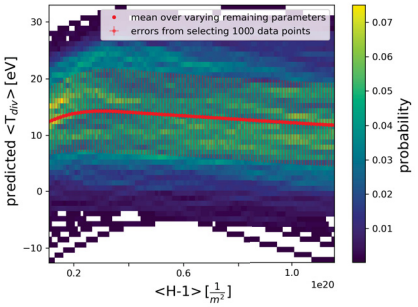
(b) Dependency of neural network predictions using all inputs on the strength of the toroidal magnetic field. The red dots indicate mean and standard deviation of the column-wise normalized distributions.



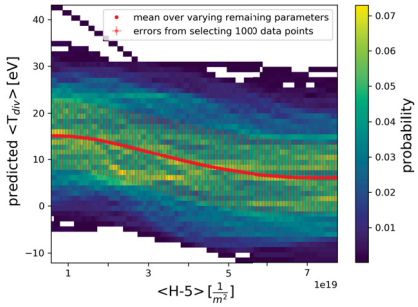
(c) Dependency of neural network predictions using all inputs on the deposited heating power. The red dots indicate mean and standard deviation of the column-wise normalized distributions.



(d) Dependency of neural network predictions using all inputs on the radiated power above the x-point. The red dots indicate mean and standard deviation of the column-wise normalized distributions.

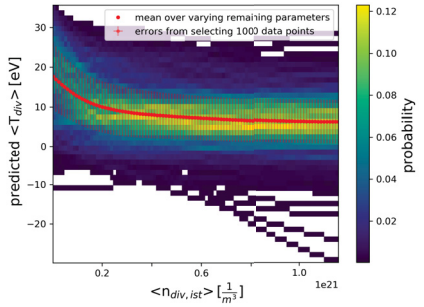


(e) Dependency of neural network predictions using all inputs on the line-integrated electron density in the core. The red dots indicate mean and standard deviation of the column-wise normalized distributions.

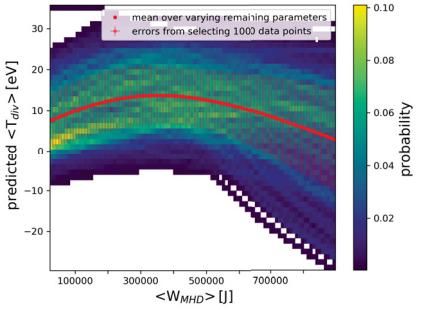


(f) Dependency of neural network predictions using all inputs on the line-integrated electron density in the edge region. The red dots indicate mean and standard deviation of the column-wise normalized distributions.

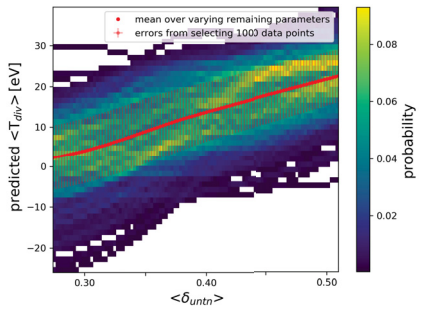
Figure 5.15: Dependencies of the neural network model using all input quantities on the various input quantities. The depicted distributions were obtained by inserting the given value of the quantity on the abscissa into 1000 randomly selected data points from the training set.



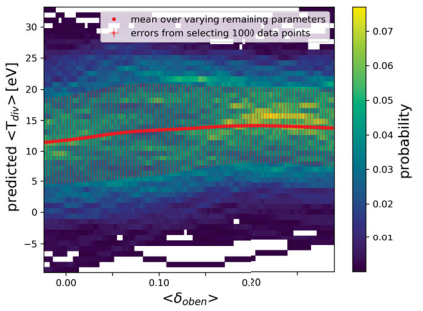
(a) Dependency of neural network predictions using all inputs on the neutral particle density in the divertor. The red dots indicate mean and standard deviation of the column-wise normalized distributions. [59]



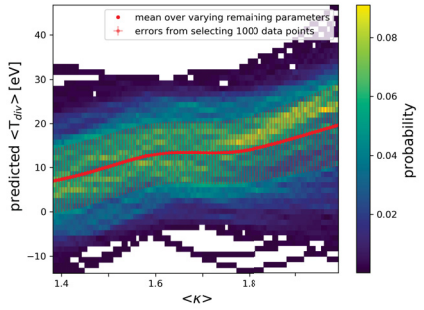
(b) Dependency of neural network predictions using all inputs on the stored energy in the plasma. The red dots indicate mean and standard deviation of the column-wise normalized distributions.



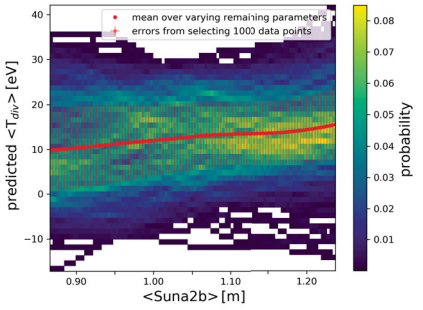
(c) Dependency of neural network predictions using all inputs on the lower triangularity. The red dots indicate mean and standard deviation of the column-wise normalized distributions. [59]



(d) Dependency of neural network predictions using all inputs on the upper triangularity. The red dots indicate mean and standard deviation of the column-wise normalized distributions.

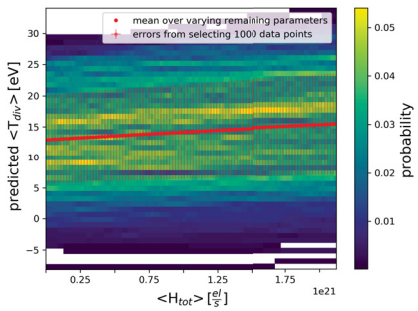


(e) Dependency of neural network predictions using all inputs on the elongation of the plasma cross section. The red dots indicate mean and standard deviation of the column-wise normalized distributions. [59]

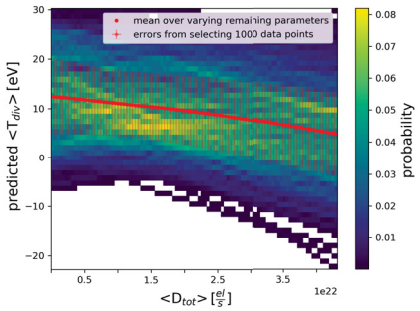


(f) Dependency of neural network predictions using all inputs on the position of the strike line. The red dots indicate mean and standard deviation of the column-wise normalized distributions.

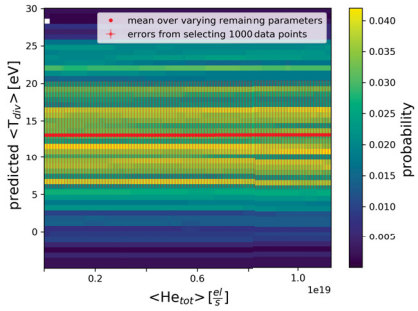
Figure 5.16: Dependencies of the neural network model using all input quantities on the various input quantities. The depicted distributions were obtained by inserting the given value of the quantity on the abscissa into 1000 randomly selected data points from the training set.



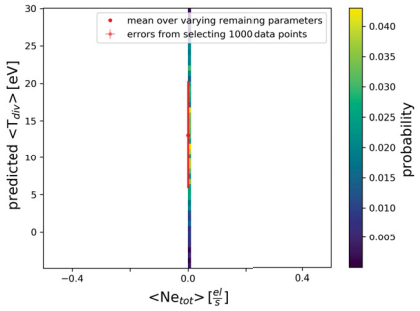
(a) Dependency of neural network predictions using all inputs on the hydrogen throughput. The red dots indicate mean and standard deviation of the column-wise normalized distributions.



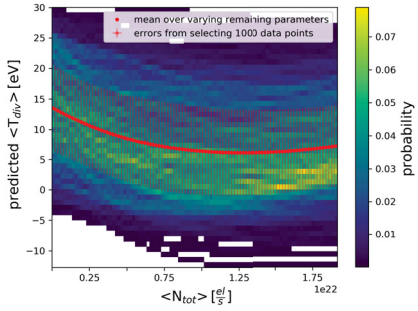
(b) Dependency of neural network predictions using all inputs on the deuterium throughput. The red dots indicate mean and standard deviation of the column-wise normalized distributions. [59]



(c) Dependency of neural network predictions using all inputs on the helium throughput. The red dots indicate mean and standard deviation of the column-wise normalized distributions.



(d) Dependency of neural network predictions using all inputs on the neon throughput. The red dots indicate mean and standard deviation of the column-wise normalized distributions.

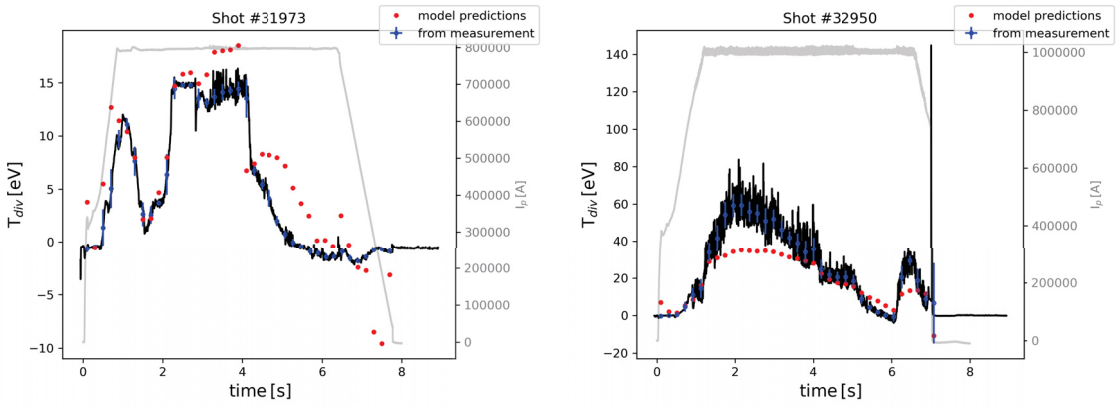


(e) Dependency of neural network predictions using all inputs on the nitrogen throughput. The red dots indicate mean and standard deviation of the column-wise normalized distributions. [59]

Figure 5.17: Dependencies of the neural network model using all input quantities on the various input quantities. The depicted distributions were obtained by inserting the given value of the quantity on the abscissa into 1000 randomly selected data points from the training set.

Another important difference between the RF models and the NN model is also visible in these plots. The NN can predict T_{div} values beyond the limits imposed on the data. This might make this model more interesting than the RF for extended analyses, provided that predictions outside the range of the training data ideally come with an estimate of the model's uncertainty.

An evaluation of the NN model on the same exemplary discharges as the RF, once again disregarding the data selection criteria, is shown in figure 5.18. Figure 5.18(a) shows that the NN predictions deviate more significantly from the values from the experiment than the RF predictions. However, the NN also manages to capture the falling trend from the 4 s mark onwards, even if the absolute values of the model predictions do not match the experimental values. This result is in line with the observation that the NN tends to perform worse on the actual test data, as indicated by figure 5.14. In contrast to this, figure 5.18(b) shows a similar accuracy of the NN in comparison to the results of the RF. The predictions between ~ 2 s and ~ 4 s also indicate that the NN can predict values beyond the limit of $T_{div} = 30$ eV in contrast to the RF models.



(a) [59] (b)

Figure 5.18: Predictions of the neural network model (red dots) for some exemplary discharges selected from the test data. The black line shows the measured T_{div} curve and the blue points indicate the extracted averages over 0.2 s (without applying any cuts to the data). The standard deviations stem from the variation within 0.2 s. The gray curve shows the plasma current

Hyperparameter Variation

As the NN has a lot of hyperparameters determining the model's structure and its performance, the effects of varying some of these parameters were analyzed in an additional study. To this end, different configurations of NNs were trained on the same training data as before and their performances will be compared with regard to the validation data set in the following. Again, every configuration was trained three times and from each of

these three iterations the minimum loss on the validation data was determined. To finally compare different NN configurations, the average of these three values will be compared.

The hyperparameters varied in this study are the number of layers in the NN, the number of neurons in each layer, the constant of the regularization term, the size of the mini batches and the activation function applied in the hidden layers. On top of this, the aforementioned technique of dropout (see the end of section 3.2.1) as well as batch normalization [78] were tested in this analysis. To test all these different settings the Optuna library [79] was used.

For this study, the values of the hyperparameters under investigation were chosen at random within set bounds. The boundary values for numerical parameters as well as the set of possible selections for categorical parameters are given in table 5.1. Note that the number of neurons was not limited to be the same in every layer. The selection of possible activation functions applied in the hidden layers contains the aforementioned ELU and *Rectified Linear Unit* (ReLU) activation functions as well as the *Scaled Exponential Linear Unit* (SELU) [80] and the leaky ReLU activation functions. The leaky ReLU activation function is similar to the ReLU activation function except for a constant prefactor if the input to the function is negative. This is supposed to allow for small gradients for negative inputs to the function instead of completely cutting off the gradient. The SELU activation function is equal to the ELU activation function times a prefactor. This factor is chosen so that the mean and variance of the inputs to the function do not shift between layers. This is supposed to address the same problem as batch normalization, namely the change in statistical properties of the inputs while passing the layers of the network. Thus, in this test, batch normalization was never used in combination with the SELU activation function.

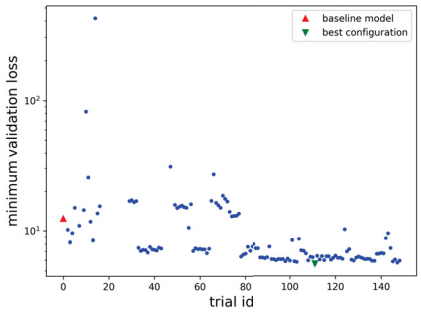
Parameter	Values
Number of layers	[2, 7]
Number of neurons	[10, 200]
Constant of regularization term	[0.001, 0.1]
Mini batch size	[20, 200]
Activation function	(ELU, SELU, ReLU, Leaky ReLU)
Batch normalization	(yes, no)
Dropout	(yes, no)
Dropout rate	[0.2, 0.7]

Table 5.1: Hyperparameters varied in the study. The boundary values for numerical parameters and the set of possible selections for categorical parameters are given.

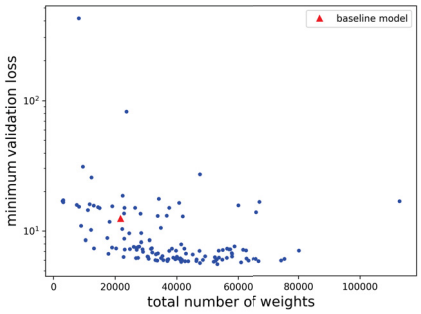
The results of this variation of hyperparameters are depicted in figures 5.19 and 5.20. Figure 5.19(a) shows the resulting averages of minimum validation losses for the different NN configurations tested. The red triangle indicates the value obtained with

the baseline model presented above. The tendency of a saturating loss with increasing number of tested configurations that can be observed is due to the way Optuna selects the parameters for every configuration. This parameter selection process is based on a tree-structured Parzen estimator [81] and, thus, utilizes previous observations to find optimal parameters. The further results indicate that a small constant of the regularization term, the use of ReLU activation functions, batch normalization and no dropout tend to be beneficial for the model's performance. The beneficial effect of the ReLU activation function might be essentially caused by the chosen initialization scheme. As the He initialization is specifically designed for rectified activation functions, such as ReLU and leaky ReLU, the improvement in model performance could be caused by the interplay of activation function and initialization. Including the initialization scheme as a further parameter in the study of hyperparameters of the NN could be a worthwhile extension to the presented analysis.

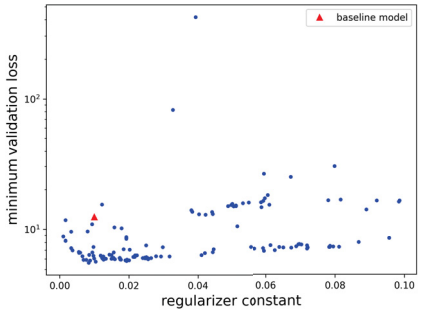
The best configuration in this analysis, according to the criterion of averaged minimum validation loss, consists of 4 hidden layers employing ReLU activation functions with 187, 196, 49 and 75 neurons, has a regularization constant of 0.008, a mini batch size of 120, uses batch normalization and no dropout. This configuration was retrained three times, like the baseline model presented above. Of these three training iterations, the model with the smallest minimum loss on the validation data was then evaluated on the test data. The resulting distribution of model predictions versus values extracted from the experiment is shown in figure 5.21. The distribution of model predictions is more focused around the line of a 1:1 relation than that observed for the baseline model (see figure 5.14). This is also supported by the median absolute difference of the model predictions and the experimental values. For the presented model with optimized hyperparameters this value is $1.7^{+2.8}_{-1.2}$ eV. Thus, the model's performance is tentatively better than that of the baseline model but still on a comparable level, taking into account the uncertainties on the median absolute difference. However, the optimiztaion of the model's hyperparameters also improved the accuracy of the NN to the level of that of the RF model.



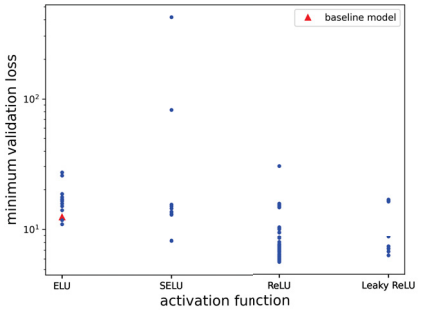
(a) Average minimum validation loss of different neural network configurations. The trial id enumerates the configuration.



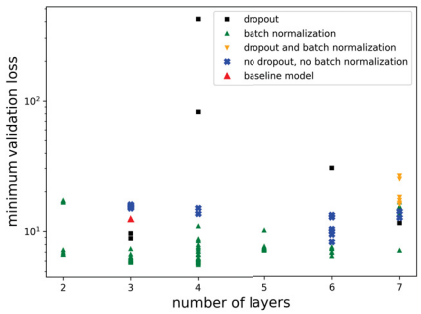
(b) Average minimum validation loss as a function of the total number of weights in any given neural network configuration.



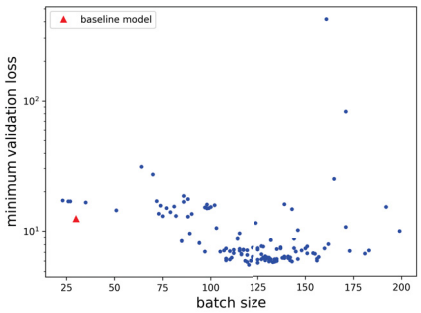
(c) Average minimum validation loss as a function of the constant of the regularization term in any given neural network configuration.



(d) Average minimum validation loss as a function of the activation function applied in the hidden layers of any given neural network configuration.

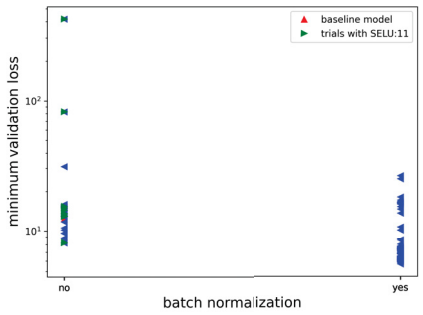


(e) Average minimum validation loss as a function of the total number of layers in any given neural network configuration. The colors and marker styles indicate which configuration applied batch normalization and/or dropout.

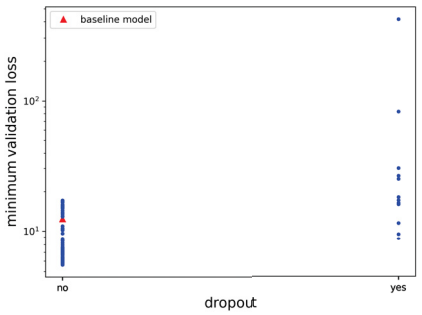


(f) Average minimum validation loss as a function of the size of the mini batches in any given neural network configuration.

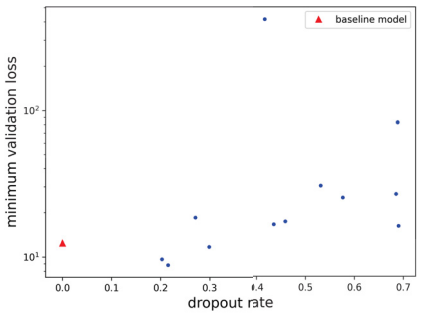
Figure 5.19: Comparison of different hyperparameter selections. Depicted is the average of the minium validation losses of three training processes for a given neural network configuration. The red triangle indicates the minimum validation loss of the baseline model presented above.



(a) Average minimum validation loss of different neural network configurations and whether or not these configurations used batch normalization. Marked in green are configurations with SELU activation functions as batch normalization was always turned off for these.



(b) Average minimum validation loss of different neural network configurations and whether or not these configurations used dropout.



(c) Average minimum validation loss as a function of the dropout rate. Only the results from neural networks that used dropout and the result from the baseline model are depicted.

Figure 5.20: Comparison of different hyperparameter selections. Depicted is the average of the minium validation losses of three training processes for a given neural network configuration. The red triangle indicates the minimum validation loss of the baseline model presented above.

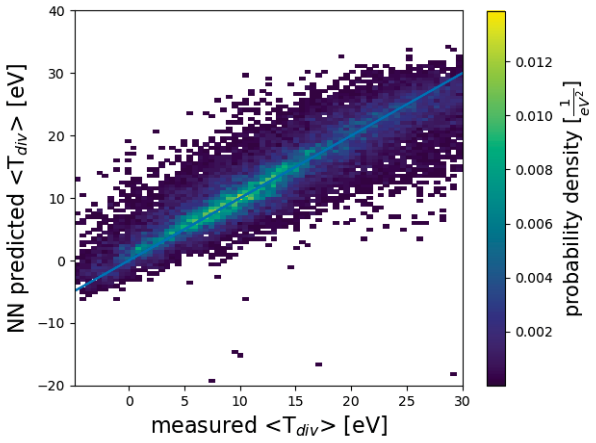


Figure 5.21: Predictions of the NN model with optimized hyperparameters versus T_{div} values from the test data. The blue line indicates where the model predictions would perfectly fit the data obtained from the experiments.

5.2.2 *Neural Networks based on a reduced set of input quantities*

The NN based approach was also tested with the reduced set of bold-faced input quantities of table 4.2. The baseline model for this approach is the same NN as before, i.e. a network with three hidden layers with 100 neurons each, ELU activation functions in the hidden layers and a linear activation function in the output neuron. Due to the reduced number of inputs, this NN has 600 neurons less than the baseline model utilizing all input quantities. The same loss function and optimizer were used to train the model, namely a loss function consisting of the MSE term and an additional L1 regularization term and the Adagrad optimizer. All other properties were chosen to be the same as for the baseline NN presented in the last section.

This NN was trained in the same way as the baseline model before. So, the NN was optimized three times on the same training data with 30% of the data held out as additional validation set. Then the best model was chosen according to the minimum validation loss achieved among the three training iterations and this model was evaluated on the separate test data in its state of minimum validation loss.

The evolution of the chosen model's loss as a function of training epochs is depicted in figure 5.22. The red data points show the model's loss on the training data and the blue dots show the loss on the validation data. Only the value of the MSE term of the loss function is depicted. In comparison to the NN with all input quantities, the overall value of the loss is larger. This already indicates that this model's performance is worse than that of the NN with all inputs. Again, a gap between training and validation data can be observed and should be expected.

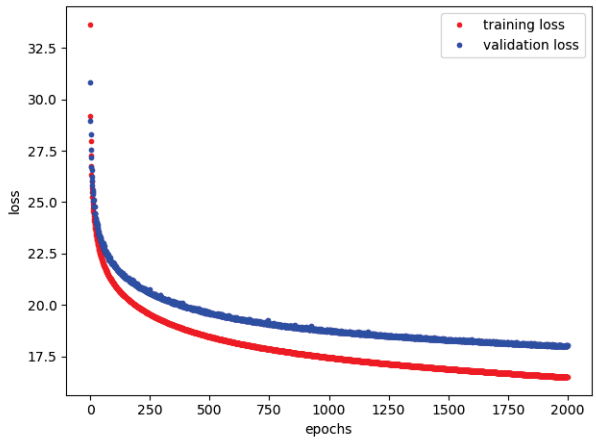


Figure 5.22: Evolution of the neural network loss during the training process. The red dots indicate the loss on the training data and the blue dots that on the validation data. Depicted is only the MSE term of the loss without the regularization term. [59]

The model's performance, evaluated on the test data, is depicted in figure 5.23. The figure shows the distribution of NN predictions versus the actual values obtained from the experimental data. The blue line indicates a 1:1 relation. In comparison to the results obtained for the NN with all input quantities, see figure 5.14, the distribution of model predictions obtained from the approach with a reduced set of input quantities is less focused around the line of a 1:1 relation and shows more significant deviations from this line. Overall, the model still shows a tendency to match the desired results obtained from the experiment, but less accurately than the NN with all inputs. This is confirmed by the median absolute difference between model predictions and experimental values. For the NN using the reduced set of inputs this value amounts to $2.7^{+3.8}_{-1.9}$ eV.

An investigation of the learned dependencies of the model shows some differences caused by the reduction of input quantities. The partial dependencies of this NN using the reduced set of input quantities are shown in figures 5.25 and 5.26. First, the model still depends linearly on the plasma current. This dependency was observed for all previously presented models and is in accordance with the expectation from physics analyses and the TPM. However, notable differences to the dependencies obtained from the NN with all inputs can be observed as well. A difference already noted in the analysis of the RF models is that removing the neutral particle density in the divertor from the set of inputs causes the deuterium throughput to take over the role of the neutral particle density. This tendency also shows in the analysis of the NN dependencies when comparing figures 5.16(a), 5.17(b) and 5.26(c) and is also in accordance with physics analyses and the observation of a strong correlation between deuterium throughput and the neutral particle density, see figure 4.4.
A further difference between the observed dependencies of the NN models shows in the dependency on the line-integrated electron density in the core, <H-1 >. The NN using

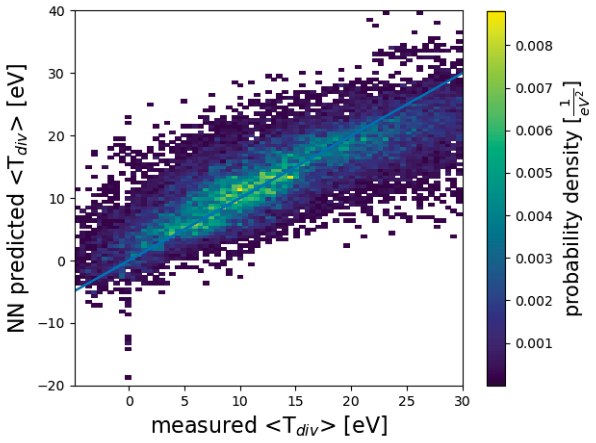
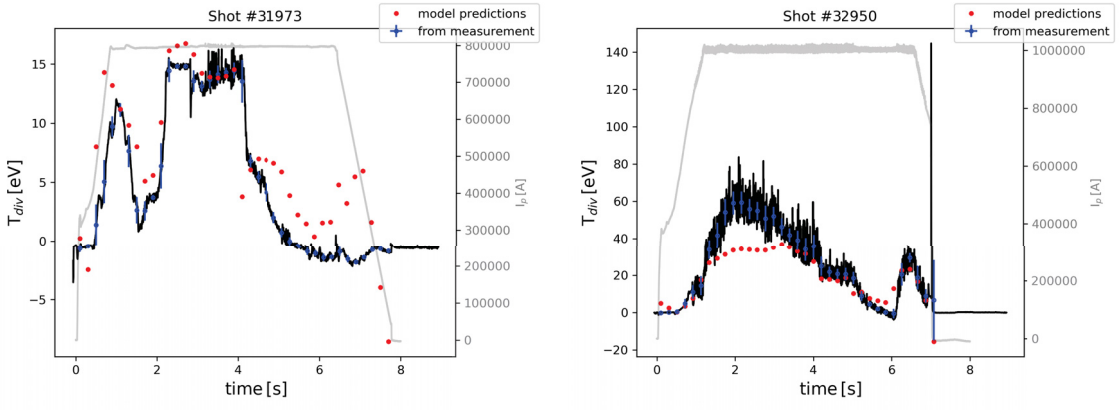


Figure 5.23: Predictions of the neural network model versus T_{div} values from the test data. The blue line indicates where the model predictions would perfectly fit the data obtained from the experiments. [59]

all inputs does not show a significant dependency on this parameter. In contrast to this, the NN with the reduced set of inputs shows a dependency on the core density more closely resembling the results shown in figure 2.3.
The dependency on the lower triangularity, $< \delta_{untn} >$, also differs between the NN with all inputs and the one with the reduced set of inputs. For the model with a reduced set of inputs a similar dependency to that of the RF models can be observed. As stated before, this dependency on the lower triangularity is not quite clear and could be caused by cross correlations of the lower triangularity and the heating power.
Further similarities of the NN models show in the dependencies on the elongation of the plasma cross section and the nitrogen throughput. Both models show decreasing predictions with increasing nitrogen throughput, as could be expected. The dependency on the elongation is less obvious. Since the training data mainly contains data around an elongation of ~ 1.7, see figure D.2(f), the learned dependency outside this area might be spurious. Other analyses suggest that $\lambda_q \propto (1+\kappa^2)^{1.2}$ [10] and, thus, $q_{target} \propto (1+\kappa^2)^{-1.2}$, at least for an attached plasma. From the TPM it would follow that $T_t \propto (1+\kappa^2)^{-12/7}$, which is not the dependency the NN models show.
The lack of dependencies on the hydrogen, helium and neon throughputs is, once again, caused by the small amount of available data in the data base.

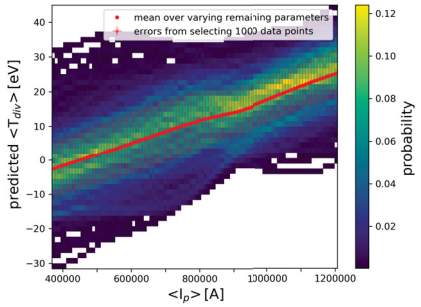
To further evaluate the model's performance, it was tested on the same exemplary discharges as the RF and the NN models presented before. The results of two of these tests are shown in figure 5.24. Again, the black line shows the measured signal of T_{div}, the blue dots are averages of this signal over 0.2 s with the standard deviation within these 0.2 s indicated by the error bars. The red dots show the model's predictions and the gray line depicts the plasma current. In this test, as before, the selection criteria were omitted for the test data. Overall, the results confirm the observation that the NN using

the reduced set of inputs performs worse than the model with all inputs. This can be seen in figure 5.24(a). Many of the predictions of the NN deviate more from the values extracted from the experiment than those of the NN with all inputs. Interestingly, the 8 data points in the plateau region of T_{div} between 2 s to 4 s seem to reverse their roles when comparing the results of the NN with all inputs with those of the NN with reduced inputs. The data points where the predictions of the NN with all inputs were bad now show a better result for the NN with reduced inputs and vice versa. Investigating the cause for this might shed some light on which inputs should be included to optimize this behavior. Both NN approaches predict the falling tendency of T_{div} in discharge number 31973 from the 4 s mark onwards. However, both models yield predictions that significantly deviate from the expectation. Figure 5.24(b) shows a similar result to that of the NN with all inputs. The predictions from $\sim$ 2 s to $\sim$ 4 s deviate from the expectation due to the limit imposed on T_{div} in the training data. Nevertheless, the NN is capable of predicting values above the limit of 30 eV.

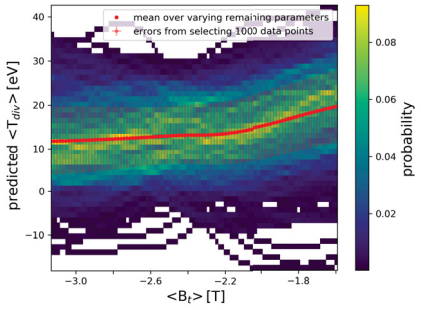


(a) [59] (b)

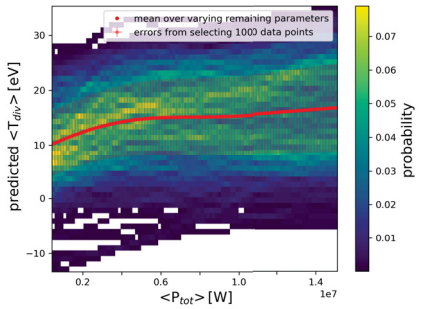
Figure 5.24: Predictions of the neural network model using the reduced set of inputs (red dots) for some exemplary discharges selected from the test data. The black line shows the measured T_{div} curve and the blue points indicate the extracted averages over 0.2 s (without applying any cuts to the data). The standard deviations stem from the variation within 0.2 s; the gray curve shows the plasma current.



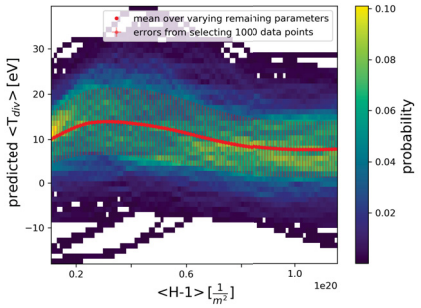
(a) Dependency of neural network predictions using the reduced set of inputs on the plasma current. The red dots indicate mean and standard deviation of the column-wise normalized distributions. [59]



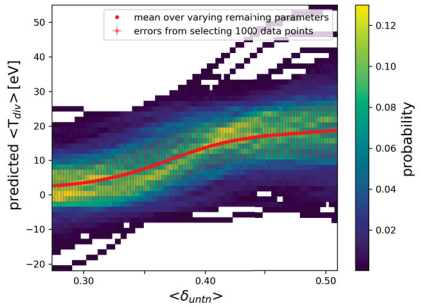
(b) Dependency of neural network predictions using the reduced set of inputs on the strength of the toroidal magnetic field. The red dots indicate mean and standard deviation of the column-wise normalized distributions.



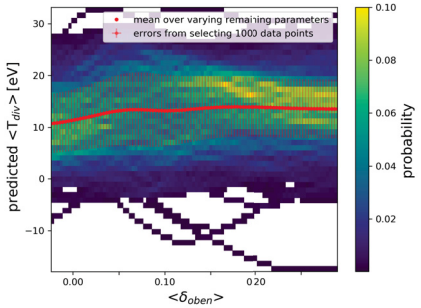
(c) Dependency of neural network predictions using the reduced set of inputs on the deposited heating power. The red dots indicate mean and standard deviation of the column-wise normalized distributions.



(d) Dependency of neural network predictions using the reduced set of inputs on the line-integrated electron density in the core. The red dots indicate mean and standard deviation of the column-wise normalized distributions.

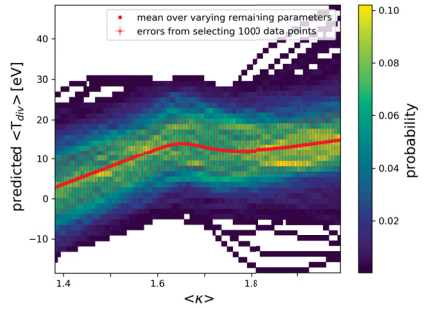


(e) Dependency of neural network predictions using the reduced set of inputs on the lower triangularity. The red dots indicate mean and standard deviation of the column-wise normalized distributions. [59]

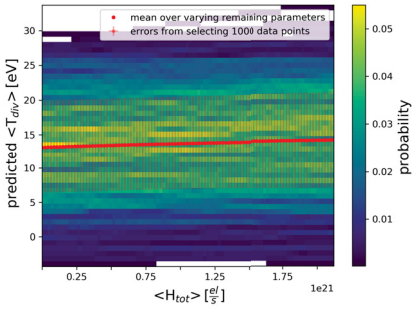


(f) Dependency of neural network predictions using the reduced set of inputs on the upper triangularity. The red dots indicate mean and standard deviation of the column-wise normalized distributions.

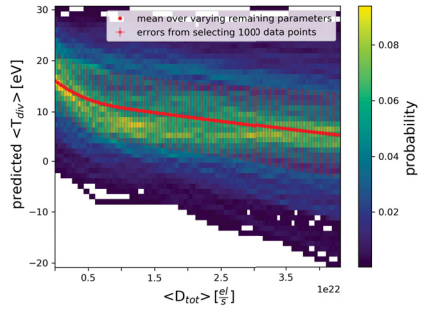
Figure 5.25: Dependencies of the neural network model using the reduced set of input quantities on the various input quantities. The depicted distributions were obtained by inserting the given value of the quantity on the abscissa into 1000 randomly selected data points from the training set.



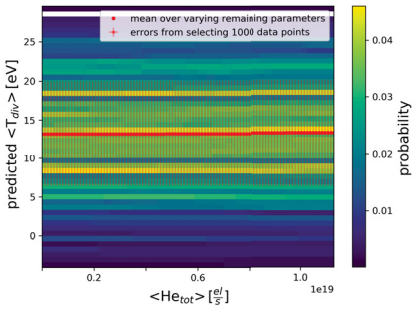
(a) Dependency of neural network predictions using the reduced set of inputs on the elongation of the plasma cross section. The red dots indicate mean and standard deviation of the column-wise normalized distributions. [59]



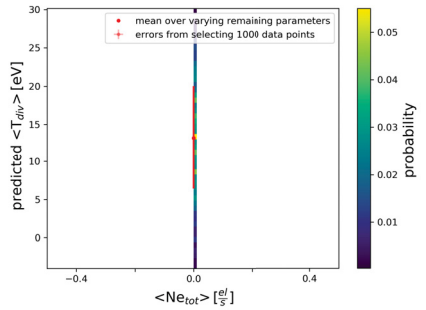
(b) Dependency of neural network predictions using the reduced set of inputs on the hydrogen throughput. The red dots indicate mean and standard deviation of the column-wise normalized distributions.



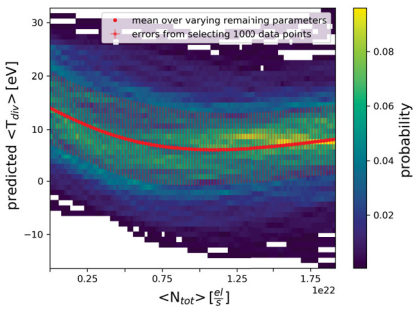
(c) Dependency of neural network predictions using the reduced set of inputs on the deuterium throughput. The red dots indicate mean and standard deviation of the column-wise normalized distributions. [59]



(d) Dependency of neural network predictions using the reduced set of inputs on the helium throughput. The red dots indicate mean and standard deviation of the column-wise normalized distributions.



(e) Dependency of neural network predictions using the reduced set of inputs on the neon throughput. The red dots indicate mean and standard deviation of the column-wise normalized distributions.

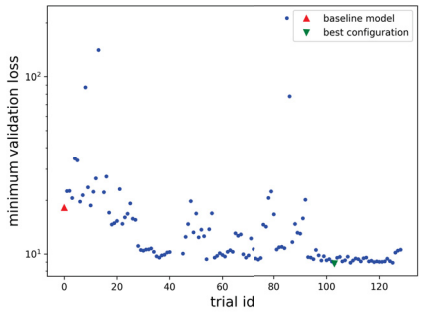


(f) Dependency of neural network predictions using the reduced set of inputs on the nitrogen throghput. The red dots indicate mean and standard deviation of the column-wise normalized distributions. [59]

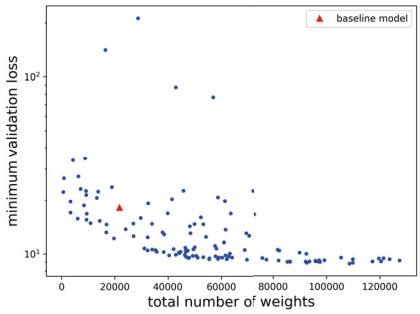
Figure 5.26: Dependencies of the neural network model using the reduced set of input quantities on the various input quantities. The depicted distributions were obtained by inserting the given value of the quantity on the abscissa into 1000 randomly selected data points from the training set.

Hyperparameter Variation

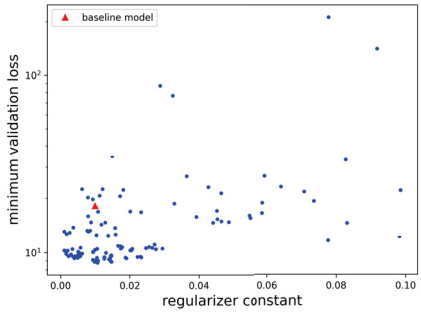
The hyperparameters of the NN with reduced inputs were also varied to analyze the effects of different NN configurations. In this test the same hyperparameters as presented in the analysis of the NN with all inputs were varied, see table 5.1. Again the hyperparameters were selected randomly using the Optuna [79] library. The results of this test are depicted in figures 5.27 and 5.28.
The overall conclusions from this test are similar to those obtained for the NN with all input quantities. The beneficial effect of using ReLU activation functions can be observed in figure 5.27(d). Once again, the cause for this might be the selected initialization scheme. Furthermore, a small constant of the regularizing term seems to improve the model's performance, as does the use of batch normalization. With this reduced set of inputs the difference in model performance between models with dropout and models without dropout is less significant than is the case with all input quantities. However, comparing the total number of weights in the model, the approach with the reduced set of inputs seems to require more than twice the number of weights of the optimal NN configuration with all inputs.
The observed trends can be confirmed by identifying the optimal NN configuration among those tested. With the reduced set of inputs the optimal configuration consists of 6 hidden layers with 141, 139, 148, 198, 156 and 37 neurons, corresponding to a total of $\sim$ 108000 weights. This configuration uses ReLU activation functions in all hidden layers. Moreover, this optimal configuration used a regularization constant of 0.01 during the training process, a mini batch size of 164, batch normalization and no dropout. Retraining and evaluating this configuration on the test data resulted in the distribution of model predictions shown in figure 5.29. The distribution of model predictions is less focused around the line of a 1:1 relation with the values extracted from the experiment than in the case of the NN with all inputs. Nevertheless, the overall tendency of matching the expectation is still visible. The number of significant outliers is reduced in comparison to the baseline model with reduced inputs. The median absolute difference between the model predictions and the experimental values amounts to $2.0^{+3.6}_{-1.5}$ eV. Thus, this NN with optimized hyperparameters performs better than the unoptimized version and achieves a slightly better performance than the corresponding RF model with reduced inputs. However, within the uncertainties on the median absolute difference the model performance is still comparable with the results obtained with other models. Nevertheless, the similar performances of the NNs with optimized hyperparameters further suggests that the reduced set of input quantities might suffice to obtain accurate predictions of the power loads at the divertor targets prior to performing an experiment.



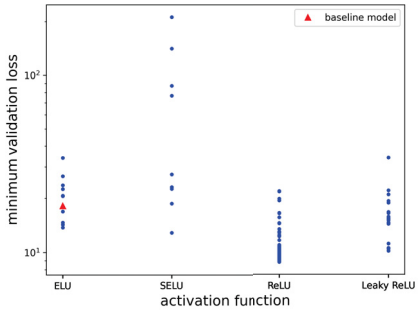
(a) Average minimum validation loss of different neural network configurations. The trial id enumerates the configuration.



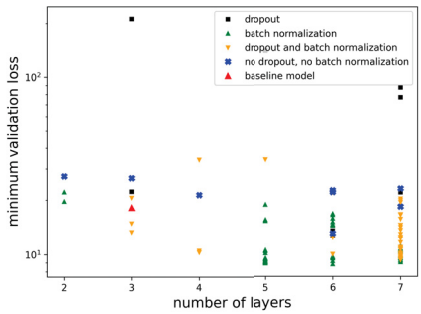
(b) Average minimum validation loss as a function of the total number of weights in any given neural network configuration.



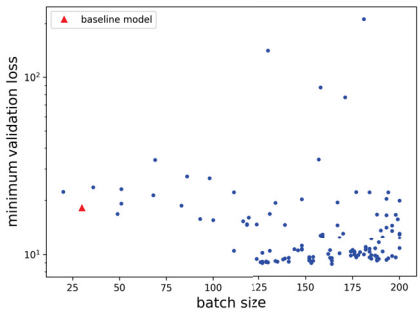
(c) Average minimum validation loss as a function of the constant of the regularization term in any given neural network configuration.



(d) Average minimum validation loss as a function of the activation function applied in the hidden layers of any given neural network configuration.

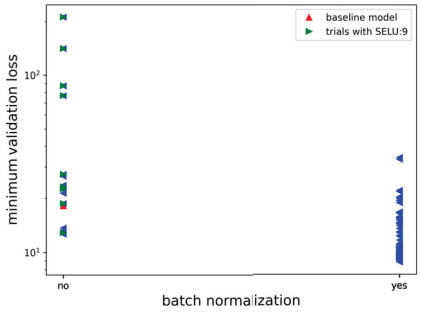


(e) Average minimum validation loss as a function of the total number of layers in any given neural network configuration. The colors and marker styles indicate which configuration applied batch normalization and/or dropout.

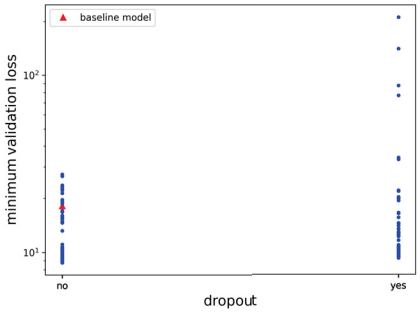


(f) Average minimum validation loss as a function of the size of the mini batches in any given neural network configuration.

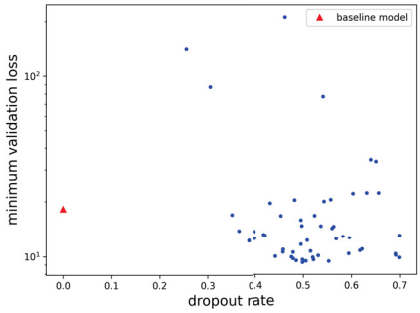
Figure 5.27: Comparison of different hyperparameter selections. Depicted is the average of the minium validation losses of three training processes for a given neural network configuration. The red triangle indicates the minimum validation loss of the baseline model presented above.



(a) Average minimum validation loss of different neural network configurations and whether or not these configurations used batch normalization. Marked in green are configurations with SELU activation functions as batch normalization was always turned off for these.



(b) Average minimum validation loss of different neural network configurations and whether or not these configurations used dropout.



(c) Average minimum validation loss as a function of the dropout rate. Only the results from neural networks that used dropout and the result from the baseline model are depicted.

Figure 5.28: Comparison of different hyperparameter selections. Depicted is the average of the minium validation losses of three training processes for a given neural network configuration. The red triangle indicates the minimum validation loss of the baseline model presented above.

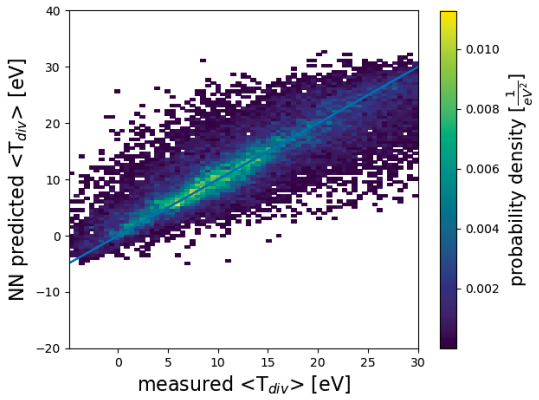


Figure 5.29: Predictions of the NN with optimized hyperparameters versus T_{div} values from the test data. The blue line indicates a perfect match of model predictions and the data.

5.2.3 *Conclusion*

Similar results to those of the RF models have been obtained for an approach based on NNs. Again both, the full set of inputs of table 4.2 and the reduced set of inputs, result in models that manage to generally predict the target quantity rather well.
Some of the observed dependencies of the model are in line with expectations from other analyses. Foremost, the observed linear dependency of the model predictions on the plasma current is in accordance with the expectations from [10] and the TPM. Moreover, a similar behaviour of the dependencies on the deuterium throughput and the neutral particle density in the divertor was observed as for the RF models.
In terms of resources needed, the NN based approach is slightly more expensive than the RF models. Training the NNs took about 2 hours on a single GPU for the model with all inputs and 1.5 hours for the model with the reduced set of inputs. However, evaluating the models only required a few minutes. Thus, obtaining new predictions once the models are set up is comparatively cheap. The final models only require about 200 kB of storage. Hence, the NN based approach, like the RF based approach, results in very inexpensive, yet quite accurate, models for divertor power load predictions.

5.3 MIXTURE DENSTIY NETWORKS FOR DIVERTOR POWER LOAD PREDICTION

This section presents results of training *Mixture Density Network* (MDN) based models on the task of predicting divertor power loads. The application of MDNs is motivated by the fact that NNs commonly only predict the conditional average of the distribution of the target quantity [50] and a more complete view on the predictive distribution might be necessary. For example, the input quantities selected for this analysis might not cover all parameters relevant to the problem of particle and power transpot in the *Scrape-Off Layer* (SOL). Thus, any given input vector might actually result in several possible predictions, thus inducing multimodality in the predictive distribution. This can

not be covered by only obtaining the conditional average of the predictive distribution. Furthermore, by obtaining the full predictive distribution an assessment of the model uncertainty can be included. This would be especially beneficial in parameter regions outside the ones observed during the model training.
Once again, the MDN was trained and tested with both, the full set of inputs and the reduced set of inputs. The primary goals of this analysis are to investigate the potential of predicting divertor power loads with MDNs and to attempt a first assessment of the necessity and benefits of this method for obtaining an approximation of the full predictive distribution.

5.3.1 *Mixture Density Networks based on all input quantities*

The MDN used for this analysis consists of two hidden layers with 100 neurons each. In the hidden layers a tanh activation function is applied. This change in activation function in comparison to the NNs presented before was necessary to limit the value range of activations in order to obtain convergence. As mentioned in the introduction on MDNs, section 3.2.2, the structure and the activation functions of the output layer also needed to be changed in comparison to the aforementioned NNs. In this first test of MDNs the model uses three Gaussian mixture components to approximate the predictive distribution. Thus, the output layer of the MDN consists of 9 neurons. The output neurons predicting the expectation values, μ_i, of the Gaussian components still employ linear activation functions. To predict the mixture coefficients, α_i, the softmax activation function, equation 3.13, is used. Finally, the ELU activation function, with a constant of 1 added to ensure non-negativity, is used to predict the standard deviations, σ_i, of the mixture components. The split in training and test data was chosen to be the same as for the NNs. Again, 30% of the training data was held out during training to be used as validation set. The loss function used to optimize the model weights was set to the negative log likelihood, equation 3.12. No regularization term was included. The same initialization schemes used for the NNs were also used for the MDNs. The optimizer used is the *Root Mean Square Propagation* (RMSProp) optimizer [82]. This optimizer employs an adaptive learning rate by scaling the constant learning rate with the running average of past gradients for every single weight. The size of the mini batches was set to 30 and the MDN was trained for a maximum of 2000 epochs with early stopping of the training in case that the loss on the validation data had not improved over 100 epochs by at least 0.01. This scheme was included to reduce the effects of overfitting.

Like the NNs, the MDN was trained three times to estimate the influence of the random weight initialization. The minimum validation losses achieved in each of the three iterations were 2.42, 2.43 and 2.44. Note that these values are not comparable to the losses of the NN as two different loss functions have been used. Of these three models, the one with the lowest minimum validation loss was evaluated on the test data, again in its state of minimum validation loss.
The evolution of this model's loss on the training data (red) and on the validation data

(blue) during training is depicted in figure 5.30. As can be seen in the figure, the model was not optimized for the maximum 2000 epochs, since the validation loss saturated after about 70 epochs. The gap between the loss on the training data and the loss on the validation data once again indicates a difference in performance on the two data sets, as could be expected.

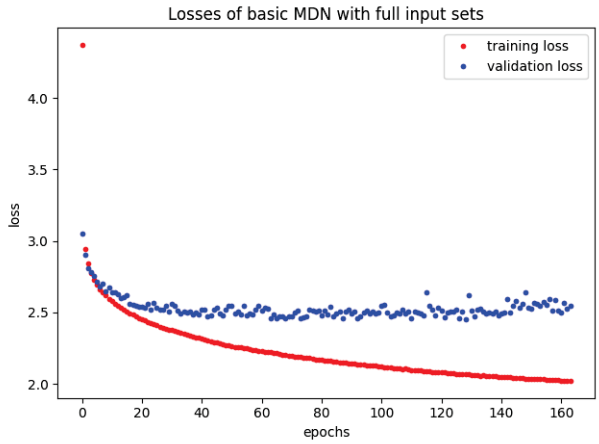


Figure 5.30: Evolution of the mixture density network loss during the training process. The red dots indicate the loss on the training data and the blue dots that on the validation data.

The results obtained by evaluating the model on the test data are depicted in figure 5.31. The model's prediction of T_{div} for any given input vector, $\mathbf{x}$, was taken to be the expectation value of the mixture component with the strongest mixing coefficient $\alpha_i(\mathbf{x})$. This prevents predictions that would correspond to sparsely or even completely unpopulated areas of the predictive distribution and might, thus, be unphysical solutions. The figure shows that the model predictions follow the trend of a 1:1 relation, indicated by the blue line. The median absolute difference of model predictions and values from the experiment was determined to be $2.1^{+3.2}_{-1.5}$ eV. This indicates a slightly better performance than the baseline NN model with all inputs. Generally, however, the two models show a comparable performance within the uncertainties on the median absolute difference.

To analyze the necessity for this approach of modeling the full predictive distribution, the distribution of mixture coefficients on the test data was analyzed. The resulting mixture coefficients, $\alpha_i(\mathbf{x})$, for each data point $\mathbf{x}$ in the test data are depicted in figures 5.32(a) and 5.32(b). The latter shows the same distribution as the first but with removed outermost bins. The distribution indicates that the third component mostly has a weak influence on the model as most data points result in small values of α_3. Thus, two components could suffice to model the problem at hand. Note that the data points span a 19 dimensional space. Even with the rather rudimentary data selection applied in this analysis and the resulting comparatively large number of data points, the density of the data points might still be small, reducing the effects of multimodality.

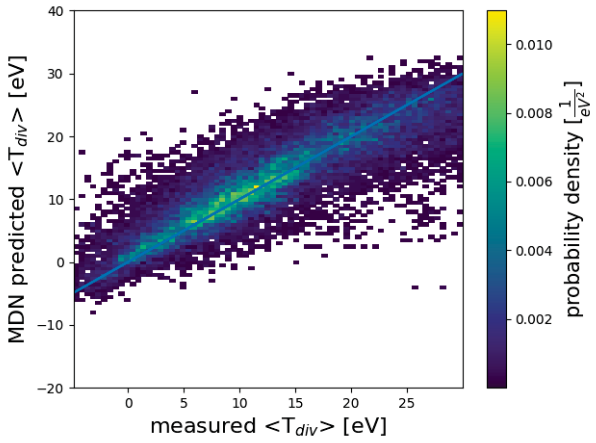


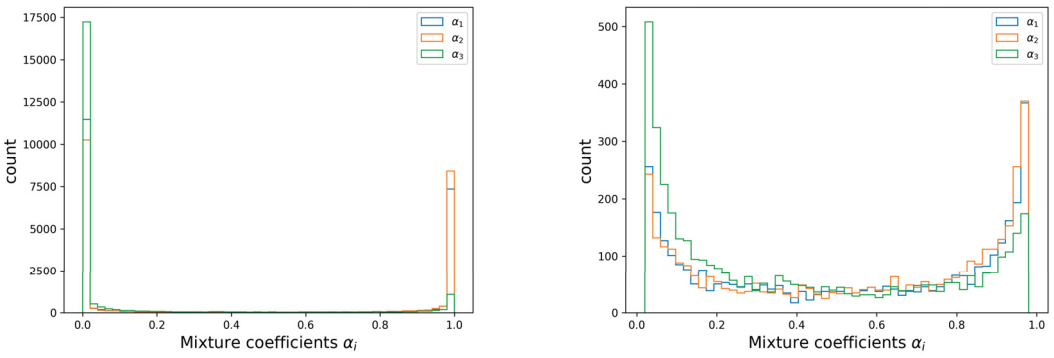
Figure 5.31: Predictions of the mixture density network model versus T_{div} values from the test data. The blue line indicates where the model predictions would perfectly fit the data obtained from the experiments.

To further analyze the distribution of mixture coefficients figure 5.33 shows the values of α_2 versus those of α_1 obtained on the test data. In this way, it is more clearly visible what the values of each mixture component are for any data point. Obviously, along the lines at which eiher α_1 or α_2 are zero, the remaining two coefficients still need to sum to a total value of one. This also holds for the diagonal line where $\alpha_2 = 1 - \alpha_1$, indicated by the blue line, and, thus, $\alpha_3 = 0$. So, the lines of the resulting triangle indicate which data points result in only two of the mixture components being non-zero. All data points inside the triangle structure cause all three components to have non-zero mixture coefficients. This visualization shows that a large portion of data points can be mostly described by at most two mixture components. This further underlines the observations from the previous results. For this approach, based on the full set of inputs, the number of data points in the test set, where all mixture coefficients are larger than 10^{-10}, amounts to 15118, which is about 68% of the data points. For the remaining test data at least one of the mixture coefficients is smaller than 10^{-10} and could, thus, be deemed unnecessary for the given data point.

The MDN approach intrinsically allows for an estimate of the model uncertainty for any given input vector, $\mathbf{x}$, via the standard deviation of the predictive distribution. This is an advantage over the previously presented NN approach in which only a point estimate of T_{div} could be obtained. The variance of the predictive distribution function is given by [50]:

$$s^2(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i(\mathbf{x}) \left(\sigma_i^2(\mathbf{x}) + \left| \mu_i(\mathbf{x}) - \sum_{l=1}^{m} \alpha_l(\mathbf{x}) \mu_l(\mathbf{x}) \right|^2 \right) \tag{5.1}$$

where m indicates the number of mixture components, i.e. m= 3 in this analysis.



(a)

(b) Excerpt of figure 5.32(a) with removed outermost bins.

Figure 5.32: Distribution of mixture coefficients $\alpha_i(\mathbf{x})$ obtained on the test data.

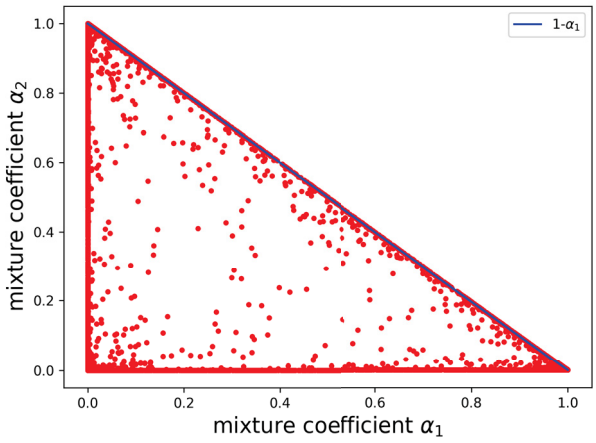


Figure 5.33: Scatter plot of mixture coefficients α_1 and α_2. The blue line indicates $\alpha_2 = 1 - \alpha_1$ and, thus, $\alpha_3 = 0$. The mixture coefficients were calculated on the test data.

Figure 5.34 shows the resulting distribution of the calculated standard deviation of the predictive distribution, termed predicted uncertainty, at the test data points versus the actual absolute mismatch of MDN predictions and the T_{div} values extracted from the experiments. The dashed blue line indicates where the standard deviation of the predictive distribution matches the actual error of the model.

The approach tends to underestimate the model uncertainty for data points with a large mismatch of predictions and observations. This can be explained by the fact that the variance of the predictive distribution is minimized by the expectation value of that distribution. Thus, comparing the standard deviation with the scatter around a different point of the distribution, in this case the expectation of the strongest mixture component, gives a skewed result.

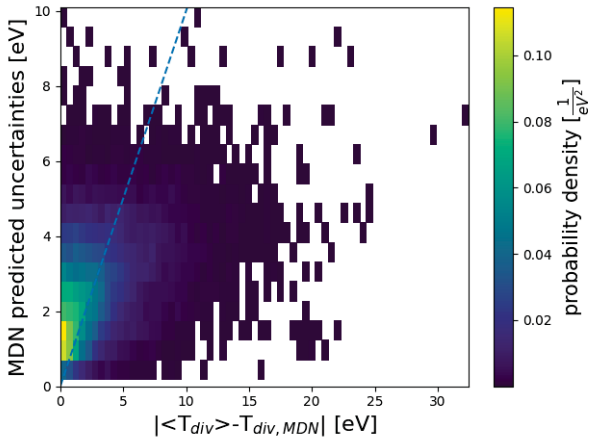


Figure 5.34: Distribution of the standard deviation of the MDN predictive distribution for each data point in the test set versus the actual absolute mismatch of MDN prediction and experimental value.

Since this approach is based on the simplified use of the expectation value of the strongest Gaussian component as the model's prediction, a better view of the model's performance and a more sophisticated assessment of the predicted uncertainty might be obtained by using the most probable value of the predictive distribution as the model's prediction. This could be part of a further analysis of the MDN based approach.

5.3.2 *Mixture Density Networks based on a reduced set of input quantities*

The MDN based approach was also tested with the reduced set of input quantities from table 4.2.

The network structure and the details of the training process were the same as for the approach with the full set of inputs. Again, a mixture of three Gaussian components has been assumed.

Figure 5.35 shows the evolution of the loss on the training data (red) and on the validation data (blue) of the MDN during the training process. As could be observed for the NNs, reducing the number of inputs to the model tends to increase the loss on the validation data. Once again, the values of this loss function are not to be compared with the values of the *Mean Squared Error* (MSE) shown for the NNs since the MDN optimization is based on the log likelihood loss function, equation 3.12. As for the MDN with all inputs, the training was stopped early because the validation loss stopped improving.

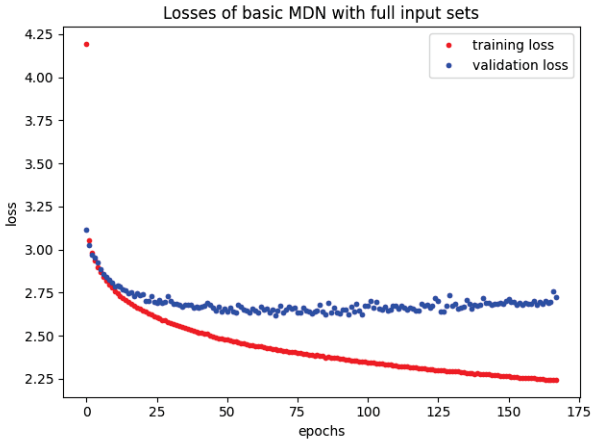


Figure 5.35: Evolution of the MDN loss during the training process. The red dots indicate the loss on the training data and the blue dots that on the validation data.

The results of evaluating the MDN in its state of minimum validation loss on the test data is depicted in figure 5.36. The distribution of model predictions, once again taken to be the expectation value μ of the mixture component with the largest mixture coefficient α, is plotted against the experimental values. The blue line indicates a 1:1 relation of model predictions and experimental values. Similar to the NN model, the MDN with the reduced set of inputs still manages to give generally accurate predictions, even though the distribution of model predictions shows a stronger scatter around the line of a 1:1 relation than could be observed for the corresponding MDN model with all inputs. The median absolute difference between model predictions and experimental values for this model is $2.4^{+3.9}_{-1.8}$ eV. Thus, the model shows a tentatively inferior performance than the MDN with all inputs but a slightly better performance than the baseline NN with reduced inputs. However, within the uncertainties on the median absolute difference, all models have a comparable performance.

To estimate the influence of the multimodality of this approach, the mixture coefficients obtained on the test data have been analyzed as before. The resulting distributions of the mixture coefficients are depicted in figure 5.37. In comparison to the results of the MDN with all inputs, figure 5.32, the observed distribution of mixture coefficients seems similar. Again, one mixture component takes on mostly small values of the mixture coefficient.
To further investigate the distribution of mixture coefficients, the scatter plot of the values of α_1 and α_2 obtained on the test data is shown in figure 5.38. Again, the outlines of the triangle structure indicate that one of the three mixture components is zero for all data points lying on the given line. For all data points inside the triangle structure all three mixture coefficients are non-zero. In comparison to the results observed for the MDN with all inputs, it is evident that more data points lie inside the triangle structure for this approach. This is also confirmed by investigating the number of data points in

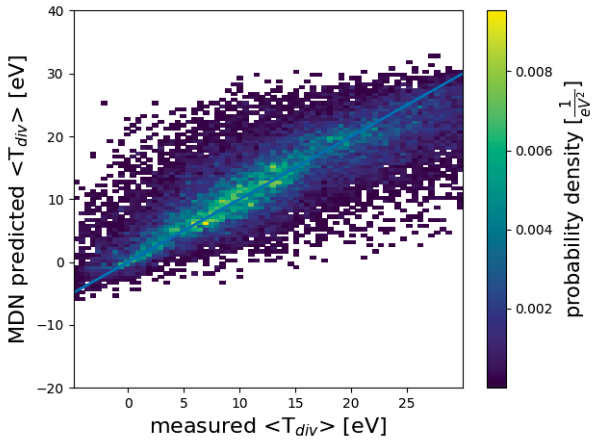
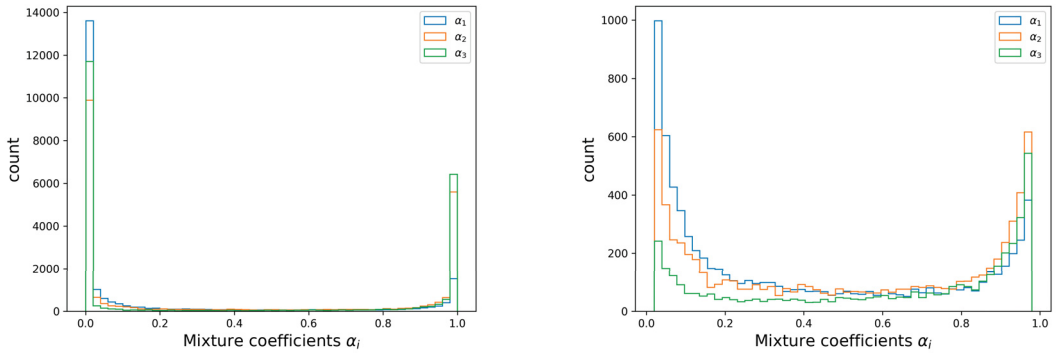


Figure 5.36: Predictions of the MDN model versus T_{div} values from the test data. The blue line indicates where the model predictions would perfectly fit the data.

the test set where all mixture coefficients are larger than 10^{-10}. For the approach based on the reduced set of inputs, this number amounts to 19901. Thus, almost 4000 more data points lead to all mixture coefficients being non-zero than was the case for the approach based on all inputs. Hence, about 89% of the data points in the test set cause all mixture components to be non-zero for the approach with the reduced set of inputs, as compared to only 68% in case of the full input set. This trend could be expected as the dimensionality of the input space is reduced in comparison to the approach based on all inputs. Intuitively this increases the necessity for a multimodal modeling approach, reflected in the observed results.



(a)

(b) Excerpt of figure 5.37(a) with removed outermost bins.

Figure 5.37: Distribution of mixture coefficients $\alpha_i(\mathbf{x})$ obtained on the test data.

Analyzing the predicted uncertainties, calculated from 5.1, on the test data reveals comparable results to those obtained with the model with all inputs. Figure 5.39 shows

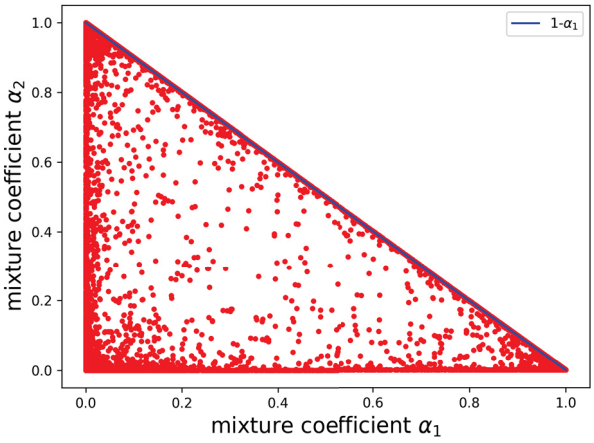


Figure 5.38: Scatter plot of mixture coefficients α_1 and α_2. The blue line indicates $\alpha_2 = 1 - \alpha_1$ and, thus, $\alpha_3 = 0$. The mixture coefficients were calculated on the test data.

the distribution of the standard deviation of the predictive distribution against the absolute mismatch of MDN predictions and experimental values. Again, the standard deviation of the distribution underestimates the absolute mismatch of model predictions and experimental values in the region where the deviation of the two is large. However, this is also caused by using the expectation value of a single mixture component as the model's prediction. Thus, comparing the absolute difference between model predictions and the experimental values with the standard deviation of the predictive distribution only gives a skewed estimate of the plausibility of the model's uncertainty. As suggested before, this might be alleviated by using the most probable value of the predictive distribution as the model's prediction and could be investigated in further analyses.

5.3.3 *Conclusion*

The analysis of the MDN models shows that again both approaches, with a full and a reduced set of input quantities, give accurate predictions. The additional benefit of modeling the full predictive distribution mainly allows for an intrinsic estimate of the model uncertainty. The analysis of the resulting mixture coefficients indicates that the problem at hand might not be multimodal in nature due to the large input space.
Further analyses with different model predictions obtained from the predictive distribution might shed more light on this approach as the presented analysis is partially biased by the choice of using the expectation value of the strongest mixture component as the model's prediction. Furthermore, the number of mixture components should be part of additional analyses as it was not investigated in the presented analysis. The number of mixture components could, for example, be chosen via Bayesian methods, see e.g. [83].
In terms of computational requirements the MDN models are comparable to the NNs presented before. Due to the faster convergence of the network training, the training

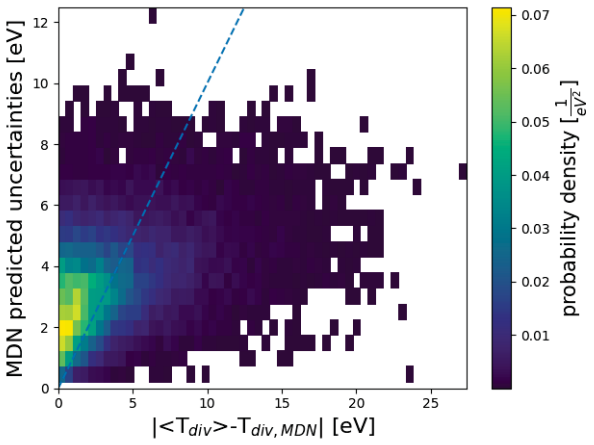


Figure 5.39: Distribution of the standard deviation of the MDN predictive distribution for each data point in the test set versus actual absolute mismatch of MDN prediction and experimental value.

process took only about 30 minutes on a single GPU. The storage requirements of the optimized models are, again, on the order of $\sim 200\,\mathrm{kB}$. Thus, the MDNs also constitute an inexpensive approach to modeling the divertor power loads and have the advantage of yielding probabilistic predictions rather than the simple point estimates obtained by the NNs.

5.4 GAUSSIAN PROCESS REGRESSION FOR DIVERTOR POWER LOAD PREDICTION

As an additional approach to obtaining a fully probabilistic model for the prediction of divertor power loads, *Gaussian Process Regression* (GPR) was tested. This section presents the result obtained in the analysis. The benefit of this modeling approach lies in its fully Bayesian nature. Thus, obtaining uncertainties on the model predictions is an inherent part of the model, similar to MDNs.
As mentioned in section 4.3 the GPR models were only trained on 7% of the available experiments. This limitation was set since the computational requirements for both, training and evaluation, of GPR models tend to grow prohibitively with the number of training data points. The analysis is focused on first evaluations of the performance of the resulting models and an investigation of different kernels. Again, approaches with the full set of inputs and the reduced set of inputs have been tested.

The GPR models presented in this analysis have been set up and evaluated with scikit-learn [44] version 0.20.3.

5.4.1 *Gaussian Process Regression based on all input quantities*

The baseline model for the GPR based approach was chosen to employ a rational quadratic kernel, see equation 3.17, with an additional noise term and a prefactor.

After optimizing the model on the training data the evaluation on the test data resulted in the distribution of model predictions shown in figure 5.40. The figure shows the prediction obtained from the optimized model against the actual values determined from the experiments. The blue line indicates a 1:1 relation. Similar to the RFs and NNs, the GPR generally gives quite accurate predictions. However, in comparison to the previous models based on all input quantities the distribution is more broadly spread around the line of a perfect match of model predictions and experimental values. This observation is confirmed by the median absolute difference between model predictions and experimental values. The median absolute difference for this model is $2.7^{+3.9}_{-1.9}$ eV. This indicates an inferior performance than that of the corresponding RF, NN and MDN models. However, it has to be noted, once again, that the training data used for the optimization of the GPR consists of only $\sim \frac{1}{10}$ the number of data points used for the optimization of the other models.

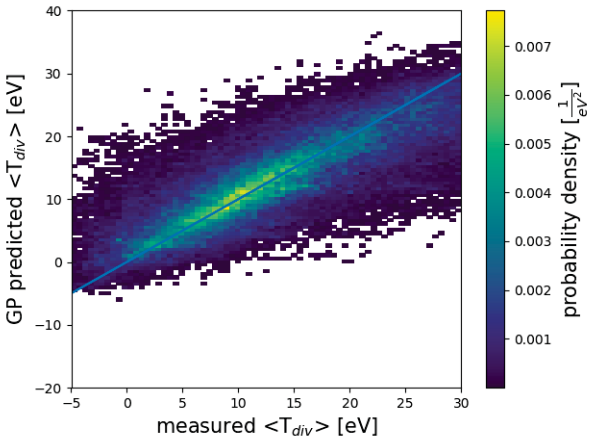
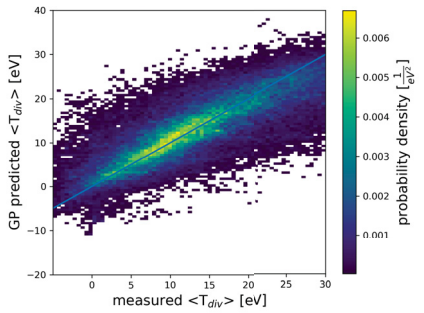


Figure 5.40: Predictions of the GPR model versus T_{div} values from the test data. The blue line indicates where the model predictions would perfectly fit the data.

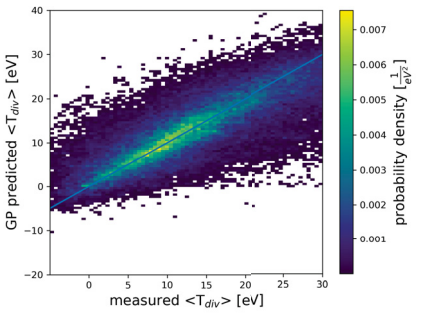
To investigate the influence of different covariance functions on the GPR, several kernels were tested. The kernels tested in this analysis are a Matérn kernel with different values of the constant ν and a *Radial Basis Function* (RBF) kernel with one length-scale parameter for each input quantity. All kernels included a prefactor and an additional noise term. The resulting model predictions on the test data are shown in figure 5.41. The primary observation of this analysis is that the tested kernels do not have a significant beneficial influence on the model's performance on the test data, as indicated by the median absolute difference between model predictions and experimental values. Only the two Matérn kernels with the smallest ν show a slight improvement over the rational

quadratic kernel presented above.
Another observation that can be made is that the model predictions seem to get cut off below 0 eV for the Matérn kernel functions with larger ν and especially for the RBF kernel. This might be linked to the differentiability of the resulting functions of the *Gaussian Process* (GP). As the functions resulting from a GP with Matérn kernel are k-times differentiable when $\nu > k$ [58], the functions tend to become smoother with increasing ν. This increased differentiability limits the flexibility of the model and, thus causes a limitation in the model predictions.
Finally, the model based on the RBF kernel seems to make a default prediction of around 0 eV for a large number of data points. Hence, this approach seems to be the least adequate for the problem at hand.
Of the tested GP models, most seem to generally make reasonably accurate predictions, except for the model with a RBF kernel. However, a rather rough function seems to be needed to describe the data, as indicated by the better performance of the Matérn kernels with small ν.
An added benefit of GPR based models over, e.g. the fully-connected NNs presented above, is the inherent treatment of the model uncertainty. To evaluate the uncertainty predicted by the GPR based model, the standard deviation of the predictive distribution at the test data points has been evaluated. The resulting values are compared with the absolute difference between model predictions and experimental values in figure 5.42. The blue line indicates where the standard deviation would match the absolute difference of model predictions and the data. The comparison is based on the results of the GP with a rational quadratic kernel, i.e. the baseline model presented above.
The comparison shows that the predicted uncertainties systematically underestimate the actual mismatch of the model predictions and the data. Thus, the uncertainties obtained from the GPR model are hardly adequate to quantify the expected error of the model. A further analysis could investigate other means of determining uncertainties from the model, see e.g. [84] and [85]. An additional point might be the inclusion of a mean of the GP, which was set to zero in this analysis.

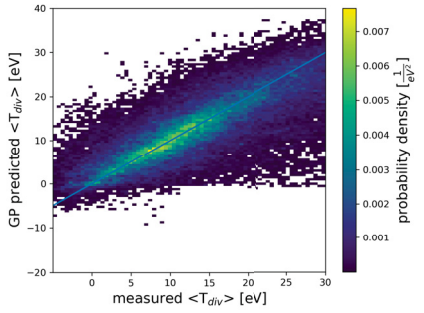
In addition to the analysis presented above, the baseline GPR approach was also tested with a larger set of training data. Optimizing the model based on 14% of the experiments yields a median absolute difference of model predictions and experimental values on the test data of $2.4^{+3.6}_{-1.7}$ eV. Thus, the larger training data slightly improves the model's performance, albeit not on a significant level. However, the runtime of the analysis increased by a factor of ~ 5. Hence, a small improvement in model performance comes at a large cost in terms of computational resources.



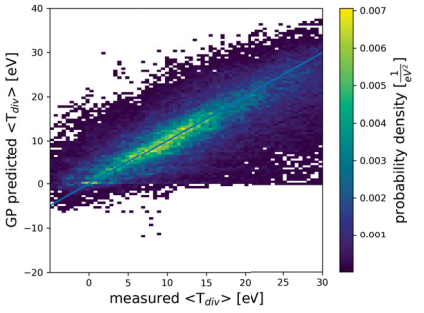
(a) Predictions of the GPR model with a Matérn kernel with $\nu = 0.5$ versus T_{div} values from the test data. The median absolute difference is $2.6^{+3.9}_{-1.9}$ eV.



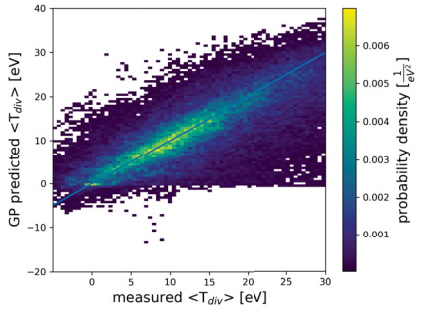
(b) Predictions of the GPR model with a Matérn kernel with $\nu = 1$ versus T_{div} values from the test data. The median absolute difference is $2.6^{+4.0}_{-1.9}$ eV.



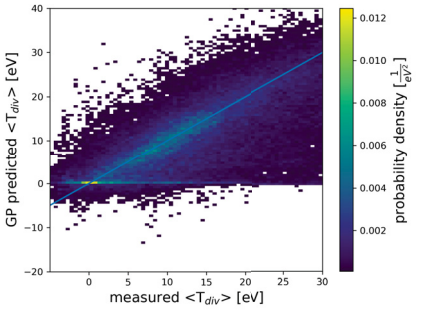
(c) Predictions of the GPR model with a Matérn kernel with $\nu = 1.5$ versus T_{div} values from the test data. The median absolute difference is $2.7^{+4.2}_{-2.0}$ eV.



(d) Predictions of the GPR model with a Matérn kernel with $\nu = 2$ versus T_{div} values from the test data. The median absolute difference is $2.8^{+4.4}_{-2.1}$ eV.



(e) Predictions of the GPR model with a Matérn kernel with $\nu = 2.5$ versus T_{div} values from the test data. The median absolute difference is $2.8^{+4.6}_{-2.1}$ eV.



(f) Predictions of the GPR model with a RBF kernel with one length-scale parameter per input quantity versus T_{div} values from the test data. The median absolute difference is $3.2^{+5.7}_{-2.4}$ eV.

Figure 5.41: Predictions of the GPR models with different kernels versus T_{div} values from the test data. The blue line indicates where the model predictions would perfectly fit the data.

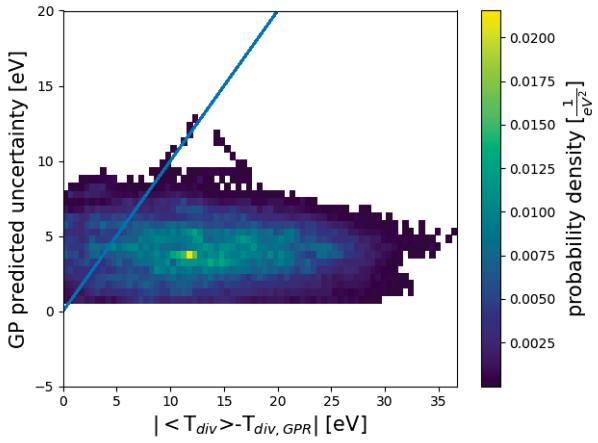


Figure 5.42: Distribution of the standard deviation of the GPR predictive distribution for the test data versus actual absolute mismatch of GPR prediction and experimental value.

5.4.2 *Gaussian Process Regression based on a reduced set of input quantities*

The GPR model tested for an approach based on the reduced set of input quantities employs the same kernel as the previous GP, i.e. a rational quadratic kernel with prefactor and an additive noise term.

The performance of the optimized model is depicted in figure 5.43. Similar to the previous model, the overall tendency of the predictions matching the experimental values can be observed. On top of this, the distribution of model predictions against experimental values seems to show a similar spread around the line of a 1:1 relation as the NN model with reduced inputs. Nevertheless, the performance of the GP is slightly worse, as indicated by the median absolute difference of model predictions and experimental values. This value amounts to $3.1^{+4.6}_{-2.3}$ eV.

As varying the kernels of the GP did not show an improvement in model performance before, this test was not repeated for the reduced set of inputs.

However, the uncertainties predicted by the GP are shown against the absolute mismatch of model predictions and experimental values in figure 5.44. As before, the predicted uncertainties systematically underestimate the error of the model. Thus, a more thorough analysis of the predicted uncertainties might be needed.

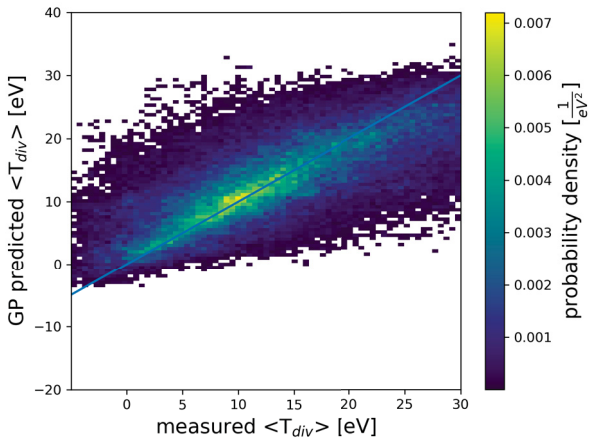


Figure 5.43: Predictions of the GPR model versus T_{div} values from the test data. The blue line indicates where the model predictions would perfectly fit the data.

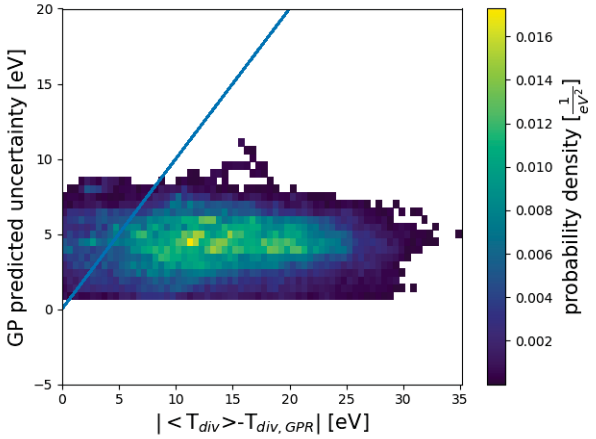


Figure 5.44: Distribution of the standard deviation of the GPR predictive distribution for each data point in the test set versus actual absolute mismatch of GPR prediction and experimental value.

5.4.3 *Conclusion*

The predictions of the GPR models showed a tentatively worse performance than most of the other models investigated. However, the general tendency of the predictions still matches the expectation.
Varying the covariance function of the underlying GP did not show an improvement in model performance in the approach based on all input quantities. Moreover, the uncertainties predicted by the GP models systematically underestimate the actual mismatch of model predictions and experimental values. Hence, a further analysis with special

regard to the model uncertainties might be justified.
An analysis of an increase in the size of the training data set with the GPR utilizing the full set of inputs did not show an improvement of the model performance that was worth the additional runtime. Doubling the fraction of experiments used for the model optimization only resulted in a minor improvement of the model performance but increased the runtime for training and evaluating the model by a factor of ~ 5 and the storage required by the model by a factor of ~ 4.

In terms of resources the GPR based models are more expensive than the models presented previously. Training and evaluating the baseline model with data from 7% of the experiments took roughly 1.5 hours on a single CPU for both, the full and the reduced input set. This includes 9 restarts of the optimizer to find the optimal kernel parameters. However, due to the fact that the model needs all training data points to make predictions both, storing the model and making predictions, can become quite expensive. The GPR models require about 244 MB of storage each, exceeding the RF models by a factor of 4. Hence, the GPR models still constitute a comparatively lightweight model for divertor power load predictions but tend to fall off in comparison to the other models investigated in this analysis. Of course, the model performance might also be significantly improved by more elaborate choices for the covariance function, which could be the topic of further studies.

6 CONCLUSIONS AND OUTLOOK

In this thesis several approaches to modeling steady-state divertor power loads based on machine learning methods have been tested. To this end a data base comprising almost 6 years of data from the *ASDEX Upgrade* (AUG) experiment was established. The analysis of the data suggested that the overall comparatively rough data selection suffices to reproduce general trends expected from dedicated physics analyses. Based on this data, several machine learning methods have been optimized and evaluated. All methods were tested with a more extensive and a reduced set of input quantities that is closer to the actual set of parameters known prior to performing experiments. These tests indicate that this smaller set of inputs might be enough to obtain accurate predictions of the power load at the divertor targets.

Among the methods tested, the *Neural Network* (NN) with optimized hyperparameters showed the best performance for both the full set of inputs and the reduced set of inputs. The *Random Forest* (RF) models, despite being rather simple in nature, also showed convincing predictive qualities. Of the probabilistic methods, i.e. *Mixture Density Network*s (MDNs) and *Gaussian Process Regression* (GPR), that have the added benefit of an inherent estimate of the model's uncertainty, only the MDN attained a predictive performance on the same level as the NN and RF. However, the predicted uncertainties of both probabilistic approaches systematically underestimated the actual mismatch of model predictions and experimental data. For the MDNs this might be caused by the definition of the model's prediction and could potentially be improved by selecting the most probable value of the predictive distribution as the model's prediction. For the GPR other methods might be required, e.g. additional covariance functions could be tested.

The analysis of the MDNs indicated that the problem at hand is not severely affected by multimodality, despite the selected input quantities not covering all properties that could be relevant to the operation of a tokamak. Nevertheless, the test with the reduced set of inputs showed that an approach able to handle multimodal data can be beneficial when reducing the number of inputs.

The investigation of the learned dependencies of the RF and NN models revealed some unexpected effects potentially caused by spurious correlations but also showed that the models extracted important trends from the experimental data that were expected from physics. For example, a linear dependency of the model predictions on the plasma current was found for all RF and NN models. This is in accordance with expectations from other analyses and the *Two Point Model* (TPM). Moreover, the RF and the NN models using all input quantities showed a trend of decreasing model predictions with increasing neutral density in the divertor, as could be expected. Another result in accordance with other analyses is the interplay of the deuterium throughput and the neutral density in the divertor. As already indicated by a strong correlation coefficient in the data, the models' dependencies also show that the deuterium throughput can substitute

the neutral density in the divertor as an input quantity of the model.
However, the observed dependencies of the models on the lower triangularity were not expected. As the energy and particle confinement tend to improve with increasing triangularity, the observed increase in the model predictions seems counterintuitive. This behavior might be caused by correlations among the input quantities. Another noteworthy observation concerns the dependency of the NN with all inputs on the energy stored in the plasma. As could be expected, an increase of the model predictions with increasing stored energy was observed. However, this dependency showed a change with increasing stored energy up to a point at which the model predictions started decreasing with a further increase in the stored energy. An analysis of the plasma confinement did not show any evidence that this behavior might be caused by a change in the confinement regime. Further analyses could shed more light on these unexpected dependencies.

All models presented are computationally rather inexpensive in comparison to simulation codes. Therefore, further investigations of the presented approaches to modeling divertor power loads seem justified. These further analyses could include the investigation of different output quantities of the models. Since the presented analysis relied on an AUG specific quantity, it might be worthwhile to investigate similar approaches based on other signals, e.g. measurements from Langmuir probes. This would naturally ease another topic for future analyses: the inclusion of data from other tokamaks. Including data from several tokamaks would have the advantage of potentially allowing for a size scaling across different machines. However, the task is hampered by the necessity of matching different measurements across the different tokamaks and a sparsely covered parameter space in terms of machine size.
Furthermore, a time-resolved analysis could handle the effects of *Edge Localized Mode*s (ELMs) better than the presented approach. In the current analysis the ELMs might distort the actual steady-state signals, interfering with the analysis.
As additional source of data and to mitigate the effects of measurement uncertainties, data from simulations could be included in similar analyses.
In conclusion, these results allow for a broad variety of possible extensions to the presented analysis from both the methodological side and from physics.

Part II

APPENDICES

A

THE BIAS-VARIANCE DECOMPOSITION

This section gives an introduction to the bias-variance decomposition to clarify these terms and important underlying concepts. It summarizes chapter 3.2 in [48].

When fitting a model to data, it is necessary to quantify the mismatch of a given model and the actual model underlying the data. However, since the actual model is usually unknown, the distribution of observed data has to be used as a proxy instead.
Assume a given data set $\mathcal{D}$ containing observations of input vectors $\mathbf{x}$ and corresponding target values t. A common choice to quantify the mismatch of a model, $y(\mathbf{x})$, and the data generating model is the squared loss function given by

$$\mathcal{L}(t, y(\mathbf{x})) = (y(\mathbf{x}) - t)^2 . \tag{A.1}$$

For this loss function it can be shown that the optimal prediction is given by the conditional expectation

$$h(\mathbf{x}) = \int t p(t|\mathbf{x}) dt . \tag{A.2}$$

The common goal of fitting a model to data is the minimization of the average loss, which can, in this case, be written as

$$\mathbb{E}[\mathcal{L}] = \int (y(\mathbf{x}) - h(\mathbf{x}))^2 p(\mathbf{x}) d\mathbf{x} + \iint (h(\mathbf{x}) - t)^2 p(\mathbf{x}, t) d\mathbf{x} dt . \tag{A.3}$$

The first part of the integrand of the second summand represents the noise in the data and is independent of the choice of model $y(\mathbf{x})$. Hence, the average loss can not be reduced beyond this value by model selection and the optimization process seeks to achieve the minimum of the first summand.
To relate this to a model bias and variance, consider multiple data sets $\mathcal{D}_i$, each of size N with data drawn from the joint distribution $p(\mathbf{x}, t)$. Fitting models to each of these data sets will result in an ensemble of models, since the optimal choice of model parameters is driven by the data. Thus, the model also depends on the selected data set, $y(\mathbf{x}, \mathcal{D}_i)$. Each of these models results in a different value of the loss function and the overall performance can be measured by averaging over all data sets. This, in particular, holds for the first summand in equation A.3. Averaging the first part of the integrand over the data sets and rewriting the result gives

$$\begin{aligned} \mathbb{E}_{\mathcal{D}}[(y(\mathbf{x}, \mathcal{D}_i) - h(\mathbf{x}))^2] =& (\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}, \mathcal{D}_i)] - h(\mathbf{x}))^2 \\ &+ \mathbb{E}_{\mathcal{D}}[(y(\mathbf{x}, \mathcal{D}_i) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}, \mathcal{D}_i)])^2] \end{aligned} \tag{A.4}$$

where $\mathbb{E}_{\mathcal{D}}$ denotes the average over the data sets.
The first summand on the right-hand side is the square of the model's bias. It describes

the deviation of an average prediction of the model from the desired optimal prediction given by $h(\mathbf{x})$. The second summand is the variance of the model and describes how strongly the model varies when the selected data set changes.

A simple, i.e. strongly regularized, model tends to have a larger bias as the model might be too rigid to properly adapt to the observed data. A very complex model, e.g. a model with many parameters, usually has a smaller bias but a larger variance as the model can adapt to features caused by the data selection, which might be present in the selected data set but not representative for other data. An illuminating example of the bias-variance tradeoff is also given in chapter 3.2 of [48].

B

SOME BASICS OF BAYESIAN REGRESSION

This section serves as a reminder on the basics of Bayesian regression, primarily to clarify terminology. More detailed explanations on this section's contents can be found in e.g. [48], [58] and [86], parts of which are summarized here.

The Bayesian framework allows for fully probabilistic regression approaches. An important advantage of this is the inherent treatment of uncertainties and a consistent approach to parameter estimation. At the heart of Bayesian approaches is Bayes' theorem. Applied to a general regression problem with a set of given data denoted by $\mathcal{D} = \{\mathbf{x}_i, y_i\}$, where i indicates the data point, and a model class $\mathcal{H}$ with a set of parameters θ, it reads

$$p(\theta|\mathcal{D}, \mathcal{H}, I) = \frac{p(\mathbf{y}|\theta, \mathbf{X}, \mathcal{H}, I)p(\theta|\mathcal{H}, I)}{p(\mathbf{y}|\mathbf{X}, \mathcal{H}, I)} \tag{B.1}$$

where the notation of [86] was used, so that I denotes additional background information. $\mathbf{y}$ is the vector of all y_i and $\mathbf{X}$ denotes the matrix composed of all input vectors $\mathbf{x}_i$. Thus, equation B.1 allows to infer the probability of any set of parameters of the model under the condition that the data, the model class and further background information are known. This probability is $p(\theta|\mathcal{D}, \mathcal{H}, I)$ which is called *posterior distribution*. This posterior distribution can be calculated from the probability of obtaining the given data set from a certain model with fixed parameters, $p(\mathbf{y}|\theta, \mathbf{X}, \mathcal{H}, I)$, times an inital guess on the probability of any given set of parameters, $p(\theta|\mathcal{H}, I)$. The former is called *likelihood function* and the latter is the *prior distribution*. These two terms embody the principle of Bayesian approaches: an initial hypothesis, the prior distribution, is modified based on observations and their relation to the model under investigation, the likelihood. The normalizing term $p(\mathbf{y}|\mathbf{X}, \mathcal{H}, I)$ is commonly called *marginalized likelihood*[1] or *evidence*.

In order to obtain the most probable set of model parameters, θ_*, the posterior distribution needs to be maximized. As the denominator in equation B.1 is only a normalizing constant, commonly only the product of likelihood and prior is maximized. For the considerations in this thesis, it is mostly beneficial to consider the case of a Gaussian distribution of the data y_i around some model function $f(\mathbf{x}, \theta)$, including additive Gaussian noise with variance σ_n^2. Under the assumption of independent data points this leads to the likelihood function

$$p(\mathbf{y}|\theta, \mathbf{X}, \mathcal{H}, I) = \prod_i \frac{1}{\sqrt{2\pi}\sigma_\mathrm{n}} \exp\left(-\frac{(y_i - f(\mathbf{x}_i, \theta))^2}{2\sigma_\mathrm{n}^2}\right) \tag{B.2}$$

1 The name already implies its relationship to the likelihood, i.e. $p(\mathbf{y}|\mathbf{X}, \mathcal{H}, I) = \int p(\mathbf{y}|\theta, \mathbf{X}, \mathcal{H}, I)p(\theta|\mathcal{H}, I)d\theta$.

with the exponential function exp. Applying a logarithm to this function makes the optimization computationally easier and shows that the resulting optimization problem is closely related to the well known minimization of the sum of squared errors. It is

$$\mathcal{L} = \ln(p(\mathbf{y}|\theta, \mathbf{X}, \mathcal{H}, I)) = \sum_i -\frac{1}{2\sigma_\mathrm{n}^2}(f(\mathbf{x}_i, \theta) - y_i)^2 - \ln(\sigma_\mathrm{n}) - \frac{1}{2}\ln(2\pi) \tag{B.3}$$

with the natural logarithm ln. Hence, maximizing the likelihood of a Bayesian approach with a Gaussian likelihood with respect to the model paramters θ corresponds to minimizing the sum of squared errors and, thus, also to minimizing the *Mean Squared Error* (MSE) which was used in this thesis for the optimization of the *Neural Network*s (NNs).

The prior term describes ab initio knowledge (or the lack thereof) about the model parameters. As an illuminating example consider a normal prior distribution, so that

$$p(\theta|\mathcal{H}, I) = \mathcal{N}(\theta|0, \lambda^{-1}\mathbb{I}) \tag{B.4}$$

with some parameter $\lambda > 0$ and the identity matrix $\mathbb{I}$. This encodes the expectation that the model parameters are normally distributed around zero. Incorporating this prior term in the optimization of the posterior results in an additional term $\frac{\lambda}{2}\theta^T\theta$ in equation B.3. Thus, this additional term corresponds to an L2 regularization term and leads to the optimization scheme also known as ridge regression. Maximizing the product of likelihood and prior then results in the *Maximum Posterior* (MAP) estimate of the best fit parameters θ_*. Usually, these best fit parameters are then used to obtain predictions from the model on new data.

In theory, the same approach could also be utilized to determine the best set of hyperparameters of a model, i.e. the parameters that determine the model class $\mathcal{H}$. In this case, the marginalized likelihood from equation B.1 would take the role of the likelihood and a new prior distribution would be required. However, calculating the marginalized likelihood from equation B.1 is often very complicated due to the integration over all potential parameter values of the model.

C

ADDITIONAL RESULTS OF THE GAUSSIAN PROCESS REGRESSION ANALYSIS

This section is used to present additional results of the analysis based on *Gaussian Process Regression* (GPR). All results presented in this section constitute further original contributions.

Figure C.1 shows the distribution of T_{div} values obtained from the data used to train and test the GPR based models. The blue line depicts the distribution in the training data and the yellow line shows the distribution in the test data. Due to the small size of the training set (only 7% of all discharges), the distributions show a stronger mismatch than that observed for the data split used in the other models. Of course, this also skews the comparison between GPR and the other models. However, the computational cost of making predictions with a GPR based model increases with the number of data points used for the optimization process. Hence, the number of training data points was limited.

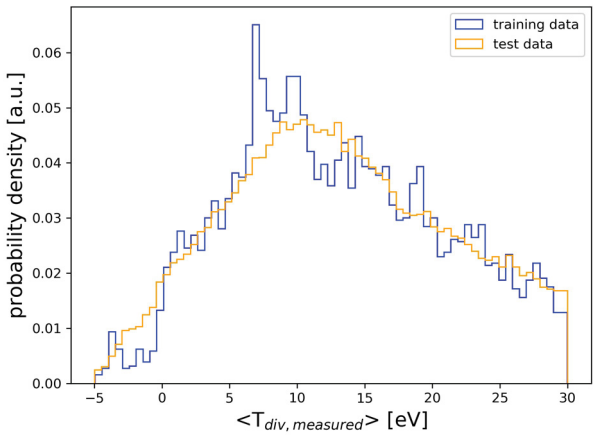


Figure C.1: Distribution of T_{div} extracted from the experiments for training and test set used for the Gaussian Process Regression.

The matrix of pairwise Pearson correlation coefficients, obtained from the training data, shown in figure C.2 also shows slight changes caused by the different data split when compared to the correlation matrix obtained from the training data used for the *Random Forest* (RF) and *Neural Network* (NN) based approaches.

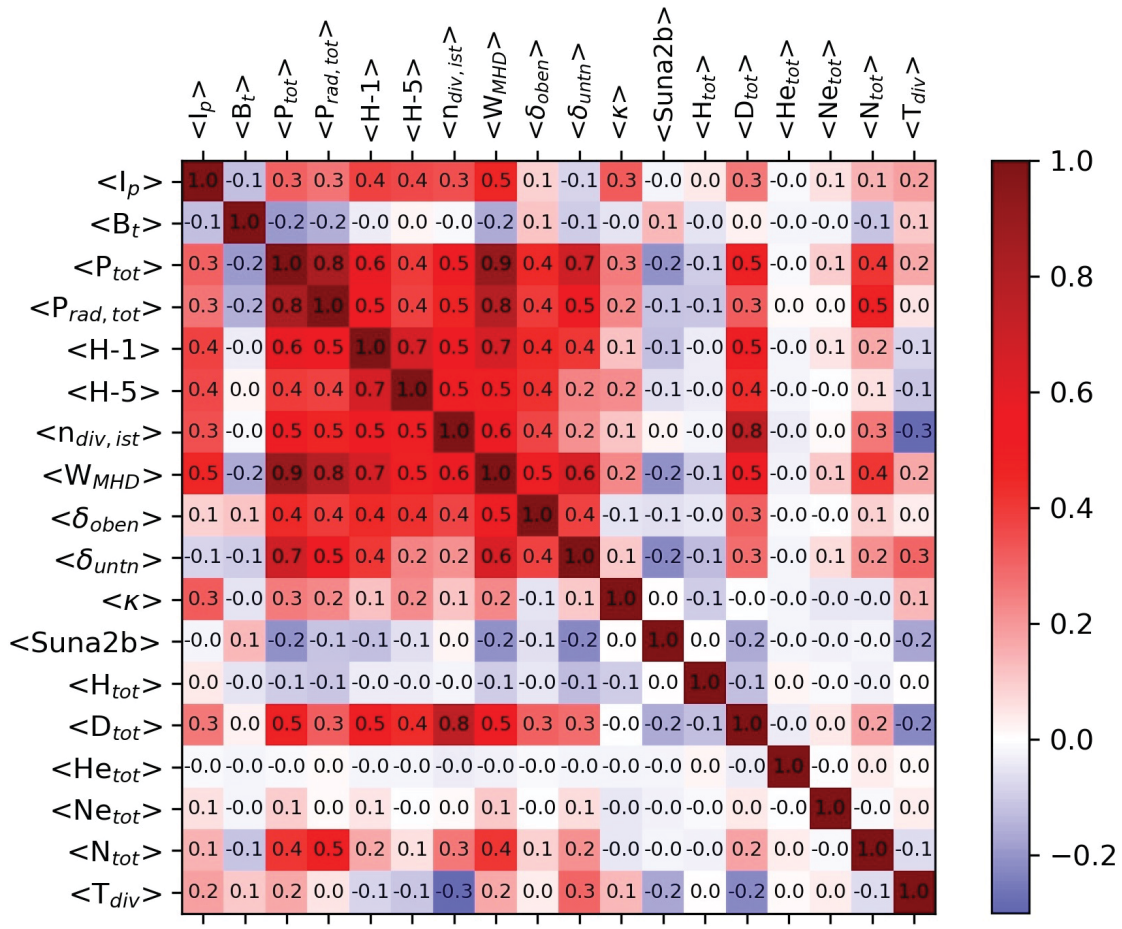


Figure C.2: Matrix of pairwise Pearson correlation coefficients of all input quantities and the target quantity, T_{div}, calculated fom the training data used for the Gaussian Process Regression approach.

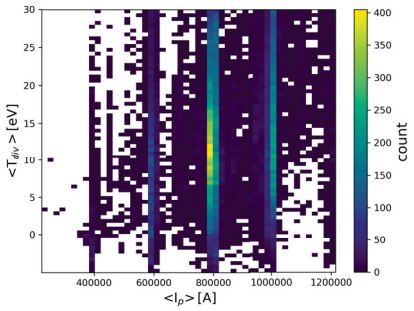
D

ADDITIONAL ANALYSIS OF THE DATA

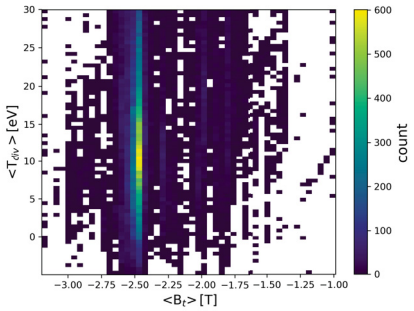
This section contains some additional information on the data used for the optimization of the *Neural Network*s (NNs) and the *Random Forest*s (RFs) as introduced in section 4.3.

Figures D.1, D.2 and D.3 show the distributions of T_{div} values measured in the analyzed experiments in the training data used for the optimization of the NNs and the RFs versus the corresponding values of the input parameters.
These plots give an indication of the parameter ranges covered in this analysis. Some noteworthy features of the collected data are, e.g. the dominant values of the plasma current, I_p, and the strength of the toroidal magnetic field, B_t. Furthermore, the distributions of the hydrogen, helium and neon throughputs show that there were very few data points with a significant contribution of these species. In total the training data contained 1207 data points with non-zero hydrogen throughput, 1023 with non-zero helium throughput and only 395 data points with non-zero neon throughput.
Moreover, the distributions show that the rather crude data selection applied in order to obtain a large data base does not eliminate all outliers in the data. This is especially evident in the distributions of the lower triangularity and the elongation where there are some data points that vary significantly from the bulk of the distribution indicating the usual operational parameter values.

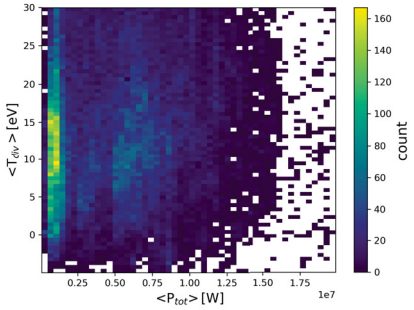
There are no immediate dependencies of T_{div} on any of the given parameters visible in the plots. However, these graphics only correspond to one dimensional projections of a potentially multidimensional function and, in contrast to the plots of the partial dependencies shown in chapter 5, do not average over all other input quantities that are not on the abscissa of each plot.



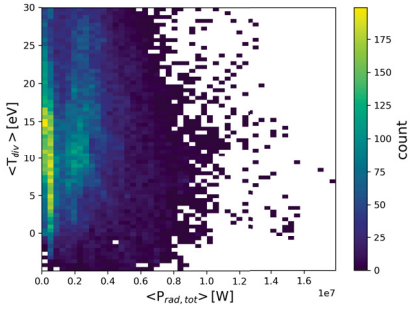
(a) Distribution of T_{div} values versus plasma current in the training data of the neural networks and the random forest.



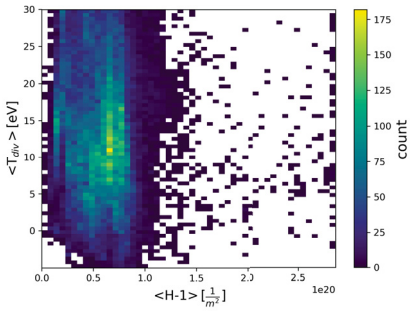
(b) Distribution of T_{div} values versus strength of the toroidal magnetic field in the training data of the neural networks and the random forest.



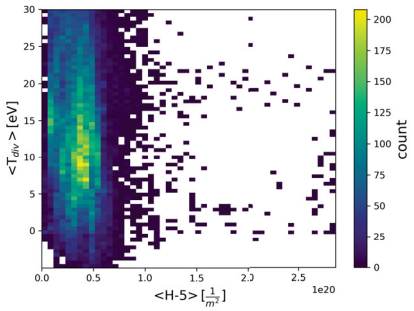
(c) Distribution of T_{div} values versus total deposited heating power in the training data of the neural networks and the random forest.



(d) Distribution of T_{div} values versus total radiated power above the X-point in the training data of the neural networks and the random forest.

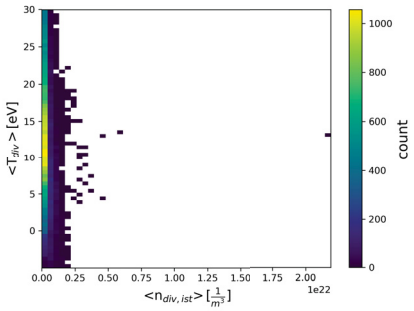


(e) Distribution of T_{div} values versus line-integrated electron density in the core in the training data of the neural networks and the random forest.

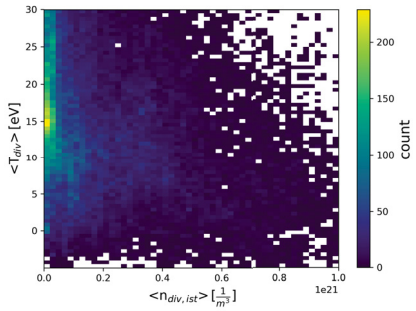


(f) Distribution of T_{div} values versus line-integrated electron density in the edge region in the training data of the neural networks and the random forest.

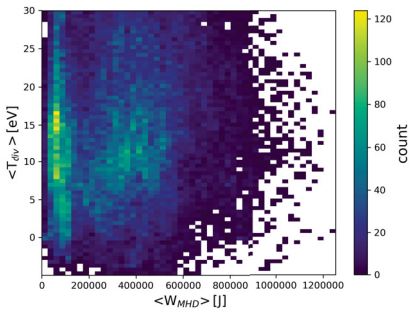
Figure D.1: Distributions of T_{div} values versus the model input parameters obtained from the training data used for the neural networks and the random forest.



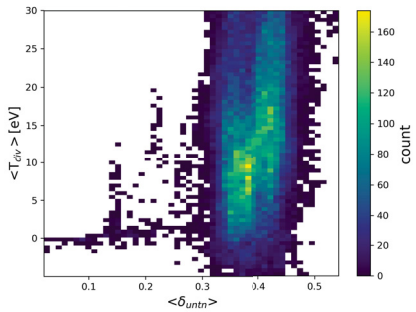
(a) Distribution of T_{div} values versus density of neutral particles in the divertor in the training data of the neural networks and the random forest.



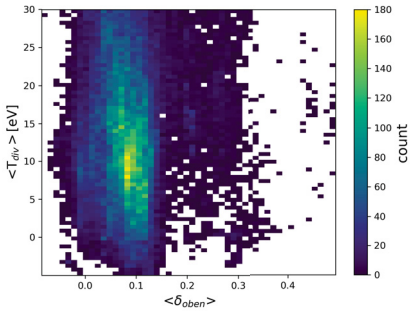
(b) Excerpt of the distribution of T_{div} values versus density of neutral particles in the divertor in the training data of the neural networks and the random forest.



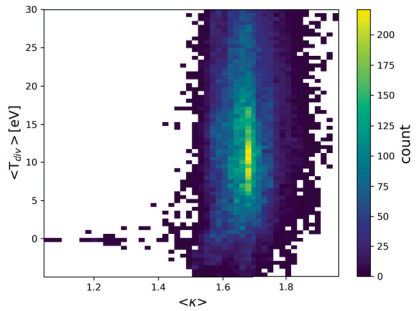
(c) Distribution of T_{div} values versus energy stored in the plasma in the training data of the neural networks and the random forest.



(d) Distribution of T_{div} values versus lower triangularity of the plasma cross section in the training data of the neural networks and the random forest.

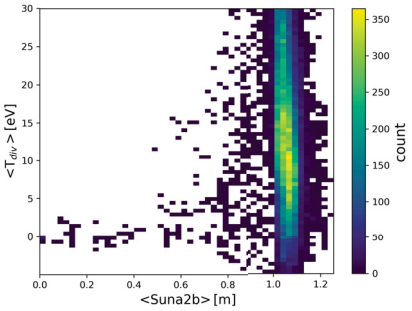


(e) Distribution of T_{div} values versus upper triangularity of the plasma cross section in the training data of the neural networks and the random forest.

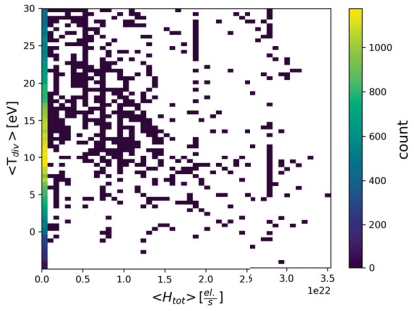


(f) Distribution of T_{div} values versus elongation of the plasma cross section in the training data of the neural networks and the random forest.

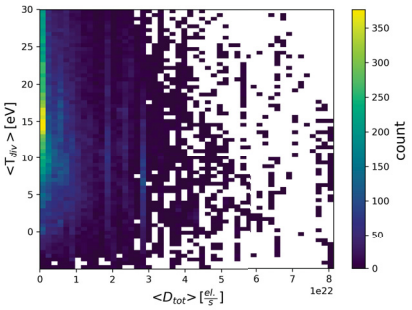
Figure D.2: Distributions of T_{div} values versus the model input parameters obtained from the training data used for the neural networks and the random forest.



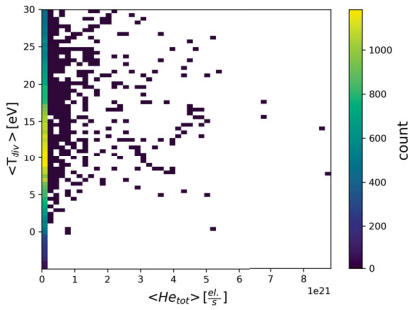
(a) Distribution of T_{div} values versus position of the strike line in the training data of the neural networks and the random forest.



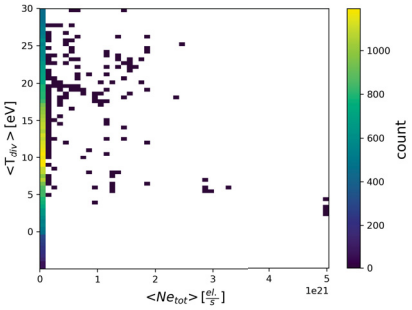
(b) Distribution of T_{div} values versus hydrogen throughput in the training data of the neural networks and the random forest.



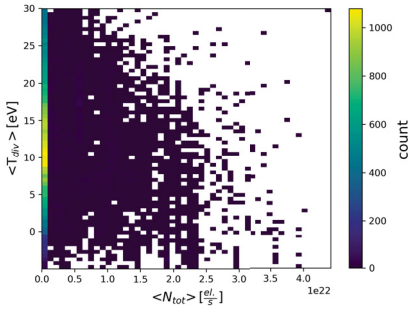
(c) Distribution of T_{div} values versus deuterium throughput in the training data of the neural networks and the random forest.



(d) Distribution of T_{div} values versus helium throughput in the training data of the neural networks and the random forest.



(e) Distribution of T_{div} values versus neon throughput in the training data of the neural networks and the random forest.

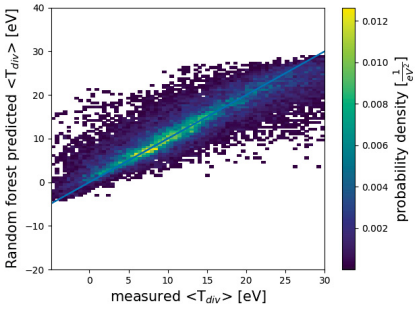


(f) Distribution of T_{div} values versus nitrogen throughput in the training data of the neural networks and the random forest.

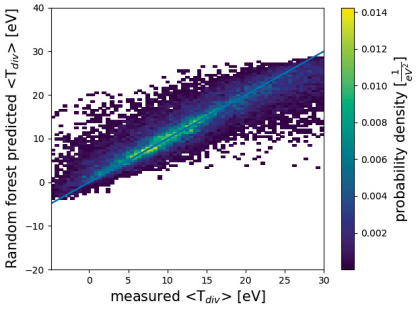
Figure D.3: Distributions of T_{div} values versus the model input parameters obtained from the training data used for the neural networks and the random forest.

E

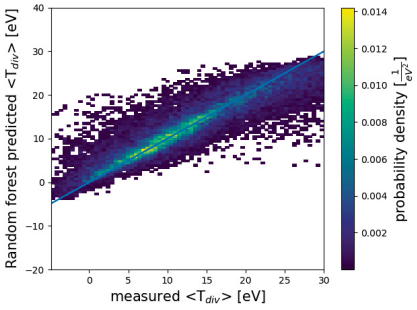
FURTHER RESULTS OF THE RANDOM FOREST MODEL USING ALL INPUTS



(a) Distribution of random forest predictions versus T_{div} values; 20 trees.

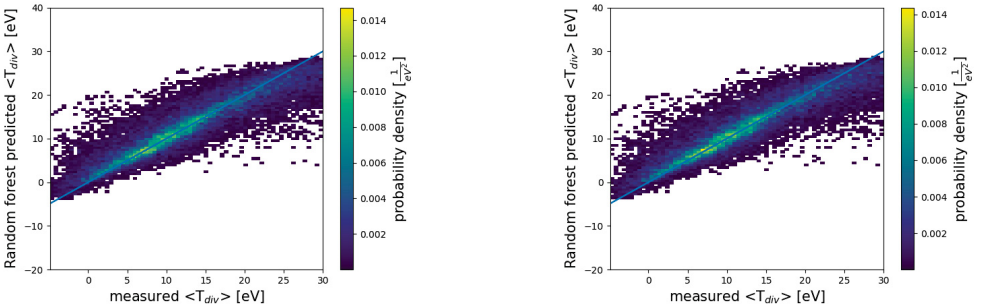


(b) Distribution of random forest predictions versus T_{div} values; 50 trees.



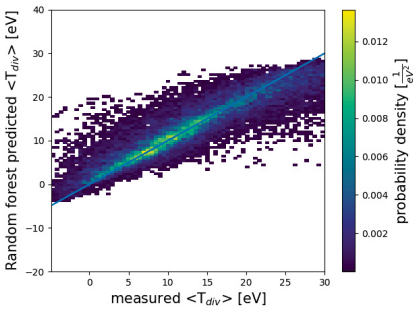
(c) Distribution of random forest predictions versus T_{div} values; 100 trees.

Figure E.1: Predictions of random forest models with different numbers of trees versus T_{div} values from the test data. The blue line indicates where the model predictions would perfectly fit the data obtained from the experiments.



(a) Distribution of random forest predictions versus T_{div} values; 200 trees.

(b) Distribution of random forest predictions versus T_{div} values; 500 trees.



(c) Distribution of random forest predictions versus T_{div} values; 1000 trees.

Figure E.2: Predictions of random forest models with different numbers of trees versus T_{div} values from the test data. The blue line indicates where the model predictions would perfectly fit the data obtained from the experiments.

F

FURTHER ANALYSIS OF THE NEURAL NETWORK DEPENDENCIES

The dependeny of the *Neural Network* (NN) with all inputs on the energy stored in the plasma observed in figure 5.16(b) shows a counterintuitive behavior. At first, an increase of the model predictions with increasing stored energy can be observed, as could be expected. However, at $<W_{MHD}> \approx 350000\,J$ the dependency changes and the model predictions start to decrease with increasing energy stored in the plasma. To investigate whether this might be caused by a transition of the plasma to a state of increased confinement in the plasma core, the ratio of the deposited heating power and the threshold power needed for such a transition, P_{L-H}, was analyzed. P_{L-H} was calculated according to [63]. Thus, if the ratio is larger than one, the core plasma is in a state of increased confinement, which could potentially explain a decrease in T_{div}.
Figure F.1 shows an excerpt of the resulting distribution of the calculated ratios against the corresponding values of $<W_{MHD}>$ for the training data. Here, data points with a ratio of the powers larger than ten have been removed. The distribution was normalized per column and the red dots show the mean values of the power ratio per column. The red dashed line indicates where the ratio of deposited power and P_{L-H} is one.
According to this distribution, the value of $<W_{MHD}>$ that coincides with most data points, on average, reaching the regime of improved confinement lies below 200000 J. Thus, this change in confinement does not seem to be the potential explanation of the change in the dependency of the NN predictions on the energy stored in the plasma.

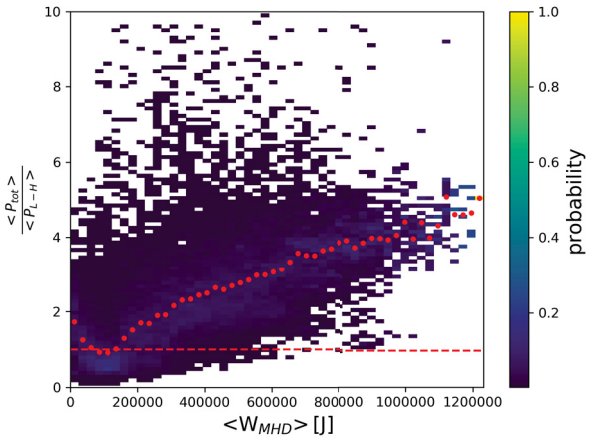


Figure F.1: Distribution of the ratio of deposited heating power and power necessary to improve the plasma confinement versus corresponding values of the energy stored in the plasma for the training data. The distribution is normalized column-wise and the red dots indicate the mean per column. The red dashed line indicates the value one.

G

GLOSSARY OF MACHINE LEARNING TERMS

- **Activation function** - (Non-linear) function employed by the neurons of a *Neural Network* (NN) to obtain a (non-linear) mapping of inputs to outputs.
- **Backpropagation** - Algorithm to calculate the derivative of a NN's loss function with respect to model parameters by applying the chain rule.
- **Dropout** - A technique to combat overfitting by reducing the number of neurons and connections in a NN during training.
- **Early stopping** - A technique to combat overfitting by stopping the training process early, i.e. when a given metric stops improving.
- **Epoch** - One whole training step of a NN, consisting of using all mini batches and, thus, all training data once to calculate model updates.
- **Hyperparameter** - Parameter determining the model class, e.g. number of layers of a NN or maximum depth of a decision tree.
- **Learning rate** - Parameter that determines the step size of the gradient-based optimization of model parameters, especially in NNs.
- **Mini batch** - A random subsample of the training data used for the calculation of the loss' gradient.
- **Overfitting** - Overadaptation of a model to noise in the training data.
- **Test data** - Data used to evaluate a model's performance on previously unseen data.
- **Training data** - Data used to optimize a model.

H

ACRONYMS

AUG *ASDEX Upgrade*

ELM *Edge Localized Mode*

ELU *Exponential Linear Unit*

GP *Gaussian Process*

GPR *Gaussian Process Regression*

MAP *Maximum Posterior*

MDN *Mixture Density Network*

MSE *Mean Squared Error*

NN *Neural Network*

RBF *Radial Basis Function*

ReLU *Rectified Linear Unit*

RF *Random Forest*

RMSProp *Root Mean Square Propagation*

SELU *Scaled Exponential Linear Unit*

SGD *Stochastic Gradient Descent*

SOL *Scrape-Off Layer*

TPM *Two Point Model*

I

BIBLIOGRAPHY

[1] J. Ongena et al., *Magnetic-confinement fusion*, *Nature Physics* **12**.5 (2016) 398–410.

[2] OpenStax, *Nuclear Binding Energy*, OpenStax CNX, 22. Dez. 2020, Accessed: 03.02.2021 12:45, URL: http://cnx.org/contents/94efefac-6c2f-4754-a330-3292820e2b71@8.

[3] H.-S. Bosch and G. M. Hale, *Improved formulas for fusion cross-sections and thermal reactivities*, 1992 611–631, ISSN: 0029-5515, DOI: 10.1088/0029-5515/32/4/i07.

[4] T. Tanabe, *Tritium: Fuel of Fusion Reactors*, Springer Japan, 2016, ISBN: 9784431564607, URL: https://books.google.de/books?id=tvqoDQAAQBAJ.

[5] J. Wesson and D.J. Campbell, *Tokamaks*, 3rd edition, Clarendon Press, 2004, ISBN: 9780198509226.

[6] S. Li et al., *Optimal tracking for a divergent-type parabolic pde system in current profile control*, *Abstract and Applied Analysis*, vol. 2014, Hindawi, 2014.

[7] S. I. Krasheninnikov and A. S. Kukushkin, *Physics of ultimate detachment of a tokamak divertor plasma*, *Journal of Plasma Physics* **83**.5 (2017).

[8] R. A. Pitts et al., *Physics basis for the first ITER tungsten divertor*, *Nuclear Materials and Energy* **20** (2019) 100696.

[9] A. C. C. Sips et al., *Advanced scenarios for ITER operation*, *Plasma physics and controlled fusion* **47**.5A (2005) A19.

[10] T. Eich et al., *Inter-ELM power decay length for JET and ASDEX Upgrade: measurement and comparison with heuristic drift-based model*, *Physical review letters* **107**.21 (2011) 215001.

[11] M. Bernert et al., *Power exhaust by SOL and pedestal radiation at ASDEX Upgrade and JET*, *Nuclear Materials and Energy* **12** (2017) 111–118.

[12] A. Kallenbach et al., *Impurity seeding for tokamak power exhaust: from present devices via ITER to DEMO*, 2013 124041, ISSN: 0741-3335, DOI: 10.1088/0741-3335/55/12/124041.

[13] A. W. Leonard, *Plasma detachment in divertor tokamaks*, 2018 044001, ISSN: 0741-3335, DOI: 10.1088/1361-6587/aaa7a9.

[14] Y. Shimomura et al., *Characteristics of the divertor plasma in neutral-beam-heated ASDEX discharges*, *Nuclear Fusion* **23**.7 (1983) 869.

[15] D. P. Coster, *Exploring the edge operating space of fusion reactors using reduced physics models*, 2017 1055–1060, ISSN: 2352-1791, DOI: 10.1016/j.nme.2016.12.033.

[16] C. Rea et al., *Disruption prediction investigations using Machine Learning tools on DIII-D and Alcator C-Mod*, *Plasma Physics and Controlled Fusion* **60**.8 (2018) 084004, DOI: 10.1088/1361-6587/aac7fe, URL: https://doi.org/10.1088%2F1361-6587%2Faac7fe.

[17] J. Kates-Harbeck, A. Svyatkovskiy, and W. Tang, *Predicting disruptive instabilities in controlled fusion plasmas through deep learning*, *Nature* **568**.7753 (2019) 526–531.

[18] K. L. van de Plassche et al., *Fast modeling of turbulent transport in fusion plasmas using neural networks*, *Physics of Plasmas* **27**.2 (2020) 022310.

[19] S. I. Braginskii, *Transport Processes in a Plasma*, *Reviews of Plasma Physics* **1** (Jan. 1965) 205, URL: https://ui.adsabs.harvard.edu/abs/1965RvPP....1..205B.

[20] P. C. Stangeby et al., *The plasma boundary of magnetic fusion devices*, vol. 224, Institute of Physics Pub. Philadelphia, Pennsylvania, 2000.

[21] K. U. Riemann, *The Bohm criterion and sheath formation*, *J. Phys. D: Appl. Phys.* **24** (1991) 493–518, ISSN: 0022-3727, DOI: 10.1088/0022-3727/24/4/001.

[22] R. Chodura, *Physics of Plasma–Wall Interactions in Controlled Fusion*, ed. by D E Post and R Behrisch, 1986, 99.

[23] K. Miyamoto, *Plasma Physics and Controlled Nuclear Fusion*, vol. 38, 2005, DOI: 10.1007/3-540-28097-9, URL: https://ui.adsabs.harvard.edu/abs/2005ppcn.book.....M.

[24] S. Brezinsek et al., *Chemically assisted physical sputtering of Tungsten: Identification via the* $^6\Pi \rightarrow ^6\Sigma^+$ *transition of WD in TEXTOR and ASDEX Upgrade plasmas*, 2019 50–55, ISSN: 2352-1791, DOI: 10.1016/j.nme.2018.12.004.

[25] P. C. Stangeby, *Basic physical processes and reduced models for plasma detachment*, 2018 044022, ISSN: 0741-3335, DOI: 10.1088/1361-6587/aaacf6.

[26] H. Zohm, *Edge localized modes (ELMs)*, 1996 105–128, ISSN: 0741-3335, DOI: 10.1088/0741-3335/38/2/001.

[27] B. LaBombard, *Experimental investigation of transport phenomena in the scrape-off layer and divertor*, 1997 149–166, ISSN: 0022-3115, DOI: 10.1016/s0022-3115(96)00502-8.

[28] F. Reimold et al., *Divertor studies in nitrogen induced completely detached H-modes in full tungsten ASDEX Upgrade*, 2015 033004, ISSN: 0029-5515, DOI: 10.1088/0029-5515/55/3/033004.

[29] A. J. Wootton et al., *Fluctuations and anomalous transport in tokamaks*, 1990 2879–2903, ISSN: 0899-8221, DOI: 10.1063/1.859358.

[30] T. Eich et al., *Scaling of the tokamak near the scrape-off layer H-mode power width and implications for ITER*, 2013 093031, ISSN: 0029-5515, DOI: 10.1088/0029-5515/53/9/093031.

[31] B. Sieglin et al., *Investigation of scrape-off layer and divertor heat transport in ASDEX Upgrade L-mode*, 2016 055015, ISSN: 0741-3335, DOI: 10.1088/0741-3335/58/5/055015.

[32] X. Bonnin et al., *Presentation of the New SOLPS-ITER Code Package for Tokamak Plasma Edge Modelling*, 2016 1403102–1403102, ISSN: 1880-6821, DOI: 10.1585/pfr.11.1403102.

[33] S. Wiesen et al., *The new SOLPS-ITER code package*, 2015 480–484, ISSN: 0022-3115, DOI: 10.1016/j.jnucmat.2014.10.012.

[34] B. J. Braams, *Radiative Divertor Modelling for ITER and TPX*, 1996 276–281, ISSN: 0863-1042, DOI: 10.1002/ctpp.2150360233.

[35] D. Reiter, M. Baelmans, and P. Börner, *The EIRENE and B2-EIRENE Codes*, 2005 172–186, ISSN: 1536-1055, DOI: 10.13182/fst47-172.

[36] E. Kaveeva et al., *SOLPS-ITER modelling of ITER edge plasma with drifts and currents*, 2020 046019, ISSN: 0029-5515, DOI: 10.1088/1741-4326/ab73c1.

[37] E. Kaveeva et al., *Speed-up of SOLPS-ITER code for tokamak edge modeling*, 2018 126018, ISSN: 0029-5515, DOI: 10.1088/1741-4326/aae162.

[38] I. Veselova et al., *SOLPS-ITER drift modelling of ITER burning plasmas with narrow near-SOL heat flux channels*, 2021 100870, ISSN: 2352-1791, DOI: 10.1016/j.nme.2020.100870.

[39] A. S. Kukushkin et al., *Finalizing the ITER divertor design: The key role of SOLPS modeling*, 2011 2865–2873, ISSN: 0920-3796, DOI: 10.1016/j.fusengdes.2011.06.009.

[40] F. Reimold et al., *Experimental studies and modeling of complete H-mode divertor detachment in ASDEX Upgrade*, *Journal of Nuclear Materials* **463** (2015) 128–134.

[41] S. Wiesen et al., *Plasma edge and plasma-wall interaction modelling: Lessons learned from metallic devices*, 2017 3–17, ISSN: 2352-1791, DOI: 10.1016/j.nme.2017.03.033.

[42] L. Breiman, *Random forests*, *Machine learning* **45**.1 (2001) 5–32.

[43] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, Springer Science & Business Media, 2009.

[44] F. Pedregosa et al., *Scikit-learn: Machine Learning in Python*, *Journal of Machine Learning Research (2011)* (Jan. 2, 2012), arXiv: 1201.0490v4 [cs.LG].

[45] L. Li, *Classification and Regression Analysis with Decision Trees*, 15. May 2019, Accessed: 07.04.2021 12:10, URL: https://towardsdatascience.com/https-medium-com-lorrli-classification-and-regression-analysis-with-decision-trees-c43cdbc58054.

[46] L. Breiman et al., *Classification and Regression Trees*, 1984, ISBN: 978-0-412-04841-8, DOI: 10.4135/9781412950589.n88.

[47] S. K. Murthy, *Automatic construction of decision trees from data: A multi-disciplinary survey*, *Data mining and knowledge discovery* **2**.4 (1998) 345–389.

[48] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2009, ISBN: 978-1-4939-3848-8, DOI: 10.1007/978-0-387-45528-0.

[49] M. A. Nielsen, *Neural networks and Deep Learning*, *Determination Press* (2015).

[50] C. M. Bishop, *Mixture density networks*, *Neural Computing Research Group Report* (1994), URL: https://publications.aston.ac.uk/id/eprint/373/1/NCRG_94_004.pdf.

[51] *Course notes on Stanford CS class CS231n: Convolutional Neural Networks for Visual Recognition*, Accessed: 11.04.2021 11:30, URL: https://cs231n.github.io/neural-networks-1/.

[52] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)*, 2015, arXiv: 1511.07289v5 [cs.LG].

[53] X. Glorot, A. Bordes, and Y. Bengio, *Deep Sparse Rectifier Neural Networks*, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ed. by Geoffrey Gordon, David Dunson, and Miroslav Dudík, vol. 15, Proceedings of Machine Learning Research, Fort Lauderdale, FL, USA: PMLR, 2011, 315–323, URL: http://proceedings.mlr.press/v15/glorot11a.html.

[54] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning representations by back-propagating errors*, 1986 533–536, ISSN: 0028-0836, DOI: 10.1038/323533a0.

[55] L. Bottou, *Stochastic Gradient Learning in Neural Networks*, *Proceedings of Neuro-Nimes* (1991).

[56] X. Glorot and Y. Bengio, *Understanding the difficulty of training deep feedforward neural networks*, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (2010).

[57] N. Srivastava et al., *Dropout: a simple way to prevent neural networks from overfitting*, *The journal of machine learning research* **15**.1 (2014) 1929–1958.

[58] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, Adaptive Computation and Machine Learning, Cambridge, MA, USA: MIT Press, Jan. 2006, 248.

[59] M. Brenzke et al., *Divertor power load predictions based on machine learning*, 2021 046023, ISSN: 0029-5515, DOI: 10.1088/1741-4326/abdb94.

[60] A. Kallenbach et al., *Divertor power load feedback with nitrogen seeding in ASDEX Upgrade*, *Plasma Physics and Controlled Fusion* **52**.5 (2010) 055002.

[61] G. M. Staebler and F. L. Hinton, *Currents in the scrape-off layer of diverted tokamaks*, 1989 1820–1824, ISSN: 0029-5515, DOI: 10.1088/0029-5515/29/10/017.

[62] A. Kallenbach et al., *Partial detachment of high power discharges in ASDEX Upgrade*, 2015 053026, ISSN: 0029-5515, DOI: 10.1088/0029-5515/55/5/053026.

[63] F. Ryter et al., *H-Mode operating regimes and confinement in ASDEX-Upgrade*, 1995 643–646, ISSN: 0031-8949, DOI: 10.1088/0031-8949/51/5/017.

[64] A. Mlynek et al., *Design of a digital multiradian phase detector and its application in fusion plasma interferometry*, 2010 033507, ISSN: 0034-6748, DOI: 10.1063/1.3340944.

[65] Y. Camenen et al., *Impact of plasma triangularity and collisionality on electron heat transport in TCV L-mode plasmas*, 2007 510–516, ISSN: 0029-5515, DOI: 10.1088/0029-5515/47/7/002.

[66] L. Giannone et al., *Improvements for real-time magnetic equilibrium reconstruction on ASDEX Upgrade*, 2015 519–524, ISSN: 0920-3796, DOI: 10.1016/j.fusengdes.2015.07.029.

[67] S. S. Henderson et al., *An assessment of nitrogen concentrations from spectroscopic measurements in the JET and ASDEX upgrade divertor*, 2019 147–152, ISSN: 2352-1791, DOI: 10.1016/j.nme.2018.12.012.

[68] A. Kallenbach et al., *Parameter dependences of the separatrix density in nitrogen seeded ASDEX Upgrade H-mode discharges*, 2018 045006, ISSN: 0741-3335, DOI: 10.1088/1361-6587/aaab21.

[69] A. F. Siegel, *Robust regression using repeated medians*, 1982 242–244, ISSN: 0006-3444, DOI: 10.1093/biomet/69.1.242.

[70] C. Molnar, *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*, https://christophm.github.io/interpretable-ml-book/, 2019.

[71] J. Stober et al., *Effects of triangularity on confinement, density limit and profile stiffness of H-modes on ASDEX upgrade*, 2000 A211–A216, ISSN: 0741-3335, DOI: 10.1088/0741-3335/42/5a/324.

[72] T. Luda et al., *Integrated modeling of ASDEX Upgrade plasmas combining core, pedestal and scrape-off layer physics*, 2020 036023, ISSN: 0029-5515, DOI: 10.1088/1741-4326/ab6c77.

[73] M. Abadi et al., *TensorFlow: A system for large-scale machine learning*, 2016, arXiv: 1605.08695v2 [cs.DC].

[74] F. Chollet et al., *Keras*, https://keras.io, 2015.

[75] K. He et al., *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*, 2015, arXiv: 1502.01852v1 [cs.CV].

[76] X. Glorot and Y. Bengio, *Understanding the Difficulty of Training Deep Feedforward Neural Networks*, 2010.

[77] J. Duchi, E. Hazan, and Y. Singer, *Adaptive Subgradient Methods for Online Learning and Stochastic Optimization*, *Journal of Machine Learning Research* (2011).

[78] S. Ioffe and C. Szegedy, *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, 2015, arXiv: 1502.03167v3 [cs.LG].

[79] T. Akiba et al., *Optuna: A Next-generation Hyperparameter Optimization Framework*, *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[80] G. Klambauer et al., *Self-Normalizing Neural Networks*, 2017, arXiv: 1706.02515 [cs.LG].

[81] J. Bergstra et al., *Algorithms for Hyper-Parameter Optimization*, Dec. 2011.

[82] G. Hinton, *Lecture 6e rmsprop: Divide the gradient by a running average of its recent magnitude*, Accessed: 12.06.2021 12:20, URL: https://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf.

[83] U. von Toussaint, S. Gori, and V. Dose, *Invariance priors for Bayesian feed-forward neural networks*, 2006 1550–1557, ISSN: 0893-6080, DOI: 10.1016/j.neunet.2006.01.017.

[84] J. Wågberg et al., *Prediction performance after learning in Gaussian process regression*, 2016, arXiv: 1606.03865v3 [stat.ML].

[85] C. Fiedler, C. W. Scherer, and S. Trimpe, *Practical and Rigorous Uncertainty Bounds for Gaussian Process Regression*, 2021, arXiv: 2105.02796v1 [cs.LG].

[86] D. Sivia and J. Skilling, *Data analysis: a Bayesian tutorial*, OUP Oxford, 2006.

J

ACKNOWLEDGEMENTS / DANKSAGUNG

First, I would like to thank Prof. Dr. Alexander Pukhov for agreeing to supervise and referee this thesis. I would also like to thank Prof. Dr. Yunfeng Liang for being the secondary referee for this thesis and for his valuable input that helped improve the data selection.
Furthermore, I want to thank Dr. Sven Wiesen for the day-to-day supervision of this thesis and his help with organizational matters.
For valuable discussions and support I want to thank Dr. Matthias Bernert, Dr. David Coster, Dr. Jenia Jitsev and Dr. Udo von Toussaint. Moreover, I would like to thank Dr. Dirk Reiser for our discussions and his interest in machine learning and statistical methods.
Furthermore, I want to thank the ASDEX Upgrade team for their work and for granting me access to their experimental data.
I would also like to thank Dieter Boeyaert for providing occasional diversion during some dull office hours.
For their continued and steadfast support and friendship I want to thank Jan Paul Koschinsky, Fabian Lange and Tim Sprenger. Our time together was the best part of university.
For proof reading this thesis I would like to thank Dr. Dirk Reiser, Dr. Sven Wiesen, Frederik Brenzke, Fabian Lange and Tim Sprenger.

Abschließend gilt mein größter Dank meiner Familie, deren Unterstützung mir im Studium und auch während dieser Arbeit immer eine verlässliche Konstante war. Insbesondere möchte ich meinen Eltern, Brigitte und Alfred, für alles, das sie für mich getan haben, und für das Umfeld, in dem ich dank ihnen aufwachsen durfte, danken. Außerdem gilt ein besonderer Dank meinem Bruder Frederik, der mir schon zu Schulzeiten ein Vorbild war und mich durch meine akademische Laufbahn hindurch motiviert und unterstützt hat.
Meiner ganzen Familie gilt mein tiefster und herzlichster Dank für ihre Hilfe und den uneingeschränkten Rückhalt, der mich maßgeblich dazu befähigt hat mein Studium und diese Arbeit zu absolvieren.