Universiteit Leiden

Master Computer Science

The Bigger Fish
A comparison of state-of-the-art QSAR models on low resourced Aquatic Toxicity Regression Tasks

Name:	Thalea Schlender
Student ID:	s2970775
Date:	07/08/2022
Specialisation:	Artificial Intelligence
1st supervisor:	Dr. Jan Van Rijn
2nd supervisor:	Prof.Dr. Holger Hoos
External Supervisors:	
:	Prof.Dr. Willie Peijnenburg,
:	Dr. Markus Viljanen,
:	Dr. Albert Wong

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Quantity Structure-Activity relationship (QSAR) models aim to predict a biological activity of a chemical structure on a certain target. This makes QSAR models vital in many areas of society: they are a key step in drug development and provide risk assessment for environmental protection. One prevalent use here is for aquatic toxicity prediction, as the European Union requires chemicals exposed to the environment to be registered with aquatic toxicity information via (simulated) experiments. For this, a chemical's impact on e.g. mortality or mobility must be considered for different aquatic species. As the European Union suggests the use of QSAR models, there is a dire need for predictive models that can model the threat to the aquatic ecosystem better. These models can then take a step towards mitigating the need for in vivo experiments on aquatic species.

Aquatic toxicity dataset collections consist of many related tasks, each predicting the toxicity of compounds for a given species. Since many of these tasks are inherently low-resource, i.e. hold few associated compounds, the problem lends itself to meta-learning techniques that make use of information captured across tasks. In our work, we use ten state-of-the-art QSAR models with a focus on knowledge sharing between species. Specifically, we employ and compare: transformational machine learning, model agnostic meta-learning, fine-tuning, as well as multitask models.

Furthermore, an in-depth investigation of the methods' ability to perform in low-resourced situations is performed, in which we find that knowledge sharing techniques outperform the single-task approaches. Finally, we advise the use of the multitask random forest model for aquatic toxicity QSAR modelling, but also for low-resource QSAR modelling in general. The multitask random forest model is never statistically outperformed in our general comparisons, and performs robustly in the low-resource settings. With this, we successfully build a multitask model on a species level which predicts toxicity for multiple species across phyla with flexible exposure duration and on a large chemical applicability domain.

Contents

Abbreviations

- QSAR: Quantity Structure-Activity Relationship
- TML: Transformational Machine Learning
- MAML: Model Agnostic Meta-Learning
- LC50: Concentration of a compound at which 50% of a species dies
- RMSE: Root Mean Squared Error

1 Introduction

With the advent of machine learning, the field of Cheminformatics has flourished by using data science techniques on physical-chemical problems. One such problem is the modelling of the bio-activity related to molecular compounds. Known as Quantitative Structure-Activity Relationship (QSAR) modelling, the field aims to prioritise *in vivo* - in test tube - and *in vitro* - in organism - experiments via cost effective *in silico* simulated approaches. QSAR models relate chemical structures to their biological activity in a certain target domain, from full organisms to specific proteins and even to specific genes. The research in this field has been motivated for decades by the aim of reducing experiments that are expensive in terms of life, cost, and time (see, e.g., [8]).

A large field benefiting from QSAR modelling is drug design. When searching molecular compounds for suitable candidates for therapeutic purposes, QSARs are used to predict further molecular descriptors such that they can be used as input for virtual screening processes [48]. Further, predicting the effect of a compound on a protein in the human body aids in the discovery of lead compounds in medicines. In the pre-clinical medicinal chemistry phase of drug development, QSARs can also be used to optimise the binding affinity of a candidate drug to a target in the body of an organism [52]. In general, in silico experiments can help select in vivo or in vitro experiments with the most chances of success.

The biological activities which QSAR models aim to predict are myriad and domain-specific. This thesis will, henceforth, focus on toxicity as a biological effect solely. Toxicity can be measured by the impact a compound has on a species' mortality, reproduction, mobility or growth. Specifically, the aquatic toxicity causing mortality in species is addressed in our work. The prediction of aquatic toxicity as a biological activity has its prevalent use in risk assessment for environmental protection. It aims to define safe quantities of chemicals to be released into the environment with little impact on the aquatic ecosystem. With the increasing amount of industrial chemicals being developed, the European Union Regulation for the Registration, Evaluation, Authorisation and Restriction of Chemical Substances (REACH) requires an investigation into the aquatic toxicity of a chemical [30]. Due to this regulation, there is a dire need for better performing aquatic toxicity QSAR models, that predict the toxicity of various aquatic species from water flees (so-called Daphnia) over algae to fish.

Research in machine learning that aims to develop approaches to share knowledge across datasets has become more sophisticated over the last decades. Meta-learning, for instance, shares knowledge by building priors across datasets that can be used to address a new task with. Another knowledge-sharing technique is multitask learning, in which multiple tasks are learnt jointly to share knowledge during training a model. The focus on sharing knowledge between related tasks has led to research considering its use in QSAR modelling. We believe the use of these techniques could be beneficial in utilising and predicting the many low-resource datasets inherent to aquatic toxicity. As such, we investigate the state-of-the-art approaches to QSAR modelling and apply these methods to a species-level aquatic toxicity prediction model.

Specifically, we aim to model the toxicity of many aquatic species individually in a general applicability model, which makes no restriction on its chemical input. With the use of a diverse set of state-of-the-art approaches, single-species models and multi-species models are compared and investigated, as well as the benefit of using meta-learning techniques. We advise future QSAR developers by investigating the impact of low-resourced situations on the modelling techniques, as well as recommend QSAR models to use for aquatic toxicity.

This is done, specifically, by answering the following questions:

- What is the current state of the art in QSAR modelling?
- How do state of the art QSAR methods compare when applied to aquatic toxicity regression problems?
- To what extent does the number of resources, in terms of the number of assays per species and the number of species themselves, impact the performance of QSAR models?
- How well do the QSAR models perform on actual low-resource datasets?

We address these questions by using an array of methods to predict the real-valued concentration at which 50% of an aquatic species dies (LC50). Using a dataset collection gathered from ECOTOX consisting of 24 816 assays, 351 separate species, and 2674 chemicals, a general comparison of the state-of-the-art models is made on internal and external validation. To investigate the models robustness towards low-resource situations, we artificially down sample datasets to imitate modelling situations that have few assays per species, as well as few species to share knowledge between.

By doing so, we find that knowledge sharing models outperform single-task models. As such, we propose a general model on a species level which predicts toxicity with flexible exposure duration on a large chemical applicability domain. Following the in-depth learning curve analysis, the gained insight into the individual methods' capacities to model low resource situations is not only useful for aquatic toxicity prediction, but also for QSAR modelling in general.

The rest of the thesis is structured as follows: first, a problem statement is given defining the issue being addressed and its surrounding guidelines. Then, the background of our work is given, which includes explaining the intuition of meta-learning, before elaborating on its use in QSAR modelling. We discuss the research into aquatic toxicity modelling and present the data used in our work. Following this, the solutions used are given in the methodology section, and the experiments performed are outlined. Finally, the results are discussed and conclusions are drawn.

2 Problem Statement

This section further elaborates on the aquatic toxicity problem our work deals with, before addressing the domain specific OECD guidelines that guide QSAR model development.

2.1 Aquatic Toxicity Problem

In 2007 the European REACH guidelines introduced the necessity that toxicological information must be given for a chemical at its registration, for instance through QSAR models [20]. The REACH program requires chemicals to be registered along with toxicity information, in order to assess the risk presented by the chemicals' output into the environment. For this, there is a dire need for regulatory models that predict the threat to the aquatic ecosystem. With the aim of reducing experiments on organisms, *in silico* tools should predict the toxicity of a compound from the exposure duration and the species that it is tested on. Representing the aquatic ecosystem, regulatory tools may provide toxicity levels for three representative groups: 'acute fish toxicity', 'acute daphnid toxicity', and 'alga toxicity' [39, 53]. Our work

aims to build a model that predicts toxicities on a species level across fish, daphnid and algae groups. Specifically, using data derived from the ECOTOX dataset collection, 351 species are predicted, for which phylum and class information is available. Phyla are larger subgroups of the species, whereas classes are a more finely-grained taxonomy.

Further, the QSAR models developed in our work are aimed at having a large applicability domain for compounds, that is that a large chemical space should be predicted. Additionally, the concentrations of compounds which are mortal to 50% of a species are predicted across exposure durations. As such, our model predicts acute and chronic toxicities for all species, as well as leveraging more data across different exposure duration whilst training the models.

Moreover, a model with adaptable exposure duration on the species level could also be used for modelling species sensitivity distributions. This distribution relates the concentration of a compound to the percentage of aquatic species affected. For this statistical approach, it is thus vital to have 50% mortality concentration values from many aquatic species. A model with adaptable exposure duration for many aquatic species can, thus, provide multiple concentration values. Using the species sensitivity distributions curve, the concentration at which 5% of the aquatic species is affected can be found. With sufficiently many species, the species sensitivity distribution can represent the aquatic ecosystem and the 5% mark may be used in environmental risk assessment [16]. These models typically use experimental data across different experimental conditions and thus suffer from uncertainty due to the variance in these conditions. By using a species model, one could model the toxicity of many species and derive multiple values for a species by changing the experimental duration. Although species sensitivity distributions are an important tool in ecotoxicological risk assessment, the application of our model for this is left as future work.

In summary, the aquatic toxicity model built is formally solving the following problem: given a compound c, a duration d and a target species s, predict the lethal concentration of c for 50% of s. Each compound is represented by physical-chemical features and molecular embeddings, whereas each target species has taxonomical information on its phylum and class group.

Ours is the first work to create models that are capable of predicting multiple species across phyla on a large chemical domain with adjustable exposure durations. Further, a comparison between all major QSAR approaches is made, many of which are novelly applied to the aquatic toxicity domain. To further advise model development in aquatic toxicity prediction, an in-depth focus is put on how well models use low-resource data. The conclusions drawn are not only helpful for models within the aquatic toxicity domain but are likely applicable to other QSAR modelling areas too.

2.2 OECD Guidelines

With the heightened relevance of the QSAR models through the REACH guidelines, the need for validated QSAR models of high quality grew. Addressing this, the OECD principles [39] were introduced and adapted, presenting points that regulatory QSAR models should adhere to. Although our work does not aim to present a model for regulatory purposes but rather aims to advise future development, it is aware of the OECD QSAR principles and will address the principles in this section.

To ensure that researchers can assess the potential use of a validated QSAR model, the first OECD principle advises an endpoint to be specified. An endpoint in QSAR modelling is

the subject that is being predicted, such as biological effects, which are often found through specific test protocols. Declaring an endpoint allows the QSAR model to be used only for the endpoint it was trained on and designed for. When focusing on aquatic toxicity, the endpoint for the ECOTOX QSAR model is in the category of ecological effects and is included in the endpoints needed for regulatory assessment. Specifically, the endpoints are 'acute fish toxicity', 'acute daphnid toxicity', 'alga toxicity' and 'long term aquatic toxicity' [39]. These endpoints are addressed in a 'general (Q)SAR model(s) based upon a common toxic effect' [39] of aquatic species, which predicts the LC50 concentration.

Further, a validated QSAR model needs to provide reproducibility and transparency. To this end, the OECD principles require the description of an 'unambiguous algorithm': the algorithm that connects the molecular activity, in our case the ecological effects, and the chemical descriptors [39]. This requires QSAR models to not only describe the employed algorithm itself, but also the dataset with the chemical descriptors and their derivation. The latter is addressed in section 4.3, whereas the models are described in the section 5. As mentioned earlier, black-box models are employed, specifically the neural network-based methods. By nature, these methods do not give unambiguous transparency of how a prediction is derived. However, the OECD guidance document mentions artificial neural networks as complying with this OECD principle and advises paying close attention to the validation processes of these methods to monitor their reliability. Moreover, all datasets with chemical descriptions, as well as the implementations of the models, will become publicly available. An additional emphasis is also placed on the reproducibility of the training data, which is an invaluable aspect in QSAR modelling [39].

To define when a QSAR model may validly be employed, a QSAR model should include a domain of applicability. The domain of applicability - defined in the chemical structure space - should be determined systematically to ensure that a model is not forced to extrapolate into unintended domains. Ideally, the domain of applicability is defined prior to the QSAR modelling, so that the training set of the model is designed for its domain. Our work, however, addresses the issue that QSAR models are used outside of their applicability domain for low-resource datasets, which do not have enough resources to build a model on. Hence, our work deliberately aims to develop a general applicability model on the given datasets by including different experimental setups and all applicable chemicals. It is important to note that the training set of a QSAR model always induces a domain of applicability. This domain of applicability is defined by the similarity of a chemical and the training set and is, thus, dependent on both the chemical descriptors and similarity measures chosen [39]. Although measuring the domain of applicability is left as future work, it is interesting to note that toxicology datasets have natural biases. Under the REACH programme, for instance, chemicals of over 1-ton production volume need to be registered with toxicological information [59]. Hence, datasets include biased information on chemicals that are produced at a higher volume, whereas chemicals under the threshold avoid testing, although their acute toxicity may be more concerning [59].

Finally, a QSAR model's performance must be measured and validated soundly, paying special attention to robustness and predictive capacity. To assess the stability of predictions, a model's robustness is typically measured by building partial models on the training set [39]. A model's predictive capacity is seen by its performance when extrapolating to an external test set not used for model development. In our work, 20% of the dataset is held out as an external test set, whereas 80% of the dataset remains for internal validation. The dataset is split on chemicals, such that each split has mutually exclusive chemicals. This is done to test

the QSAR models on the real-world challenge of predicting the toxicity of unseen chemicals, for instance at their registration. The internal validation is done by 5-fold cross-validation, a technique which splits the training set into 5 mutually exclusive chemical groups. Iterating over all sections, the model is then built on four sections, whilst the fifth section is predicted by the partial model. The external test set is predicted by the model trained on the whole internal training set. The performances are measured by the root mean squared error and the percentage of estimates within factors of ten and two of the actual LC50 value (see section 6.1).

To reiterate, the development of QSAR pipelines must be described and validated rigorously to ensure that they are robust, transparent and reproducible. All of the resources needed to recreate our work are available under https://git.liacs.nl/s2970775/The_Bigger_Fish-Aquatic_Toxicity_QSAR_models.

3 Background

Having presented the motivation and details of the aquatic toxicity problem investigated in our work, the background of QSAR modelling, in general, is introduced in this section. To address the low-resourced QSAR domains, the intuition behind modern branches of machine learning, such as meta-, transfer- and multitask- learning are presented, before its use in QSAR modelling so far is elaborated on. Finally, we return to the aquatic toxicity domain, where we show its solutions and advances thus far.

3.1 Definitions

Throughout the QSAR literature, terminology has been used in an ambiguous and potentially contradicting fashion. To establish a common basis to continue with, we now clarify key terminology and their definitions as assumed in our work.

- *QSAR Task*: The notion of a task has been used to discriminate various QSAR problems. Some refer to a task to be different datasets with different endpoints and targets, whereas others describe tasks as different endpoints exclusively. We specify a task to be a QSAR problem that predicts a biological effect e for molecular compounds on a target t. Varying either the effect e and/ or the target t would thus constitute a new task. In our problem of aquatic toxicity prediction, the endpoint remains the same, whereas multiple target species are to be predicted. Hence, each new target species is a further task.
- *Fingerprint*: A widely-used group of molecular embeddings in the form of real-valued or binary vectors.
- *Assay*: An assay refers to individual bio-activity tests performed. In the aquatic toxicity prediction in our work, an assay refers to the combination of chemical compounds, exposure duration, and target species.

3.2 Quantitative Structure Activity Relationship Modelling

QSAR models aim to predict a biological activity of a chemical structure on a certain target. This makes QSAR models vital in many areas of society: they are a key step in drug development and provide risk assessment for environmental protection. As such, many QSAR tasks exist, that predict myriad molecular attributes, or effects on targets ranging from

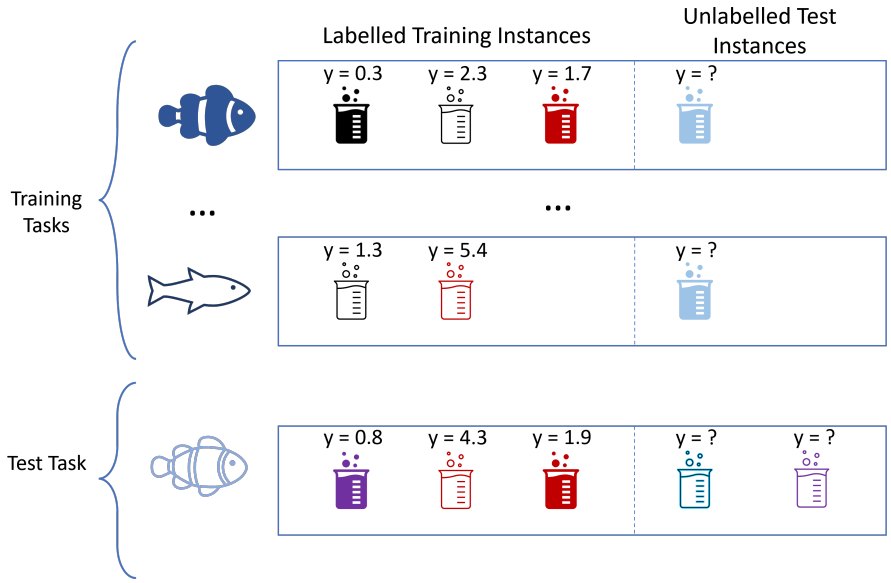


Figure 1: Aquatic toxicity QSAR tasks: The set up of the individual aquatic species tasks. The image shows how the tasks can be used for meta-learning: using the tasks in the training set to acquire prior knowledge for the test task to be trained.

proteins to entire organisms. QSAR tasks can either be cast into classification problems that typically predict whether a compound is *active* or not, or be cast as regression problems that typically give a real-valued concentration of a compound to achieve a given biological effect. Although the classification of active compounds is a popular research problem, the classification problem requires an established threshold between activity classes, whereas the regression task outputs can be interpreted as needed.

Over decades traditional methods dominated QSAR modelling. This typically involved using relationships of physical-chemical attributes of compounds inherent to a biological activity. These relationships were then captured in hand-crafted linear regression models. More open-source data and the rise of machine learning have consequently led to the use of further data science techniques for QSAR modelling. Building a model for each task, many approaches have been applied, compared, and have become the new standard. Mayr et al. [2018] compare different QSAR models using machine learning methods. Specifically, they compare the use of the most popular QSAR algorithms: k-nearest neighbour algorithm, support vector regression, random forests, and artificial neural networks. These QSAR methods have also been applied in regulatory tools [2].

Although the use of traditional single-task machine learning models has become well established, the many related and low-resourced tasks inherent to QSAR modelling call for techniques to share knowledge amongst them [52, 6]. In the next section, the domain of meta-, transfer-, and multitask learning is introduced, before its application to QSAR modelling so far is described. Finally, work related to the aquatic toxicity prediction is presented.

3.3 Meta-Learning, Transfer Learning, and Multitask Learning

Although machine learning models have accomplished (beyond) human-level performance in some domains, their deficiency lies in their cost: Machine learning models typically require an abundance of data, as well as substantial training time to perform well. Meta-learning

attempts to address this issue by asking *how to learn to learn tasks?* For this, meta-learning borrows intuition from how humans learn and solve problems. Instead of learning each task independently and anew, humans approach each challenge with prior knowledge [28].

A simple example of this is learning to recognise a new object, say a certain breed of dog. A human could use its knowledge in fundamental things like object detection and depth perception but also its knowledge of how different animals and dogs in general appear and could, thus, learn to recognise the new breed after seeing a few examples. In contrast to a human, which essentially uses its collection of prior knowledge to finetune on certain tasks [28], a machine learning algorithm would need to learn these things from scratch. Hence, the idea of meta-learning is to also develop some prior knowledge to approach new problems with.

Aiming to learn across datasets, meta-learning uses priors from previously learned tasks to learn a new task more efficiently. Figure 1 shows an example of what the many separate tasks for meta-learning could look like. Ideally, there would be many related tasks in the training tasks set, such that the test tasks can be learned quickly and effectively using the priors established.

A traditional approach of meta-learning is algorithm selection, in which a model predicts which algorithm to use for a new task given some task features. More formally, may the problem space P contain all related QSAR tasks available and the corresponding feature space F hold descriptions on those separate QSAR tasks. Note that these descriptions can comprise features that are domain-specific to the task or may hold dataset information, such as its size. Then, given an algorithm space A containing the candidate algorithms, and a performance metric y, algorithm selection aims to select the algorithm $a \in A$ yielding the highest value of y for a given QSAR task $p \in P$ given its features $f \in F$ [40].

More recently, three main fields of meta-learning have developed: metric-, model- and optimisation-based meta-learning. Similar to the k-nearest neighbour algorithm, metric-based meta-learning uses distances between input samples to predict the label that the closest known input has. For this, the approach typically learns an embedding space to encode the input samples into. In this space, the distances between a new sample and all labelled samples can be measured. An example of this approach is Siamese Networks, which use two neural networks to encode the input sample and a known sample into an embedding space [23].

The next branch is model-based meta-learning, which is typically a black box neural network model. The model is specifically designed to learn new tasks fast. To do so, the models typically read the task samples sequentially and can then adapt the state of the model to the new task quickly [23]. Therefore, the prior of the other tasks is in the model itself.

Finally, optimisation-based meta-learning adjusts the way models are optimised for new tasks. Typically, this is done by optimising for a task specifically, whilst also optimising the optimisation process across tasks [23]. Possibly the most popular optimisation-based meta-learning technique is model agnostic meta-learning (MAML), proposed by Finn et al. [2017] and used in our work. MAML can be used on any model that uses gradient descent to optimise its parameters. MAML tries to find good initialisation parameters, such that when a model is built for a new task, the initialisation parameters can quickly adapt to good parameters for the new problem. As such, MAML optimises for a task itself, but also optimises the initialisation parameters used. The knowledge of prior related tasks is then encapsulated in the initialisation parameters.

Similar to optimisation-based meta-learning, transfer learning [57] uses prior knowledge of tasks embedded into the parameters of a model. Typically using neural networks, transfer learning trains a model on source tasks and then finetunes the model on a further task [58]. For instance, an approach called fine-tuning trains a neural network on source tasks, such that the feature extraction and internal representation are captured by the weights of the network. Then, the (head of the) network can be finetuned on a specific task from the given weights efficiently [22].

In contrast to meta- and transfer-learning techniques that learn related tasks sequentially, multitask learning is an approach that learns multiple tasks simultaneously. The intuition behind this is that by learning multiple tasks with the same representation jointly, the knowledge learnt can be generalised and of use to all tasks [7]. Although this approach does not use prior information, it shares related task knowledge during training, by learning multiple tasks jointly. That is that by sharing a (large part of a) model, all tasks may make use of the same internal representations and insights.

Often this approach is embedded in a neural network, which may have one output for all tasks, but typically has an output node for each individual task. These neural networks aim to learn multiple tasks simultaneously, by sharing the input and hidden layers of the network between tasks, and, thus, sharing the weights and representations of the input, here: the molecular compounds. When using multiple output nodes, the connections to the output layers can then capture task-specific information.

Although these methods have been applied successfully, an important restriction to make is that meta-, transfer- and multitask learning perform best when the tasks are related. When tasks are too unrelated, and present new phenomena respectively, prior experience cannot be leveraged [58].

3.4 Advances of Meta-, Transfer-, and Multitask-Learning in QSAR Modelling

With the success of transfer learning techniques in, e.g. natural language processing or image analysis, its potential use in QSAR modelling has been recognised [52, 6]. Since training data in the bio-activity domain is gathered by performing in vitro (in test tube) or in vivo (in organism) experiments, the cost of acquiring additional data is high. In fact, QSAR data is highly skewed: in the aquatic toxicity domain, for instance, laboratories use the cheapest (in terms of effort, time and cost) representative of a group of species to test on. This leads to cheaper (sub)species having an abundance of data, whilst other species merely have a handful of toxicity tests associated with them. Given the many low-resource datasets in QSAR modelling, the research community has long called for transfer learning techniques [52, 6] to combine knowledge between datasets.

QSAR modelling lends itself well to transfer learning approaches, as there are many small related tasks, that differ only in a specific endpoint to be predicted, or differ on different targets, within which a chemical's effect is measured (see Figure 1). Due to the fact that QSAR tasks have very similar structures, the aim is to share and generalise the knowledge learnt in the tasks to aid in the general performance across tasks. The case of aquatic toxicity prediction examined in our work aims to predict the mortality of a species caused by a chemical. This issue is split into many tasks: each task refers to a unique target species the effect is to be measured on. The molecular input features and the endpoint remain the same throughout these tasks, such that meta-learning could potentially utilise this task

relatedness and result in better generalisation.

In the following, an overview of work on applying these techniques to QSAR modelling is given.

3.4.1 Algorithm Selection

One approach to utilising the many related QSAR tasks is to deduce which algorithm will work well on a new QSAR problem. Olier et al. [2018] apply algorithm selection to QSAR modelling, specifically to predict biological effects on protein targets gathered from a subset of the open-source ChEMBL dataset collection [19]. Chemical compounds tested on each protein target varied from 10 to 6000 compounds respectively. The regression tasks included predicting the concentrations of a compound causing potency or inhibitory effects on a target protein. Using 8000 QSAR tasks, Olier et al. [2018] predict which algorithm and corresponding pre-processing steps would perform best on the task. Included in the algorithm space were 18 popular machine learning methods for QSAR problems, including linear regression, neural networks, support vector machines, k-nearest neighbour algorithm, and random forests. Olier et al. [2018] find that for the majority of tasks a random forest was suggested and outperformed other methods. Our work takes inspiration from this analysis and includes the single-task random forest model as a representative of the popular single-task machine learning approaches.

3.4.2 Multitask Learning

Erhan et al. [2006] first use a multitask neural network in QSAR modelling in their work on collaborative filtering. Collaborative filtering was initially made to predict ratings users give items in the recommender system domain. Inspired by this, Erhan et al. [2006] propose casting biological targets as users, molecular compounds as items, and the resulting biological activity as ratings. As such, JRank is investigated, a collaborative filtering algorithm in combination with a neural network. Further, a simple multitask network with multiple output nodes is employed. Both approaches aim to predict binary biological activities from Astrazeneca's library. This dataset collection was made using high throughput screening, a technique which allows to automate performing a large number of tests. Astrazeneca's library also included descriptors for the target proteins on which the compounds were tested. Overall, their work found that using these target descriptors aided in the performance of the multitask neural network, but that the single-task learners were strong competitors [13]. A few years later, Dahl et al. [2014] utilise multitask learning to win a binary activity classification challenge in 2015. Predicting both biochemical (in test tubes) and cell type (in cell cultures) assays, they paired their QSAR model with methods to inhibit overfitting.

Ramsundar et al. [2015] also predict binary biological activity using 'massively' multitask neural networks built on over 200 tasks with over 40 million experimental values and varying endpoints. Investigating the performance on each task using their multitask model further, it was found that its performance continues to climb or plateaus with the addition of more datasets. Additionally, the performance benefits from more data, even when the tasks are fixed [45]. As our work will also examine the influence of the number of tasks and amount of data respectively, this result is good to note. However, although this is insightful, no baseline algorithms are shown and the evaluation of training and test splits are not done such that compounds in the respective sets are mutually exclusive. This type of evaluation is overly optimistic and does not reflect real-world requirements.

The development of multitask neural network solutions in QSAR modelling has received a lot of attention, however, later work by Ramsundar et al. [2017] criticises that the research is not suitable for biotech companies and chemical regulators to use. Specifically, the models proposed each have their own implementation and setups, requiring adequate expertise from biotech organisations for its use. Most of the work presented on multitask learning so far uses different QSAR modelling domains, different test settings and varying scales of multitask learning. Additionally, asserting the robustness of multitask models needs to be improved for these models to be used. Ideally, multitask models should always outperform single or baseline methods, although it is acknowledged that research has not achieved this yet [46]. To address the lack of consistency in proposed multitask models and their inherent complexity for biotech domains, Ramsundar et al. [2019] provide their own multitask network implementation as part of the open-source library deepchem. Inspired by open-source projects in the neural network domain, for instance, PyTorch [42], that allowed the use of neural networks to become more accessible, the deepchem project aims to make chemical networks more widely available. It does this by providing support with importing molecular datasets, calculating molecular features and providing QSAR models, such as multitask networks. For this, Ramsundar et al. [2017] provide both multitask models for classification and regression. We adapt the multitask neural network regression model from deepchem to use in our work as a representative of multitask neural networks with multiple output nodes.

Multitask learning can also be performed with other models than neural networks. Sadawi et al. [2019] use multitask learning via random forest models, that have shown to be effective in single-task cases [40]. Using a subset of the ChEMBL dataset collection [19], the tasks in their work predict inhibitory and potency effects on target proteins. Specifically, the concentration of compounds that result in a biological effect on the target protein is predicted as a real value. From ChEMBL, groupings of the target proteins were given, such that over 400 target protein groups existed. Their approach then built a multitask random forest model on each of their protein groups individually. Thus, multiple protein targets of a related protein target group are predicted within one random forest model. Additionally, Sadawi et al. [2019] enhanced their methods by considering a measure of relatedness between proteins in a protein group. For this, each protein pair within a group received an additional input feature that described how similar the protein pairs were based on their amino-acid sequences. They found that these additional features outperformed both single-task models and models without the additional features [50].

Similar to their work, we use a multitask random forest model predicting multiple target species jointly. In contrast to their work, we use a random forest predicting all target species available and do not use different models for each class or phylum of target species. By using the class and phylum taxonomical features, we hope to provide some indication of task relatedness that can be used by the multitask random forest model.

In 2021, Olier et al. [2021] propose a transformational machine learning approach, which takes inspiration from multitask learning, transfer learning and ensemble learning. Using data from the ChEMBL dataset collection [19], multiple inhibitory effects are predicted on various target proteins. The approach aims to learn multitask-specific compound representations. This embedding shares knowledge between all tasks, by encapsulating the general consensus on biological activity. To use the embedding to make predictions for a certain task, a further model can be trained on the embeddings for the given task. With this, the model can make use of the task-specific sensitivities [41]. This approach is included in our work and will be explained in-depth in section 5.3.

The use of many model predictions in the embeddings gives the transformational machine learning approach its similarity to ensemble-learning, an approach that uses many models to form consensus predictions. The fact that all tasks are learnt jointly via the embeddings resembles multitask learning. Its similarity to transfer learning, in which a new task is predicted with knowledge of previously seen tasks, comes from the fact that the transformational representation is formed by many tasks and the final predictive model is used for the new task only [41]. Since ensemble models have performed well for QSAR tasks [40, 51], as well as multitask learning approaches [46, 50], this method is very interesting to include in our work. Although it offers flexibility in which machine learning model to use, we use transformational machine learning in combination with random forest models.

3.4.3 Non-numeric molecular descriptors

All approaches presented as related work so far have been using physical-chemical molecular descriptors or expert-made molecular embeddings, as their input. However, with the overwhelming success of deep learning methods on images, audios and other spatially structured inputs, a substantial amount of research has focused on utilising graphical molecular descriptors in QSAR modelling. These features offer the advantage that end-to-end models can be built meaning that the model itself deduces and selects features from the 'raw' chemical input.

Instead of using embeddings that were developed by domain experts, Jaeger et al. [2018] propose Mol2Vec, a molecular embedding derived in an unsupervised manner. Its inspiration is the Word2Vec [34], which capture the meaning and semantic relationships of words from their context and translate these to real-valued vectors. In Mol2Vec, like in its natural language equivalent, the embedding places molecules with similar structures close to each other in the vector space [25]. Word2Vec embeddings are made by training a neural network on predicting a word given its context. This task can easily be set up and does not require any supervised labels. Once the neural network with one hidden layer is trained, the weights of the hidden layer are used as the embeddings. To create the Mol2Vec embeddings, instead of words, substructures of chemicals are found via the Morgan algorithm [36]. The Morgan algorithm is also used in the development of the molecular fingerprints used in our work and will be addressed later in 4.3. The found substructures are then used like words in the original Word2Vec algorithm. That is that a substructure is predicted given its surrounding substructures, and the weights of the solving neural network are taken as Mol2Vec embeddings. To create compounds, substructure embeddings can be summed together. When compared in downstream QSAR modelling tasks, Jaeger et al. [2018] find that these embeddings can outperform traditional embeddings. A major advantage of the Mol2Vec vector is that, in contrast to the typically sparse handmade embeddings, they offer density [25].

The most common representation of molecules is SMILES (Simplified Molecular-Input Entry-System) [60]. The SMILES representation, developed in the 1980s, can typically be retrieved with a molecule from any biochemical database. Most physical-chemical features, molecular descriptors and molecular embeddings can be accessed from open source software using the SMILES representation. The representation is a string, which describes 2d molecular structures by defining letters or symbols for different atoms, bonds, branches and cyclic structures [60]. Whilst exploring the use of different molecular descriptors, Mayr et al. [2018] propose a long short-term memory model built on SMILES. It was found, however, that these models perform rather poorly.

Moreover, Li & Fourches [2020] use SMILES in their proposed Molecular Prediction Model Fine Tuning *MolPMoFit*: a large pre-trained neural network that can be fine-tuned on a specific learning task. The model was pre-trained on one million compounds from the open-source ChEMBL dataset collection [19] in an unsupervised manner. The network can then be taken to be finetuned on classification and regression tasks.

With the emergence of more sophisticated deep learning techniques, more research has investigated the possibility to apply machine learning techniques directly to the 2-dimensional graphical representation of molecules by using graph neural networks.

Altae-Tran et al. [2017] use an iterative refinement long short-term memory model on a one-shot problem [1]. Using graph-convolutional layers to use the 2-dimensional graphical molecule directly, the model is trained for a one-shot classification problem. A one-shot problem learns to solve problems with one try only. During training, an endpoint and a corresponding support set are sampled. The loss is then calculated on the rest of the endpoints samples. When testing the model, endpoints not used in the training are sampled. Three datasets are used to additionally test the transfer of endpoints of different datasets. Altae-Tran et al. [2017] find that although one-shot learning is successful within some datasets, it struggles when unseen molecular substructures are tested. Further, between some datasets, the transfer of the model trained on one dataset to predict another, performed very poorly [1].

Using a parameter-based transfer learning approach, Nguyen et al. [2020] explore end-to-end GNN further. Using the ChEMBL dataset collection [19], 5 QSAR endpoints are considered: ADME (absorption, distribution, metabolism, and excretion) endpoints, Toxicity, Physicochemical properties, Binding, and Functional properties [38]. To predict these endpoints, graph neural network models are built. A solution for a task can then be defined as the weights of the neural network that solve the task well. The goal of their research is to find good initialisation parameters for a QSAR model via optimisation-based meta-learning. Nguyen et al. [2020] include Model Agnostic Meta-Learning [15], as well as two of its variants, in addition to fine-tuning baselines. How this approach works will be explained in the methodology section, as MAML is also applied to the aquatic toxicity predictions in our work.

Nguyen et al. [2020] test the application of MAML and its variants, in twofold situations: the first tests MAML on tasks predicting the same endpoint, and the latter tests MAML on tasks that predict unseen endpoints. Over 800 tasks with over 128 instances each are used for training. 20 test tasks were held out to test in-distribution performances on the same endpoint, whereas 3 test tasks were held out to test the ability to extrapolate to other endpoints. As all tasks had over 128 instances, it should be noted that the amount of data available is considerably higher than in aquatic toxicity dataset collections. A brief investigation into the effect of low-resource datasets on MAML is made by the authors via sub-sampling of the training data. Specifically, to simulate low-resource settings, the 800 tasks are randomly subsampled to include only 16 instances each. They find MAML outperforms other baselines in both in and out of distribution tasks, as well as across different task sizes [38]. It can be noted, however, that when the task sizes increase, the fine-tuning baselines improve their performances rapidly to be similar to MAML's performances.

3.5 QSAR models for predicting Aquatic Toxicity

The aim of our work is to develop models to predict aquatic toxicity. In other words, this means that the toxicity of a chemical on a species is modelled. For this, the concentration

at which 50% of the species die is measured and, as such, this problem is cast as a regression problem.

Traditionally, aquatic toxicity was modelled via physical-chemical attributes of a compound with a relationship to toxicity. A popular regulatory QSAR model proposed by the Environmental Protection Agency of the United States is called ECOSAR: a regulatory model that uses a linear relationship between chemicals and their toxicity based on the octanol-water coefficient. Building different linear regressions on groups of chemicals, ECOSAR is a non-species-specific tool for aquatic toxicity. It is based on the idea that a chemical that dissolves into the water rather than octanol (fatty substance) will be absorbed less into a species' body and will therefore show fewer signs of toxicity. A large deficit of the model is that the tool requires expertise to handle the output. Since different chemical groups have different linear models, a chemical belonging to more than one group will get multiple toxicity predictions to manually combine. Further, the predictive capability is limited due to the fact that for entire groups of chemicals the toxicity prediction will be the same for all given species. Not only do the predictions of ECOSAR need to be improved [53], large safety factors need to be added to the predictions for their use in risk assessment.

With the rise of machine learning, aquatic toxicity models, like other branches of QSAR modelling, used machine learning models built for singular tasks. Using machine learning models, the Toxicity Estimation Software Tool (T.E.S.T) by the environmental protection agency of the United States is a deployed set of QSAR predictors [31]. Specifically, the regulatory tool allows predictions on specific aquatic toxicity tests, as well as oral rat toxicity, and physical-chemical attributes related to aquatic toxicity. The tool implements acute aquatic toxicity tests on three species: fathead minnow, representing fish, daphnia magna, representing daphnia, and tetrahymena pyriformis representing Algae [31]. Users of T.E.S.T can choose between several methods: hierarchical modelling, in which the weighted average of models made for different chemical substructures is predicted, the use of a single multitask model on the compounds for a task, which can use two types of molecular descriptors, and the nearest neighbour method, which predicts the average of the closest chemicals. Further, an ensemble method of all models above may be used, as well as a model that predicts the mode of action of a compound, before using a model made for that mode of action to predict the toxicity. A further employed regulatory tool is Vega [2], which extends some multiple linear regressions from T.E.S.T, but also implements a neural network, Support Vector Regression and Random Forests for the fathead minnow [2]. Zhou et al. [2021] compare the performance of different open source tools in classifying acute fish and daphnid toxicity and find that ECOSAR's performance is stable. Using Vega, the highest accuracies were achieved. The authors, however, call for research in more sophisticated models [63].

In researching QSAR models, Wu & Wei [2018] apply multitask learning to a toxicity context, including oral rat toxicity and aquatic toxicity. The work criticises that graphical representations of molecules become infeasible for large molecular systems and that topological descriptors summarising the molecule reduce the geometric information drastically. To solve this issue, Element Specific Topological Descriptors are proposed, a molecular descriptor that aims to 'retain crucial biological information during the topological simplification of geometric complexity [of molecules]' [61]. Using multiple models, including the multitask neural network, the performance of the descriptors is compared to the T.E.S.T [31] models. The same species from daphnia, fish, and algae used in T.E.S.T is selected to train a model on, respectively. Although the T.E.S.T baseline compares well against the new methods, the multitask neural network and a combination of this method and gradient boosting trees

perform well [61].

Regulatory tools T.E.S.T and Vega use models to predict the acute toxicity of a representative species for the fish, daphnids, and algae. Work has also gone into investigating models of more general applicability. As such, Lunghini et al. [2020] propose to build a model on fish, daphnids, and algae, respectively. All assays are only associated with the high-level category, such that the species of an assay cannot be determined anymore. For each of these groups, a support vector regressor with a radial basis function kernel is employed. To find the best descriptors in a physical-chemical descriptor space, in addition to optimal hyperparameters, a genetic algorithm-based optimizer is used. All data points for a model have been acquired by using the same experimental setup, but instead of considering just one species, multiple species in that group are considered. Leveraging more data like this, the authors used a combination of all available datasets in literature to test the model, as well as testing the models on an unseen industrial dataset. Their model outperforms ECOSAR, T.E.S.T, and Vega on an unseen industrial dataset of toxicity data.

Lunghini et al. [2020] also find that the ecotoxicological dataset qualities impact the models' performances heavily, a concern also found in other work [56]. Aquatic toxicity values have been gathered over decades in various laboratories causing variation among experimental values. For instance, Raimondo et al. [2010] find that 57%, 86%, 94% of minimum and maximum toxicity values given for the same chemical-species pairs are within a factor of 2, 5, 10, respectively. Hence, next to the inherent experimental bias in the data, the aquatic toxicity datasets can be of low quality. This could be a limitation in contrast to QSAR models in different domains, in which methods like high throughput screening have enabled recent testing of chemicals at a high volume. As we use a subset of the data used in Lunghini et al. [2020]'s work, it is important to note this limitation.

Similar to Lunghini et al. [2020], Sheffield & Judson [2019] build an ensemble learner predicting any fish species. They aim to build a model with large general applicability. For this, they predict both acute and chronic duration, many different experimental setups, as well as all fish species in one model. Using the weighted average of three models, the yielded predictions clearly outperform T.E.S.T and Vega models [51]. Similar to their work, we aim to build a large general applicability model, such that all durations, all experimental setups and all viable compounds can be predicted within one model. In addition to this, our work does not just include fish but also includes aquatic species from other phyla. Inspired by their state-of-the-art performance, our work implements their method and will discuss its exact approach in the methodology section.

Proposing the first multi-species model, Singh et al. [2014] use a model that is trained on a given species, but can extrapolate to different species in different classes. As such, the training set used consists of an algae species, whereas the model is tested on an external set of the algae species, in addition to a new algae, daphnid and fish species. Utilising boosted random forests, they find that predictions are worse for all species not in the training set, but that the performance is still comparable to literature. The finding that extrapolating to a new species from a different phylum with comparable performances [53] could indicate that a multi-species model across phyla could be beneficial for sharing and utilising common knowledge.

Our work aims to build a general applicability species-level model to predict the toxicity of all aquatic species. Although Gajewicz-Skretna et al. [2021] find in a study for classifying aquatic toxicity that models built on a local chemical compound space perform better than

using large chemical spaces, they agree with the added value of large models. Hence, we make little restrictions to the chemical space predicted by the models in our work. Recent research has also evaluated the use of graphical features for compounds. Jiang et al. [2021], however, find that 'descriptor-based models outperform the graph-based models in the predictions of a variety of molecular properties in terms of predictive accuracy and computational efficiency'. Hence, molecular fingerprints in combination with physical-chemical attributes are used in our work.

To the best of our knowledge, ours is the first work to build a general applicability multi-species aquatic toxicity model across phyla for felxible exposure duration and on a large chemical applicability domain. Considering the recent research done in meta-learning in QSAR modelling, 10 state-of-the-art models representing recent developments are adapted and applied for aquatic toxicity prediction.

The single-task random forest model is taken as a representative of a successful single-task QSAR model [40], whereas a multitask random forest model is inspired by Sadawi et al. [2019]'s work. Moreover, the state-of-the-art transformational machine learning method [41] and the ensemble stacked model from Sheffield & Judson [2019] are implemented. Using deepchem [46], a multitask neural network with multiple output nodes is employed. A neural network predicting all species with one output node is also implemented. Further, fine-tuning methods that finetune all or just the head of the trained neural network on a specific task. Finally, MAML is also applied.

We believe that with the array of methods and different solutions employed, a meaningful comparison of performances can be made. Our contribution involves applying many advanced methods to the aquatic toxicity domain for the first time. Further, since the main motivation behind sharing knowledge between tasks is the many low-resourced problems in aquatic toxicity prediction, we emphasise the analysis of how different methods are affected by low-resource situations. By doing so, we aim to advise the development of regulatory tools in the aquatic toxicity domain and ultimately aim to take a step towards mitigating the need for in vivo experiments on aquatic species.

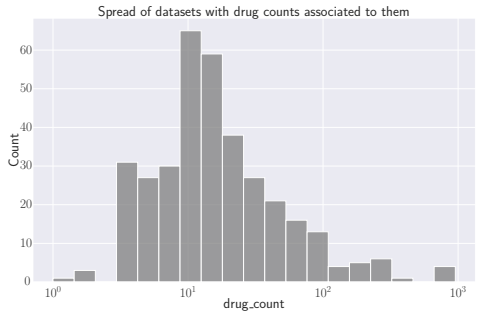
4 Data

This section presents the ECOTOX dataset collection, which was used to make the QSAR aquatic toxicity models explained in the following section. Assembled by the United States Environmental Protection Agency, the ECOTOX database is a large database of toxic effects automatically gathered from publications since 1970. Via the QSAR toolbox[1], the ECOTOX dataset collection was searched for aquatic toxicity data, retrieved and manually pre-processed. From this, we retrieved 24 816 assays altogether, which included 351 aquatic species and 2674 chemicals.

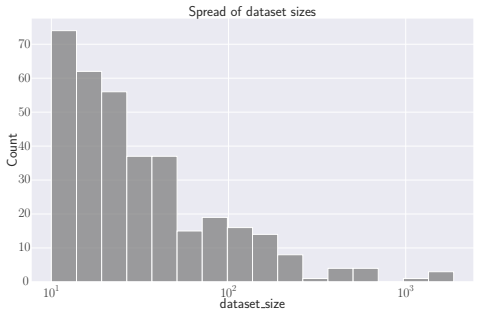
4.1 Endpoint

The toxicity endpoint is given by the concentration of a chemical needed to trigger a certain toxic effect on a species. In our work, the toxic effect observed is mortality, in the endpoint *LC50*. The toxicity target LC50 gives the concentration at which 50% of the species experiences a lethal effect after having been subjected to the chemical for a certain duration. Other toxic effects can include the concentration at which no effect is observed for 50% of

[1] https://www.oecd.org/chemicalsafety/risk-assessment/oecd-qsar-toolbox.htm



(a) The number of datasets with a certain drug count in their assays. Note that the x-axis is on a log-scale.



(b) The number of datasets with a certain number of assays. Note that the x-axis is on a log-scale.

Figure 2: Dataset sizes and drug counts in the retrieved ECOTOX dataset.

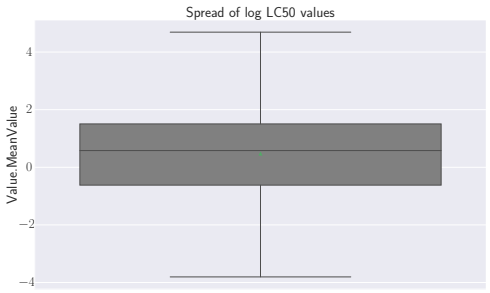


Figure 3: After applying the log transform, the spread of LC50 values of the retrieved ECOTOX dataset can be seen in this Figure.

the species, at which reproduction is harmed for 50% of the species, or at which 50% of the species becomes immobile. The latter toxic effect can be seen as a lethal effect for some species, due to the fact that the survival of some organisms is presumed impossible when they become immobile. Although it is possible to broaden the endpoint by including these cases, this thesis solely investigated LC50 values.

The LC50 values are standardised into the $\frac{mg}{L}$ unit where possible and dropped if not. Due to the spread of the LC50 target, the $\log_{10}(LC50)$ is predicted as a real number. The spread of the log values can be seen in Figure 3.

Endpoints that involve a quantifier or ranges are disregarded. These bounds occur when, for instance, more of the substance cannot be dissolved into the water. This can cause detection limits of the analytical techniques used to verify the actual exposure concentration during toxicity testing.

4.2 Preprocessing

An assay describes an experiment on a species that tests the toxicity of a chemical for a certain duration. Each assay in the dataset corresponds to a unique combination of species, chemicals and duration resulting in a toxicity target. Experiments, which were performed under the same experimental conditions multiple times, are collapsed into one unique assay

via the experimental mean, as suggested by the REACH guidance document [12]. Note that by combining multiple toxicity targets for the same experiment, the variance of the experiments is no longer captured and noise can be added to the data.

Species are identified via their Latin name and are grouped into taxonomies. Each of the 351 species belongs to a phylum and a class. There are 9 phyla present in the data, with the large majority of species belonging to either the Chordata (fish) or Arthropoda (daphnia) phylum. The classes divide the species more finely into 20 subsets. Here, the majority of species belong to the Actinopteri class, which holds ray-finned fishes. Since the Chordata and Arthropoda phylum mainly holds fish and daphnia, respectively, this dataset is well suited for predicting the endpoints needed for chemical regulation, specifically the 'acute fish toxicity', and 'acute daphnid toxicity' [39]. The QSAR model predicts the toxicity of chemicals on a species level. Thus, any subspecies of a species were combined into one species via their experimental mean.

The chemicals are identified via their CAS (Chemical Abstracts Service) registration number and their SMILES (Simplified Molecular-Input Entry-System) representation. As a result of format error, the CAS registration numbers were derived faulty in part. For this, it was ensured that the SMILES representations uniquely identified a compound. The SMILES representation gives a 2-dimensional string of a compound, from which the chemical descriptors (described in the following section) are found and derived.

Further, the SMILES were examined to ensure that chemicals not suited for modelling were removed. In this process, SMILES referring to inorganic chemicals or a combination of inorganic and organic chemicals are disregarded. This is due to the presence of metals or metal salts, which directly account (at least partly) for toxicity. Other chemicals that were not able to be represented by a single SMILES were also omitted. To ensure that the SMILES representation is consistent for all chemicals, Kekulé SMILES are used. The consistent SMILES representation can then be used to derive the chemical descriptors.

Although it is common to specify one test type and duration to use for modelling, our work aims to build a large applicability domain model. Thus, similar to Sheffield & Judson [2019]'s work, all experimental setups are included in the dataset and are only defined by their duration. With this, short-term (acute) and long-term (chronic) toxicity can be modelled together. As acute and chronic periods vary for each species, it was decided to not further classify a duration as acute or chronic. Duration values are converted into days wherever possible and disregarded wherever a duration is not specified.

In the light of building a generally applicable model with few restrictions, no outlier removal was performed.

4.3 Chemical Descriptors

Although recent research has also evaluated the use of graphical features for compounds, our work uses chemical embeddings based on Jiang et al. [2021]'s finding that descriptor-based models outperform graph-based models in the predictions of a variety of molecular properties in terms of predictive accuracy and computational efficiency' [26]. Hence, the compounds in the ECOTOX QSAR models are described by chemical fingerprints and relevant physical-chemical properties.

Embeddings that aim to capture two-dimensional chemical structures are called 2D fingerprints. Three main types of fingerprints exist: dictionary-, path- and circular-based embed-

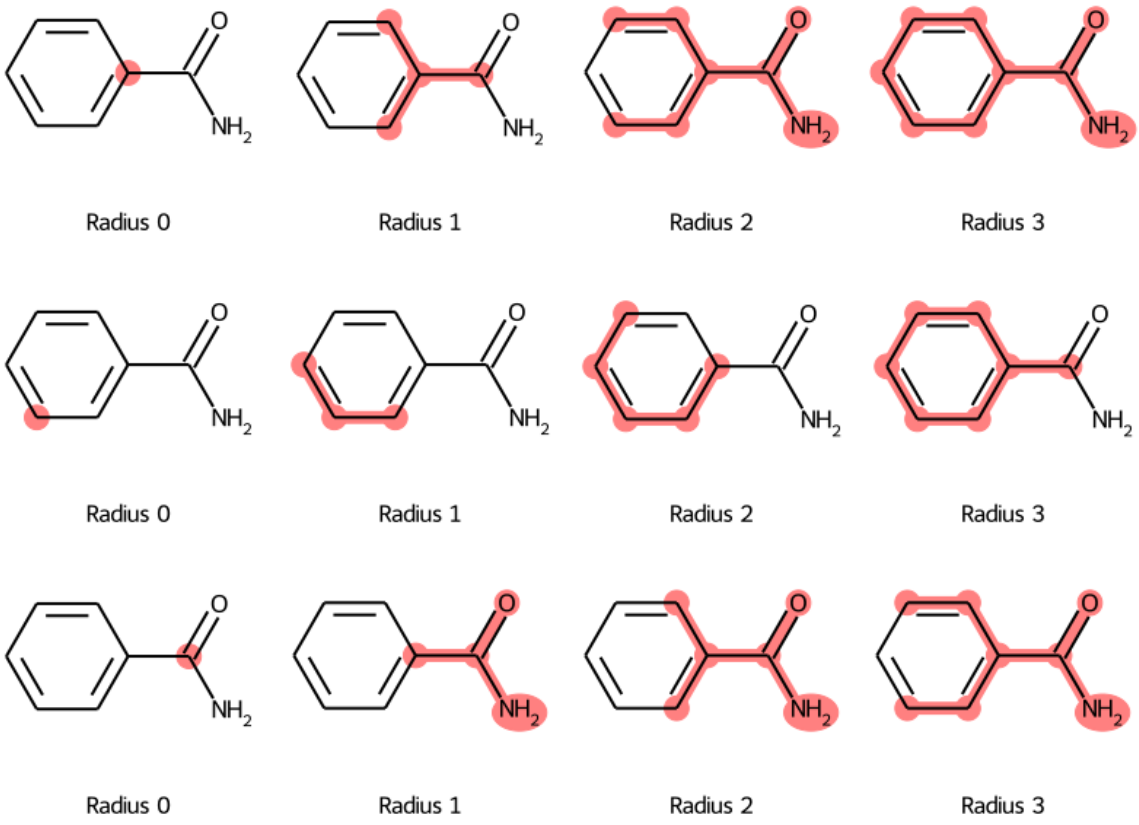


Figure 4: ECFP Generation for Benzoic acid amide: The figure shows how the EFCP algorithm expands the atom identifier of one atom (shown at radius 0) with each iteration/radius through the molecule. For instance, at radius 2 the atom identifiers reach up to 2 bonds away. Please note that these are three out of nine atom starting points. When calculating the fingerprints, all nine atom starting points are considered. This image is inspired by [49].

dings. Briefly, the first acts as a look-up table for certain chemical features of substructures in a key set. A typical example is a 166-bit fingerprint from MACCS (Molecular ACCess System) [11]. The latter two refer to traversing the chemical structure in a certain way. A path-based method will note atomical features in a path through the chemical, whereas circular fingerprints typically traverse the structure recursively on many paths.

Our work uses extended connectivity fingerprints (ECFP), which belong to the class of circular fingerprints. In contrast to other chemical representations, the ECFP embeddings were specifically designed for QSAR modelling [49]. The fingerprints are based on the Morgan algorithm [36], an algorithm that was developed to recognise duplicate graphical chemical structures in a database.

To generate the ECFP, each atom of a molecule is given an identifier, which holds some information on that specific atom. Figure 4 shows along which paths the atom identifiers are combined during an iterative process. In this iterative process, the atom identifiers are replaced by a combination of an atom's own identifier and the identifier of its direct neighbours. This set of identifiers is hashed into a new identifier for that atom. Hence, the atom identifier does not only incorporate information on the starting atom (radius 0 in Figure 4), but also incorporates its direct neighbours (radius 1 in Figure 4). This concludes an iteration, after which the algorithm begins to combine neighbouring identifiers anew. Once a preset number of iterations has been completed, the set of identifiers, that now describe various substructures of varying lengths, make up the chemical fingerprint [49].

A fingerprint may capture the presence of each final atom identifiers in binary vectors or may give a count of each final atom identifier in real numbered vectors. For the standard binary ECFP fingerprint used in our work, any duplicates in final atom identifiers are omitted. The atom identifiers left are collapsed into a vector by dividing the identifiers by the wanted length of the fingerprint. Then, the position given by the remainder is increased by one or the count, depending on which format is desired. The length of a fingerprint, thus, also acts as a trade-off between collisions of identifiers and too sparse vectors.

By choosing the information an atom identifier holds, different variants of the ECFP can capture different features. Our work uses the original 1024-bit binary ECFP4 fingerprints, which aims to capture precise atom environment sub-structural features' [49]. The 4 in the name specifies the so-called diameter of the fingerprint. The radius - half of the fingerprint's diameter - is equal to the number of iterations the algorithm runs for. As such, the length of the largest substructures captured is given by the radius [49].

Using the open-source chem-informatics package RdKit[2], the fingerprints are calculated via their SMILES representation. Note that through standardisation of the SMILES all compounds are treated equally in RDKit, and fingerprints will not differ based on the method with which the SMILES were constructed.

Certain physical-chemical attributes may also yield important information on a molecule. To add to the chemical fingerprints, relevant physical-chemical attributes were gathered from PaDEL [62], open-source software that allows the calculation of molecular descriptors from the SMILES representation. The attributes gathered were suggested by a domain expert, and include constitutional and hydrophobic attributes. A selection of these attributes was made by ensuring that at least 90% of the assays had information on a given feature and that the feature had sufficient variance throughout the dataset. Via the uni-variate feature selection

[2]RdKit Info.

measures by Scikit-learn [43], it was found that a global binary missing value indicator was the most beneficial indicator of missing values, which were imputed by the mean. Features that had a notably low contribution score based on the F-statistic, mutual information and random forest importance were dropped.

Finally, the structural properties included in the QSAR modelling are counts of carbon, oxygen, nitrogen, and aromatic atoms (in a molecule), as well as counts of (hetero) rings, rotatable bonds, hydrogen bond acceptors and hydrogen bond donors. Further, the molar refractivity, polarizability, ionization and topological polar surface area of the molecule were added.

Attributes that are specifically beneficial for aquatic toxicity are the octanol-water partition coefficient, *log P*, the octanol/air partition coefficient *KOA*, the octanol-water distribution coefficient, *logD55* and *logD74*, in addition to the vapour pressure and the water solubility of a molecule. These attributes traditionally aid in ecotoxicological QSAR models. The ECOSAR model, for instance, predicts the toxicity of a chemical via a linear regression dependent on the octanol-water partition coefficient.

5 Methodology

In this section, the employed QSAR solutions are elaborated further. The section is split as follows: first we describe the single-task set-up and the single-task models used, before elaborating the multitask methods employed. Finally, the transfer learning and optimisation-based meta-learning methods are given. Additionally, a special focus is placed on the hyperparameter optimisation of each method.

5.1 Single-task Models

The single-task models approach each dataset individually without using any knowledge of other datasets. As such, they cannot make use of any additional information on the species and their taxonomies. The single-task models used in this thesis are the mean baseline and random forest models.

5.1.1 Single-task Mean

The simplest baseline used is predicting the mean of the seen train set for any unseen data. Any model that claims to be a good predictive model should be able to outperform this prediction. This is due to the fact that the mean baseline does not utilise any additional information, and naively estimates the mean for all further data points. The single-task mean model predicts the mean of all toxicity concentrations seen for a given single species in training.

5.1.2 Single-task Random Forest

Originally proposed by Ho in 1995 and expanded by Breiman in 2001, random forest models have been applied in many domains successfully [21, 5]. Random forest models are ensemble models, which means that they consist of multiple models that predict a consensus value. Containing multiple decision trees, random forest regressors predict the average of all values predicted by the individual decision trees. This aids the model to avoid overfitting.

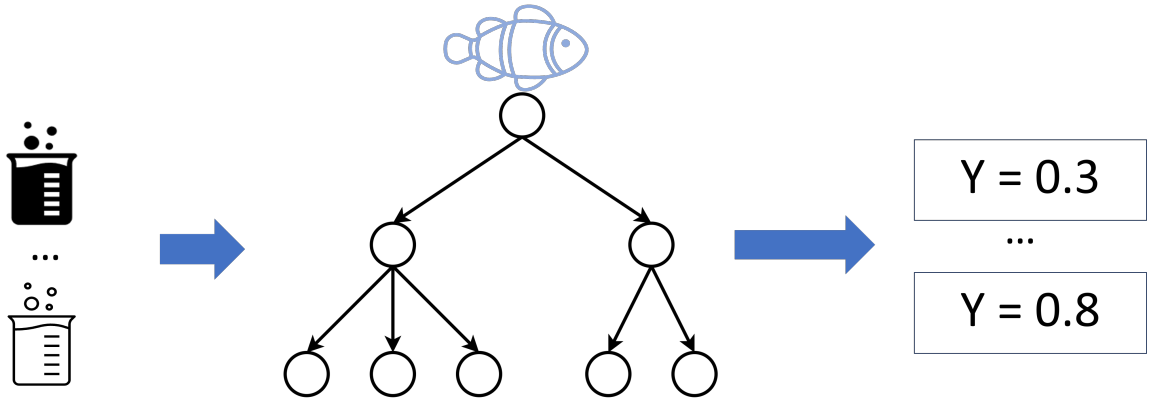


Figure 5: Single-task Model: A random forest model is fitted for all target species respectively. The features used are the molecular descriptors.

To build a random forest model, each decision tree is built on a randomly selected subset of the data available, where the number of decision trees in the random forest model is a hyperparameter. The subset of data a decision tree is trained on is selected via a bagging schema, i.e. sampling training points with replacement [4]. A decision tree, which can model non-linearity, then splits the data at each node of the tree, such that a loss function is minimised. At each node in a decision tree, only a subset of features are selected as candidates to be split on. Note that this forces trees to be diverse, which leads to more diverse predictions, and a better consensus value. At test time, each tree predicts a new sample, and the average of all predictions is the final output.

Inspired by Olier et al. [2018]'s work on algorithm selection for QSAR modelling, a single-task random forest regressor is implemented, as it was found to be the best performing single-task model. As seen in Figure 5, the random forest model uses chemical descriptors and their exposure duration to predict the toxicity of a given species. As the random forest model is task-specific, it cannot make use of the target species' class and phylum information.

We perform hyperparameter optimisation for the single-task random forest models. The additional costs that occur when hyperparameter optimising all single-task models individually, however, is infeasible in our work. Instead, we find data-driven hyperparameter defaults. For this, we find the hyperparameter configuration that leads to the best average performance across all single-task random forest configurations. The hyperparameter optimisation set up can be seen in Table 1. The cross-validation folds are subsets of chemicals, such that the hyperparameter optimisation is performed on the same use case as the model will be employed in. Using 3 folds here is a trade off between the variance caused by the specific training data and computational cost.

5.2 Multitask Learning Models

The multitask learning models learn the separate tasks jointly as to share knowledge between them during training. As such, these models can utilise the different species and their taxonomies, which are categorical variables.

Single-task random forest

Search Algorithm	Random Search
Search Space	Adapted from Autosklearn [14]
Iterations	50
Cross Validation Folds	3
Performance criteria	RMSE across all single task models
Performed	At every model build

Table 1: Hyperparameter Optimisation for the single-task random forest models. Note that the cross validation folds are always defined on subsets of chemicals.

Multitask random forest

Search Algorithm	Random Search
Search Space	Adapted from Autosklearn [14]
Iterations	50
Cross Validation Folds	3
Performance criteria	RMSE
Performed	At every model build

Table 2: Hyperparameter Optimisation for the multitask random forest models.

5.2.1 Multitask Mean

The multitask mean naively predicts the mean of all toxicity concentrations (including all species) seen in training. This method does not make use of any input features.

5.2.2 Multitask Random Forest

Sadawi et al. [2019] successfully use random forest models for multitask learning in QSAR modelling. Although they apply a separate model for groups of targets, the second random forest model in our work is implemented for all aquatic species jointly. Sadawi et al. [2019] find that within this multitask approach, the addition of a measure of target distance was beneficial. Although there is no distance measure for the aquatic species available, the target species are categorised into two taxonomy levels (phyla and classes), which may help the model's performance. This multitask random forest model uses these additional features to predict all aquatic species simultaneously.

The hyperparameter optimisation for the multitask random forest model can be seen in Table 2. The procedure is nearly identical to the single-task random forest's hyperparameter optimisation, however, instead of searching for data-driven defaults of the hyperparameters for the many single-task models, we are searching for an optimal hyperparameter configuration for the multitask random forest model here.

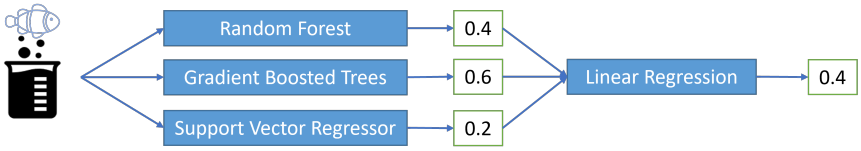


Figure 6: Stacked Ensemble Learning method: Given a compound, its exposure duration and a species, the three baselearners predict the aquatic toxicity. Then, a linear regressions tacks these predictions and combines them into one consensus value.

5.2.3 Multitask Stacked Ensemble Learner

A further multitask approach used is the stacked ensemble learner from Sheffield & Judson [2019]. Their method, shown in Figure 6, was originally proposed to model the aquatic toxicity in fish species, whereas here the model is applied to all aquatic species. All base learners use the information of the target species. A stacked ensemble learner improves on traditional ensemble learning, which predicts the mean of its base models predictions, by learning how to best combine the predictions of the models. Sheffield & Judson [2019] achieve good performance by using a linear regression model to combine the three base models: support vector regression, gradient boosted trees and a random forest.

The first base learner, the random forest model, is equivalent to the multitask model mentioned earlier. Where the random forest model trains its decision trees in parallel, gradient boosted trees, the second base learner, builds each decision tree sequentially. It does this with the aim of correcting the mistakes previous decision trees have made. The last base learner is the support vector regression model, which draws a hyperplane in multi-dimensional space via a kernel that maps lower-dimensional to high-dimensional data [54]. This hyperplane then predicts the lethal concentration values.

To train the stacking linear regression, the training data is split into 5 folds. Using 4 out of 5 folds iteratively, a partial model is built for each of the base learners to make predictions for the held-out fold. Subsequently, each point in the training data has been predicted by a partial model of all three base learners, such that each training point has 3 predictions - one from each base learner. A linear regression model can then learn how to weigh the predictions of each base learner to combine or stack the predictions of the base learners into the best consensus value. For this, the linear regression is fitted on the full training data, using the three predictions of each base learner as features. To use the model at test time, all three base learners are trained on the full training data. At test time, each of the base learners, which were trained on the full data set, predict the unseen compound for a given species. The linear regression model then stacks these predictions into one final consensus value.

The hyperparameter optimisation set up can be seen in Table 3. Note that the hyperparameter optimisation is done for all three base learners individually.

5.2.4 Multitask Neural Networks

Neural Networks take inspiration from the human brain. As such, they consist of a network of nodes, imitating neurons. Each of these nodes can forward an activation so that signals can travel through the network.

The activation of each of the nodes is determined by its input vector x weighted by the

Stacking ensemble

Search Algorithm	Random Search
Search Space	Adapted from Autosklearn [14]
Iterations	50
Cross Validation Folds	3
Performance criteria	RMSE
Performed	On all three base learners individually at every model build

Table 3: Hyperparameter Optimisation for the stacking ensemble model.

adaptable weight vector of the node w, in addition to an adaptable bias b. Using a non-linear activation function, here: ReLu, the activation of a node can be calculated with $y = max(0, (b + \sum \theta_{i,j} x_i))$. Nodes are organised in layers, such that a typical neural network has an input layer, one or multiple hidden layers and an output layer. The combination of activations of nodes of different layers leads to the network being able to make predictions.

Two types of multitask neural networks were implemented and are elaborated on in this section.

One Output Node The neural network presented here is trained on all of the tasks, but only uses one output node in the output layer, see Figure 7a. In addition to this, we also employ a neural network with an output node for each task it is predicting, see Figure 7b. To distinguish between the two models, the simple neural network is referred to as a neural network, whereas the multiple output node neural network is named multitask neural network.

Next to the architecture of the neural network, the weight matrix noted as θ is the essence of the neural networkf_θ. When training the neural network, the weights are optimised, such that a trained neural network solution is defined by its weights. To optimise these weights, gradient descent algorithms are used. In a gradient step, the weights are updated via $\theta' = \theta - \alpha \Delta_\theta L_T(f_\theta)$, where α denotes the learning rate, which controls how finely the weights should be updated, and the loss function L_T is given in Equation 1.

As the aquatic toxicity problem is cast as a regression task in our work, the MSE loss is employed to update the weights of the network. Equation 1 borrows notation from [15] and shows the loss function used. L_T is the loss for the collection of tasks modelled in this neural network calculated by sampling $x^{(j)}, y^{(j)}$ pairs. Equation 1 shows how the loss is calculated, where $||.||_2$ denotes the L^2 norm.

$$L_T(f_\phi) = \sum_{x^{(j)}, y^{(j)} \sim T} ||f_\phi(x^{(j)}) - y^{(j)}||_2^2 \tag{1}$$

To avoid overfitting on the training set, early stopping is implemented. Essentially, this technique splits the training set into a training and validation set. The neural network is trained on the test set as usual, but the validation set allows us to monitor when the

neural network seems to be overfitting. For this, the training error and validation error are observed. Once the validation error starts to rise and has not recovered for a certain number of iterations, training is halted. The number of iterations to wait is called *patience* and is a hyperparameter. The validation set is chosen to be 20% of the training set.

Since this technique uses parts of the training set, and the QSAR problem is naturally low-resourced, the training is repeated on the whole training set to ensure that all of the available data is used. In the second training phase, the neural network is trained for as many epochs as the early stopping phase ran for. Typically, it is also possible to run for the number of epochs minus the patience. The additional training data can, however, benefit from the extra iterations as well. We simply rerun the neural network training for the same amount of epochs.

To optimise a neural network, a good neural architecture needs to be found, before the hyperparameters of the learning algorithm are tuned.

First, a neural architecture needs to be found that works well with the problem to solve. The neural architecture search involves finding how deep and wide the network should be, such that performance can be optimised. For this, the number of layers (to find the network's depth) and the number of nodes per layer (to find the network's width) need to be found. Further, the use of regularisation techniques that are embedded into the network needs to be evaluated. Preventing the model to overfit, the regularisation techniques considered here are dropout [55] and the addition of batch normalisation [24]. Dropout is a technique that does not use a fraction of the nodes during training iterations to prevent overfitting. Normalising the output of a layer, batch normalisation scales the output such that the output has a mean of zero and a standard deviation of one.

To perform a neural architecture search, the neural network intelligence (NNI) from Microsoft is used [33], as it is compatible with Pytorch [42], which is used to implement the neural networks themselves. Using the Tree-structured Parzen estimator (TPE) [3], the use of 3 or 4 layers is evaluated, in addition to the number of nodes in a layer, the scale of dropout, and the addition of batch normalisation. TPE belongs to the class of sequential-based model optimisation methods, which train independent models. We use 50 iterations of TPE on 5-fold cross-validation, in which no two folds had common molecular compounds. The use of a more sophisticated search method for the neural architecture may provide an advantage over other models. The neural architecture search, however, is run only once and this optimised neural architecture is used in all further neural networks that train on multiple tasks, e.g. fine-tuning.

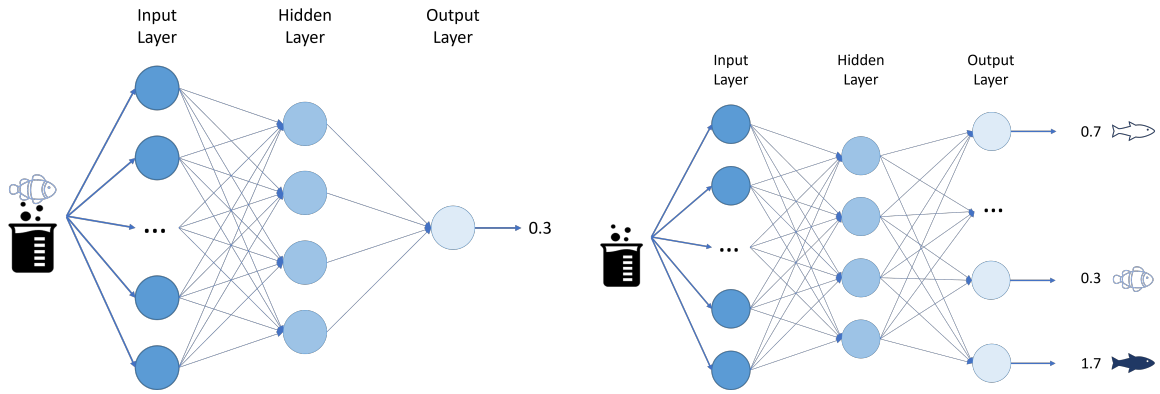
The second area that needs to be optimised are hyperparameters of the optimiser, such that the weights can be learned effectively. For optimising the weights of the network, Adam [27], an alternative to the classic stochastic gradient descent is used. Adam has been shown to handle sparse gradients well, a problem which may arise with the sparse binary fingerprint inputs. The hyperparameter optimisation set-up is given in 4. Again, the use of the Tree-structured Parzen estimator (TPE) [3] for the hyperparameter optimisation may give this method an advantage. The hyperparameter configuration, which is found after performing hyperparameter optimisation once, is used in the fine-tuning methods.

Multiple Output Nodes The neural network presented in the previous section (and in Figure 7a) used one model and one output node to predict the toxicities in all n tasks, whereas the multitask neural network in this section, see Figure 7b, predicts the toxicities

Multitask neural network one output node

Search Algorithm	Tree-structured Parzen estimator (TPE) [3]
Hyperparameters optimised	Learning rate, weight decay, batch size and patience
Iterations	50
Cross Validation Folds	5
Performance criteria	RMSE
Performed	Once

Table 4: Hyperparameter Optimisation for the multitask neural network with one output node.



(a) Classic neural network with one hidden layer and one output node. All aquatic species are predicted via the one output node.

(b) This Multitask neural network predicts the aquatic toxicity for reach task with a separate output node.

Figure 7: Multitask Neural Networks

Multitask neural network with multiple output nodes

Search Algorithm	grid search
Hyperparameters optimised	Learning rate, weight decay
Iterations	16
Cross Validation Folds	5
Performance criteria	RMSE
Performed	Once

Table 5: Hyperparameter Optimisation for the multitask neural network with multiple output nodes.

of all n tasks using n output nodes. Essentially, this allows the neural network to share the internal feature extraction and representation part embedded in the hidden layers of the neural network, whereas the task-specific dependencies can be captured in the weights toward the task-specific output nodes.

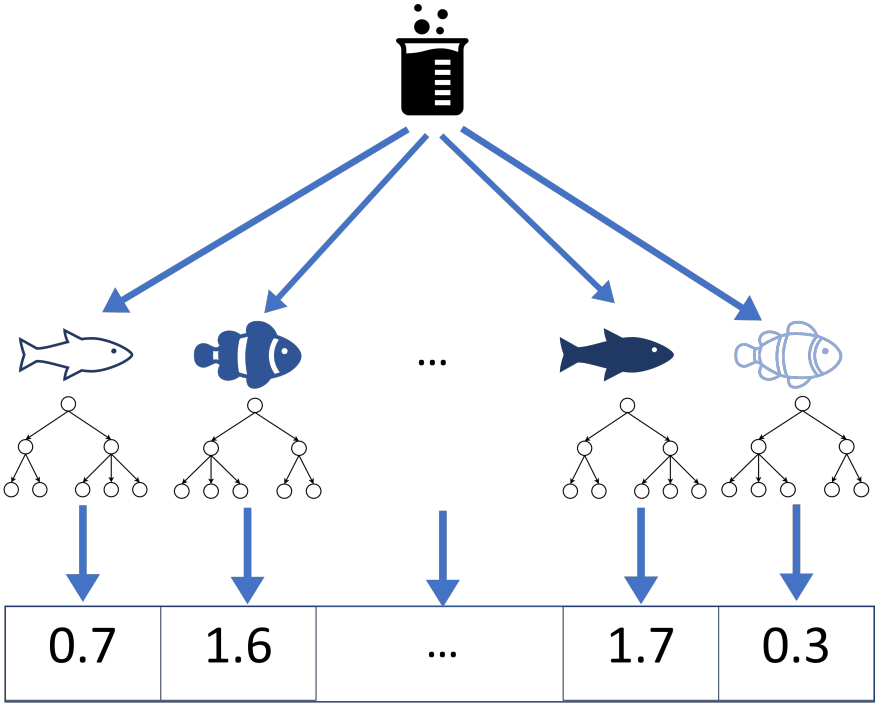
The training of a neural network with multiple outputs differs in that weights need to be provided to select from which node(s) the output is known and the error can be backpropagated. That is that the weights passed need to indicate missing values in chemical, duration and species pairings.

When implementing the multitask neural network with multiple output nodes, two main designs can be considered: the neural network can be trained with taxonomical information of the species, as well as the exposure duration and compound descriptors, or the neural network can be trained on the exposure duration and compound descriptors only. To train the first option, each prediction the network makes only passes through one output node, whereas when using the second approach multiple output nodes are used when passing a certain chemical and duration. In both cases, whenever a chemical and exposure duration pair has not been tested for a species, the output node for that species on the unseen chemical and exposure duration pair is weighted with zero. We try both options and notice little difference in performance, thus, we concentrate solely on the network including taxonomical information.

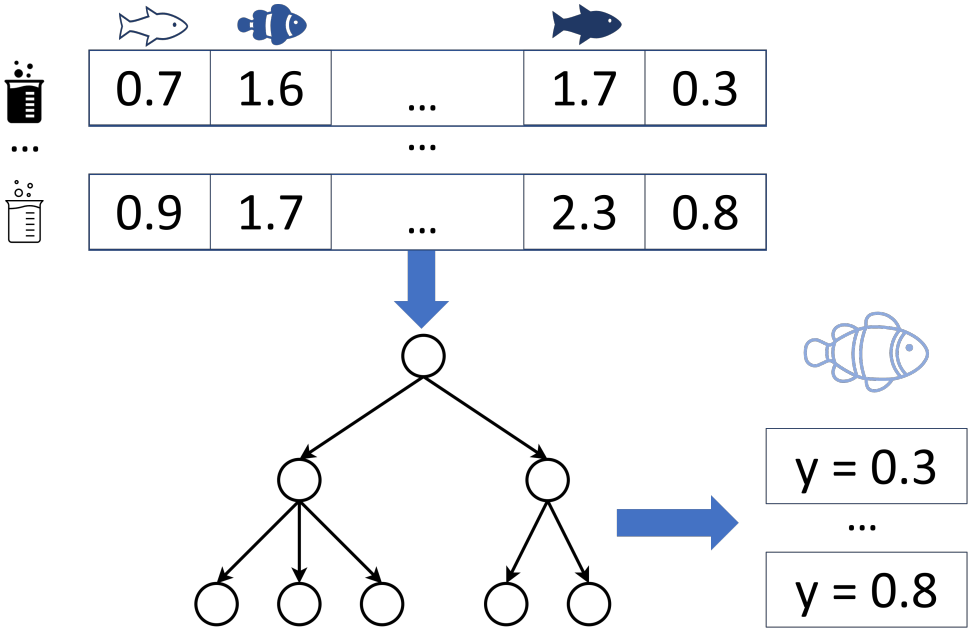
As Ramsundar et al. [2019] criticise the variance in multitask QSAR neural network models, they propose a standard adaptable model implemented in DeepChem. The neural architecture found for the neural network with one output node is used, although the output nodes are updated. The hyperparameter optimisation is presented in Table 5.

5.3 Transformational Machine Learning

Combining aspects of ensemble-, multitask-, and transfer learning, Olier et al. [2021] proposed a new approach: Transformational Machine Learning (TML). This approach aims to use the many related tasks to build a shared representation of the input, which can then be used to train further single-task models with. The shared embedding should represent a general consensus on the toxicity of a compound. The shared embedding can also enable predictions by identifying similarities between the sensitivities of the aquatic species.



(a) Transformational Embedding: To create the embedding for a chemical, the chemicals toxicity is predicted by all of the single-task models. The vector containing all of the predicted toxicity values is then the embedding used.



(b) Final model: A random forest model is fitted for all target species, respectively. The features used are the transformational embeddings of a chemical and its exposure duration found in Figure8a.

Figure 8: Transformational Machine Learning [41].

TML's methodology combines elements from ensemble -, multitask -, and transfer learning. Its resemblance to multitask learning comes from the fact that multiple tasks are used to share knowledge and to learn jointly. Although both multitask learning and TML use no priors, the multitask learning approach typically shares the knowledge among tasks in the model itself, whereas TML creates a shared representation to utilise knowledge across tasks. The use of TML has similarities with stacking multiple base learners using a meta-model, similar to the approach in section 5.2.3. The difference is that TML builds many base learners that are single-task learners built on different subsets of data [41].

TML can be split into two parts: the first part fits a transformational embedding, whereas the second part uses that embedding to build a single-task machine learning model.

To create the transformational embedding, a single-task model (here: random forest model) is fitted for the training instances of each target species. As seen in Figure 5, for each aquatic species, a separate random forest model is built that takes the chemical descriptors and the exposure duration as input and predicts how toxic this is for the given species. Once all single-task models have been built, an embedding for each chemical and exposure duration can be created.

Figure 8a shows how an embedding is made for a specific compound with given exposure duration. All single-task models predict the toxicity of the same compound with the given exposure duration. These predictions are then placed in a vector, which will be our transformational embedding. This embedding now holds information on the general consensus of how toxic it is predicted to be for all species. A toxic compound will therefore have an embedding showing lower concentrations predicted by most species.

Once the embeddings for each compound and exposure duration have been created, the final model can be built. Note that this need not be the same type of model used to create the embeddings. In this case, a single-task random forest is used again. The choice of the random forest method is again due to its good performance found by Olier et al. [2018], and it allows a direct comparison to be made with the single-task random forests employed. In contrast to the first single-task random forest built for the species, the input features are now the embeddings created, see Figure 8b. By training the random forest model with the training set for that species, the model can learn to use the general consensus of the embedding. Further, with sufficient examples, the model can learn which aquatic species sensitivities it resembles.

This thesis uses two types of end models. The first trains a single-task random forest on the transformational embeddings, whereas the second combines the prediction with the single-task random forest model trained on the chemicals in the first step. We refer to the two models as *TML*, and *TML Stacked*, respectively.

The transformational machine learning approaches uses the data-driven hyperparameter defaults found for the single-task random forest models.

5.4 Fine-tuning

Fine-tuning techniques are a simple way to perform transfer learning. The idea behind fine-tuning is to use all related tasks to train a neural network to extract knowledge from the input features well and build an internal representation. Using all tasks, this can be done with more training data. To then use the model for a specific task, the model is adapted further to the specific task solely. By doing so, the neural network can leverage much more

Fine-tuning

Search Algorithm	grid search
Hyperparameters optimised	fine-tuning iterations, learning rate, weight decay
Iterations	48
Cross Validation Folds	2
Performance criteria	RMSE
Performed	Once for both finetuning approaches, respectively

Table 6: Hyperparameter Optimisation for the fine-tuning approaches.

data and knowledge from other related tasks, compared to training a neural network on a single-task from scratch.

Fine-tuning is done in two stages. In the first stage, a neural network is trained on multiple tasks as described in section 5.2.4. Once training is completed, the network's weights are fixed. Then, during the latter stage, (a selection of) the weights are adapted to the end task specifically. That is, for a given aquatic species, a neural network is trained on all species, before (a part of) the network is trained on the given aquatic species solely.

Our work uses two types of fine-tuning. The first common type of finetuning is *fine-tuning top*, in which the network head's (the final layer's) weights are adapted to the specific species. Next to this, we also employ *fine-tuning all*, in which all weights are unfrozen and finetuned. Although early stopping is employed, the latter approach may suffer from the loss of weight information in all layers as they are retrained. The hyperparameters used in the finetuning phase are optimised via the set-up in Table 6, whereas the hyperparameters for the general training are taken from the multitask neural network with one output node.

5.5 Model Agnostic Meta-Learning

The optimisation-based meta-learning technique Model Agnostic Meta-Learning (MAML) was proposed by Finn et al. [2017]. As it is model agnostic, it can be used with many models and merely requires that a model uses gradient descent in its training. In our work, MAML is used in combination with a simple neural network.

Typically, when a neural network is initialised, the weights are assigned random values that require a substantial amount of resources in terms of training time and training data, to adapt into weights that work well for the given task. MAML aims to encapsulate knowledge of related tasks it has trained a neural network on into good initialisation parameters. That is, to observe which weights worked well for the related tasks to suggest good initial weights to initialise a neural network for a new task with. In this context, *good* initialisation parameters for the neural network are initialisation parameters that can be adapted to work well for a new task quickly. In contrast to fine-tuning, which adapts weights found optimal for all tasks to a single-task, MAML aims to find initialisation weights that allow for quick adapting to all tasks. Figure 9 shows which initialisation parameters MAML aims to find in a simple example taken from [23]. MAML does this by not only optimising the weights on a given task

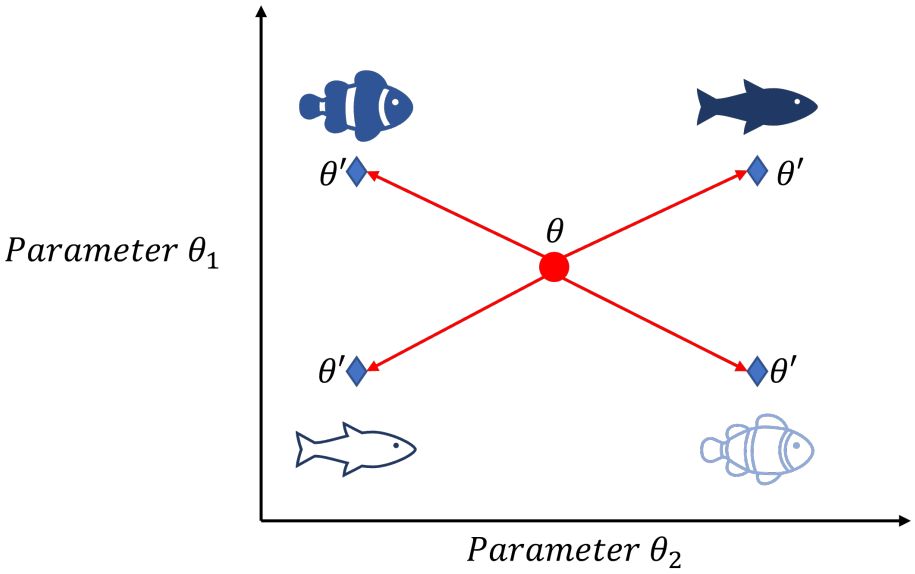


Figure 9: Intuition behind MAML [15]: Let the model used have two parameters θ_1 and θ_2, that are initialised to θ. The blue points in the plot show the optimal configuration of parameters for a specific aquatic species task, θ'. Then, MAML aims to find initialisation parameters θ, such that the optimal configuration for each task can be reached quickly. The optimal configuration, here, would be the point in red, from which all blue points can be reached equally fast. Example taken from [23].

but by simultaneously optimising the initial weights across tasks. Hence, MAML uses two optimisers: one task-specific, and one for the initialisation weights. The following explanation draws heavily from [15].

The initialisation weights found by MAML are initialised at small random values. Let a distribution of tasks $p(T)$, for instance, the distribution of aquatic species datasets, be given. From this distribution, a task T_i is sampled, on which a neural network f_θ is initialised with the current initialisation weights θ. This neural network is trained for this task for n training iterations, after which the weights have been adapted to θ_i'. For simplicity, one training iteration is chosen, such that the weights are updated via Equation 2, in which α is the learning rate for the task-specific optimiser.

$$\theta_i' = \theta - \alpha \cdot \Delta_\theta L_{T_i}(f_\theta) \tag{2}$$

Once training on that task is completed, the loss of the network on a new test sample of the task dataset is computed. This loss, shown in Equation 1, is calculated on the trained network, thus on the trained weights, θ'. Note that this loss is needed when updating the initialisation weights in the next step, but first, this procedure is repeated for m more tasks in the batch size.

With the losses of the m trained networks, the initialisation weights θ can now be updated. This is done with a step of the separate optimiser for the weights in Equation 3, in which β denotes the learning rate for the optimiser of the initialisation parameters.

$$\theta \leftarrow \theta - \beta \cdot \Delta_\theta \Sigma_{T_i \sim p(T)} L_{T_i}(f_{\theta_i'}) \tag{3}$$

MAML

Search Algorithm	grid search
Hyperparameters optimised	meta learning rate, learning rate, weight decay
Iterations	34
Cross Validation Folds	5
Performance criteria	RMSE
Performed	Once

Table 7: Hyperparameter Optimisation for MAML.

This update concludes an iteration of the MAML algorithm, and m tasks are sampled anew [15].

MAML has several hyper parameters: the learning rate for the task-specific optimiser, α, the learning rate for the initialisation parameter optimiser, β, the number of adaptation steps the task-specific optimiser trains on a task n and the batch size m of tasks trained before the initialisation weights are updated.

As MAML learns good initialisation weights for a neural network to adapt quickly to a new task, we search for a good default single-task neural architecture. For this, the same set-up for the multitask neural architecture search, see 5.2.4, is employed, which searches different neural architectures using 2 or 3 layers is evaluated, rather than 3-4 layers. The performance criteria used was the averaged performances of all single-task neural networks with the same architecture.

Once the default architecture is found, a brief hyperparameter optimisation is done for MAML in the form of a grid search shown in Table 7. The MAML training procedure uses the same early stopping approach as explained earlier. It uses a patience of 25 iterations, as MAML's training is typically more unstable than directly training simple neural networks. We chose to use one training step for all new tasks to adapt from the initialisation weights from MAML.

6 Experiments

Having outlined the data and the models that were used in our work, the next section elaborates on the experiments that were performed. First, the metrics used for performance assessment are given, after which the experiments are presented.

6.1 Performance Metrics

In this section, the performance metrics used to evaluate the models are presented: the within factors of a prediction metric and the root mean squared error. These metrics can be calculated by considering different groups of predictions. Specifically, we can aggregate performance across species, such that the performance per species is considered, across chemicals, or across experimental folds.

6.1.1 Within Factors of a Prediction

This metric was designed to be interpretable for domain experts by evaluating the regression task as a classification task. That is that we count how many predictions are 'correct' given a lenience factor. Ranging from 0-100%, the metric describes what percentage of all predictions fall within a factor of x of the actual toxicity value. Two factors are measured here: a factor of two and a factor of ten. A factor of two from the actual value is said to be similar to the expected experimental variance in toxicity tests, whereas a factor of ten from the actual value is considered an acceptable prediction of toxicity. The exact calculation can be seen in Equation 4, where x denotes the factor in which the predictions should fall (hence $x \in 2, 10$). Note that these metrics are calculated on the original LC50 values (and not on the log scale).

$$Within\_Factor_x = \sum_{i=0}^{n} \frac{((\hat{y}_i - y_i) \leq x)}{n} \tag{4}$$

6.1.2 Root Mean Squared Error

The second performance metric used to evaluate the predictions made by the model is the root mean squared error (RMSE), which is calculated on the log transformed LC50 values. The RMSE, calculated by Equation 5, describes the standard deviation of the residuals, where residuals describe the distance of actual toxicity values to their predicted points. Typically, the RMSE error has been used to evaluate QSAR regression performances(in aquatic toxicity prediction [61, 51, 30, 53]), and, hence, is employed in our work as well. As the RMSE is more nuanced than the previous metric, we use the RMSE to optimise the neural architecture structures and hyperparameters on.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{5}$$

6.1.3 Statistical Test

For further statistical analysis, the rank of each method on each task is computed. That is, the performances (measured by the RMSE metric) on a given task are ranked across models. The model with the lowest RMSE for the given task receives a 1, the second-lowest RMSE receives a 2, whereas the highest RMSE receives the highest number. Ties are resolved by using assigning methods the mean rank between them, such that if two models are tied for the first rank, both receive 1.5.

To compare the performance of multiple algorithms on multiple datasets to each other statistically, Demšar [2006] suggests using the Friedman test [17]. The Friedman test assesses the hypothesis that the mean of each population is equal. For comparing multiple algorithms on different tasks, the null hypothesis is that the mean of the ranks of a model (over all tasks) is equal to all other models' rank means. If we can reject this null hypothesis, we can investigate which models' means are significantly different to each other.

Once it has been rejected that the mean of all method's ranks is equal, the Nemenyi test [37] can be performed to determine which ranks are significantly different. The test results can be visualised in a critical distance plot. The critical distance describes the distance of mean ranks that two methods must have such that they are significantly different. Once it has been rejected that the mean of all method's ranks is equal, the Nemenyi Test [37] can be performed

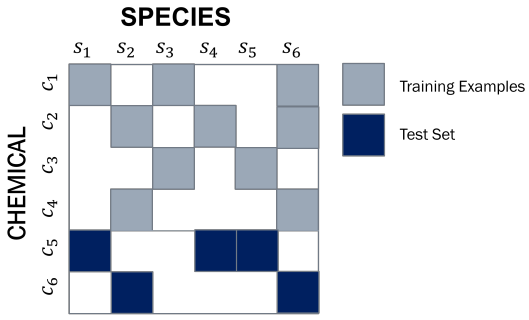


Figure 10: Test scenario: The rows represent chemicals, whereas the columns represent the test species. When splitting the datasets to train and test on respectively, the data is split into two mutually exclusive subsets of chemicals.

to determine which ranks are significantly different. The test results can be portrayed in a critical distance plot. The critical distance describes the distance of mean ranks that two methods must have such that they are significantly different.

6.2 Test Sets

In the typical use case, chemicals to be registered under the REACH ruling will not have been tested in surplus before, as otherwise the modelling of the chemical is irrelevant, since its experimental values suffice. To imitate this situation, all of the test splits in this body of work are done on subsets of chemicals. It is vital to consider the real-world regulatory use since the standard cross-validation split is overly optimistic here.

To ensure that all test scenarios, including test sets and cross-validation folds, have mutually exclusive sets of chemicals, the test sets are formed by splitting on the chemicals. Consider the simplified data set shown in Figure 10. In the matrix representation, each column represents an aquatic species (a separate task), whereas each row refers to a chemical. The shaded squares represent when an aquatic toxicity value of a chemical for an aquatic species is present in our dataset. Note that in our case, the matrix would extend into a third dimension, which would give the exposure duration of the assay, but for simplicity, this is omitted in the figure. When splitting the given experimental values (shaded squares), the train and test set are defined by two distinct subsets of all chemicals. When considering the third dimension of exposure duration, the train and test splits would cut the space into two cubes along the chemical axis. In practice, splitting train and test splits on chemicals can in special cases lead to a species not being seen before in training.

6.3 General Comparison

For the general comparison, the internal- and external validation performances between methods are compared. The internal validation uses 80% of the ECOTOX data, which is split into 5 distinct subsets of chemicals. Cross-validation is then performed on the internal set, such that at each iteration of 4 distinct subsets of chemicals are used to train a model, which is tested on the final subset. The internal validation experiments can show the robustness of the models to their training input. To advise the use of a model, its performance should be stable across the partial models on different folds.

The external validation is done on the held out 20% of the ECOTOX data. It is impor-

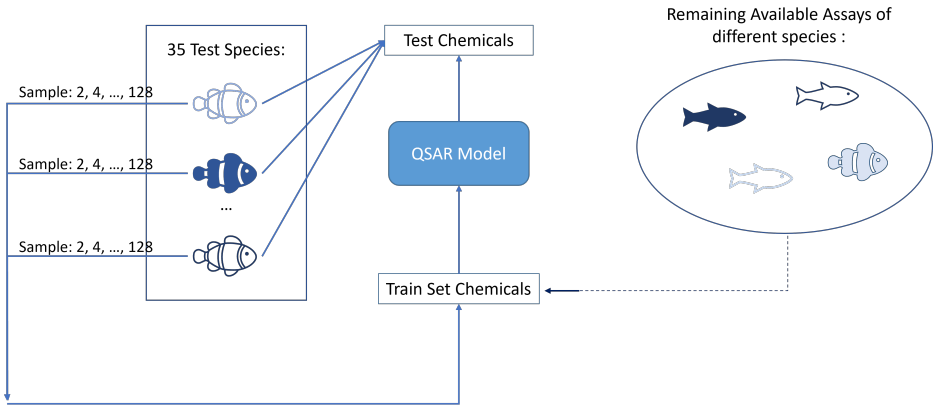


Figure 11: Assay Learning Curve: Using 35 test species in our train and test set, the number of assays per species is downsampled to create a learning curve. A harsh low-resource situation is simulated with the training set containing only the downsampled species, whereas the second scenario adds the remaining assays from other species to the training set too.

tant to note that the external validation set shares no common chemicals with the internal validation set, such that at no point information can be leaked through. All models are trained on the full internal validation set and tested on the held-out external validation set. For these experiments, the hyperparameters of all models are optimised as described in the methodology section.

6.4 Learning curves

To investigate the impact of low-resource data on the QSAR modelling techniques, two experiments are performed. For these experiments, datasets are artificially downsampled to imitate controlled low-resource scenarios, which are captured in learning curves. Learning curves can model the performance of machine learning algorithms with respect to a number of resources [35]. Here, we use empirical learning curves to model performances with respect to the number of assays for the test species and the number of species, respectively. Hence, the first experiment varies the number of samples per dataset whilst keeping the number of datasets fixed, whereas the latter experiment shows the impact of varying the number of datasets included.

6.4.1 Sampled Learning Curve

To imitate a low-resource situation, the amount of assays is undersampled for a fixed number of test species. Using a static test set, the test species are grouped to build the training dataset. Then, two scenarios are considered. The first scenario limits the training set only to the undersampled union of test species. The second scenario shows a less harsh low-resource scenario, in which the training set has access to all other species that are not selected for the test species set. Note that this second scenario does not use hyperparameter optimisation at run time. Both scenarios are shown in Figure 11.

35 test species are selected and scaled down according to the sequence of the square root of the powers of two, i.e. $\lfloor \sqrt{2^n} \rfloor$, with $n = 2, ..., 14$. By sampling this sequence beginning with 2, 4, 5, 8, 11..., the lowest resource situation tested had 2 assays per species, whereas the highest resource situation saw all species have 128 assays. In the first scenario, no external

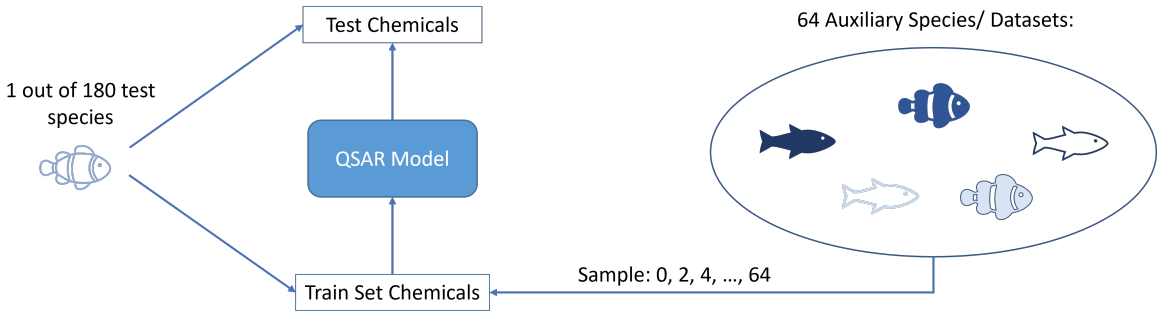


Figure 12: Meta-learning Curve: Iterating over 180 test species, a test species, its training and test set, is selected. Sampling 0,2,4...,64 auxiliary species into the training set, a new QSAR model is built on the test species training data with the additional auxiliary species data. With this, the impact of adding more species to aid in learning a test species is shown.

training data is added, so that the learning curves reveal not only the impact of adding more data for a task but also the impact of adding more assays to the models in general. The second scenario adds additional training resources and focuses on the impact of adding more data for a specific task. It is important to note that the sampled assays are a superset of the assays sampled for the prior iteration. Hence, the assays sampled at a training size of 2 samples per species will be included in the assays sampled at a training size of 4 assays per species, and so on. To reduce the effect of selecting the assays in a certain order, three random seeds are used that induce different sampling orders. Through all repetitions of the experiments, the test chemicals remain the same throughout ensuring that the difficulty of the test set remains constant. Further, to prevent any information leak, it is of utmost importance for the experiments that no compounds overlap in any of the training sets and the test sets.

6.4.2 Meta-Learning Curve

The meta-learning curve experiment investigates to what extent the number of species included in the training sets benefits QSAR algorithms. The hypothesis is that the algorithms that utilise knowledge between datasets will perform better with more species' datasets added. Its set-up is shown in Figure 12: a given species' test set is predicted from its training set. Iteratively, more species from a predefined auxiliary set of species are added to the training set, with which new QSAR models are built and the same test set is predicted. With this, the effect on the performance can be measured. Moreover, in each iteration, the added auxiliary datasets are a superset of the auxiliary datasets added in the prior iteration.

It is important to note that the set of test species (180 species) and the set of auxiliary species (64 species) are mutually exclusive. Further, the union of chemicals in the auxiliary datasets is mutually exclusive from the test chemicals set for the target species. In fact, as the union of all chemicals in the auxiliary datasets included the majority of chemicals, only 75% of the chemicals are included in the auxiliary datasets. This is done so that there are sufficient unseen chemicals to form test sets of the species to be predicted.

The number of samples in each auxiliary dataset varied from containing 64 samples to containing over one thousand samples. With this, a change in performance could also be attributed to the fact that many data samples were added to the training set, and not primarily to the addition of additional species to infer knowledge from. To reduce the variance in the

effect of sampling different sized auxiliary species, all species in the auxiliary dataset are sampled down to contain the same number of samples. By doing so, the effect of adding more samples to the dataset is constant. The experiments are repeated over 3 different seeds, that induce different downsampled auxiliary datasets. Note that the meta-learning curve does not use hyperparameter optimisation at run time.

The specific setup, see Figure 12, is then as follows: 180 species are predicted separately, such that each species has at least 3 unseen chemicals to predict. The auxiliary set of datasets to be sampled includes 64 datasets that are all scaled down to have 64 assays for each species. Each of the 180 species' test sets is predicted respectively, first by utilising only the species' own training examples. Then, 2 datasets are sampled from the auxiliary set of species and added to the training set, with which the models are trained anew. The species' test set is predicted to measure the performance of the models with the new training set. In the next iteration, more datasets are sampled from the auxiliary set, added to the training set, and the performance of the models on this new training set is measured. The number of *additional* species in the training set follows the sequence of the square root of the powers of two, i.e. $\lfloor\sqrt{2^n}\rfloor$, with $n = 2, ..., 12$. This sequence begins with 2, 4, 5, 8, 11 assays.

6.5 Low-resource datasets

Although the learning curves already investigate to what extent modelling techniques are impacted by less data, it is important to note that these experiments were performed on artificially downsampled datasets. Specifically, to show learning curves with long sampling sequences, the species with the most smaples were downsampled. Using large datasets to model low-resource situations has two main issues: the way the datasets are downsampled imposes a bias, and the assumption that downsampled large datasets behave like low-resource datasets can be false.

To show why this assumption may be false, it is vital to take a look at why there are disparities in the number of toxicity values per species. Eco-toxicity datasets are typically created on an ad-hoc basis, meaning that chemicals are typically tested once their production value exceeds the threshold and the REACH ruling requires toxicity values for the chemicals to be provided. For this, toxicity measures for daphnia, algae, and fish are required, although no exact species of these groupings is dictated [30]. Thus, it is typical that testing facilities use low-cost - in terms of time, expense and effort - species to acquire toxicity values. This leads to collections of toxicity datasets in species having a large variance in the number of samples: low-cost species have many samples, whereas species that are tested for a specific use case only are only available in small samples. Since there is a certain selection bias when choosing the species to test chemicals on, there is reason to believe that modelling of downsampled and low-resource datasets can not be assumed to behave equally.

Hence, section 7.3 focuses on real low-resource datasets and whether the trend found in the learning curves can be seen here, too. For this, the external validation of the QSAR models is taken, which sees the models training on 80% of the data and being tested on an external test set. We observe the datasets that have less than a certain number of compounds in their training set.

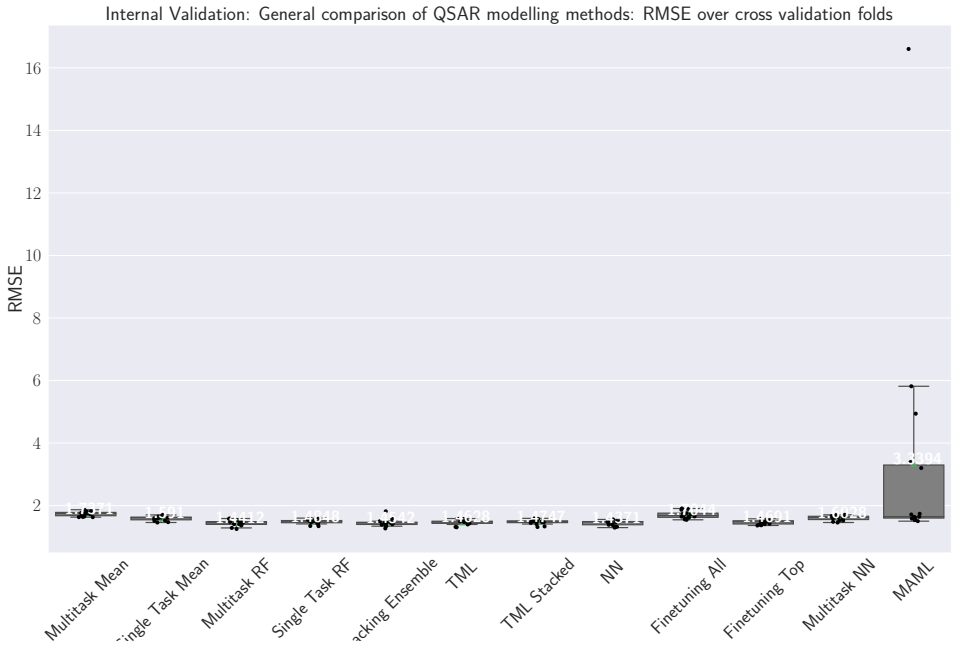


Figure 13: Internal Validation: Experimental results over 5 fold cross validation performed on 3 seeds. The performances are RMSE values averaged over partial models of each fold.

7 Results

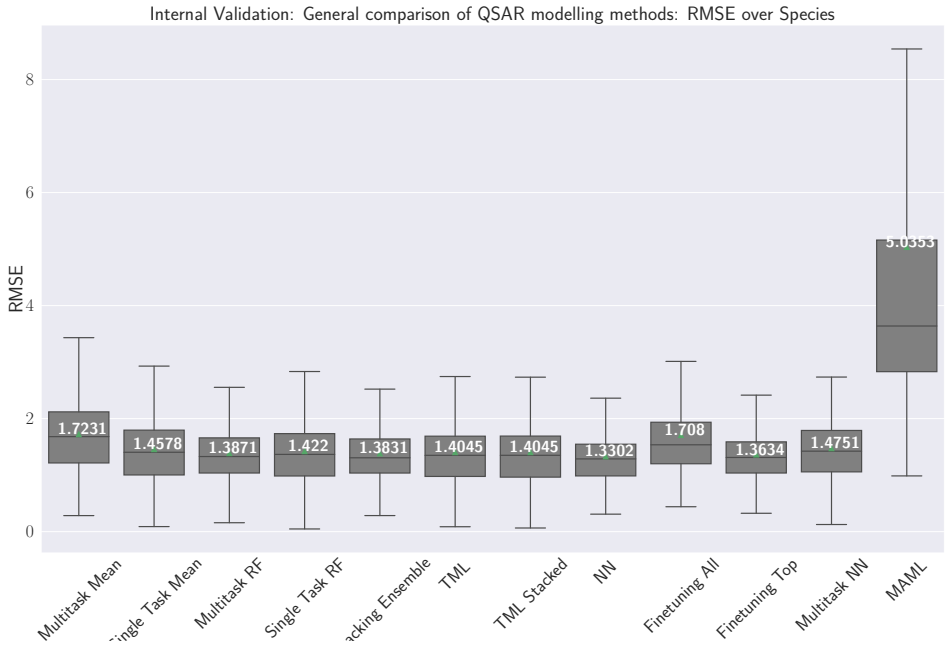
Having described how the experiments were performed, this section is dedicated to showing and discussing their results.

7.1 General Comparison

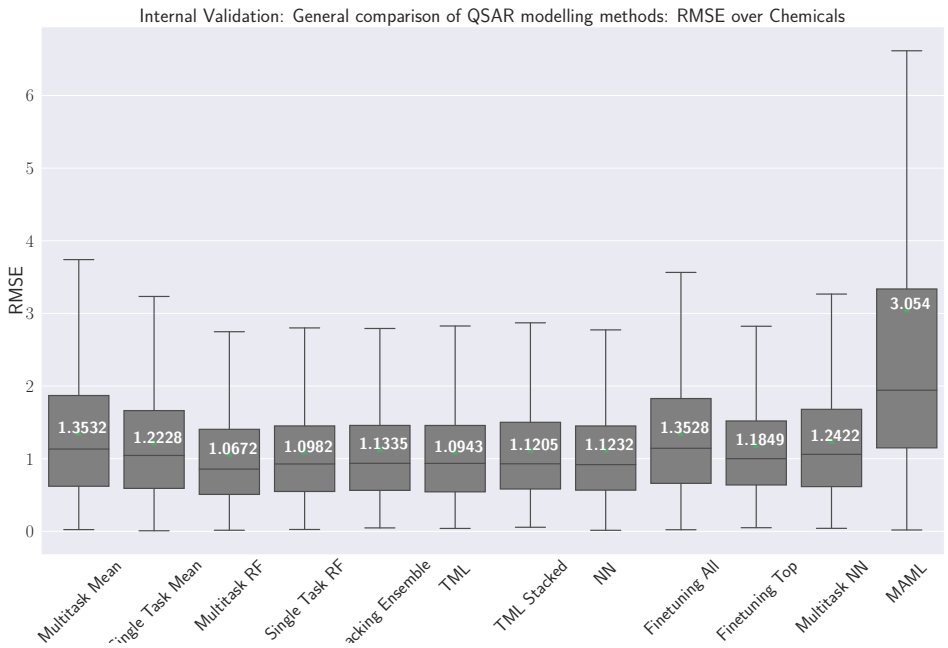
Figure 13 shows the RMSE performances over different cross-validation folds. This plot, therefore, does not only provide an indication of which algorithms perform well, but also of how stable the performances are. It can easily be seen that MAML's performances are very unstable. One partial MAML model, for instance, has an RMSE score of over 16, whereas the median of the performances is below an RMSE value of 2. This shows us that MAML is very sensitive across different partial models, and is not suitable for our problem. The best RMSE score of 1.43 over internal validation folds is achieved by the neural network.

Further results of the internal validation are shown in Figure 14. Averaging over species, the neural network achieves the best mean RMSE score of 1.33, whereas averaging over chemicals, the multitask random forest achieves the best mean RMSE score of 1.06. Further performance results of the general comparison can be found in the appendix.

After performing the Friedman test, it was found that the hypothesis that all mean ranks are the same could be rejected. As such, Figure 15 shows the results of the Nemenyi test in critical distance plots, such that when methods are not statistically different in their mean rank, a bar is drawn between them. Three critical distance plots are drawn: the critical distance plot over experimental folds in Figure 15a, over target species in Figure 15b, and over compounds in Figure 15c. The equivalent results for the external validation are shown in Figure 16.

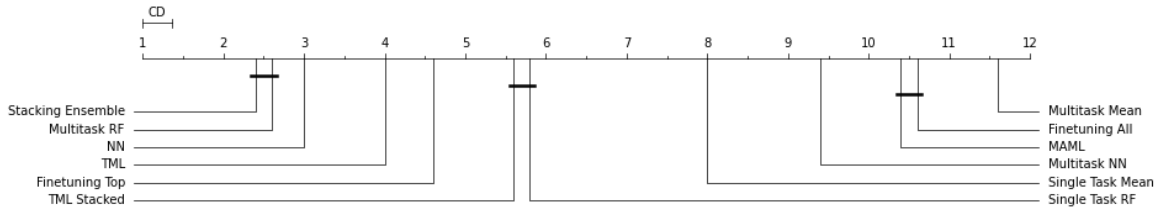


(a) Performances averaged over species.

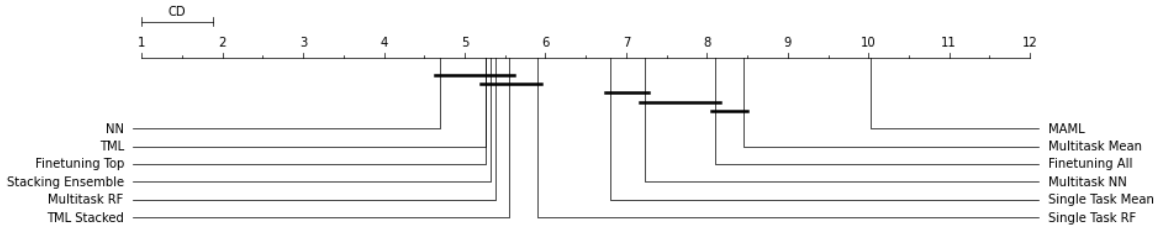


(b) Performances averaged over chemicals.

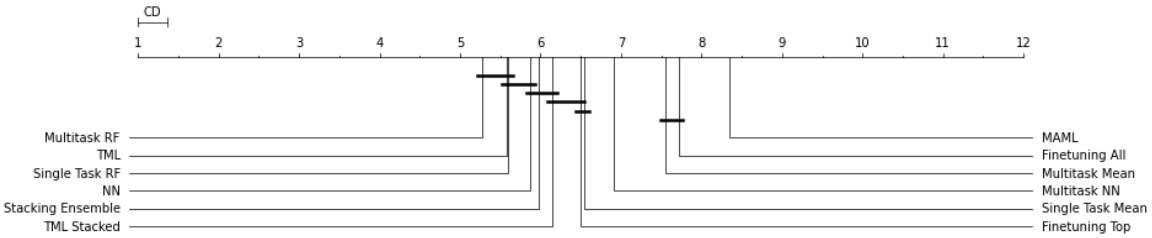
Figure 14: Performances of different algorithms in the internal validation. Performances are measured in RMSE.



(a) Statistical significant differences in mean rank over experimental folds evaluated via the Nemenyi Test. The critical distance that must be between two mean ranks to be significantly different is 0.36.

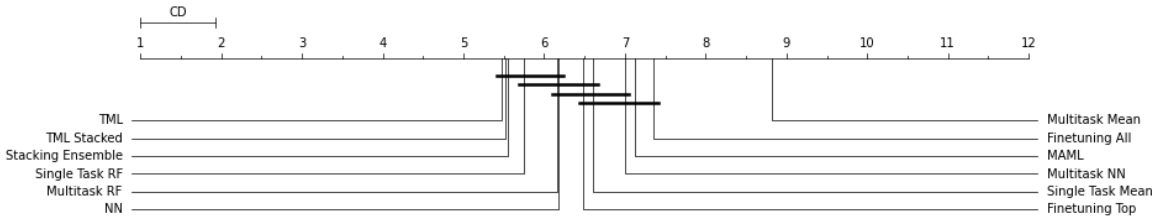


(b) Statistical significant differences in mean rank over target species evaluated via the Nemenyi Test. The critical distance that must be between two mean ranks to be significantly different is 0.89.

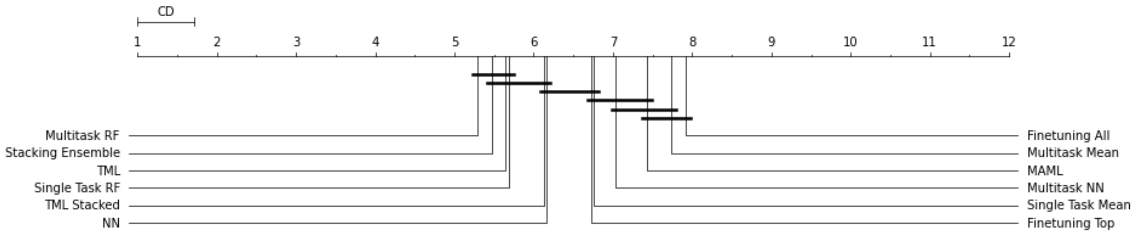


(c) Statistical significant differences in mean rank over chemicals evaluated via the Nemenyi Test. The critical distance that must be between two mean ranks to be significantly different is 0.36.

Figure 15: Critical distance plots for statistical differences of mean ranks in the *internal validation* with a significance value of $\alpha = 0.05$.



(a) Statistical significant differences in mean rank over target species evaluated via the Nemenyi Test. The critical distance that must be between two mean ranks to be significantly different is 0.93.



(b) Statistical significant differences in mean rank over chemicals evaluated via the Nemenyi Test. The critical distance that must be between two mean ranks to be significantly different is 0.72

Figure 16: Critical distance plots for statistical differences of mean ranks in the *external validation* with a significance value of $\alpha = 0.05$..

The methods MAML, fine-tuning all, multitask mean, multitask neural network (with multiple output nodes) and the single-task means are often statistically worse than other methods. The best-performing methods in the general comparison so far have been the multitask neural network with one output node, the stacked ensembling method, and the multitask random forest. The neural network is found to be significantly worse than the multitask random forest when considering performances over experimental folds, and both the neural network and the stacked ensemble method are significantly worse than the multitask random forest when averaging performances over chemicals, whereas the multitask random forest is not found to be significantly worse than any other method in any critical distance plot diagram.

7.2 Learning Curves

Having presented the results of the general comparison, the learning curves, investigating the impact of imitated low-resource situations are shown in this section.

7.2.1 Sampled Learning Curve

Figure 17 shows the impact of downsampling assays per species with no auxiliary data available. Figure 27 in the Appendix B splits the figure to enhance the readability of the individual trends. The curves nicely show improvement in performances with more data. In general, it can be seen that the multitask methods, such as the multitask random forest or the stacking ensemble model, outperform single-task methods, as these models can share knowledge between the test tasks. In an extremely low-resource setting, which can be seen to the left of the curves, the multitask ensemble models work superior. The neural network with one output node improves rapidly with more data samples.

It can also be seen that the transformational machine learning (TML) methods outperform

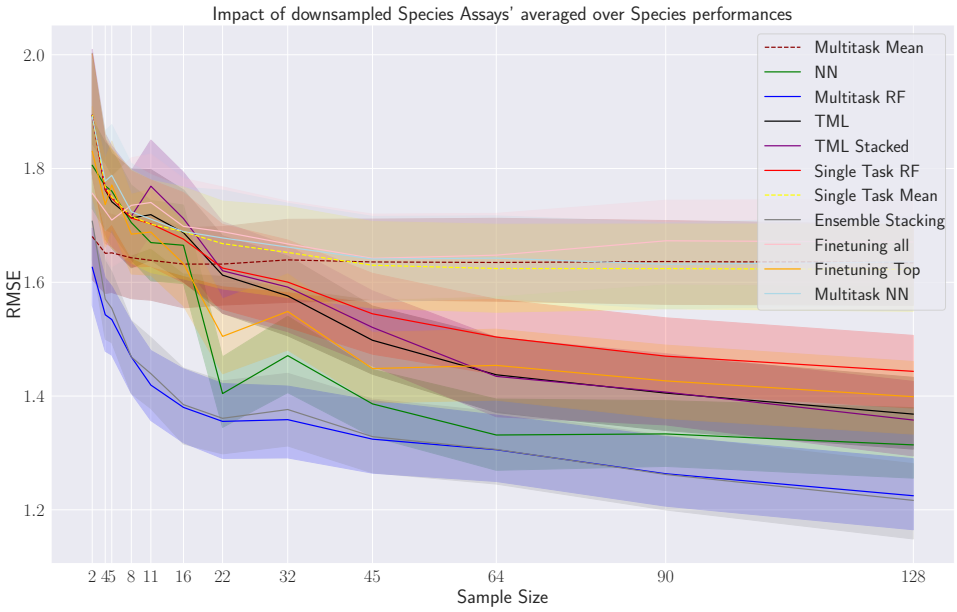


Figure 17: Learning curve performances when downsampling assays for test species. The perfomances are given in RMSE and are averaged over species.

the single-task random forest, once more data is gathered per test species. In general, the correlation between the single-task random forest models can be seen in the fact that when the single-task random forest model improves, the transformational embedding is also of higher quality and TML can outperform the single-task random forests. However, when the single-task random forest model does not perform well, TML suffers from its poor quality embeddings.

Fine-tuning all, the multitask neural network, and both mean baselines flatten at above 1.6 RMSE performances, indicating that both methods do not perform well.

The next learning curve experiment investigates the downsampling of the test species with auxiliary training data available, the results of which can be seen in Figure 18 and in Figure 29 in Appendix B. For the single-task mean and random forest, this is equivalent to the prior experiment, as they do not make use of the additional training data. It can be seen, however, that TML performs worse than the single-task random forest, showing that the addition of auxiliary training data of other species causes the transformational embedding to lose quality. This could be due to the fact that the embeddings are now larger (the size of all species in the training data), but that (some of) the added single-task random forests on the auxiliary species perform poorly. The poor performances could be caused by the fact that the single-task random forest performances seem to flatten from 64 assays per species each, and it is likely that many auxiliary species in the dataset have considerably fewer assays.

In Figure 18a, a clear separation can be seen between two groups of methods. As with the previous learning curve, Figure 28 in the Appendix B splits the figure to enhance the readability of the individual trends. The better performing methods are the multitask random forest, the multitask neural network with one output node, stacked ensemble and fine-tuning

top models - all of which can make use of the auxiliary data provided. To the very left of the learning curves of these methods, it can be seen that, when the test species only have two assays associated with them, the performances have improved from above 1.6 RMSE with no auxiliary data to around and below 1.4 RMSE with auxiliary data. The best performing method for the lowest sample sizes is fine-tuning top, which trains the neural network with one output node, and then finetunes its head on the 2 or 4 samples of a test species available. With more of the added data available the method performs worse, whereas the performances of the other three methods continue to increase.

When averaging the results over chemicals, as in Figure 18b, the same methods perform the best, although the ensemble stacking method is outperformed by the single-task random forest model.

7.2.2 Meta-learning Curve

Following the analysis of the impact of downsampling test species' assays, this section investigates the impact of adding more auxiliary species to the training set of a test species. The results can be seen in Figure 20 and in Appendix B in Figures 30 and 31, in which the single-task random forest model and single-task mean have constant performances, as they do not make use of the additional data. That being said the single-task random forest model performs very strongly.

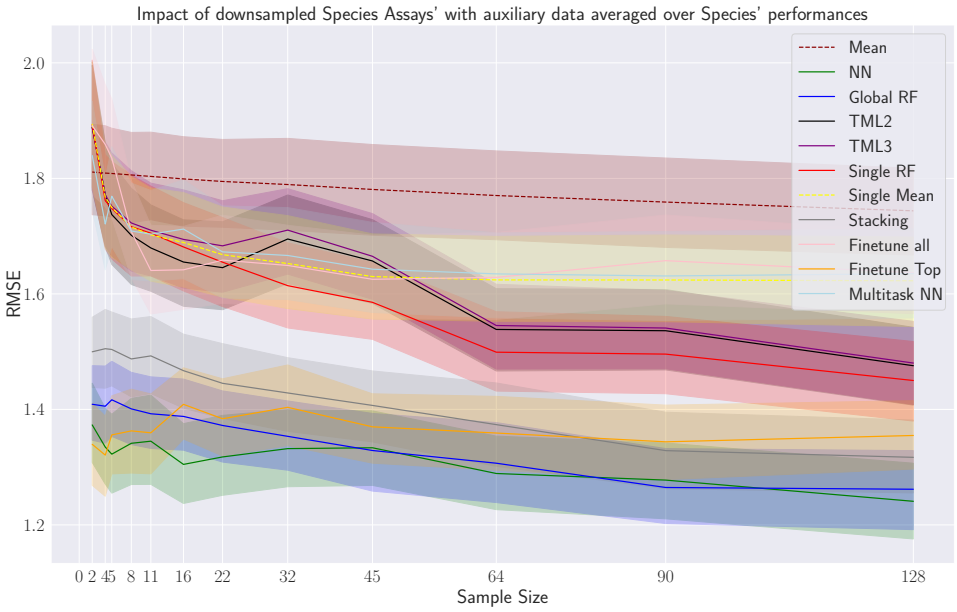
The good performance of TML in Figure 20a can be expected: as soon as sufficient auxiliary species (that all have a fixed size of 64 assays, with which a single-task random forest typically shows good performances) are added for the transformational embedding to be long enough, the TML outperforms the single-task random forest and becomes the best predictive method averaged over species. In general, the TML methods depend too much on the performance of the single-task random forests for the methods to be used robustly in regulatory tools.

The neural network with one output node and the related fine-tuning top method improve rapidly when adding more data. This is intuitive, since neural models typically require more data to perform well.

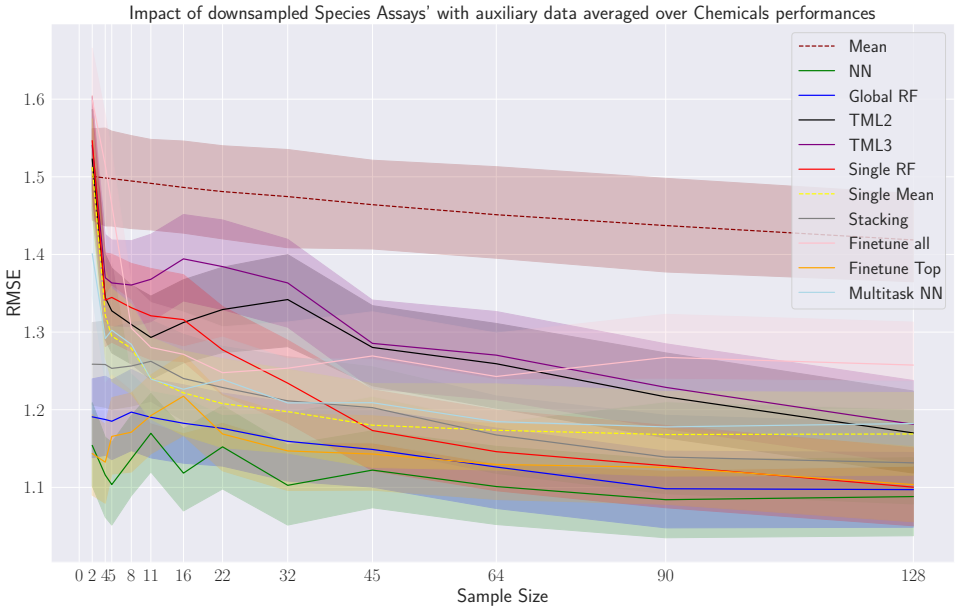
Perhaps most interesting to note is that the multitask random forest and stacking ensemble model, which have performed very well so far, perform worse with more added species in Figure 20a. In contrast, when considering the performances over chemicals in Figure 20b, these models outperform most models again.

It is important to consider performance results averaged over species or chemicals with caution here, as significant differences can be observed. A hypothesis addressing these differences may be different instance weightings in between single- and mulit-task models. Single-task (species-specific) and multitask models weigh their input differently. To achieve a generally good performance, a model aims to predict the majority of assays well. Due to the large differences in the number of assays with a certain chemical or species, the multitask model may aim to predict the largest groups of chemicals or species better, such that the average performance increases. A single-task model, however, inherently weights each species equally, as a separate model is built for each task.

Due to the difference in weighting, the single-task models optimise for good performance over species, whereas when the models are averaged over chemicals, the single-task models also aim to predict the majority group of chemicals most accurately.



(a) RMSE Performances averaged over species. Note the rift between two groups of methods.



(b) RMSE Performances averaged over chemicals.

Figure 18: Learning Curves showing the effect of downsampling the test species with auxiliary training data available.

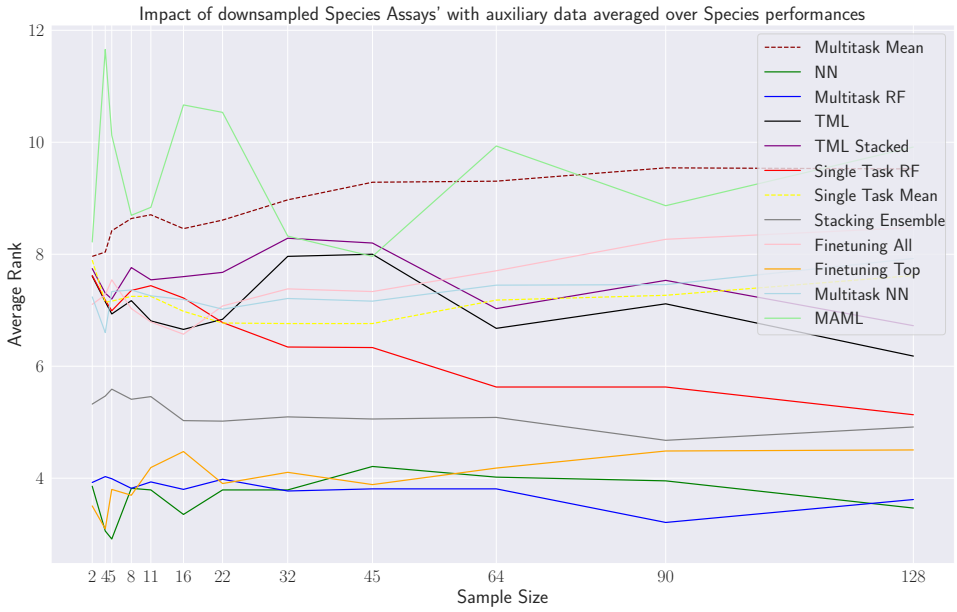


Figure 19: The mean ranks over different sample sizes calculated over species.

Future work should investigate how the choice of evaluation affects the relative order and the reasons behind the differences more. Further, it may be interesting to experiment with instance weighting explicitly by weighting training instances whilst building a model.

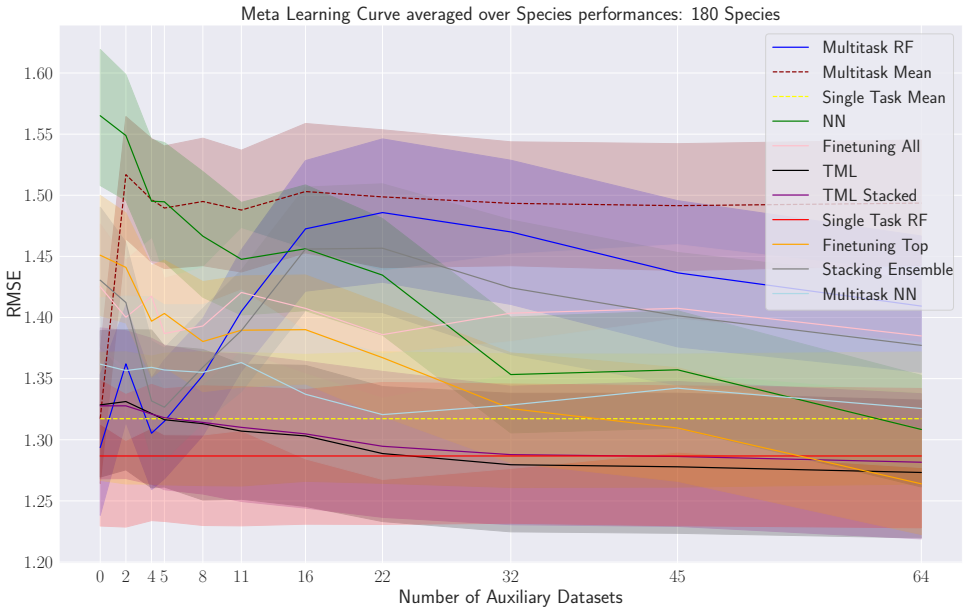
7.3 low-resource Datasets

This section takes a look at the performances of actual low-resource datasets that are predicted using the complete ECOTOX training set as auxiliary data. Only the methods that have been interesting so far and the mean baselines for comparison are shown in Figure 21. It can be seen that although the multitask mean is a decent predictor when merely one or two compounds have been seen for a species, it quickly is outperformed by the other models once more compounds have been seen. As observed in the sampled learning curve with auxiliary data, the fine-tuning top method performs very well with extremely low-resource datasets. Fine-tuning the neural network when only one compound has been seen so far seems to calibrate the neural network well, such that performance improves by over 1 RMSE. Once more samples are added, the fine-tuning top is soon outperformed by the neural network.

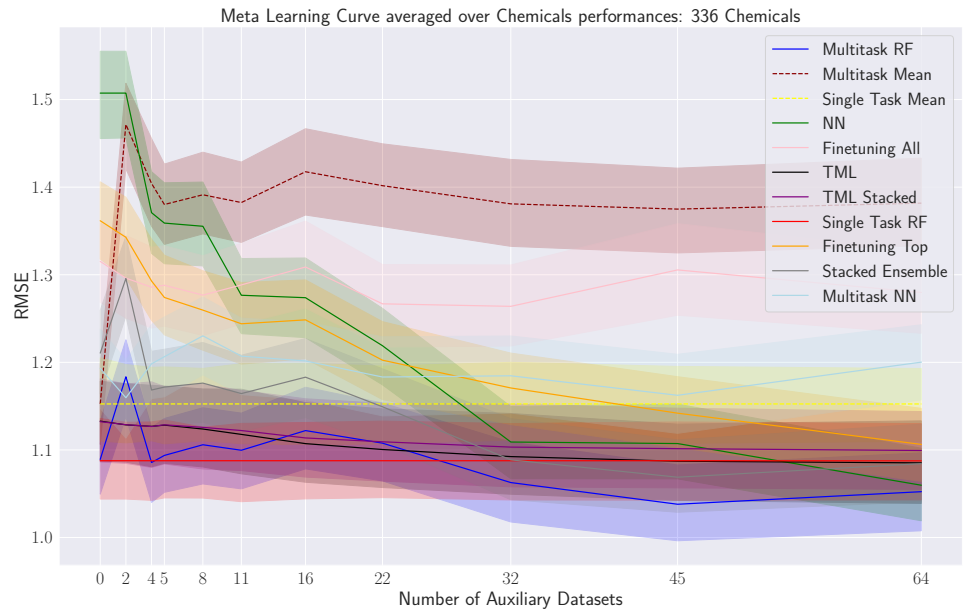
The single-task mean and random forest are very similar in their average ranks and continuously improve as more compounds are added to the training set. Generally, it can be seen that the methods' performances on low-resource datasets are rather noisy and conclusions should be drawn with caution. As such, towards higher compound counts, the ranks of the separate methods seem to grow more similar.

8 Conclusion

Our work has addressed modelling LC50 values of different aquatic species, specifically using 24,816 assays and 2674 unique chemicals of 351 separate species. We pay special attention

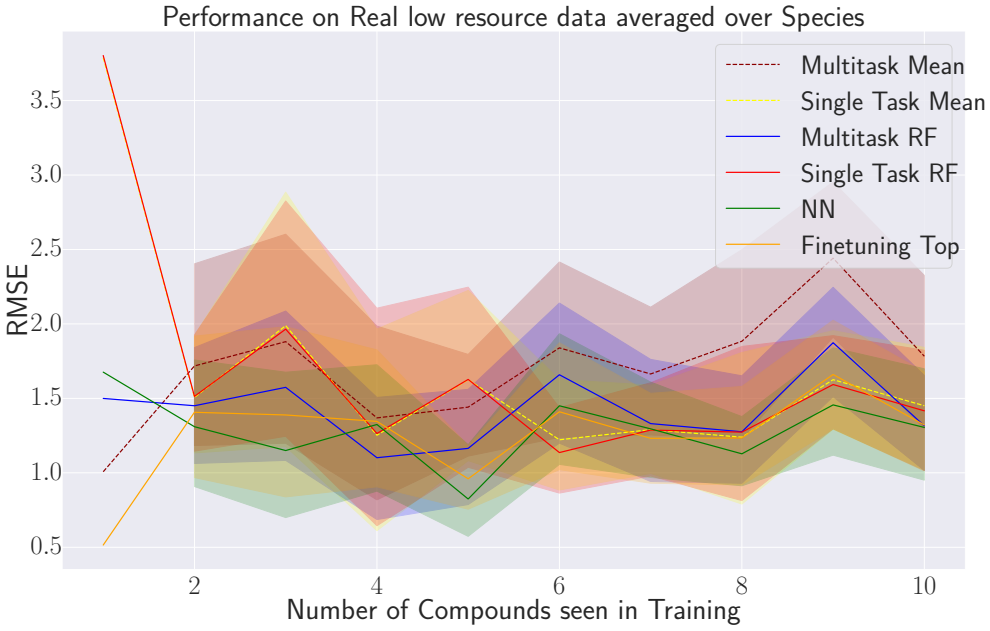


(a) RMSE Performances averaged over species.

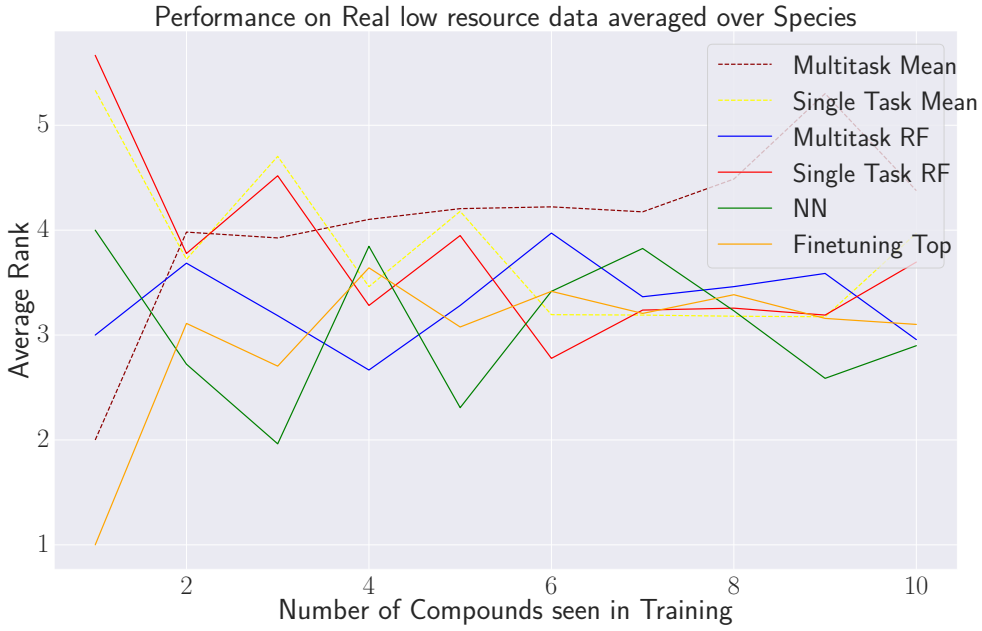


(b) RMSE Performances averaged over chemicals.

Figure 20: Meta-learning Curves showing the effect of adding more auxiliary species to the training set of a test species.



(a) RMSE Performances averaged over species for different compound counts seen.



(b) Average ranks averaged over species for different compound counts seen.

Figure 21: Prediction performances on actual low-resource datasets.

to addressing domain-specific requirements via the OECD principles, and we evaluate the models on a relevant use case. As such, the models predict unseen subsets of chemicals to imitate the most prevalent use for regulatory tools. In summary, the models solve aquatic toxicity regression tasks with adaptable exposure durations and a large general applicability domain for (unseen) chemicals.

Addressing this challenge, ten state-of-the-art QSAR methods were employed. As a single-task model, the single-task random forest model was employed. Further, multitask learning is used on random forest regressors, neural networks, and Sheffield & Judson [2019]'s stacked ensemble learner. In addition to transformational machine learning [41], transfer learning in the form of fine-tuning is used, and MAML, an optimisation-based meta-learning approach, is employed.

From general comparisons of internal and external validation of the models, we find that methods of sharing knowledge between tasks outperform the single-task models. Fine-tuning all weights of a network to one task performed poorly. Moreover, the use of the optimisation-based meta-learning method MAML proved to be very sensitive to different partial models being built. As robustness is a key factor for QSAR modelling, this method is not applicable to our aquatic toxicity problem. In general, all multitask models perform well. We hypothesise that the multitask models can utilise a species' phylum and class as a measure of task relatedness to enhance knowledge sharing.

Using learning curves, the performance of the methods was modelled with respect to the number of assays per species and the number of species, respectively. Intuitively, predictive performances for a species increase with more assays available for a given species. The multitask neural network with one output node and the fine-tuning method, in particular, see large increases in their performances with more data, as is typical for neural network models. When faced with few assays for a species, the knowledge-sharing models can benefit from additional auxiliary data and increase their predictive performance for the test species by an RMSE in excess of 2.

The use of transformational machine learning, in which transformational embeddings captured chemical-exposure duration pairs, proved very sensitive to the performance of the single-task models the embedding was built on. Especially in low-resource situations when the embeddings were short and built on fewer species, the single-task models have a higher impact on the transformational machine learning's performance. This sensitivity makes its use for QSAR modelling with low-resources, in terms of the number of targets and assays, infeasible.

On actual low-resource datasets, similar tendencies can be seen in the artificial downsampling. With the auxiliary training data, the multitask neural network with a single output performs well. In fact, when fine-tuning its head on extreme low-resource species, its predictions outperform all other models.

Finally, we advise the use of the multitask random forest model for aquatic toxicity QSAR modelling, as well as for low-resource QSAR modelling in general. Its performance is robust over both internal and external validations and performance analysis averaged on chemicals and species. Furthermore, the multitask model performs well in imitated low-resource situations.

The proposed multitask random forest model is the first aquatic species model that predicts toxicity on a species level for multiple species between multiple phyla. To investigate the

impact of our species-level model further, its use in species sensitivity distributions should be explored.

As we believe that the inclusion of class and phyla information aids the multitask models, we hypothesise that a continuous distance measure between the species could further enhance these models. As such, in future work, different, potentially more easily obtainable measures of target relatedness could be investigated. Furthermore, our investigation into low resource situations via learning curve has given more insight into the singular approaches. A future investigation with more data available could further control the effect of singular chemicals on the learning curves.

To conclude, we successfully propose multitask models on a species level which predict toxicity on flexible exposure duration and a large chemical applicability domain, showing promising results for models with general chemical applicability as well as applicability across phyla.

In our work, we bring popular knowledge sharing techniques to the aquatic toxicity domain. Further, we investigate the individual methods on their capability to handle different resource scenarios. The insight gained here is valuable for not only the aquatic toxicity domain, but QSAR modelling in general. The use of multitask learning shows promising results, as well as its combination with simple transfer learning for low-resource situations. With this research, we hope to not only take a step towards mitigating the need for in vivo experiments, but also hope to inspire the use of knowledge-sharing approaches for other low resource QSAR problems.

References

[1] Altae-Tran, H., Ramsundar, B., Pappu, A. S., and Pande, V. Low data drug discovery with one-shot learning. *ACS central science*, 3(4):283–293, 2017.

[2] Benfenati, E., Manganaro, A., and Gini, G. C. Vega-QSAR: AI inside a platform for predictive toxicology. In *PAI@ AI* IA*, pp. 21–28, 2013.

[3] Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. Algorithms for hyper-parameter optimization. volume 24, 2011.

[4] Breiman, L. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

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

[6] Cai, C., Wang, S., Xu, Y., Zhang, W., Tang, K., Ouyang, Q., Lai, L., and Pei, J. Transfer learning for drug discovery. *Journal of Medicinal Chemistry*, 63(16):8683–8694, 2020. doi: 10.1021/acs.jmedchem.9b02147. URL https://doi.org/10.1021/acs.jmedchem.9b02147. PMID: 32672961.

[7] Caruana, R. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[8] Cherkasov, A., Muratov, E. N., Fourches, D., Varnek, A., Baskin, I. I., Cronin, M., Dearden, J., Gramatica, P., Martin, Y. C., Todeschini, R., et al. QSAR modeling: where have you been? where are you going to? *Journal of medicinal chemistry*, 57(12): 4977–5010, 2014.

[9] Dahl, G. E., Jaitly, N., and Salakhutdinov, R. Multi-task neural networks for QSAR predictions, 2014.

[10] Demšar, J. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, 7:1–30, 2006.

[11] Durant, J. L., Leland, B. A., Henry, D. R., and Nourse, J. G. Reoptimization of MDL keys for use in drug discovery. *Journal of Chemical Information and Computer Sciences*, 42(6):1273–1280, 2002. doi: 10.1021/ci010132r. URL https://doi.org/10.1021/ci010132r. PMID: 12444722.

[12] (ECHA), E. C. A. Guidance on information requirements and chemical safety assessment, 2008.

[13] Erhan, D., L'heureux, P.-J., Yue, S. Y., and Bengio, Y. Collaborative filtering on a family of biological targets. *Journal of chemical information and modeling*, 46(2): 626–635, 2006.

[14] Feurer, M., Klein, A., Eggensperger, Katharina Springenberg, J., Blum, M., and Hutter, F. Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems 28 (2015)*, pp. 2962–2970, 2015.

[15] Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

[16] Fox, D., van Dam, R., Fisher, R., Batley, G., Tillmanns, A., Thorley, J., Schwarz, C., Spry, D., and McTavish, K. Recent developments in species sensitivity distribution modeling. *Environmental Toxicology and Chemistry*, 40(2):293–308, 2021.

[17] Friedman, M. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92, 1940.

[18] Gajewicz-Skretna, A., Gromelski, M., Wyrzykowska, E., Furuhama, A., Yamamoto, H., and Suzuki, N. Aquatic toxicity (pre) screening strategy for structurally diverse chemicals: global or local classification tree models? *Ecotoxicology and Environmental Safety*, 208:111738, 2021.

[19] Gaulton, A., Hersey, A., Nowotka, M., Bento, A. P., Chambers, J., Mendez, D., Mutowo, P., Atkinson, F., Bellis, L. J., Cibrián-Uhalte, E., et al. The ChEMBL database in 2017. *Nucleic acids research*, 45(D1):D945–D954, 2017.

[20] Gramatica, P. Principles of QSAR models validation: internal and external. *QSAR & combinatorial science*, 26(5):694–701, 2007.

[21] Ho, T. K. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pp. 278–282. IEEE, 1995.

[22] Huisman, M., van Rijn, J. N., and Plaat, A. A preliminary study on the feature representations of transfer learning and gradient-based meta-learning techniques. In *Fifth Workshop on Meta-Learning at the Conference on Neural Information Processing Systems*, 2021.

[23] Huisman, M., van Rijn, J. N., and Plaat, A. *Metalearning for deep neural networks*, pp. 237–267. Springer International Publishing, Cham, 2022. ISBN 978-3-030-67024-5. doi: 10.1007/978-3-030-67024-5_13. URL https://doi.org/10.1007/978-3-030-67024-5_13.

[24] Ioffe, S. and Szegedy, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

[25] Jaeger, S., Fulle, S., and Turk, S. Mol2vec: unsupervised machine learning approach with chemical intuition. *Journal of chemical information and modeling*, 58(1):27–35, 2018.

[26] Jiang, D., Wu, Z., Hsieh, C.-Y., Chen, G., Liao, B., Wang, Z., Shen, C., Cao, D., Wu, J., and Hou, T. Could graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models. *Journal of cheminformatics*, 13(1):1–23, 2021.

[27] Kingma, D. P. and Ba, J. Adam: a method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[28] Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

[29] Li, X. and Fourches, D. Inductive transfer learning for molecular activity prediction: next-gen QSAR models with MolPMoFiT. *Journal of Cheminformatics*, 12(1):1–15, 2020.

[30] Lunghini, F., Marcou, G., Azam, P., Enrici, M., Van Miert, E., and Varnek, A. Consensus QSAR models estimating acute toxicity to aquatic organisms from different trophic levels: Algae, daphnia and fish. *SAR and QSAR in Environmental Research*, 31(9): 655–675, 2020.

[31] Martin, T. Toxicity estimation software tool (TEST). *U.S. Environmental Protection Agency*, 2016.

[32] Mayr, A., Klambauer, G., Unterthiner, T., Steijaert, M., Wegner, J. K., Ceulemans, H., Clevert, D.-A., and Hochreiter, S. Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chemical science*, 9(24):5441–5451, 2018.

[33] Microsoft. Neural Network Intelligence, 1 2021. URL https://github.com/microsoft/nni.

[34] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, volume 26, 2013.

[35] Mohr, F. and van Rijn, J. N. Learning curves for decision making in supervised machine learning - A survey. *CoRR*, abs/2201.12150, 2022. URL https://arxiv.org/abs/2201.12150.

[36] Morgan, H. L. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of chemical documentation*, 5(2):107–113, 1965.

[37] Nemenyi, P. B. *Distribution-free multiple comparisons*. Princeton University, 1963.

[38] Nguyen, C. Q., Kreatsoulas, C., and Branson, K. M. Meta-learning GNN initializations for low-resource molecular property prediction. In *ICML 2020 Workshop on Graph Representation Learning and Beyond*, 2020.

[39] OECD. *Guidance document on the validation of (quantitative) structure-activity relationship [(Q)SAR] models*. 2014. doi: https://doi.org/https://doi.org/10.1787/9789264085442-en. URL https://www.oecd-ilibrary.org/content/publication/9789264085442-en.

[40] Olier, I., Sadawi, N., Bickerton, G. R., Vanschoren, J., Grosan, C., Soldatova, L., and King, R. D. Meta-QSAR: a large-scale application of meta-learning to drug design and discovery. *Machine Learning*, 107(1):285–311, 2018.

[41] Olier, I., Orhobor, O. I., Dash, T., Davis, A. M., Soldatova, L. N., Vanschoren, J., and King, R. D. Transformational machine learning: learning how to learn from many related scientific problems. *Proceedings of the National Academy of Sciences*, 118(49), 2021.

[42] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: an imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

[43] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[44] Raimondo, S., Jackson, C. R., and Barron, M. G. Influence of taxonomic relatedness and chemical mode of action in acute interspecies estimation models for aquatic species. *Environmental science & technology*, 44(19):7711–7716, 2010.

[45] Ramsundar, B., Kearnes, S., Riley, P., Webster, D., Konerding, D., and Pande, V. Massively multitask networks for drug discovery, 2015. URL https://arxiv.org/abs/1502.02072.

[46] Ramsundar, B., Liu, B., Wu, Z., Verras, A., Tudor, M., Sheridan, R. P., and Pande, V. Is multitask deep learning practical for pharma? *Journal of chemical information and modeling*, 57(8):2068–2076, 2017.

[47] Ramsundar, B., Eastman, P., Walters, P., Pande, V., Leswing, K., and Wu, Z. *Deep learning for the life sciences*. O'Reilly Media, 2019. https://www.amazon.com/Deep-Learning-Life-Sciences-Microscopy/dp/1492039837.

[48] Reddy, A. S., Pati, S. P., Kumar, P. P., Pradeep, H., and Sastry, G. N. Virtual screening in drug discovery - a computational perspective. *Current Protein and Peptide Science*, 8(4):329–351, 2007.

[49] Rogers, D. and Hahn, M. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

[50] Sadawi, N., Olier, I., Vanschoren, J., van Rijn, J. N., Besnard, J., Bickerton, R., Grosan, C., Soldatova, L., and King, R. D. Multi-task learning with a natural metric for quantitative structure activity relationship learning. *Journal of Cheminformatics*, 11(1):1–13, 2019.

[51] Sheffield, T. Y. and Judson, R. S. Ensemble QSAR modeling to predict multispecies fish toxicity lethal concentrations and points of departure. *Environmental science & technology*, 53(21):12793–12802, 2019.

[52] Simoes, R. S., Maltarollo, V. G., Oliveira, P. R., and Honorio, K. M. Transfer and multi-task learning in QSAR modeling: advances and challenges. *Frontiers in pharmacology*, 9:74, 2018.

[53] Singh, K. P., Gupta, S., Kumar, A., and Mohan, D. Multispecies QSAR modeling for predicting the aquatic toxicity of diverse organic chemicals for regulatory toxicology. *Chemical research in toxicology*, 27(5):741–753, 2014.

[54] Smola, A. J. and Schölkopf, B. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[55] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[56] Thomas, P. C., Bicherel, P., and Bauer, F. J. How in silico and QSAR approaches can increase confidence in environmental hazard and risk assessment. *Integrated Environmental Assessment and Management*, 15(1):40–50, 2019.

[57] Thrun, S. and Pratt, L. Learning to learn: introduction and overview. In *Learning to learn*, pp. 3–17. Springer, 1998.

[58] Vanschoren, J. Meta-learning: a survey. *arXiv preprint arXiv:1810.03548*, 2018.

[59] Wandall, B., Hansson, S. O., and Rudén, C. Bias in toxicology. *Archives of Toxicology*, 81(9):605–617, 2007.

[60] Weininger, D. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

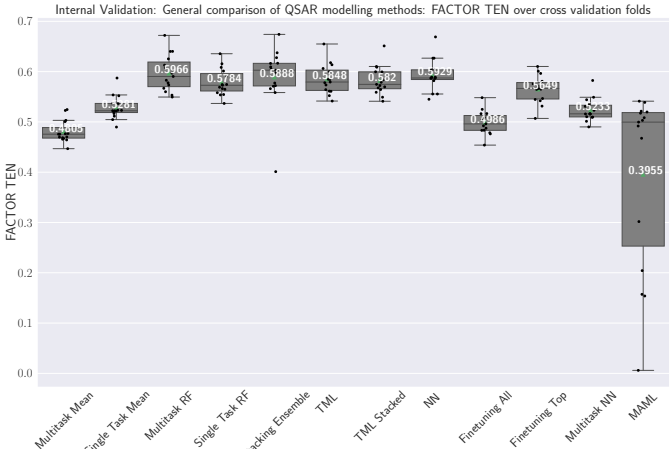
[61] Wu, K. and Wei, G.-W. Quantitative toxicity prediction using topology based multitask deep neural networks. *Journal of chemical information and modeling*, 58(2):520–531, 2018.

[62] Yap, C. W. Padel-descriptor: an open source software to calculate molecular descriptors and fingerprints. *Journal of computational chemistry*, 32(7):1466–1474, 2011.

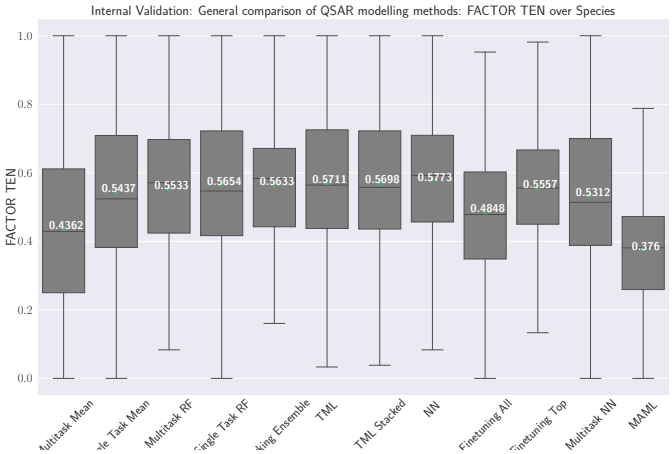
[63] Zhou, L., Fan, D., Yin, W., Gu, W., Wang, Z., Liu, J., Xu, Y., Shi, L., Liu, M., and Ji, G. Comparison of seven in silico tools for evaluating of daphnia and fish acute toxicity: case study on chinese priority controlled chemicals and new chemicals. *BMC bioinformatics*, 22(1):1–31, 2021.

A Additional Material on the general comparison

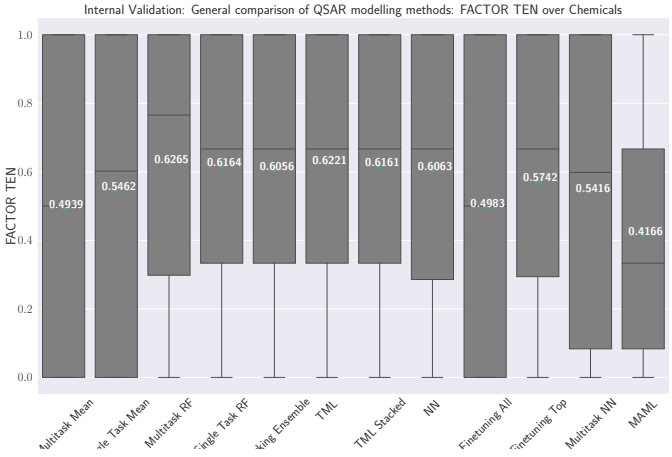
The appendix provides extra figures illustrating the results of our general comparison experiments. Specifically, the figures describe the external validation, as well as the performance values measured in percentage of predictions within factors of the actual value.



(a) Performance values averaged over folds.

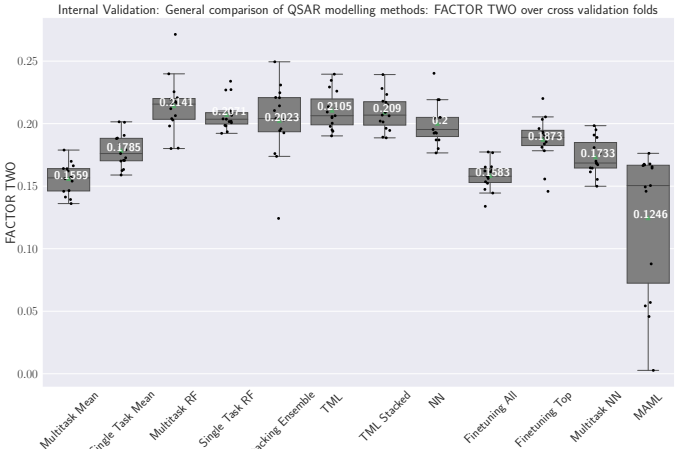


(b) Performance values averaged over species.

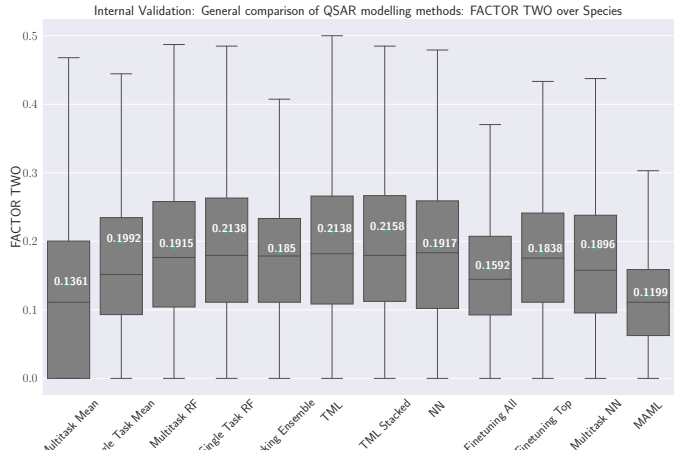


(c) Performance values averaged over chemicals.

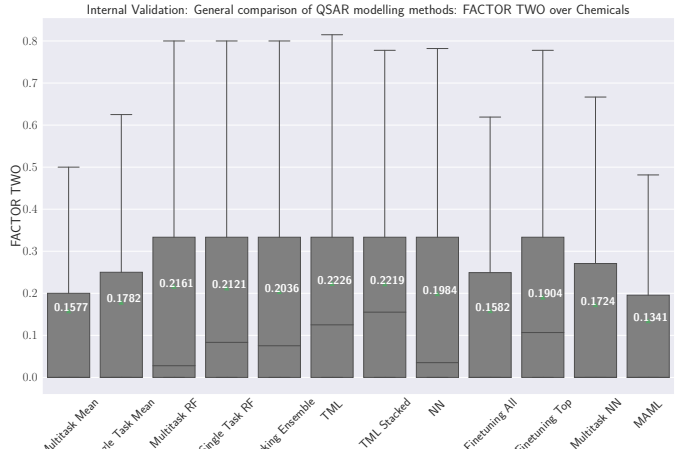
Figure 22: Performance values of different algorithms in the internal validation. Performance are measured in percentages of predictions within a factor of 10 of the actual value.



(a) Performance values averaged over folds.

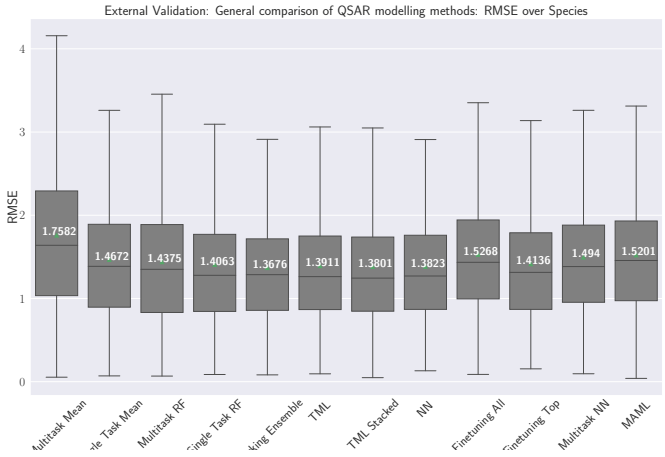


(b) Performance values averaged over species.

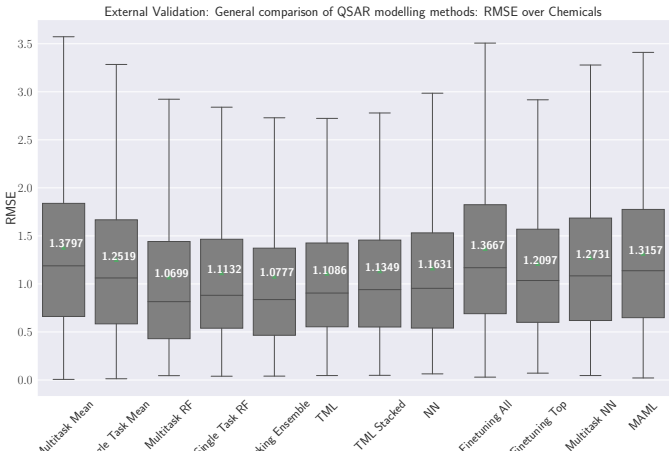


(c) Performance values averaged over chemicals.

Figure 23: Performance values of different algorithms in the internal validation. Performance are measured in percentages of predictions within a factor of 2 of the actual value.

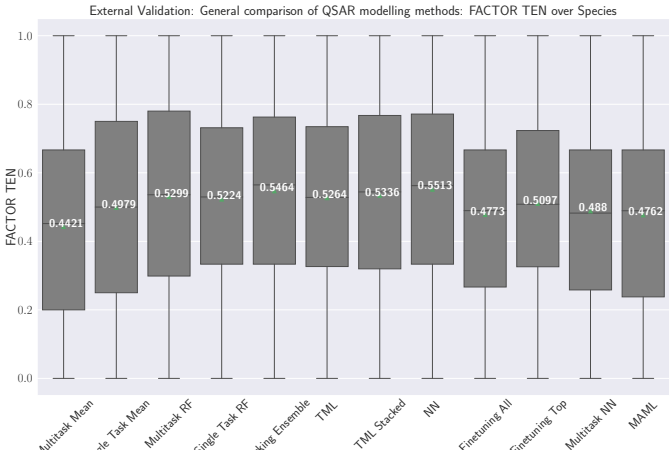


(a) Performance values averaged over species.

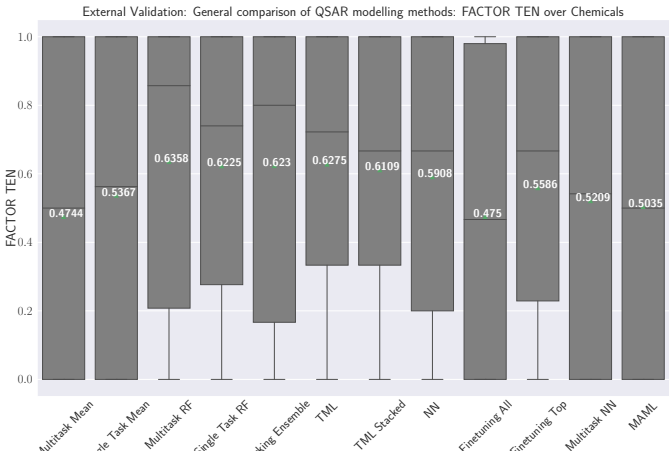


(b) Performance values averaged over chemicals.

Figure 24: Performance values of different algorithms in the external validation. Performance are measured in RMSE.

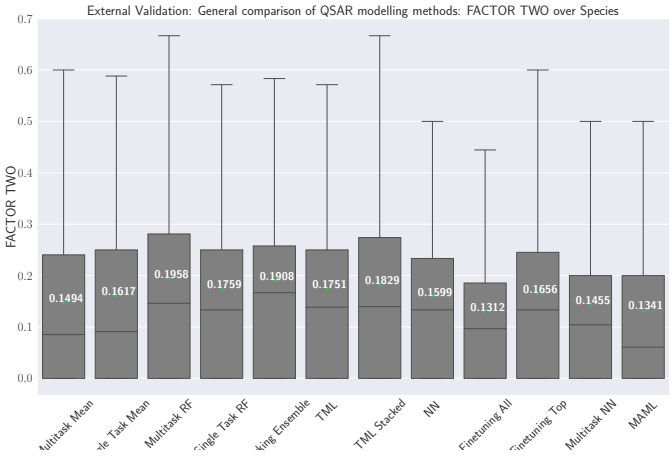


(a) Performance values averaged over species.

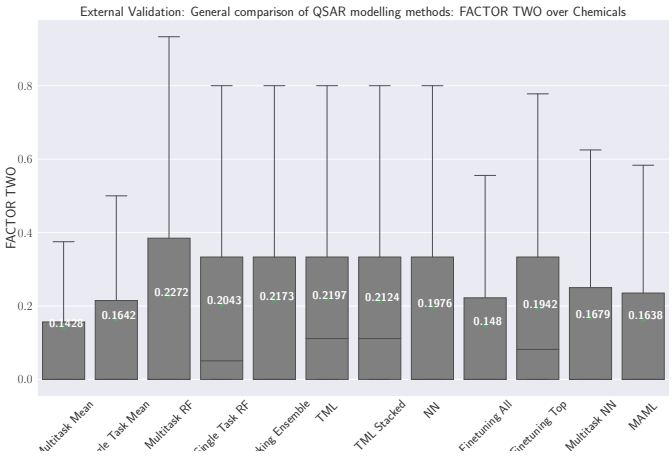


(b) Performance values averaged over chemicals.

Figure 25: Performance values of different algorithms in the external validation. Performance are measured in percentages of predictions within a factor of 10 of the actual value.



(a) Performance values averaged over species.

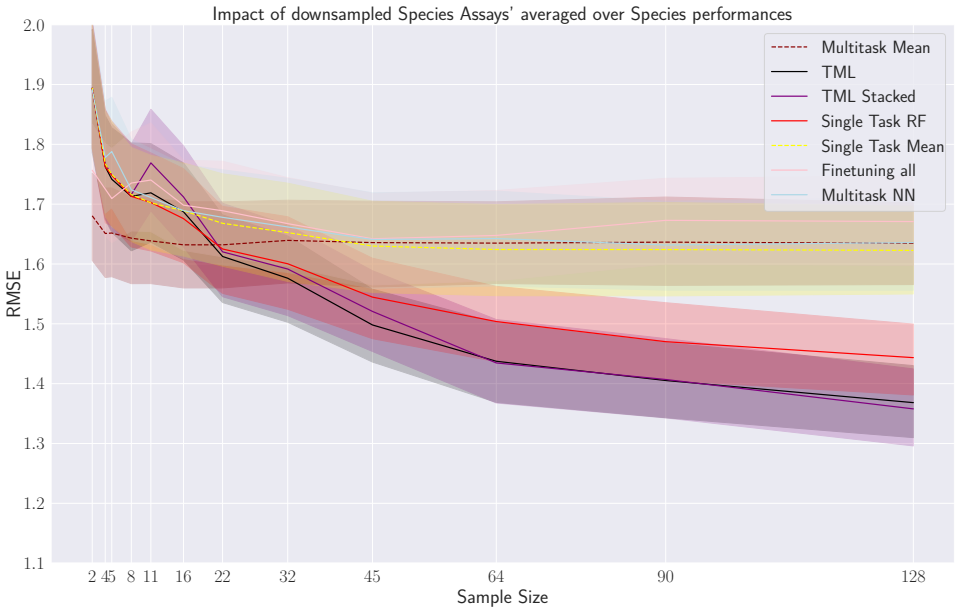


(b) Performance values averaged over chemicals.

Figure 26: Performance values of different algorithms in the external validation. Performance are measured in percentages of predictions within a factor of 2 of the actual value.

B Additional Figures on the Learning Curves

To highlight the performance of the singular methods in our learning curve experiments, the appendix provides extra figures that group the methods to increase readability.



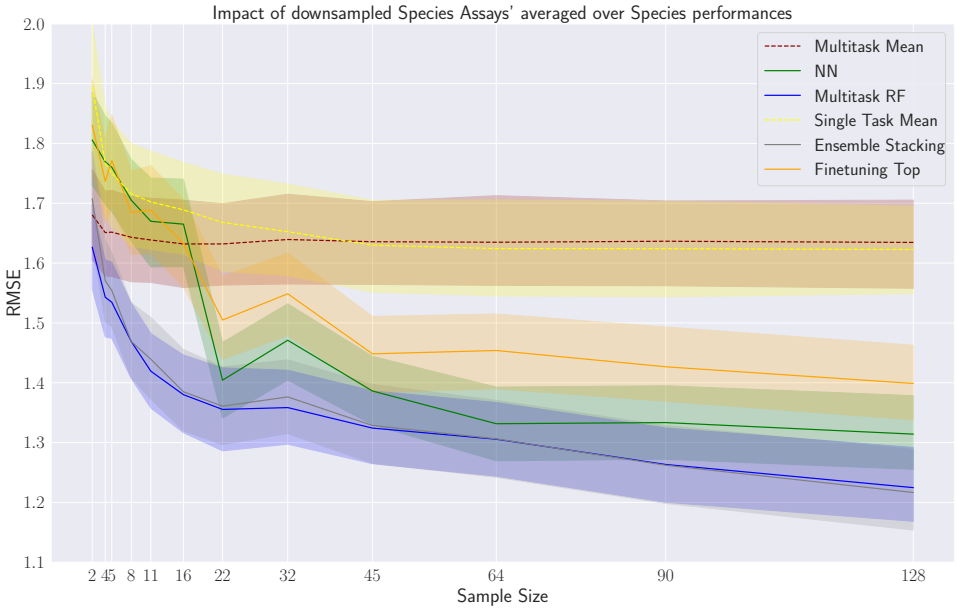
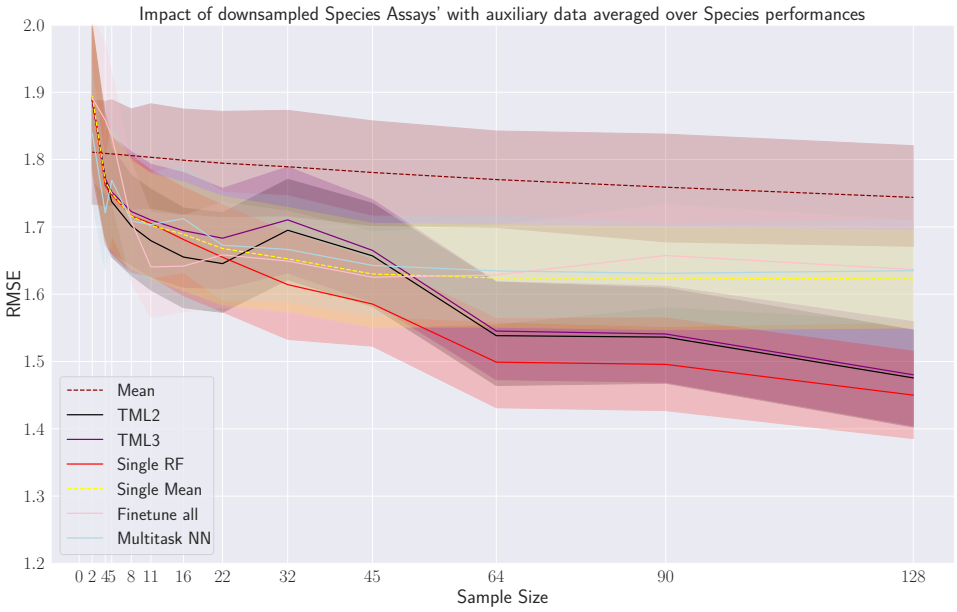
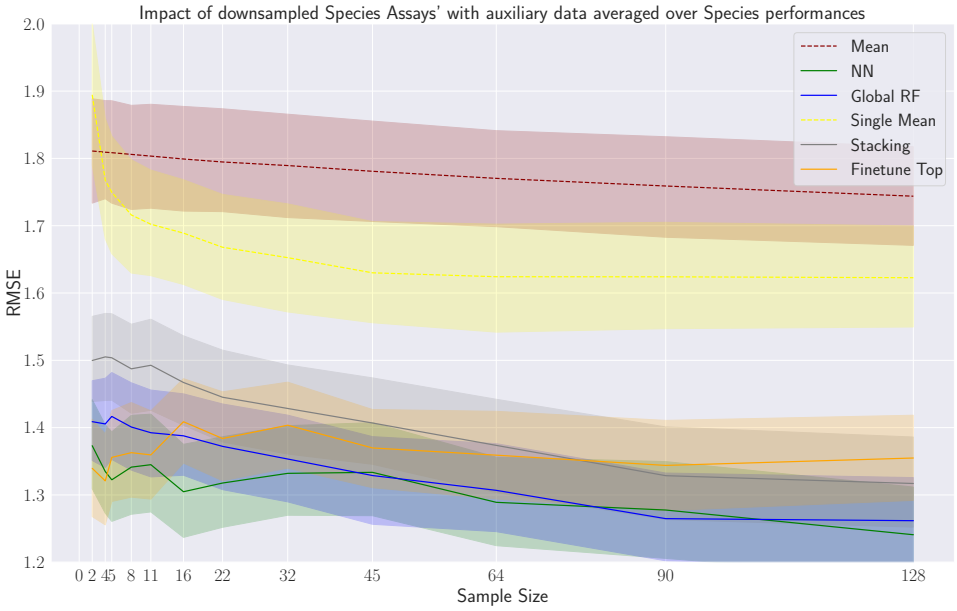


Figure 27: Learning curve performance when downsampling assays for test species. The perfomances are given in RMSE and are averaged over species.

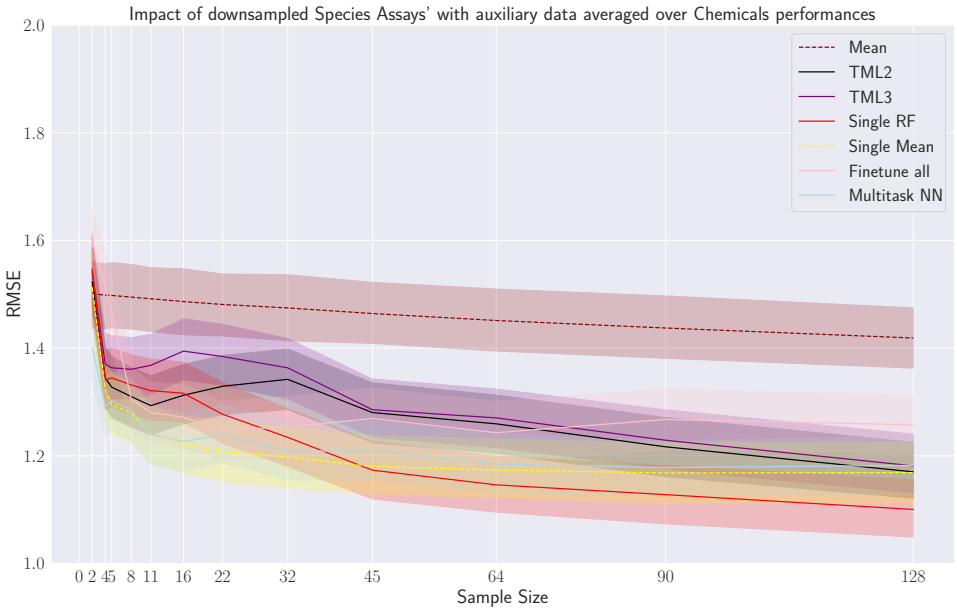


(a) group 1.

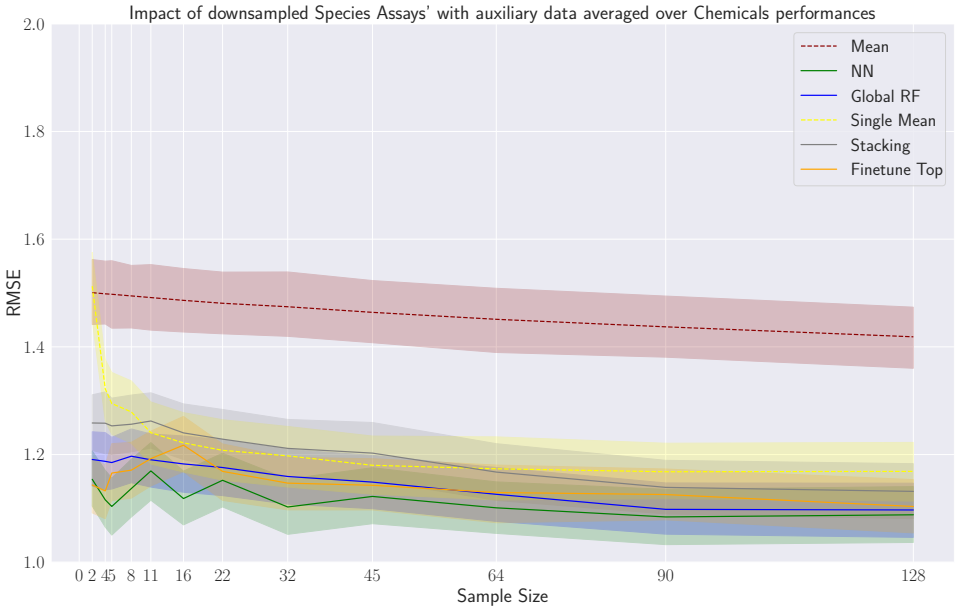


(b) group 2.

Figure 28: Learning Curves showing the effect of downsampling the test species with auxiliary training data available: RMSE Performance averaged over species.

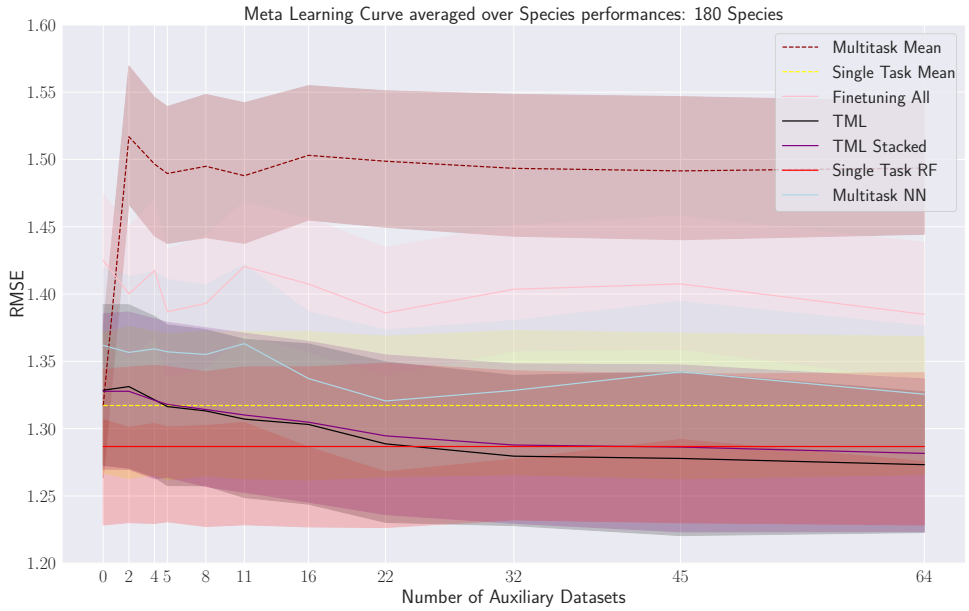


(a) group 1.



(b) group 2.

Figure 29: Learning Curves showing the effect of downsampling the test species with auxiliary training data available: RMSE Performance averaged over chemicals.



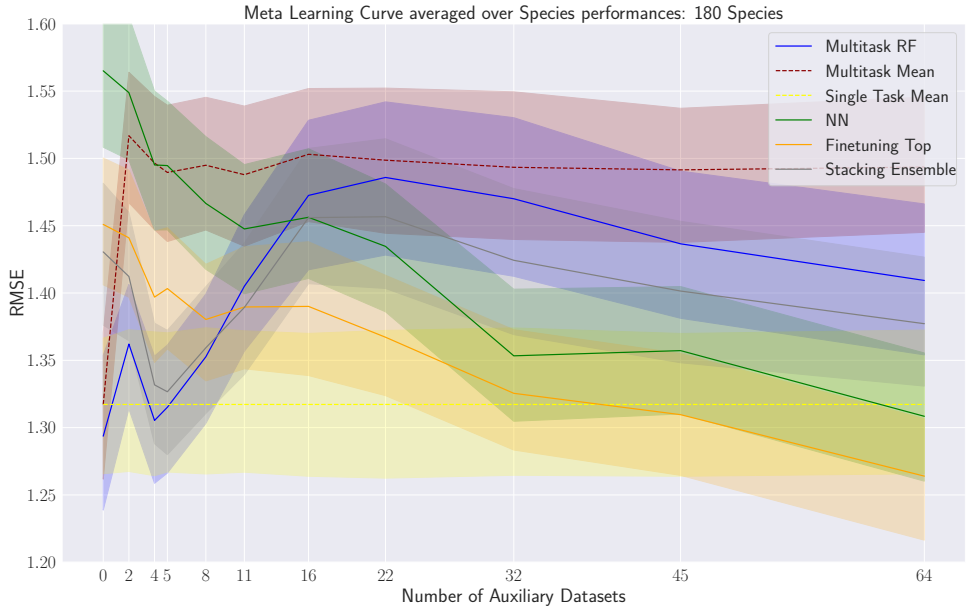
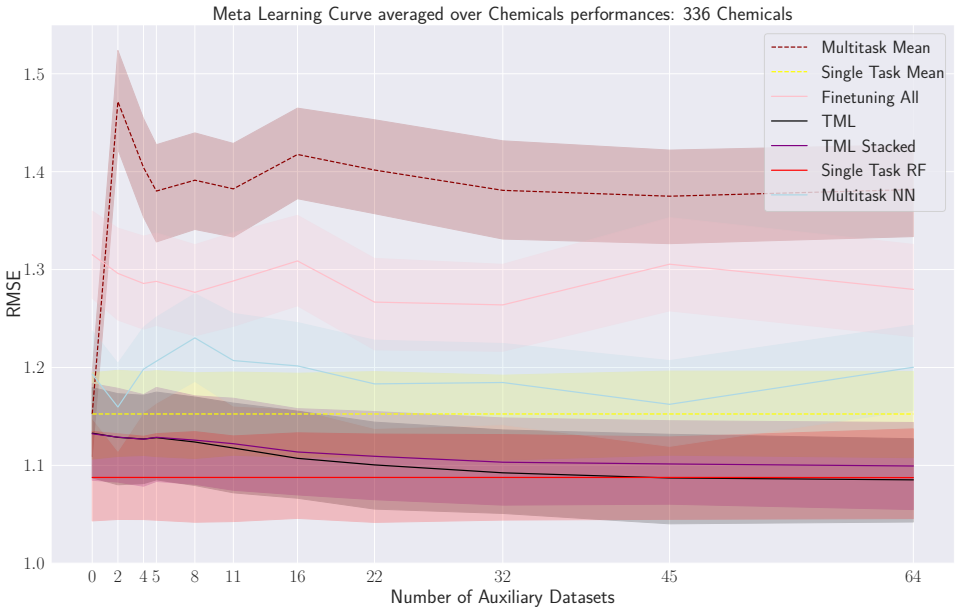


Figure 30: Meta-learning Curves showing the effect of adding more auxiliary species to the training set of a test species:RMSE Performance averaged over species.



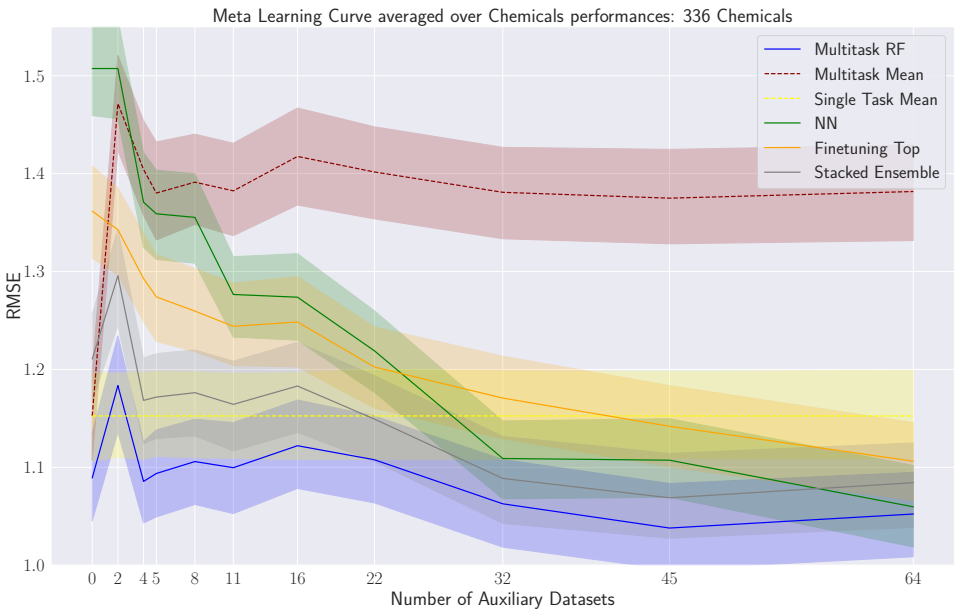


Figure 31: Meta-learning Curves showing the effect of adding more auxiliary species to the training set of a test species: RMSE Performance averaged over chemicals.