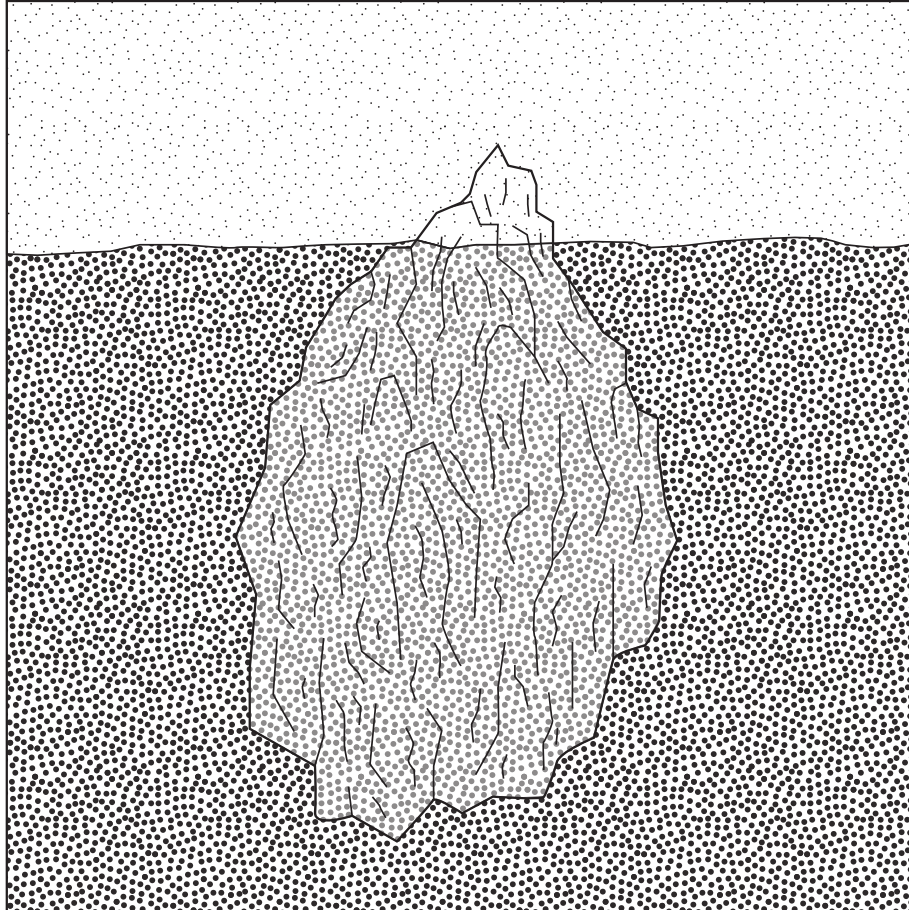




CHALMERS



GÖTEBORGS UNIVERSITET



The tip of the iceberg: using AI to identify toxic chemicals

Analysis of the TRIDENT models and exploration of the chemical space

Master's thesis in Statistical Learning and AI

Elin Edgren

DEPARTMENT OF MATHEMATICAL SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Gothenburg, Sweden 2025

www.chalmers.se

MASTER'S THESIS 2025

**The tip of the iceberg:
using AI to identify toxic chemicals**

Analysis of the TRIDENT models and exploration of the
chemical space

ELIN EDGREN



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2025

The tip of the iceberg: using AI to identify toxic chemicals
Analysis of the TRIDENT models and exploration of the chemical space
Elin Edgren

© Elin Edgren, 2025.

Supervisor: Erik Kristiansson, Department of Mathematical Sciences
Supervisor: Mikael Gustavsson, Department of Mathematical Sciences
Supervisor: Styrbjörn Käll, Department of Mathematical Sciences
Examiner: Marija Cvijovic, Department of Mathematical Sciences

Master's Thesis 2025
Department of Mathematical Sciences
Chalmers University of Technology
University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Iceberg illustration by Johanna Edgren.

Typeset in L^AT_EX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

The tip of the iceberg: using AI to identify toxic chemicals
Analysis of the TRIDENT models and exploration of the chemical space
Elin Edgren
Department of Mathematical Sciences
Chalmers University of Technology
University of Gothenburg

Abstract

This project was conducted to analyze the TRIDENT models described by Gustavsson et al. in the article *Transformers enable accurate prediction of acute and chronic chemical toxicity in aquatic organisms* [1]. The aims were to investigate the predictions made by the models, the relationship between the model's chemical space, and the predictions they make. The methods used for the analyses were a combination of TRIDENT model predictions, modeling, and visualizations. The results of which were that there is a relationship between how accurately the TRIDENT models predict and the closeness the chemical has to the TRIDENT training data as well as the density of close neighbors in the training data. We also found that there are chemicals for which the TRIDENT models predict effective concentration values that are inconsistent with the measured value (label), possibly warranting further investigation of the chemical's toxicity.

Keywords: artificial intelligence, chemical space, embeddings, statistical analysis, toxicity, transformers, TRIDENT.

Acknowledgements

I want to thank my supervisors Erik Kristiansson and Mikael Gustavsson as well as Styrbjörn Käll for their helpful discussions with me about the topic of chemicals and the use of their TRIDENT models. I am very thankful for their insights, help, and kindness.

Elin Edgren, Gothenburg, 01 2025

Contents

List of Acronyms	ix
List of Figures	xi
List of Tables	xv
1 Introduction	1
1.1 Aims	4
2 Literature Study	5
3 Methods and Theory	11
3.1 Data description	11
3.1.1 Dataset 1: Cross-validation dataset	12
3.1.2 Dataset 2: REACH data	13
3.2 Investigation of relationships between training data distance, neighborhood density, and prediction error	13
3.2.1 Heatmap	13
3.2.2 Correlation	15
3.3 Comparison of TRIDENT model-based toxicity classifications with label-based classifications	16
3.4 Model the magnitude of prediction errors	18
3.4.1 SVM	18
3.4.2 kNN	19
3.4.3 Naive kNN	20
3.5 Investigation of TRIDENT predictions on chemicals absent in training data and without labeled values	20
4 Results	23
4.1 Investigation of relationships between training data distance, neighborhood density, and prediction error	23
4.1.1 Heatmap	23
4.1.2 Correlation	25
4.2 Comparison of TRIDENT model-based toxicity classifications with label-based classifications	29
4.2.1 PCA plot analysis	29
4.2.2 PCA inconsistency analysis	31

4.3	Model the magnitude of prediction errors	31
4.3.1	SVM model	32
4.3.2	kNN	32
4.3.3	naive kNN	33
4.4	Investigation of TRIDENT predictions on chemicals absent in training data and without labeled values	34
5	Discussion and Conclusions	37
	Bibliography	41
A	Appendix 1	I

List of Figures

3.1	Illustration for the understanding of Dataset 1, showing the 10 folds different sectioning into training and validation data.	12
4.1	Heatmap illustrating the relationship between cosine similarity in the training data, the abundance of training data within the ranges of closeness, and the average error in prediction for the validation data. The heatmap is an average overall 10-fold training and validation data. The y-axis represents cosine similarity ranges for cosine similarity between the validation embeddings to the training embeddings (averaged across the 10 folds). The x-axis represents neighbor abundance within each cosine similarity range for each chemical (averaged across the 10 folds). The error values are the averaged L1 error for all validation data allocated there (averaged across the 10 folds). The count values is the corresponding number of chemicals that have been used in the mean L1 error calculation for each cell.	24
4.2	Plot showing the Spearman correlation between the cosine similarity of the k-nearest neighbors in the training data for every validation data point and the L1 error of the validation data. The k-values range from 1 to the entire training dataset. The x-axis is the k-value for the number of neighbors within the training data, the y-axis is the Spearman correlation value. Each of the 10 folds of Dataset 1 is illustrated in a unique color. The lines represent the Spearman correlation with the median cosine similarity of the k-nearest neighbors, and the filled-in region is between the 25th and the 75th percentiles of cosine similarity of the k-nearest neighbors in the training data.	26
4.3	Plot showing the Spearman correlation between the cosine similarity of the k-nearest neighbors in the training data for every validation data point and the L1 error of the validation data. The k-values range from 1 to 100 neighbors in the training data. The x-axis is the k-value for the number of neighbors within the training data, the y-axis is the Spearman correlation value. Each of the 10 folds of Dataset 1 is illustrated in a unique color. The lines represent the Spearman correlation with the median cosine similarity of the k-nearest neighbors, and the filled-in region is between the 25th and the 75th percentiles of cosine similarity of the k-nearest neighbors in the training data.	28

4.4 PCA plot for Fold 3 of Dataset 1 illustrating the placement of toxic and uncertain chemicals in the TRIDENT chemical space. Chemicals classified as non-toxic and falling within two standard deviations of the mean cosine similarity of the training data are shown as blue hexagons. The shade of blue indicates the density of chemicals in each area, with deeper shades representing higher densities and lighter shades indicating lower densities. Chemicals classified as non-toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in green. Chemicals classified as toxic within two standard deviations of the mean cosine similarity of the training data are shown in red. Chemicals classified as toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in orange. The PCA plot to the left displays classifications based on predicted values in the dataset. The PCA plot to the right displays classifications based on labeled values in the dataset. 30

4.5 PCA plot of Dataset 2 illustrating the placement of toxic and uncertain chemicals in the TRIDENT chemical space. This PCA plot is for fish with endpoint EC50, the effect mortality, and a duration of 96 hours. Chemicals classified as non-toxic and falling within two standard deviations of the mean cosine similarity of the training data are shown as blue hexagons. The shade of blue indicates the density of chemicals in each area, with deeper shades representing higher densities and lighter shades indicating lower densities. Chemicals classified as non-toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in green. Chemicals classified as toxic within two standard deviations of the mean cosine similarity of the training data are shown in red. Chemicals classified as toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in orange. The PCA plot to the left includes chemicals from the REACH excerpt both in the model training data and new data. The PCA plot to the right only includes chemicals from REACH not in the training data. 35

-
- A.1 PCA plots for Fold 1, 2, and 4 of Dataset 1 illustrating the placement of toxic and uncertain chemicals in the TRIDENT chemical space. Chemicals classified as non-toxic and within two standard deviations of the mean cosine similarity of the training data are shown as blue hexagons. The shade of blue indicates the density of chemicals in each area, with deeper shades representing higher densities and lighter shades indicating lower densities. Chemicals classified as non-toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in green. Chemicals classified as toxic within two standard deviations of the mean cosine similarity of the training data are shown in red. Chemicals classified as toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in orange. The PCA plots to the left shows the classifications of points using the predicted values in the dataset. The PCA plots to the right shows classifications using the labeled values in the dataset. I
- A.2 PCA plots for Folds 5 and 6 of Dataset 1 illustrating the placement of toxic and uncertain chemicals in the TRIDENT chemical space. Chemicals classified as non-toxic and within two standard deviations of the mean cosine similarity of the training data are shown as blue hexagons. The shade of blue indicates the density of chemicals in each area, with deeper shades representing higher densities and lighter shades indicating lower densities. Chemicals classified as non-toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in green. Chemicals classified as toxic within two standard deviations of the mean cosine similarity of the training data are shown in red. Chemicals classified as toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in orange. The PCA plots to the left shows the classifications of points using the predicted values in the dataset. The PCA plots to the right shows classifications using the labeled values in the dataset. II
- A.3 PCA plots for Folds 7 and 8 of Dataset 1 illustrating the placement of toxic and uncertain chemicals in the TRIDENT chemical space. Chemicals classified as non-toxic and within two standard deviations of the mean cosine similarity of the training data are shown as blue hexagons. The shade of blue indicates the density of chemicals in each area, with deeper shades representing higher densities and lighter shades indicating lower densities. Chemicals classified as non-toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in green. Chemicals classified as toxic within two standard deviations of the mean cosine similarity of the training data are shown in red. Chemicals classified as toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in orange. The PCA plots to the left shows the classifications of points using the predicted values in the dataset. The PCA plots to the right shows classifications using the labeled values in the dataset. III

A.4 PCA plots for Fold 9 and 10 of Dataset 1 illustrating the placement of toxic and uncertain chemicals in the TRIDENT chemical space. Chemicals classified as non-toxic and within two standard deviations of the mean cosine similarity of the training data are shown as blue hexagons. The shade of blue indicates the density of chemicals in each area, with deeper shades representing higher densities and lighter shades indicating lower densities. Chemicals classified as non-toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in green. Chemicals classified as toxic within two standard deviations of the mean cosine similarity of the training data are shown in red. Chemicals classified as toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in orange. The PCA plots to the left shows the classifications of points using the predicted values in the dataset. The PCA plots to the right shows classifications using the labeled values in the dataset. IV

A.5 Figure showing the PCA scatterplots for Dataset 2 with endpoint EC50. The plots to the left contain all chemicals in the dataset with predictions for each species group. The plots to the right contain only chemicals that were not used in the training of the model. Chemicals classified as non-toxic and within two standard deviations of the mean cosine similarity of the training data are shown as blue hexagons. The shade of blue indicates the density of chemicals in each area, with deeper shades representing higher densities and lighter shades indicating lower densities. Chemicals classified as non-toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in green. Chemicals classified as toxic within two standard deviations of the mean cosine similarity of the training data are shown in red. Chemicals classified as toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in orange. XIV

A.6 Figure showing the PCA scatterplots for Dataset 2 with endpoint EC10. The plots to the left contain all chemicals in the dataset with predictions for each species group. The plots to the right contain only chemicals that were not used in the training of the model. Chemicals classified as non-toxic and within two standard deviations of the mean cosine similarity of the training data are shown as blue hexagons. The shade of blue indicates the density of chemicals in each area, with deeper shades representing higher densities and lighter shades indicating lower densities. Chemicals classified as non-toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in green. Chemicals classified as toxic within two standard deviations of the mean cosine similarity of the training data are shown in red. Chemicals classified as toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in orange. XV

List of Tables

A.1	Table presenting chemicals located in the region of the PCA plot that is typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classifies them as toxic. Their CAS number and name are displayed for each chemical. (Fold 1.)	V
A.2	Table presenting chemicals located in the region of the PCA plot that is typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classifies them as toxic. Their CAS number and name are displayed for each chemical. (Fold 2.)	V
A.3	Table presenting chemicals located in the region of the PCA plot that is typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classifies them as toxic. Their CAS number and name are displayed for each chemical. (Fold 3.)	VI
A.4	Table presenting chemicals located in the region of the PCA plot that is typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classifies them as toxic. Their CAS number and name are displayed for each chemical. (Fold 4.)	VII
A.5	Table presenting chemicals located in the region of the PCA plot that is typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classifies them as toxic. Their CAS number and name are displayed for each chemical. (Fold 5.)	VIII
A.6	Table presenting chemicals located in the region of the PCA plot that is typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classifies them as toxic. Their CAS number and name are displayed for each chemical. (Fold 6.)	IX
A.7	Table presenting chemicals located in the region of the PCA plot that is typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classifies them as toxic. Their CAS number and name are displayed for each chemical. (Fold 7.)	IX
A.8	Table presenting chemicals located in the region of the PCA plot that is typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classifies them as toxic. Their CAS number and name are displayed for each chemical. (Fold 8.)	X
A.9	Table presenting chemicals located in the region of the PCA plot that is typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classifies them as toxic. Their CAS number and name are displayed for each chemical. (Fold 9.)	X

A.10	Table presenting chemicals located in the region of the PCA plot that is typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classifies them as toxic. Their CAS number and name are displayed for each chemical. (Fold 10.)	XI
A.11	Table presenting chemicals whose labeled values classify them as non-toxic that have a cosine similarity z-score beyond two standard deviations from the training data's mean cosine similarity for all 10 folds. Their CAS number and name are displayed for each chemical.	XII
A.12	Table displaying the validation metrics for big ($L1 \geq 2$) and small $L1 < 2$ errors for the SVM model.	XII
A.13	Table displaying the validation metrics for big ($L1 \geq 2$) and small $L1 < 2$ errors for the kNN model.	XIII
A.14	Table displaying the validation metrics for big ($L1 \geq 2$) and small $L1 < 2$ errors for the naive kNN model.	XIII
A.15	Table presenting the chemicals classified as toxic for the endpoint EC10, not present in the training data for the model. The chemical is classified as toxic for the intersection of invertebrates and algae. For each chemical, their CAS number and name are displayed.	XVI
A.16	Table presenting the chemicals classified as toxic for the endpoint EC50, not present in the training data for the model. The chemical is classified as toxic for all species groups. For each chemical, their CAS number and name are displayed.	XVI
A.17	Table presenting the chemicals classified as toxic for the endpoint EC50, not present in the training data for the model. The chemical is classified as toxic for the intersection of invertebrates and fish. For each chemical, their CAS number and name are displayed.	XVI
A.18	Table presenting the chemicals classified as toxic for the endpoint EC50, not present in the training data for the model. The chemical is classified as toxic for the intersection of algae and fish. For each chemical, their CAS number and name are displayed.	XVII
A.19	Table presenting the chemicals classified as toxic for the endpoint EC50, not present in the training data for the model. The chemical is classified as toxic for the intersection of invertebrates and algae. For each chemical, their CAS number and name are displayed.	XVII

1

Introduction

The use and abundance of chemicals in today's society is significant, but what is known about them can be likened to the tip of an iceberg; there is much to discover beneath the surface of chemical knowledge. A chemical is defined by the International Union of Pure and Applied Chemistry (IUPAC) as a "Matter of constant composition best characterized by the entities (molecules, formula units, atoms) it is composed of." [2]. Chemicals are indispensable in modern society, with applications spanning agriculture, manufacturing, medicine, and much more [3][4][5]. The exponential growth in the chemical synthesis and since the 1800s [6] has brought immense benefits within these applications, benefits such as improved crop yields, innovative materials, and advanced pharmaceuticals [3][4][5].

Despite the undeniable importance of chemicals in modern society, their use is not without risk. History is full of examples where unforeseen and often disastrous outcomes arise from insufficient understanding or inadequate testing of chemical substances. One notable case is Dichlorodiphenyltrichloroethane (DDT), an insecticide used to prevent the spread of malaria [7]. While the chemical is attributed to the saving of millions of lives from malaria, it also accumulates in the environment, most notably resulting in too-thin shells in predatory birds' eggs [7]. The use of DDT caused worldwide population decreases and ecosystem damage because of its widespread use [7]. Another significant example is the use of pesticides and their correlation with the mass death of bees, essential pollinators for ecosystems and agriculture [8][9]. The discovered harmful effect of DDT highlights and the unintended consequences of pesticide exposure on bee populations both illustrate the critical need for the rigorous testing of chemicals before their widespread use.

Since disruptions in one part of an ecosystem can cascade, this lack of knowledge of the chemical's effects has the potential to affect interconnected species and the ecosystem as a whole. Predicting and understanding these intricate interactions remains a challenge. Thus, unforeseen adverse effects from chemical use persist globally, underscoring the urgent need for proactive and predictive approaches to chemical knowledge. The unintended consequences demonstrate how chemical use can disrupt ecological balances, with implications that go far beyond the original applications. Historically, the decision to ban or regulate harmful chemicals has often come only after their adverse effects were realized [10]. This reactive approach leaves room for significant harm, ranging from environmental degradation to public health crises, before action is taken.

It can be assumed that many chemicals are currently impacting ecosystems and human health. Determining the toxicity of a chemical traditionally involves testing on animals and or other organisms to evaluate its potential harmful effects on biological functions [11]. These tests are conducted using various methodologies, such as *in vivo* testing, which examines living organisms in their natural biological environment, and *in vitro* testing, which is performed outside this context, such as in isolated cells or tissues [12]. While advancements in alternative testing methods have reduced reliance on animal testing, *in vivo* experiments remain an essential step in ensuring chemical safety [13]. However, regulatory constraints have significantly limited their use, as alternative methods are becoming a bigger part of toxicity assessment [13][14]. With the sheer number of chemicals in use today there thus exists a vast "gray area" of unknown chemical toxicity, this is in particular since many chemicals remain untested for their effects on organisms and ecosystems since regulations do not require such information for all chemicals in use in the EU [15]. Thus, not all chemicals undergo comprehensive toxicity testing. The exemptions to experimental toxicity testing create gaps in our understanding of the potential toxicity of several chemicals.

Toxicity experiments evaluate specific adverse effects, such as mortality, infertility, or intoxication, focusing on measurable endpoints [1]. One of the metrics for toxicity evaluation is the effective concentration (EC), which represents the concentration of a chemical required to produce a specified effect; an example would be EC₅₀, which signifies the death of 50% of test organisms within a given time frame for an effect [16]. While these metrics provide valuable insights, they are not without limitations. Experiments are subject to errors both in measurement and design, which can affect the reliability of the results, and experimental data is often also hard to get ahold of from regulatory agencies, and a standardized method is not always used when conducting said experiments [1]. Furthermore, the extrapolation of experimental findings to other species or broader ecological systems remains a challenge [11], thus, a chemical deemed non-toxic in one tested species may exhibit harmful effects in others, including humans.

The lack of universality of results leaves room for unforeseen adverse effects as a chemical, tested on a particular organism at specific concentrations, may still cause harm to other species or humans at similar exposure levels [11]. In addition to methodological limitations, toxicity testing is both time-consuming and costly [11]. Therefore, conducting extensive investigations to ensure safety requires substantial financial and time resources. These constraints, combined with the inherent uncertainty in extrapolating results, underscore the complexity of assessing chemical toxicity comprehensively. These challenges highlight the need for alternative approaches that are cost-effective, efficient, and capable of addressing traditional methods' limitations. Predictive models based on statistical and Artificial Intelligence (AI)-driven approaches present a promising pathway to enhance the reliability and scope of chemical toxicity assessments while reducing reliance on animal testing.

In this context, the integration of computational methods and AI offers a promis-

ing avenue. Computational methods like AI have the potential to become integral tools that can help identify patterns and predict toxicity, enabling preventive measures to mitigate harmful outcomes before they occur regarding chemical exposure. Therefore, these tools with their potential to predict chemical toxicity accurately are becoming an important research topic for the protection of public health and the environment. Advances in computational and AI toxicology methods are beginning to play an integral role as these fields offer powerful tools for analyzing vast datasets, modeling chemical behaviors, and predicting potential toxicity with high precision [1]. By using computational methods such as AI and computational methods, we can gain the possibility to address critical gaps in our understanding of chemicals and chemical safety, enabling better regulatory decisions and fostering the development of safer, more sustainable chemicals.

The rapid advancements in AI have significantly expanded its applicability across various fields, including chemistry and toxicology [17]. AI-driven approaches are increasingly being explored to understand the properties of chemicals and their potential impacts on organisms [17]. By leveraging AI, researchers aim to develop cost-efficient methods to predict and investigate the effects of chemicals on the environment without the need for traditional trials on living organisms [1]. These methods, capable of accelerating toxicity assessments, enable faster decision-making before the introduction of chemicals into the environment.

AI techniques have become increasingly prevalent in toxicology modeling, with applications including graph neural networks (GNNs) and quantitative structure-activity relationship (QSAR) models, which have proven effective in predicting toxicity within controlled environments [17]. More recently, transformer-based models have attracted considerable attention for their potential in cheminformatics [18][19][1]. Natural language processing (NLP) techniques, particularly transformers, have shown promise in advancing toxicity prediction by providing more sophisticated representations of chemical properties [1]. The adoption of AI in cheminformatics has grown rapidly due to its potential to overcome the hurdles of traditional experimental approaches. While early models, such as those utilizing QSARs, demonstrated strong predictive performance for specific application domains, recent advancements in transformer-based models have significantly expanded the possibilities for toxicity prediction [1]. Notably, the TRIDENT models exemplify how transformers can be adapted for cheminformatics tasks, further enhancing AI-driven toxicity analysis [1].

This thesis explores the intersection of computational methods like AI and chemical toxicity prediction. We will do this by analyzing a set of AI models of the TRIDENT models, described in the article *Transformers enable accurate prediction of acute and chronic chemical toxicity in aquatic organisms* by Gustavsson et al. [1]. These models, each for a combination of species group, fish, aquatic invertebrates, and algae, and model type, EC10, EC50, or EC50EC10 can, given a chemical structure, predict an effective concentration for a specified effect, endpoint, and exposure duration for three aquatic organism species groups, fish, aquatic invertebrates, and

algae [1]. Here, an effect is either developmental (DVP), growth effect (GRO), intoxication (ITX), mortality (MOR), morphological (MPH), population toxicity (POP), or reproduction effect (REP), with restrictions between the models on which effects are applicable. The endpoints are either EC10 or EC50, where the combined EC50EC10 model can predict both endpoints. The TRIDENT models are composed of two parts. The first is a transformer encoder, which for each SMILES produces an embedding vector, which then is used as input together with the information of effect, toxicity endpoint, and duration into a Deep Neural Network (DNN), which outputs a predicted effective concentration value. The embedding vector for each chemical places the chemical in the chemical room. In this analysis, we will be looking at some of these models, examining predictions on new data, and analyzing aspects of the chemical room regarding prediction accuracy.

1.1 Aims

The aims of the project are thus:

- Investigate relationships between training data distance, neighborhood density, and prediction error.
- Compare TRIDENT model-based toxicity classifications with label-based classifications.
- Model the magnitude of prediction errors.
- Examine TRIDENT predictions on chemicals absent in training data and without labeled values.

The project will begin with a literature study on the use of computational methods to perform large-scale assessments of chemical properties. After which, the methods and theory for the computational tasks will be stated, and the results will follow. The conclusion and discussion of the results will conclude the project.

2

Literature Study

The methodology for the literature study is to use the articles provided by the supervisors and relevant literature found in other databases, accessed through the Chalmers Library and other relevant sites. The search is conducted in such a way as to find appropriate articles regarding the use of computational methods to perform large-scale assessments of chemical properties in ways of cheminformatics and toxicology.

For this literature study, we will explore the use of computational methods to perform large-scale assessments of chemical properties. Despite their benefits, the continuous development and use of new chemicals is and has not been without risks. Unforeseen reactions and interactions can pose serious threats to human health and the environment [20], underscoring the need for comprehensive assessments of chemical properties to ensure their safety and responsible use. Traditional methods for evaluating chemical properties primarily rely on conducting physical experiments to test and measure the properties of interest. Such experiments directly expose the chemical to conditions that reveal its characteristics; for example, organisms may be exposed to the chemical to infer its effects [14]. To understand many toxicological properties, *in vivo* experiments are conducted on organisms to study the toxic effects of chemicals [14]. Typically, researchers use organisms vulnerable to the chemical, such as fish, invertebrates, or algae, species commonly representative of aquatic ecosystems where significant chemical accumulation takes place [21]. While traditional methods provide valuable observations of chemical toxicity, they also come with limitations [11]. Conducting physical experiments is a time-intensive and resource-demanding task, resulting in high costs related to labor and materials [11]. Such constraints have driven the development of computational methods as more efficient alternatives. Computational methods for large-scale chemical property assessments represent an evolving field within cheminformatics, characterized by its interdisciplinary nature [22]. These methods are increasingly valued for their promising ability to quickly and cost-effectively address knowledge gaps across a wide range of chemical properties [23]. As the field advances, new techniques and applications are continually being developed and investigated.

There are many possible applications for large-scale property assessment of chemicals, as they are promising for systematically understanding the complex interactions of chemicals and deriving valuable insights across diverse fields. Computational methods can address many limitations of traditional experimentation by simulations, models, and algorithms to perform large-scale chemical property assessments [12].

These methods offer several key advantages. Computational methods can prove to be a significantly more time- and cost-effective method of chemical property assessment compared to physical experimentation [12]. While traditional methods require extensive labor, materials, and time for setup and repetition, computational techniques can rapidly evaluate numerous chemical properties using pre-existing data and models [17]. Unlike physical experiments, which are often constrained by resource availability, computational methods can scale to assess thousands or even millions of chemicals [17]. This makes them particularly valuable for addressing large chemical libraries or datasets. For toxicological assessments, computational methods can reduce or possibly eliminate the need for *in vivo* experiments, thus minimizing ethical concerns associated with animal testing [13]. Computational models have the potential to simulate a variety of conditions and predict outcomes that may be difficult or impossible to recreate experimentally [12] [17]. They can also provide insights into potential chemical interactions or properties not yet observed experimentally [12][17]. By combining these advantages with ongoing advancements in computational techniques, such methods serve as valuable complements to, and in some cases, replacements for traditional physical experimentation.

One of the fields where computational methods are applicable is drug discovery. In drug discovery, assessing chemical properties is crucial to ensure desired interactions with biological targets, enabling drugs to achieve their intended therapeutic effects [17]. These methods are used to predict key chemical interactions in the body, aiding in identifying potential drug candidates [17]. This way, large-scale assessments can also detect unintended interactions, which may lead to side effects, thus, they also have the potential to improve the safety of drugs.

Computational methods can be used in other fields for discovering chemicals of interest, like in material science. In material science, large-scale assessments of chemical properties are instrumental in designing materials for specific purposes [4]. Generative chemistry, a subset of this field, involves engineering chemicals to achieve desired properties, such as enhanced durability, thermal resistance, or electrical conductivity [4]. For example, biodegradable polymers, synthesized to replace conventional materials, are an endeavor for promoting environmentally friendly solutions [4]. These assessments enable the development of innovative materials tailored to industrial, environmental, and societal needs.

Large-scale chemical property assessments also play a pivotal role in toxicology and in understanding the effects of chemicals on the environment, where understanding how chemicals impact ecosystems and organisms is an important function [24]. Ecological disturbances caused by chemicals can lead to significant environmental consequences. For example, the toxicity of chemicals to aquatic organisms is a common concern in environmental chemistry and toxicology [21]. Toxic effects may range from impaired fertility and reduced population sizes to severe intoxication or mortality [21]. Assessing the properties of chemicals that enter the environment is vital to safeguarding ecosystems, maintaining biodiversity, and ensuring long-term environmental sustainability. Many properties could be of interest depending on the goals of a study. Properties intended for certain uses like in the case of Drug Dis-

covery which involves designing chemicals with specific desired properties, such as target interaction within the body [17], or harmful properties like those relating to adverse effects of chemicals on organisms, ecosystems, or the broader environment [24].

A wide range of computational methods are employed for large-scale chemical property assessment and new techniques continue to develop. These methods leverage chemical information to make predictions, with many focusing on the chemical structure as the primary source of data. Below, we provide an overview of some commonly used approaches.

High-throughput screening (HTS) involves virtual techniques to assess chemical properties. By screening the compounds and simulating virtual experiments with datasets containing chemical information, HTS methods enable the rapid inference of chemical properties [25]. This approach is particularly valuable for screening large libraries of compounds, allowing researchers to prioritize candidates for further investigation [26]. This way, HTS methods find chemicals that have desired characteristics by conducting simulations of a large number of experiments.

One of the most established computational methods is the Quantitative Structure-Activity Relationship (QSAR) with its long time of use to infer chemical properties like toxicity endpoints [17]. QSAR methods are one of the many methods that use the chemical structure to infer and predict various chemical properties [27]. QSAR models are widely used due to their ability to predict activities and properties efficiently based on known structural features. The representations of chemical structures, such as molecular graphs, molecular descriptors (MDs), or chemical fingerprints, establish mathematical relationships between structure and properties [28]. Molecular descriptors, for example, assign numerical values based on the structural characteristics of a chemical and capture relevant structural features, translating them into numerical values for computational analysis [27]. QSAR modeling involves important aspects like model design, applicability domain, and data requirements [23]. QSAR models are constructed using curated datasets that include chemical structures and known properties [28]. Early QSAR models employed simple regression techniques to establish the relationships between structure and properties, such as toxicity [23], but following the advances in machine learning applications have broadened significantly and improved predictive accuracy as well [24].

By the guidelines of the Organization for Economic Co-operation and Development (OECD), QSAR models are limited by their applicability domains, which define the chemical space where predictions are reliable [12]. It is essential to ensure that predictions remain within this domain to maintain model validity [27]. Curated data is also crucial for building accurate QSAR models, but this is also a requirement that can limit the scope of predictions for the QSAR models [1]. Despite the many achievements using QSARs, these computational methods face challenges and limitations of their use, such as their need for a design with specific application domains in mind, creating models with narrow windows of use [29]. While this domain-focused approach improves prediction accuracy, it also limits the model's generalizability to

broader chemical spaces [29], resulting in QSAR models having narrow applicability. There is thus also a customization requirement as developing new models or retraining existing ones is necessary for different domains, which increases time and resource demands [29].

Machine learning (ML) and artificial intelligence (AI) techniques have become increasingly popular for large-scale chemical property assessments [17]. ML methods have become pivotal in large-scale computational chemical property assessment as they can process big datasets [17]. With these large datasets, ML methods train models capable of making predictions and uncovering patterns in complex chemical data [17]. Neural networks are a key category of ML algorithms, with Deep Neural Networks (DNNs) being particularly well-suited for handling complex and high-dimensional data [17]. DNNs, with their use of multiple layers of interconnected neurons, are increasingly popular for modeling intricate relationships in data using large datasets [17]. The adaptability and predictive power of AI-driven models make them well-suited for handling the diverse and intricate nature of chemical data and information [17].

Despite their many advantages, ML and AI computational methods face challenges and limitations as well that must be addressed for broader adoption and improved accuracy. One of the primary challenges in developing AI models for chemical property assessment is the scarcity of high-quality chemical data, but also the need for expertise from different disciplines within which the methods are deployed, as well as the "black-box" nature of AI-based methods also present as challenges [17]. Issues regarding data arise from various factors, including data confidentiality, where political and proprietary concerns often restrict access to detailed chemical datasets [24]. The great need for data for AI methods poses challenges of resource constraints since generating and curating large, high-quality datasets is time-consuming and expensive [23]. There could also be data errors since existing datasets frequently contain inaccuracies, inconsistencies, or missing values, adversely affecting model training and performance [23]. AI models require substantial amounts of reliable data to learn complex relationships, making the lack of comprehensive datasets a significant barrier to progress [23]. The 'black-box' nature of AI makes it difficult to understand how models generate predictions, leading to a lack of interpretability that poses several challenges. One significant issue is the uncertainty in predictions. Without clear insight into the decision-making process, confidence in the results diminishes. As a result, even when predictions are accurate, their perceived reliability is lower compared to well-understood methods [17].

Recently, transformers have become an increasingly popular way to extract information for many different purposes, they use self-supervised learning where attention and several layers can extract key information [18]. These transformer encoders can be used to extract chemical information from molecular representations, like SMILES, further improving the chemical information for computational model input. Such chemical information representation is, for example, used as input for a DNN in the TRIDENT models, as described by Gustavsson et al. [1]. The

TRIDENT models showed strong predictive performance regarding accuracy and robustness when compared to traditional QSAR models [1]. Given the successful use of transformers for chemical information extraction in TRIDENT models, it is reasonable to expect that future research to advance computational methods for assessing chemical properties will further explore transformer-based approaches.

3

Methods and Theory

The methodology and the accompanying theory for the different parts of the report are described in the following sections. The analyses will be performed using Python version 3.9.15.

In this project, we would like to explore the TRIDENT models' performance regarding aspects of its training data and examine "suspect chemicals" based on the models' predictions. Since this project is framed around the TRIDENT models, we will use Simplified Molecular-Input Line-Entry System (SMILES) as the chemical representation. The TRIDENT models are Deep Neural Network (DNN) models that use transformer encoding for the extraction of chemical information from the input SMILES. A SMILES is a 2-dimensional representation of a chemical structure and is one of the most widely used chemical representations [30]. SMILES are a linear notation of the atoms and connections between the atoms and other important features of the chemical derived from its molecular graph [30]. The order for the generalization of the 2-dimensional representation is not set, so there can be many different SMILES representations of the same chemical, but for use, standardization is often needed [30]. One such standardization can be done with, for example, RDKit's Canonicalization of SMILES, which is used in the implementation of the TRIDENT models. Transformer-based models include a special classification token (CLS) that is added to the beginning of the input sequence. During training, the model learns to produce a hidden state that captures a contextual representation of the input. For tasks such as molecular property prediction, this resulting CLS embedding vector serves as a condensed representation of the molecule [31][1]. The use of transformers, therefore, creates an embedding for each chemical, positioning it in the chemical space. This embedding vector can then be used to infer information about the chemical, which, in the case of the TRIDENT models, is toxicity. Exploration of this chemical space is useful for understanding the TRIDENT models further.

3.1 Data description

For the analysis of TRIDENT, two different datasets will be used.

3.1.1 Dataset 1: Cross-validation dataset

The first dataset for the analysis of the TRIDENT models (Dataset 1) consists of 10 train and test splits of a dataset containing a total of 3542 chemicals from the 10-fold cross-validation training of the TRIDENT model for endpoint EC50 and species group fish. The features of importance in the dataset for the analyses are predicted values from the TRIDENT model during the cross-validation training, labeled values from the physical experiments in the original database, and the embeddings that the TRIDENT model assigns to each of the chemicals. The motivation behind using this dataset for analysis of the TRIDENT models is to gain an understanding of how the models perform on validation data given the training data while still having access to the labels. As there are no chemicals with measured effective concentrations (labels) that can be used in our analysis of the final TRIDENT models that were not in the training of the models.

In Dataset 1, we denote the training set $D_{train}^F = c_{i=1}^{tN}$ and the validation set $D_{val}^F = c_{i=1}^{vM}$ for folds $F = 1, \dots, 10$. For a visual representation of Dataset 1, see Figure 3.1.

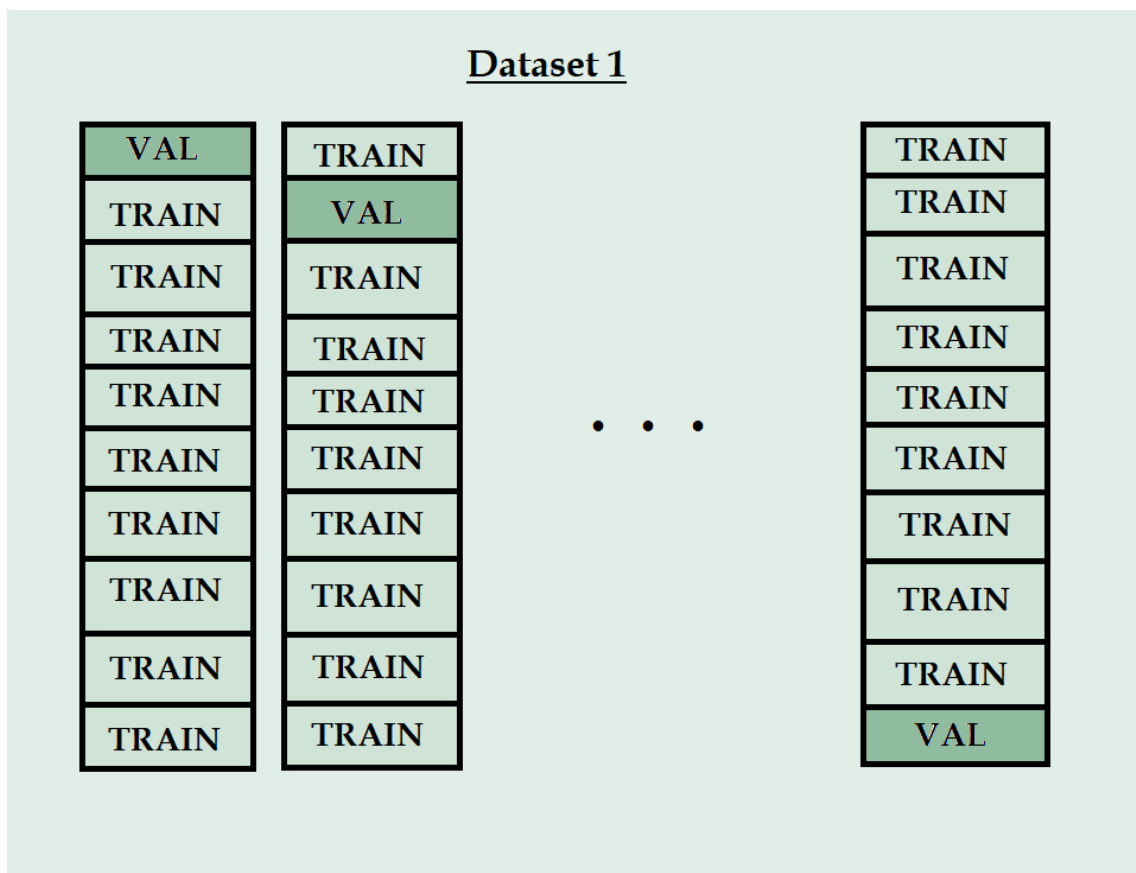


Figure 3.1: Illustration for the understanding of Dataset 1, showing the 10 folds different sectioning into training and validation data.

3.1.2 Dataset 2: REACH data

The second dataset, Dataset 2, consists of an excerpt from the REACH database from 22/05/24. The excerpt includes both chemicals on which the TRIDENT model has been trained and new chemicals on which the model has not been trained and for which there are no measured values (labels), with a total number of chemicals of 10209 with different CAS numbers. The number of chemicals with different canonical RDKit SMILES is 10014, which will be the number of chemicals used in the analysis. This dataset is interesting because it allows us to examine the new chemicals with no measured value and the predictions the model makes for them to see if any warrants further investigation.

3.2 Investigation of relationships between training data distance, neighborhood density, and prediction error

To investigate the relationship between distance to the training data, density of the neighborhood in the training data, and prediction error made by the model, we will illustrate their connection in two ways: with a heatmap and a correlation plot.

3.2.1 Heatmap

In this analysis, we aim to explore and analyze the relationship between distance to the training data, density of the neighborhood in the training data, and prediction error made by the TRIDENT model using a heatmap. As the TRIDENT models use transformers, each output gets a representation in the vector space of the domain, the representing vector is the CLS embedding of that input in the domain space [32]. We will, for this analysis, use Dataset 1 to have both the predicted and the labeled values for each chemical in the data to illustrate this relationship between closeness, density, and prediction error.

To measure the closeness of chemicals in the chemical space, we look at the embedding space of the TRIDENT model and measure closeness through a distance measure in vector space. For the heatmap, the choice of thresholds for the partitions of the heatmap is made by using the distribution of the distance measure for the embeddings so that partitions can capture important information in the values of the distance measure.

For the prediction errors in this analysis, we will use the absolute value of the residuals (L1). The L1 error is defined in Equation 3.1.

$$e(x) = | f(x) - y | \quad (3.1)$$

The closeness measure chosen for this analysis is cosine similarity. Cosine similarity is defined in Equation 3.2.

$$\text{cosine similarity}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{|\mathbf{u}||\mathbf{v}|} \quad (3.2)$$

The mean cosine similarity for each validation chemical’s embedding to the embeddings of the training set D_{train} is calculated and added as a new column in the validation set D_{val} . The cosine similarity indicates how closely a validation set chemical c_i^v , $i \in M$ aligns with the training set chemicals D_{train}^F in the chemical space [33]. Cosine similarity values range between -1 and 1, where a value of 1 indicates large similarity and a value of -1 indicates low similarity. For each validation chemical, c_i^v , the cosine similarity with every training chemical is computed, resulting in a matrix of size $|c_i^v| \times |D_{train}^F|$. The mean of these similarity values is then stored in the new column for each validation set point. This process quantifies the average closeness of a validation chemical to the training set’s chemicals in the embedding space.

For the partition boundaries of the heatmap to visualize the information contained in the cosine similarity, cosine similarity thresholds are calculated. To determine the thresholds for the heatmap in cosine similarity values, the distribution of cosine similarity values in the validation data is used. Since the cosine similarity values range between -1 and 1, thresholds are chosen to partition this range based on the percentiles of the cosine similarity values in the validation dataset. For each validation fold, percentiles of the cosine similarity values are computed within the range [-1, 1] using the desired number of thresholds, which in our case is chosen to be 10. The boundary thresholds are explicitly set to -1.0 and 1.0 to ensure the full range of cosine values is captured. Thresholds from all 10 validation folds are then aggregated into a single matrix, and the mean thresholds across folds are computed to create a consistent set of thresholds for all 10 folds in the heatmap.

To collect the values for the heatmap, two dictionaries, one for error collection and one for neighbor count collection, are used when iterating through each chemical in the validation set. For each validation chemical, its cosine similarity with all chemicals in the training set is calculated. These similarity values are compared against the previously calculated cosine similarity thresholds, and for each range defined by two consecutive thresholds, the number of training set embeddings that fall within the cosine similarity range is counted and stored in the neighbor count dictionary. The L1 error value for the current validation chemical is stored in the error dictionary along with the neighbor count for the same range. To avoid errors, if no training embeddings fall within a range, a None value is stored in the error collection dictionary, and a count of 0 is stored in the neighbor count collection. This ensures that for each validation chemical, the error and neighbor counts are collected across all similarity ranges, enabling the visualization of the relationship between the L1 error and neighbor density in the heatmap.

For the partition boundaries of the heatmap for the visualization of the information contained in the neighbor density, neighbor count thresholds are calculated. To determine the neighbor count thresholds for the heatmap, the neighbor counts collected during the previous step are processed for all 10 folds of the dataset. To

derive the thresholds, the percentiles are calculated so that for each cosine similarity range and each fold, the percentiles of the neighbor counts are computed. These percentiles divide the distribution of neighbor counts into intervals, capturing the abundance of neighbors. To derive the final thresholds, the neighbor count thresholds obtained from all folds are averaged for each percentile index. Ensuring that the final thresholds capture the overall distribution of the neighbor counts across the dataset. These averaged thresholds are the partition boundaries used in the heatmap to analyze the distribution of neighbor counts within each cosine similarity range. This way, the thresholds summarize the abundance of neighbors in the training data, allowing for a visualization of how neighbor density affects the L1 error.

To generate the final dataframe for the heatmap, the information from all folds is processed to generate a 10×10 dataframe. The data is firstly prepared so that for each fold, a dataframe containing the validation chemical indices, the cosine similarity thresholds, the L1 error values, and the neighbor counts is created. If any invalid rows occur here, they are filtered out. Each validation chemical and cosine similarity range combination is then assigned a neighbor count bin based on its value and the percentile thresholds derived earlier. These bins represent different ranges of neighbor abundance. The mean L1 error for validation chemicals within each combination of cosine threshold range and neighbor count bin is calculated. This creates a table for each of the 10 folds, with cosine thresholds as rows and neighbor count bins as columns. For the combination of the fold data into a single dataframe, the tables from all folds are averaged to produce a single dataframe representing the mean L1 error across folds for each combination of cosine similarity and neighbor count bin.

For the creation of the final heatmap, a seaborn heatmap is generated using the averaged pivot table, with cosine similarity partitions on the y-axis and neighbor count partitions on the x-axis. Annotations are added to the heatmap to show both the mean error value and the corresponding number of chemicals that have been used in that mean error for each cell. If there are empty cells they will be left unannotated. The resulting heatmap provides a visualization of how prediction error varies with neighbor abundance in the training set across cosine similarity ranges. This allows for analysis of the relationship between model performance, chemical similarity to the training set, and data density within the similarity range of the training set.

3.2.2 Correlation

In this analysis, we aim to explore and analyze the relationship between the similarity to neighbors in the training data and the prediction error made by the TRIDENT model using a correlation plot. This analysis explores the relationship between the similarity of validation embeddings to the k-nearest neighbors' training embeddings and the prediction errors for the validation data, using Dataset 1 to have both the predicted and the labeled values for each chemical in the data under the assumption that the cross-validation embeddings and predictions behave the same way as those

of the TRIDENT models.

The first step involves computing the L1 error for the validation datasets, the L1 error calculation is as shown in Equation 3.1. As we want to investigate the proximity of k-nearest neighbors in the training set for each chemical in the validation set, we measure the proximity in the embedding space. For every fold and every validation embedding within that fold’s validation set, the k-nearest neighbors are determined by sorting the training chemicals in descending order of cosine similarity and selecting the top k chemicals. We do this for a range of k values. We then compute the cosine similarity between each of the embeddings of the validation set to the k-nearest embeddings of the training set and calculate the median, 25th, and 75th percentiles of the similarity score for the k-nearest neighbors for each validation chemical for every k-value in the range of k-values. The correlation between the L1 error of each validation chemical and the cosine similarity statistic of its k-nearest neighbors (median, 25th, and 75th percentiles) is calculated using the Spearman correlation. The Spearman correlation is a rank-order correlation measure, measuring the strength and direction of a monotonic relationship between two variables [34]. We thus get a Spearman correlation value for every combination of fold, neighbor size, and median, 25th, or 75th percentile calculation. By varying k and looking at the correlation between the similarity scores and the prediction error for the validation chemicals, we aim to capture the relationship between local similarity structures in the embedding space and predictive error.

For each fold, we compute the Spearman correlation of L1 error with the median similarity of k-nearest neighbors and the Spearman correlation of L1 error with the 25th and 75th percentiles of similarity. This is then illustrated together in one single plot. The area between the 25th and 75th percentiles represents the uncertainty in the correlation value. The final plot combines results from all folds, with each fold represented by a distinct color, illustrating trends and variability in the relationship across folds.

Firstly, we do this correlation plot for k-values ranging from 1 to the size of the entire training set. The same procedure is then repeated with a smaller range of k-values $k \in [1, 100]$ to illustrate the relationship between prediction error and the proximity of k-nearest neighbors in more local neighborhoods of the training set.

3.3 Comparison of TRIDENT model-based toxicity classifications with label-based classifications

In this analysis, we want to investigate the classifications of toxicity that can be made from the TRIDENT model predictions and compare them with the classifications using the labeled values. In this analysis, we will use Dataset 1 to have both the predicted and the labeled values for each chemical in the data under the assumption that the cross-validation embeddings and predictions behave the same way as those of the TRIDENT models. For this analysis we will utilize Principal Com-

ponent Analysis (PCA) plots to visually inspect the embeddings' distribution and the alignment of labeled and predicted toxicity values. This analysis aims to help evaluate the model's performance in generalizing and distinguishing toxic chemicals. This analysis also aims to identify inconsistencies in the dataset based on label-to-projection mismatches and significant deviations in embeddings to highlight edge cases or potentially mislabeled data points that warrant further investigation.

PCA is a dimension reduction technique that will be used to reduce the dimensionality of the 768-dimensional CLS embedding vectors into two principal components that capture the greatest variation in the data. These components are used to create scatterplots of projected chemicals that visualize the distribution of embeddings in the chemical space for the validation datasets. The L1 errors are computed as described in Equation 3.1 for both the training set and the validation set to assess prediction accuracy. For each embedding in the training data, the cosine similarity within the training data is calculated and averaged, creating a mean cosine similarity within the training data; a standard deviation of similarity within the training set is also calculated.

The mean and standard deviation within the training data are then used to measure the closeness deviation from the training data. For each embedding in the validation data, the cosine similarity to the embeddings of the training data is calculated and averaged. The averaged similarities for each embedding of the validation data are standardized into z -scores to be able to identify "uncertain" embeddings, in particular, those significantly deviating from the mean, that is, $|z - score| > 2$. For the classification of a chemical as toxic, we use the threshold according to the Swedish Chemical Agency standards, where for aquatic organisms, an effective concentration of less than 0.1 mg/l is toxic for the endpoint EC50 in the CLP Regulation [35].

Two scatterplots are generated for each of the 10 validation folds in the dataset. The first scatterplot uses the concentration values predicted from the model during the cross-validation. The points are categorized according to the prediction value concerning the toxicity threshold and its embedding similarity z -score. The points are colored according to their combination of the two classifications. A point that is non-toxic and within two standard deviations of the mean cosine similarity of the training data (blue), toxic and within two standard deviations of the mean cosine similarity of the training data (red), non-toxic and beyond two standard deviations of the mean cosine similarity of the training data (green), or toxic and beyond two standard deviations of the mean cosine similarity of the training data (orange). This way, the plot shows the model's ability to separate toxic from non-toxic chemicals using its predictions. The second scatterplot instead uses the labeled concentration values, and as is done in the first scatterplot, the coloring is based on the two different classifications for each chemical in the validation set.

Following the PCA plots, there is an additional analysis. In this analysis, we identify two categories of "suspect" data points in a validation dataset based on their embeddings and associated labels. The first category is toxic-labeled chemicals located in

regions of the PCA plot that are typically occupied by non-toxic chemicals, a PC1 value lower than zero, possibly indicating a mismatch between the label and the PCA projection for the chemical. The second category is chemicals labeled as non-toxic that have a cosine similarity z-score beyond two standard deviations from the training data’s mean cosine similarity, suggesting an embedding-based uncertainty or deviation.

For the first category, chemicals labeled as toxic but whose location in the PCA plot indicates possible misclassification, the criteria are that the chemical is labeled as toxic and that it falls within the "typically" non-toxic region of $PC1 < 0$. These chemicals are then stored together. For the second category, chemicals labeled as non-toxic but whose embedding cosine similarity z-score is beyond two standard deviations of the mean, these points are collected and stored together as well. This way, "suspect chemicals" are identified and stored in dictionaries for inspection.

3.4 Model the magnitude of prediction errors

In this analysis, we want to model the error in prediction from the TRIDENT models. To investigate the potential of using embeddings and their closeness to the training data (and other features) to predict the magnitude of prediction errors, L1, for new, unobserved embeddings. Since we do not have access to new data containing labels, the analysis is performed using Dataset 1 to have both the predicted and the labeled values for each chemical in the data under the assumption that errors in the cross-validation behave in the same way as the TRIDENT models. The prediction error L1 is calculated according to Equation 3.1 and added to the datasets. A measure of closeness between embeddings in the validation and training sets is computed using cosine similarity and added to the validation set as a feature.

The validation data from all 10 validation folds are combined into a single dataset, which is then divided into training and testing subsets. The features of interest now include embeddings and a closeness measure (mean cosine similarity to the training set). Three regression models, a Support Vector Machine (SVM), a k-nearest neighbors (kNN), and a naive kNN, are trained and evaluated on their ability to predict the magnitude of the prediction error. We want to, using these different models, model the response $e(x) \approx g(x; \theta)$, where g represents the regression model of choice [32].

3.4.1 SVM

This analysis method uses the embeddings, their distances to the training data, and their neighborhood density as features. In this study, we will investigate if we can, with tuned parameters, model prediction errors using an SVM model based on these features.

SVM regression uses an ϵ -insensitive loss function, which ignores deviations smaller than ϵ from the predicted value, making the method robust to outliers [36]. The

type of SVM model we will use in this analysis is a Support Vector Regression (SVR) model. In this case of using an SVR model, we model the response $e(x) \approx g(x; \theta)$ where g represents an SVR model parameterized by θ . In this analysis, we will use the SVR model from SciKit-learn with the parameters kernel type (e.g., RBF or polynomial), the margin of tolerance term epsilon, and the regularization term C [36].

The embeddings are first normalized using a standard scaler. Similarly, the "distance to training" feature is scaled and combined with the embeddings to form the feature matrix X . The target variable is the L1 prediction error. To find optimal parameters, a grid search is conducted using the GridSearchCV function from SciKit-learn using 10-fold cross-validation with negative mean squared error as the evaluation metric. During the 10-fold cross-validation training, Pearson and Spearman correlations are calculated between the true labels and the predicted values. The best parameter combination is selected for the final model. The performance of the final model is evaluated on an independent test set that is split into large error values $L1 \geq 2$ and small error values $L1 < 2$. The model performance is evaluated using Pearson and Spearman correlation metrics and the Mean Squared Error (MSE) on both test sets.

3.4.2 kNN

The goal of the analysis for this method is to model the prediction error $e(x)$ for new chemical embeddings as a function of their embeddings and closeness to the training set, represented mathematically as $e(x) \approx g(x; \theta)$, where g is the k-Nearest Neighbor (kNN) model and θ are the model parameters. By using embeddings and their proximity to the training data, the kNN model aims to provide accurate predictions of error magnitudes for the data contained in the validation set.

The kNN regression model is used to predict the error $e(x)$ as a function of the embeddings and closeness metrics. The embeddings are first normalized using a standard scaler. Similarly, the "distance to training" feature is scaled and combined with the embeddings to form the feature matrix. The target variable is the L1 prediction error from the prediction and label values in the dataset. The kNN model uses the KNeighborsRegressor function from SciKit-learn with parameter weights set to distance and metric to cosine. The model thus assigns error values based on the weighted values of the k-nearest neighbors of a test point, using cosine similarity as the distance metric. To determine the optimal number of neighbors (k), a grid search is conducted. The k -values are evaluated by calculating the MSE across cross-validation folds. The k -value with the lowest average MSE is selected as the optimal parameter. The chosen kNN model is trained on the entire training dataset and evaluated on an independent test set that is split into large error values $L1 \geq 2$ and small error values $L1 < 2$. The metrics training and validation MSE, Pearson, and Spearman correlations are computed to assess model performance.

3.4.3 Naive kNN

For this third analysis method, a naive kNN model, we again aim to predict the magnitude of the L1 prediction error $e(x)$ for unseen embeddings in a chemical dataset. By using a combination of embeddings and a closeness metric to the training data, a naive kNN model is fit and evaluated on its potential to predict TRIDENT prediction error magnitude.

In this analysis, the embeddings are first normalized using a standard scaler. Similarly, the "distance to training" feature is scaled and combined with the embeddings to form the feature matrix X . The target variable is the L1 prediction error. We model $e(x) \approx g(x; \theta)$, where g represents the naive kNN model. The naive kNN model uses the cosine distance metric to find the k nearest neighbors of a test point in the training data. The prediction is then the mean L1 error of these neighbors. A grid search is performed over a range of k -values to identify the optimal value that minimizes the validation MSE. The process involves splitting the dataset into training and testing subsets, applying the naive kNN model, and evaluating metrics, including MSE, Spearman correlation, and Pearson correlation. The grid search identifies the optimal k -value based on the lowest validation MSE. The final optimal naive kNN model is then evaluated on an independent test set that is split into large error values $L1 \geq 2$ and small error values $L1 < 2$.

3.5 Investigation of TRIDENT predictions on chemicals absent in training data and without labeled values

In this analysis, we want to investigate the TRIDENT model's predictions on data that was not included in the model training and does not have labeled effective concentration values. For this analysis, we use Dataset 2. We will look at interesting placements and predicted values using the TRIDENT model on this new data, which the model has not trained on and whose results might spark interest in further investigating certain chemicals with interesting results. We begin the analysis by predicting effective concentration values for all chemicals in Dataset 2. We do this for the model with the best predictive performance of the TRIDENT models, that is, the combined EC50EC10 model for each of the species groups fish, invertebrates, and algae [1]. We will look at a duration of 96 hours for all species groups in this analysis. We will examine mortality as the effect on fish and aquatic invertebrates and population toxicity for algae. We will look at predictions for both endpoints EC50 and EC10.

For measuring the distance to the training data, we compute the mean cosine similarity within the training data as well as the standard deviation for each species group. We then compute the z-score for each chemical in Dataset 2 to see if the chemical deviates from the training data in the embedding space. Then, using a toxicity threshold, we classify chemicals whose effective concentration prediction value crosses that threshold. Using the threshold for toxicity for aquatic organ-

isms according to the Swedish Chemical Agency standards for EC50, an effective concentration of less than 0.1 mg/l is toxic and for EC10 a concentration of less than 0.01 mg/l [35]. We then look at PCA plots of the dataset embeddings in the chemical space created by TRIDENT projected onto a two-dimensional plot. The chemicals will be colored according to their classification of being toxic and within two standard deviations of cosine similarity closeness to the training data (red) or non-toxic and two standard deviations of cosine similarity closeness to the training data (blue) at the specific concentration and duration, as well as if they are classified as toxic and beyond two standard deviations of the mean closeness within the training data to the training data (orange), or classified as non-toxic and beyond two standard deviations of the mean closeness within the training data to the training data (green). From toxicity classification results, a list of interesting chemicals for further investigation can be gathered, which will both look at the chemicals for each species group and the intersection of them. The names of the chemicals were retrieved using their CAS number and the *get_compound* function from pubchempy.

The code for the thesis can be found at the GitHub account:
<https://github.com/EdgrenElin>

4

Results

4.1 Investigation of relationships between training data distance, neighborhood density, and prediction error

For the investigation of the relationship between distance to the training data (cosine similarity), density of the neighborhood in the training data (neighbor counts), and prediction error made by the model, we used two different methods: a heatmap and a correlation plot. Here are the resulting plots and results of this analysis.

4.1.1 Heatmap

From the TRIDENT models, each chemical gets a representation in the embedding vector space, this vector is a representation of that chemical in the chemical space. This analysis aimed to analyze the relationship between the closeness of new chemicals to the training chemicals, the density of the chemical space in different closeness ranges to the new chemical, and the prediction error of the new chemical. For this purpose, a heatmap illustrating this relationship was created. Dataset 1 was used to illustrate this relationship between closeness, density, and prediction error.

The resulting heatmap provide a visualization of how prediction error varies with neighbor abundance in the training set across cosine similarity ranges. This allows for the analysis of the relationship between model performance, chemical similarity to the training set, and data density within the similarity range of the training set. When looking for general trends in the resulting heatmap, see Figure 4.1, moving up the y-axis (decreasing cosine similarity thresholds), you have averages for training embeddings further and further away, similarity wise to each given embedding in the validation set. Here, we see that the average prediction errors are quite similar across the cosine similarity ranges, except for the range furthest away and the closest range. When we instead move across the x-axis, when moving towards increased training embedding density (towards the right), we see that the prediction errors generally are lower for density ranges 1-4, indicating that the new embeddings with moderately many training embeddings within each cosine range have lower prediction error.

The highest errors are concentrated in the top-right corner. This suggests that having a large number of training embeddings with very low cosine similarity to the

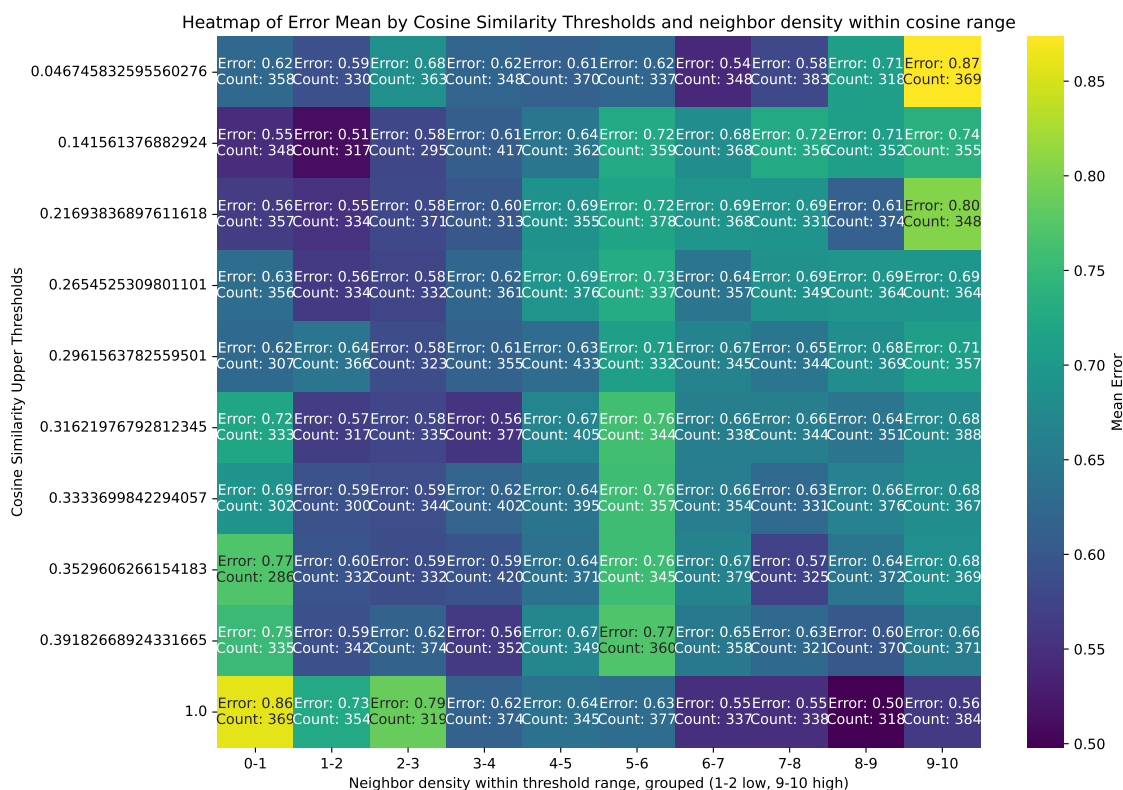


Figure 4.1: Heatmap illustrating the relationship between cosine similarity in the training data, the abundance of training data within the ranges of closeness, and the average error in prediction for the validation data. The heatmap is an average overall 10-fold training and validation data. The y-axis represents cosine similarity ranges for cosine similarity between the validation embeddings to the training embeddings (averaged across the 10 folds). The x-axis represents neighbor abundance within each cosine similarity range for each chemical (averaged across the 10 folds). The error values are the averaged L1 error for all validation data allocated there (averaged across the 10 folds). The count values is the corresponding number of chemicals that have been used in the mean L1 error calculation for each cell.

new chemical generally comes with high prediction error. So, being far away from densely populated areas is correlated with high error. High errors are concentrated in the lower-left corner as well. This suggests that having very few training embeddings that have high cosine similarity is also correlated with high prediction error. As for regions of low error, we have the bottom-right corner. This region shows some of the lowest error values, suggesting that having a large amount of training embeddings within a high cosine similarity range corresponds with better predictive accuracy. At intermediate cosine thresholds, error levels stabilize across all neighbor count bins. This suggests that by having training embeddings at moderate similarity levels, the predictive model performs consistently, regardless of neighbor abundance in the training data. The heatmap suggests an optimal zone for prediction is to focus on moderately many training embeddings falling within a high cosine similarity to the new embedding, that is, cosine thresholds > 0.3 . These conditions generally yield low L1 errors. There is a trade-off between using stricter similarity thresholds (for closer neighbors) and considering more neighbors. The heatmap highlights the importance of balancing these factors to minimize error. The dataset seems to have sufficient density in bins with high neighbor counts, as errors generally decrease when more neighbors are considered. However, sparse regions with few neighbors struggle to provide reliable predictions, suggesting that data sparsity is a critical limitation in those areas.

4.1.2 Correlation

This analysis explores the relationship between prediction errors and the closeness of k -nearest neighbors in training data embedding space, using Dataset 1. The closeness of k -nearest neighbors in the training data was calculated using cosine similarity for each chemical in the validation set. From the labels and predicted values in the validation sets, the L1 errors were calculated. To investigate the relationship between the L1 prediction errors of the validation sets and the similarity score, their correlation was calculated using the Spearman correlation. This procedure was firstly done for k -values ranging from 1 to the size of the entire training set and then repeated for k -values ranging from 1 to 100. This way, we capture the relationship between both overall and local similarity structures in the embedding space and predictive error. For each of the 10 folds, the Spearman correlation of the L1 error and the median, 25th percentile, and 75th percentile cosine similarity of k -nearest neighbors are calculated and plotted together with the area between the 25th and 75th percentile filled in with color.

Looking at the resulting plot for the full range of k -values, see Figure 4.2. When looking at general trends, we see that for most folds, the Spearman correlation between cosine similarity of k -nearest neighbors and L1 error tends to remain negative, particularly for smaller values of k . This indicates that high similarity among relatively few nearest neighbors (in terms of cosine similarity of embeddings) is generally closely associated with lower prediction errors. With increasing k (more neighbors are included), the Spearman correlations across most folds show a slight upward trend approaching zero correlation. This suggests that when considering a broader

4. Results

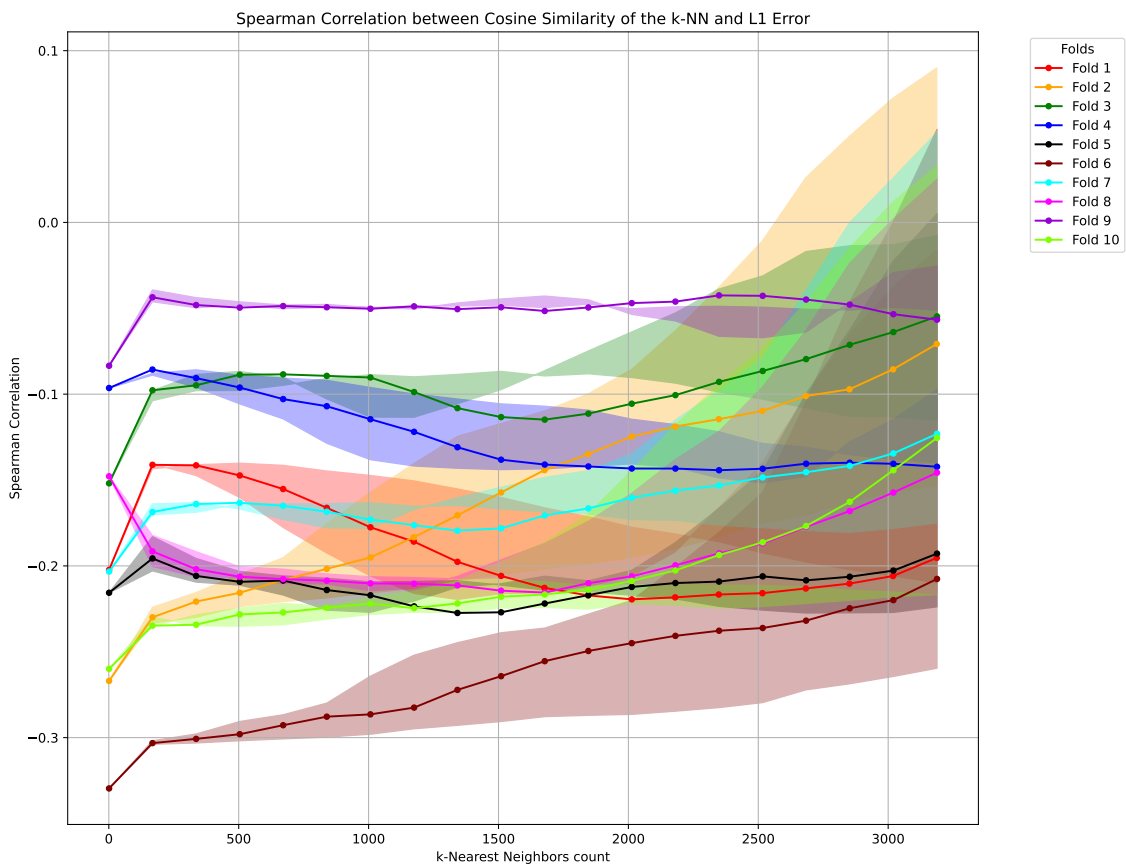


Figure 4.2: Plot showing the Spearman correlation between the cosine similarity of the k-nearest neighbors in the training data for every validation data point and the L1 error of the validation data. The k-values range from 1 to the entire training dataset. The x-axis is the k-value for the number of neighbors within the training data, the y-axis is the Spearman correlation value. Each of the 10 folds of Dataset 1 is illustrated in a unique color. The lines represent the Spearman correlation with the median cosine similarity of the k-nearest neighbors, and the filled-in region is between the 25th and the 75th percentiles of cosine similarity of the k-nearest neighbors in the training data.

neighborhood, the predictive power of the neighbors diminishes, and the relationship between similarity and L1 error becomes weaker.

From the plot, we can see that there is variability between folds, both regarding the magnitude of the correlation and the trend with increasing k-value. Fold 6 shows strong negative correlations even as the k-value rises, and both Fold 4 and 9 show weak correlations even for low k-values. The filled regions, that is, between the 25th and 75th percentiles, indicate the uncertainty and variability within each fold, and as there is an overlap between these filled regions between the folds, this suggests that there is some consistency in the correlation across folds. The width of the filled regions tends to widen as k increases, suggesting that the variability in correlation values across samples within a fold increases when more neighbors are included. In other words, bigger neighborhoods introduce more variability in the strength of

the correlation. This correlation plot thus indicates that for prediction, relying on smaller neighborhoods (lower k -values) might provide more meaningful insights into the relationship between neighbor similarity in the training set and the prediction error. The variability between folds warrants further investigation into why some folds show stronger correlations while others do not, as this might relate to how the data was initially split.

Looking instead at the correlation plot with the smaller range of k -values, see Figure 4.3, we see that similar to the larger k -range plot, folds show a negative Spearman correlation between cosine similarity of k -nearest neighbors and L1 error. This highlights further that smaller neighborhoods with high similarity are associated with lower predictive errors. For many folds, the Spearman correlation stabilizes or plateaus as k increases toward 100, suggesting that as the neighborhood size grows, the additional influence of more neighbors diminishes.

Just as with the plot for the larger range of k -values, we see that there are some differences between the folds. Some of the folds, like Fold 9 and Fold 4, have relatively weak negative correlations, indicating a weaker relationship between neighbor similarity and error for small k . Whereas other folds, like Fold 6, exhibit stronger negative correlations throughout the range of k -values. These are similar results to what we saw in the larger k -range plot. Unlike the larger k -range plot, the fold curves are more clearly separated here, suggesting more distinct fold-specific behavior for smaller neighborhoods since there is little overlap in the correlation.

The stronger negative correlations at small k values suggest that the nearest neighbors (top 1–10 neighbors) are more predictive of error trends. These small neighborhoods likely capture local structure in the embedding space that directly influences model performance. Beyond $k = 50$, the correlations tend to stabilize or show weaker changes, indicating diminishing returns in using larger neighborhoods for the error analysis. The differences between folds in both correlation magnitude and trend for smaller k -values suggest that the relationship between embeddings, similarity, and prediction error is influenced by the specific fold’s data.

Since relatively small k -values, around $k \leq 50$, exhibit relatively strong correlations and low variability in the correlation between the 25th and 75th percentiles, this range could be the most relevant for uses aiming to use local embedding structures of the training set. This small k -value range plot provides a more detailed view of correlation trends for small neighborhoods, revealing nuances that might be hidden in the broader k -range plot.

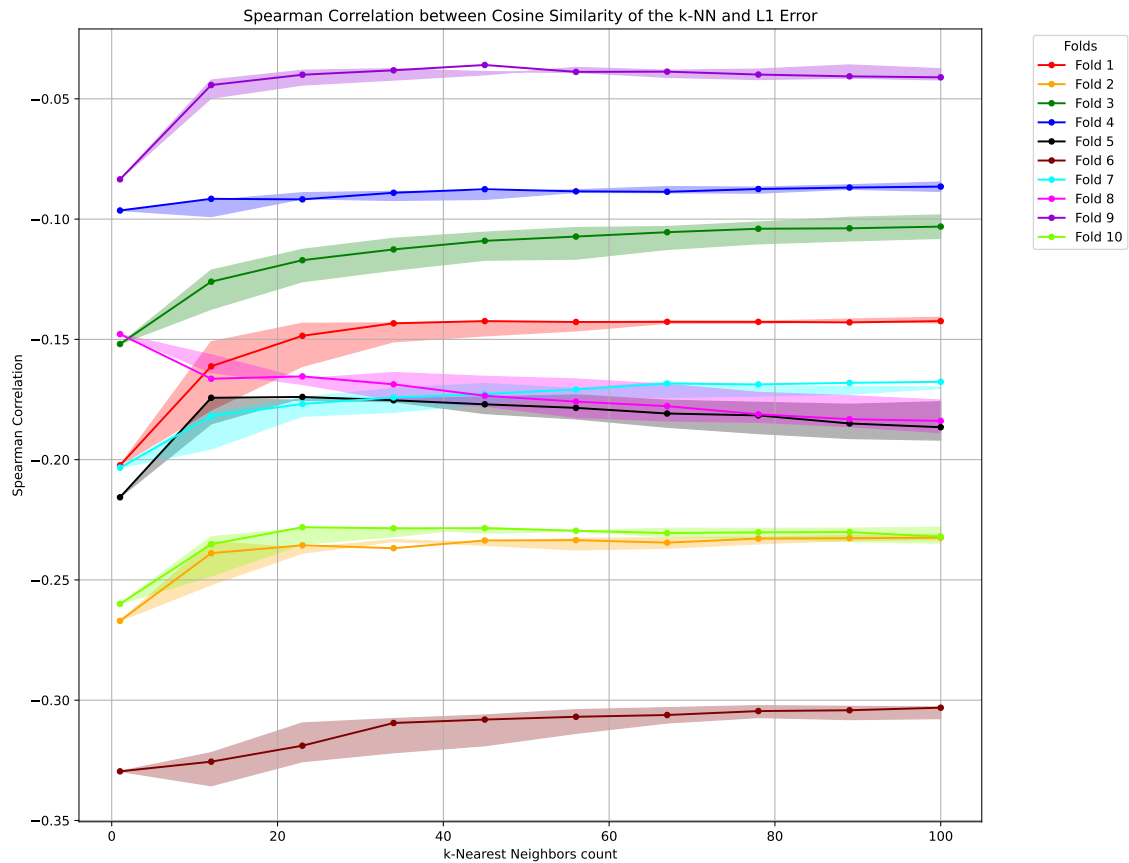


Figure 4.3: Plot showing the Spearman correlation between the cosine similarity of the k-nearest neighbors in the training data for every validation data point and the L1 error of the validation data. The k-values range from 1 to 100 neighbors in the training data. The x-axis is the k-value for the number of neighbors within the training data, the y-axis is the Spearman correlation value. Each of the 10 folds of Dataset 1 is illustrated in a unique color. The lines represent the Spearman correlation with the median cosine similarity of the k-nearest neighbors, and the filled-in region is between the 25th and the 75th percentiles of cosine similarity of the k-nearest neighbors in the training data.

4.2 Comparison of TRIDENT model-based toxicity classifications with label-based classifications

4.2.1 PCA plot analysis

In this analysis, we analyzed Dataset 1 and utilized PCA plots to visually inspect the embeddings' distribution and the alignment of labeled and predicted toxicity values. This analysis aimed to help evaluate the model's performance in generalizing and distinguishing toxic chemicals effectively.

Two scatterplots were generated for each of the 10 validation folds, the first where the effective concentration value used is the predicted value from the trained model. The points are categorized as non-toxic and within two standard deviations of the mean cosine similarity of the training data (blue), toxic and within two standard deviations of the mean cosine similarity of the training data (red), non-toxic and beyond two standard deviations of the mean cosine similarity of the training data (green), or toxic and beyond two standard deviations of the mean cosine similarity of the training data (orange). The second scatterplot instead uses the labeled effective concentration values.

We illustrate the results showing the PCA plot for one of the folds, Fold 3. The PCA plots for the other folds are presented in Figure A.1, A.2, A.3, and A.4 in the Appendix. In the PCA plot to the left, we have values predicted by the model, see Figure 4.4, the blue hexagons representing the location and density of non-toxic chemicals within two standard deviations of mean cosine similarity to the training data. These points are densely clustered in one region of the plot, indicating that there is consistency in predictions for these chemicals. The green dots represent non-toxic chemicals that are beyond two standard deviations of mean cosine similarity to the average similarity of the training data. We see that a small number are scattered, suggesting that the model predicts some chemicals as non-toxic but whose embeddings deviate from the training data distribution. The red dots represent toxic chemicals that are within two standard deviations of the mean cosine similarity to the training data. The red dots are concentrated in a distinct region, indicating that the model can position toxic chemicals in the chemical space well. The orange dots represent toxic chemicals beyond two standard deviations of the mean cosine similarity to the training data. The orange dots indicate chemicals identified as toxic with embeddings deviating significantly from the mean training embeddings, we see that they are also placed well regarding model toxicity prediction.

Similar patterns are observed for the plot to the right, using the labeled values as with the plot with the predicted values. The non-toxic chemicals are clustered in one area, while toxic chemicals mostly occupy another. The alignment between chemicals classified as toxic based on their label and the predicted counterparts in the "tail" of the PCA plot, together with the alignment with the large amount of

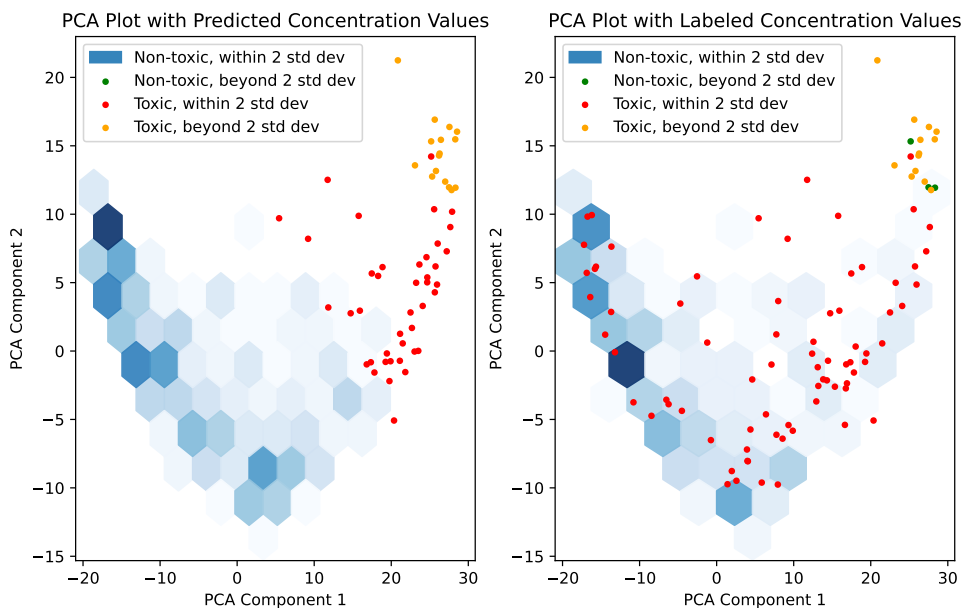


Figure 4.4: PCA plot for Fold 3 of Dataset 1 illustrating the placement of toxic and uncertain chemicals in the TRIDENT chemical space. Chemicals classified as non-toxic and falling within two standard deviations of the mean cosine similarity of the training data are shown as blue hexagons. The shade of blue indicates the density of chemicals in each area, with deeper shades representing higher densities and lighter shades indicating lower densities. Chemicals classified as non-toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in green. Chemicals classified as toxic within two standard deviations of the mean cosine similarity of the training data are shown in red. Chemicals classified as toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in orange. The PCA plot to the left displays classifications based on predicted values in the dataset. The PCA plot to the right displays classifications based on labeled values in the dataset.

chemicals shown in blue in the plots, suggests that the model is performing well in identifying toxic chemicals.

Both plots show some separation between non-toxic and toxic chemicals in the PCA-reduced space, indicating that the model embeddings capture toxicity-related features. The points that are uncertain, beyond two standard deviations, that we see can warrant further investigation, as they may be edge cases or chemicals that are difficult to classify with confidence. Since the number of chemicals in the PCA plot is around 350, where most of them are hidden in the blue hexbin visualization, then there are relatively few "uncertain" chemicals visualized in the plot. The relatively few uncertain points suggest that most chemicals fall within the expected distribution of embeddings. The points that were classified as toxic by just one of the labels or prediction values are also of interest for further investigation.

4.2.2 PCA inconsistency analysis

This analysis aimed to identify inconsistencies in the dataset based on label-to-projection mismatches and deviations in embeddings to highlight edge cases or potentially mislabeled data points that warrant further investigation.

We identified two categories of "suspect" data points in each validation dataset based on their embeddings and associated labels. The first category, chemicals that are located in regions of the PCA plot that are typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classify them as toxic, possibly indicating a mismatch between the label and the PCA projection for the chemical. These chemical names and CAS numbers are stored for each fold and displayed in Table A.1, A.2, A.3, A.4, A.5, A.6, A.7, A.8, A.9, and A.10 respectively in Appendix. The second category of "suspect chemicals" is chemicals whose labeled values classify them as non-toxic that have a cosine similarity z-score beyond two standard deviations from the training data's mean cosine similarity, suggesting an embedding-based uncertainty or deviation. These chemical names and CAS numbers for all 10 validation data folds are displayed in Table A.11

4.3 Model the magnitude of prediction errors

This analysis investigated the potential of using embeddings and their closeness to the training data (and other features) to predict the magnitude of prediction errors, L1, for new, unobserved embeddings. The analysis was performed using Dataset 1. The validation data from all 10 validation folds were combined into a single dataset, which was then divided into training and testing subsets. The features include embeddings and a closeness measure (mean cosine similarity to the training set). Three regression models, a Support Vector Machine (SVM), a k-nearest neighbor (kNN), and a naive kNN, were trained and evaluated on their ability to predict the magnitude of the prediction error.

4.3.1 SVM model

For the SVM model we get from the grid search the optimal parameter values. In the optimal parameters, the regularization parameter $C=0.2$ indicates that the optimal model minimizes overfitting by not allowing large coefficients in the model. The parameter for the margin of tolerance where no penalty is given in the training loss function for errors is $\text{epsilon}=0.4$. This large epsilon value allows the model to capture broader trends while ignoring small deviations. The parameter for kernel type, $\text{kernel}=rbf$, indicates that an RBF kernel, which is effective for capturing complex nonlinear patterns in the data, is optimal for this model. These optimal parameter values indicate that there are non-linear relationships to capture in the data but that regularization is needed for better performance.

During the model training, the correlation between the predicted values and the true labels was calculated to assess model performance during training. For the Pearson correlation, the correlation values on the training data showed a high value (around 0.6); this indicates that the model is capturing linear relationships between predicted and true values on the training data during model training. On the validation data, the correlation values are much lower (around 0.2); this suggests that the model struggles to generalize to unseen data in terms of linear correlation. When we instead look at the Spearman correlation, we see similar results. On the training data, we see a high correlation in the relative ranking of predicted and true values, and on the validation data, we see a lower ranking correlation. These results indicate that the model is having trouble generalizing and making correct predictions for new data, and the high correlation values for training data with low correlation values for validation data indicate that the model is overfitting.

The validation MSE values on the small ($L1 < 2$) and big error ($L1 \geq 2$) data can be seen in Table A.12 in the Appendix. Because the value for small errors is relatively low, this suggests the model is not entirely failing with its error predictions. However, since we have low correlation metrics, this indicates that the errors might not align systematically with the true values, even for the small errors. When we instead look at the results for the larger errors, we see that the MSE value is large. this, together with the correlation values for the larger errors being next to zero, we find that the model is not able to capture larger errors with its predictions.

4.3.2 kNN

For the kNN model, from the grid search, the optimal k-value is identified as $k = 160$; this value yields the lowest MSE on the validation set within the training folds. This suggests that using a relatively large neighborhood, and therefore smoothing out prediction errors, reduces errors compared to smaller k-values.

During the model training, the validation MSE stabilizes as k increases beyond $k = 160$, indicating that as k-values go larger, there is no improvement. The relatively low average validation Pearson correlation and Spearman correlation suggest the kNN model is struggling to capture any strong linear or rank-order relationships

between the predicted and actual errors. While the MSE is low, the correlation metrics indicate that the model might not be capturing the true structure for the errors.

During model training, the MSE on the training data and the MSE on the validation data are close, indicating that the model generalizes well and is not overfitting to the training data during training.

The final validation MSE-value for small errors (computed on the evaluation set) being consistent with the training fold MSE-value indicates that the embeddings and closeness metrics are informative features for predicting the error for small errors. These results illustrate that combining embeddings with the closeness metric helps predict the magnitude of prediction errors when errors are small. The low MSE, especially with a relatively large k , suggests that the features provide meaningful information about the underlying error distribution. Despite the low MSE, the relatively weak Pearson and Spearman correlations indicate that while the model predicts error magnitudes with reasonable accuracy, it may not always rank errors correctly or reflect the precise relationship between the input features and error magnitudes, even for small errors. This could indicate that the kNN model may not fully capture possible more complex non-linear patterns in the data or that the input features used (embeddings and closeness to the training data) may not be sufficient to explain all the variance in the error. When looking at the final validation MSE for big errors, we see a much larger value in comparison; see Table A.13 in the Appendix. We also see that for big errors, the correlations between labels and predictions are slightly larger than for small errors, indicating that the model was able to better capture the relationship between the input features and the error and rank them correctly, despite the large MSE.

4.3.3 naive kNN

For the naive kNN model, the grid search identifies $k = 90$ as the optimal value as it has the lowest validation MSE during training. This indicates that for the given dataset and its features, considering the 90 nearest neighbors provides the most accurate predictions of the L1 error on unseen data. During training, as k increases, the training MSE rises, and the validation MSE decreases initially, to then stabilize around $k = 90$, to then increase slightly for larger values of k . This illustrates the balance between overfitting (low k) and underfitting (high k) in the model during training. The validation correlation improves as k increases and peaks at $k = 90$. Beyond $k = 90$, the validation correlation starts to decrease, indicating diminishing predictive capability of higher k -values. The training correlation decreases as k increases, indicating that there is reduced sensitivity to the training data as their numbers increase. The k -value of 90 minimizes validation error while displaying a balance between training and validation performance and indicates that predictions derived from a fairly broad neighborhood of data points perform well, indicating a need for smoothing out the noise in predictions.

When looking at the MSE values for the small errors, see Table A.14, we see that

its value is quite small, indicating that the model is able to predict model error magnitude well for small errors. When looking at the correlation values for small errors we see that they are quite small, both in the Pearson correlation and the Spearman correlation, this indicates that while the model is performing well in regard to its predictions it is not able to capture the relationship between the input embeddings and features and the corresponding error or rank the errors correctly. When we instead look at the MSE values for the big errors, see Table A.14, we see that this value is large in comparison to the small error evaluation set’s MSE value. When looking at a comparison of the correlation values for the big and the small errors, we see that those for the big errors are a little larger than those for the smaller errors, both with Pearson correlation and Spearman correlation. This result indicates that the model is better at capturing the relationship between the input features and the error and ranking them correctly, despite the large MSE.

4.4 Investigation of TRIDENT predictions on chemicals absent in training data and without labeled values

In this analysis, we looked at the excerpt from the REACH database, that is, Dataset 2. We looked at interesting placements and values using the TRIDENT models on new data that the model was not trained on and whose results might spark interest in further investigation of certain chemicals with interesting results. We look at PCA plots of the new chemicals in the chemicals space created by TRIDENT projected onto a two-dimensional plot. The chemicals are colored according to their classification of being toxic or non-toxic at the specific endpoint and duration, as well as if they are beyond two standard deviations of the mean closeness within the training data to the training data (since how close they also affect the confidence with which we can say the prediction is a certain way). From the results, a list of interesting chemicals for further investigation is gathered, and the intersection of toxic chemicals between the species groups is gathered.

In Figure 4.5, we see one of the PCA plots of the dataset projected from the chemical space onto the first two principal components for endpoint EC50. We can see that these chemicals are placed in the region where toxic chemicals are normally located and that the pattern displayed is consistent with previous plots when the labels are known. The PCA plots for the two other species groups and the other endpoint are available in the Appendix; see Figure A.6 and A.5.

To look at the intersection of the chemicals classified as toxic between the species groups, tables were produced. For endpoint EC10 there were no chemicals that were classified as toxic in the intersection of all species groups, as well as the intersections of invertebrates and fish and algae and fish, that were not in the training data. The intersection for invertebrates and algae is shown in Table A.15 in the Appendix. For the EC50 endpoint, the intersection of toxic chemicals for all species groups is shown in Table A.16 in Appendix, the intersection of invertebrates and algae is shown in Table A.19 in Appendix, the intersection of invertebrates and fish is shown in Table

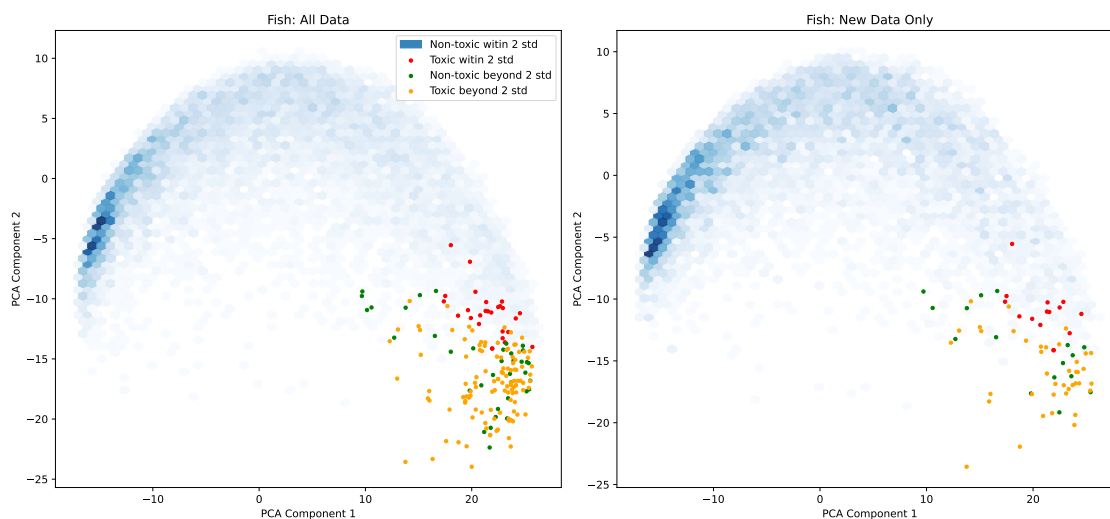


Figure 4.5: PCA plot of Dataset 2 illustrating the placement of toxic and uncertain chemicals in the TRIDENT chemical space. This PCA plot is for fish with endpoint EC50, the effect mortality, and a duration of 96 hours. Chemicals classified as non-toxic and falling within two standard deviations of the mean cosine similarity of the training data are shown as blue hexagons. The shade of blue indicates the density of chemicals in each area, with deeper shades representing higher densities and lighter shades indicating lower densities. Chemicals classified as non-toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in green. Chemicals classified as toxic within two standard deviations of the mean cosine similarity of the training data are shown in red. Chemicals classified as toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in orange. The PCA plot to the left includes chemicals from the REACH excerpt both in the model training data and new data. The PCA plot to the right only includes chemicals from REACH not in the training data.

4. Results

A.17 in Appendix, and the intersection of algae and fish are shown in Table A.18 in Appendix.

From the intersections of chemicals classified as being toxic for the different species groups and endpoints, we can see that there is some overlap. Of certain interest is the chemicals that is classified as toxic for all species groups, which for EC50 Dibutoxydibutylstannane. There is, thus, an indication that this chemical could be toxic to a variety of organisms.

5

Discussion and Conclusions

To analyze the TRIDENT models' predictions and chemical space, a couple of different analyses were performed.

To investigate the relationship between distance to the training data, the density of the neighborhood in the training data, and prediction error made by the model, we used two different methods, producing a heatmap and a correlation plot. From the heatmap, we see that there are regions of TRIDENTs' chemical space that are associated with higher prediction errors and other regions where the prediction error is lower. The results of the heatmap indicate that there is a relationship between a chemical's closeness to the training data and the prediction errors and that this also is influenced by the density of the training data. We also saw some indication that there are areas in the embedding space where sparsity seemingly affects the model's ability to accurately make predictions despite the presence of training embeddings. From the correlation plot, we see that the prediction errors are most correlated with small local neighborhoods in all of the datafolds, about $k \leq 50$. Thus, to gain insight into the prediction error of a new chemical, the most insight is given by the prediction error of the closest chemicals in the training data in the embedding space. Thus, to have the most descriptive neighborhood size for prediction error magnitude, moderately small neighborhoods are preferable. We know from these analyses that there is uncertainty to the predictions, and we can see that there is a relationship between that uncertainty and the closeness to the training data as well as the local density of the training data. Thus, from these distance and density analyses, we gain the knowledge that from where the chemical is positioned in the chemical space of TRIDENT, we can gain some understanding of its prediction error.

For the investigation of the classifications of toxic that can be derived from the TRIDENT model predictions in comparison with the labeled values, we used PCA plots together with the threshold value for aquatic toxicity provided by the Swedish Chemical Agency. From the PCA plots, we see that there is a consensus in the classification of toxicity for the majority of chemicals in the data. There is, however, a subset of chemicals whose predicted value diverges from the label. With this disagreement, there is uncertainty, as both model prediction and measurement errors are possible explanations. Therefore, further investigation of these "suspect chemicals" would be valuable for a greater understanding of their toxicity.

For modeling the error in prediction from the TRIDENT model, we made three different models and evaluated their performance. For the prediction models, we

observed that all of them perform better at predicting smaller errors and struggle with larger ones. We also observed that all of the models have difficulty in capturing the relationship between the features and the magnitude of the prediction error, suggesting that it is quite hard to capture and describe the L1 errors of the TRIDENT model using any of the three models. Possible reasons for this could be that the errors depend on something unrelated to either the chemical structure or the average closeness to embeddings in the training set. Another possibility is that another model than any of the ones tested in this analysis may be needed to capture the true relationship.

For the investigation of the TRIDENT model's predictions on new data not included in the model training, we have produced PCA plots and gathered chemicals with interesting predictions. From the PCA plots, we see that there is consensus in the placement of toxic chemicals when comparing the plot with only new data to the plot with training data included, both with the endpoint EC50 and EC10. From the results of the REACH analysis, we also find that several chemicals are being flagged as toxic to aquatic organisms when looking at each species group individually. Out of these chemicals, those that indicate toxicity for all of our species groups are of the most interest. For those chemicals that have no measured value, only one is indicated as toxic for all groups with the endpoint EC10, that is silver; methanesulfonate. This result could motivate further investigation of this chemical and its effect on aquatic life. When looking at the endpoint EC50, we see that chemicals that are indicated to be toxic are expanded to now also include dibutoxy(dibutyl)stannane as toxic to all species groups.

These different analyses verify the strengths of the TRIDENT models with their predictions on toxicity and highlight some possible ways to improve them further. According to the analysis performed, filling out the chemical room would improve the model predictions, but identifying such chemicals to increase the density in sparse areas of the chemical space could prove difficult.

As there was no data available with labels that were not used in the training of the TRIDENT models, this study used the 10-fold cross-validation data from the fish EC50 model. This way, both predicted values and measured values are available together with a validation set for each of the 10 folds. We assumed that a similar relationship is present in the final TRIDENT model as is displayed during the cross-validation. The data availability for data containing labels was thus limited.

The split of the folds could be a limitation, as it may be biased, given the differences in fold results. This study mostly focuses on fish mortality with EC50, this was also a limitation of the study as other effects, endpoints, and species groups are of interest to study as well.

For future work, it would be interesting to further investigate the performance of the TRIDENT models' predictions on new data that also have measured values to see how the actual TRIDENT models predict. It would be interesting to see if similar results are achieved using the final trained TRIDENT model on new data as is achieved with Dataset 1.

It would also be of interest to see further investigation of "suspect chemicals" as new

measured values or new predictions.

Further investigation of the other endpoints and effects would also be of interest as this study mainly focused on a single species, endpoint, and effect combination.

There are also other chemical databases than REACH whose chemical data would be of interest to investigate using TRIDENT predictions.

When it comes to future work, it could also be of interest to study whether there is an out-of-distribution problem with these models. As the models are now, the assumption is that they can make valid predictions for all chemicals. It could be of interest to investigate the possibility that there are out-of-distribution chemicals for which the model should not be assumed to make accurate predictions. There is still so much more to uncover, both about chemicals currently in use and about future chemicals. With the AI methods currently being developed, the aim of a greater understanding of the chemical space is taking leaps forward, but the majority of the chemical iceberg remains to be uncovered.

Bibliography

- [1] Mikael Gustavsson, Styrbjörn Käll, Patrik Svedberg, Juan S Inda-Diaz, Sverker Molander, Jessica Coria, Thomas Backhaus, and Erik Kristiansson. Transformers enable accurate prediction of acute and chronic chemical toxicity in aquatic organisms. *Science Advances*, 10(10), Mar 2024.
- [2] International Union of Pure and Applied Chemistry. chemical substance. IUPAC Compendium of Chemical Terminology, 3rd ed., 2006. Online version 3.0.1, 2019.
- [3] Muyesaier Tudi, Huada Daniel Ruan, Li Wang, Jia Lyu, Ross Sadler, Des Connell, Cordia Chu, and Dung T. Phung. Agriculture development, pesticide application and its impact on the environment. *International Journal of Environmental Research and Public Health*, 18(3):1112, 2021.
- [4] R. Song, M. Murphy, C. Li, K. Ting, C. Soo, and Z. Zheng. Current development of biodegradable polymeric materials for biomedical applications. *Drug Design, Development and Therapy*, 12:3117–3145, September 2018.
- [5] Scandinavian Simvastatin Survival Study Group. Randomised trial of cholesterol lowering in 4444 patients with coronary heart disease: the scandinavian simvastatin survival study (4s). *The Lancet*, 344:1383–1389, 1994.
- [6] Eugenio J. Llanos, Wilmer Leal, Duc H. Luu, Jürgen Jost, Peter F. Stadler, and Guillermo Restrepo. Exploration of the chemical space and its three historical regimes. *Proceedings of the National Academy of Sciences*, 116(26):12660–12665, Jun 2019.
- [7] Vladimir S Turusov, Vladimir N Rakitsky, and Luigi Tomatis. Dichlorodiphenyltrichloroethane (dDT): ubiquity, persistence, and risks. *Environmental Health Perspectives*, 110(2):125–128, 2002.
- [8] Ben A Woodcock, Nick J B Isaac, James M Bullock, David B Roy, David G Garthwaite, Andrew Crowe, and Richard F Pywell. Impacts of neonicotinoid use on long-term population changes in wild bees in england. *Nature Communications*, 7:12459, 2016.
- [9] Simon G Potts, Jacobus C Biesmeijer, Claire Kremen, Peter Neumann, Oliver Schweiger, and William E Kunin. Global pollinator declines: trends, impacts and drivers. *Trends in Ecology & Evolution*, 25(6):345–353, 2010.
- [10] Sheldon Krinsky. The unsteady state and inertia of chemical regulation under the us toxic substances control act. *PLOS Biology*, 15(12):e2002404, 2017.
- [11] Aysha Akhtar. The flaws and human harms of animal experimentation. *Cambridge Quarterly of Healthcare Ethics*, 24(4):407–419, 2015.
- [12] Thomas Luechtefeld and Thomas Hartung. Computational approaches to chemical hazard assessment. *ALTEX*, 34(4):459–478, 2017.

- [13] Kevin A. Ford. Refinement, reduction, and replacement of animal toxicity tests by computational methods. *ILAR Journal*, 57(2):226–233, dec 2016.
- [14] European Chemicals Agency. Animal testing under reach, 2025. Accessed: 2025-01-16.
- [15] European Chemicals Agency (ECHA). Registration information requirements under reach, n.d. Accessed: 2025-01-21.
- [16] Xiaoqi Jiang and Annette Kopp-Schneider. Summarizing ec50 estimates from multiple dose-response experiments: A comparison of a meta-analysis strategy to a mixed-effects model approach. *Biometrical Journal*, 56(3):493–512, 2014.
- [17] Nicole Kleinstreuer and Thomas Hartung. Artificial intelligence (ai)—it’s the end of the tox as we know it (and i feel fine)*. *Archives of Toxicology*, 98(3):735–754, Jan 2024.
- [18] Walid Ahmad, Elana Simon, Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta-2: Towards chemical foundation models, 2022.
- [19] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: Large-scale self-supervised pretraining for molecular property prediction. *arXiv:2010.09885 [physics, q-bio]*, Oct 2020.
- [20] European Chemicals Agency. Understanding reach, n.d. Accessed: 2025-01-14.
- [21] J. B. Pritchard. Aquatic toxicology: Past, present, and prospects. *Environmental Health Perspectives*, 100:249–257, April 1993.
- [22] John B. O. Mitchell. Machine learning methods in chemoinformatics. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 4(5):468–481, September 2014.
- [23] Hao Zhu. Big data and artificial intelligence modeling for drug discovery. *Annual Review of Pharmacology and Toxicology*, 60:573–589, 2020.
- [24] Kerstin von Borries, Hanna Holmquist, Marissa Kosnik, Katie V. Beckwith, Olivier Jolliet, Jonathan M. Goodman, and Peter Fantke. Potential for machine learning to address data gaps in human toxicity and ecotoxicity characterization. *Environmental Science & Technology*, 57(46):18259–18270, Nov 2023.
- [25] Terry W. Schultz, Robert Diderich, Chanita D. Kuseva, and Ovanes G. Mekenyan. The oecd qsar toolbox starts its second decade. In Orazio Nicolotti, editor, *Computational Toxicology: Methods and Protocols*, pages 55–78. Springer, 2018.
- [26] Gregory Sliwoski, Sameer Kothiwale, Jens Meiler, and Edward W. Jr. Lowe. Computational methods in drug discovery. *Pharmacological Reviews*, 66(1):334–395, 2013.
- [27] Francesca Grisoni, Davide Ballabio, Roberto Todeschini, and Viviana Consonni. Molecular descriptors for structure–activity applications: A hands-on approach. In Orazio Nicolotti, editor, *Computational Toxicology: Methods and Protocols*, pages 3–54. Springer, 2018.
- [28] Roger Perkins, Hua Fang, Weida Tong, and William J. Welsh. Quantitative structure-activity relationship methods: Perspectives on drug discovery and toxicology. *Environmental Toxicology and Chemistry*, 22(8):1666–1679, 2003.

-
- [29] K. K. Mak and M. R. Pichika. Artificial intelligence in drug development: present status and future prospects. *Drug Discovery Today*, 24(3):773–780, March 2019.
- [30] Orazio Nicolotti, editor. *Computational Toxicology: Methods and Protocols*, volume 1800 of *Methods in Molecular Biology*. Springer, New York, NY, 2018.
- [31] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.
- [32] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2nd edition, 2009.
- [33] Scikit learn Developers. *sklearn.metrics.pairwise.cosine_similarity - Scikit-learn 1.5.0 documentation*, 2024. Accessed: 2025-01-21.
- [34] Joost de Winter, Samuel Gosling, and J. Potter. Comparing the pearson and spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data. *Psychological Methods*, 21:273–290, 09 2016.
- [35] Swedish Chemicals Agency. CLP - Classification, Labelling and Packaging. <https://www.kemi.se/en/rules-and-regulations/clp---classification-labelling-and-packaging>. Accessed: 2025-02-06.
- [36] Scikit learn Developers. *sklearn.svm.SVR - Scikit-learn 1.5.0 documentation*, 2024. Accessed: 2025-01-21.

A

Appendix 1

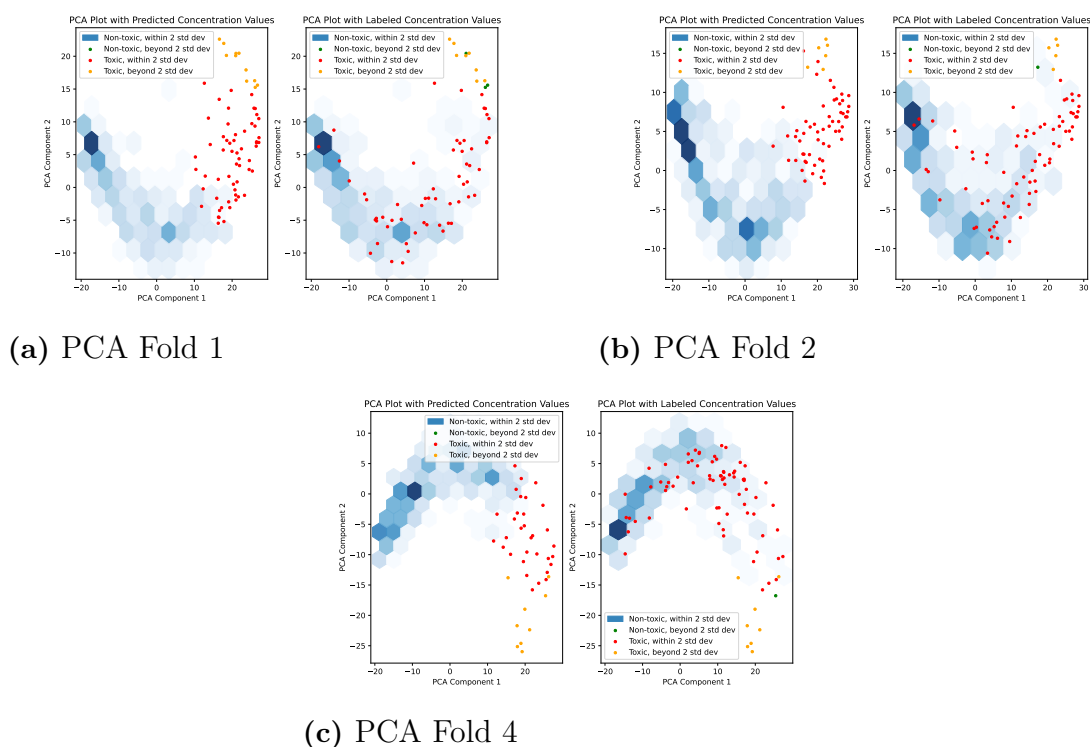
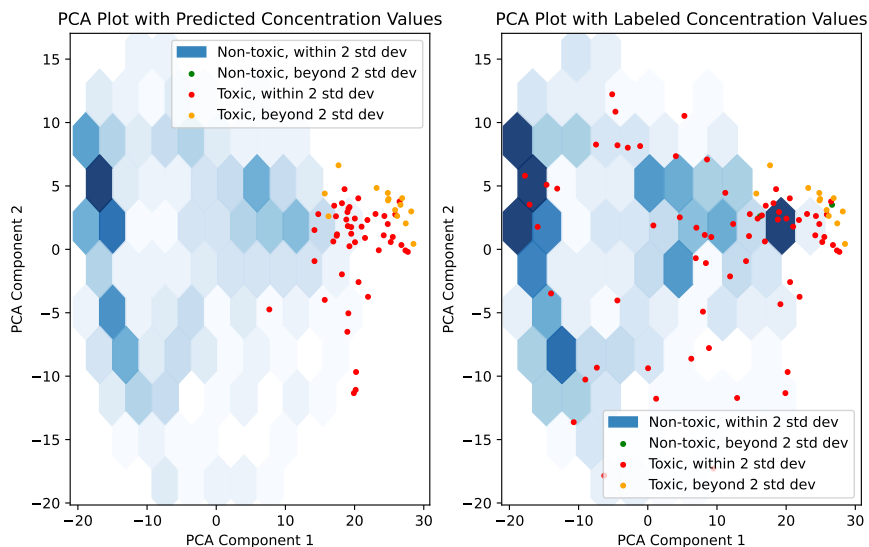
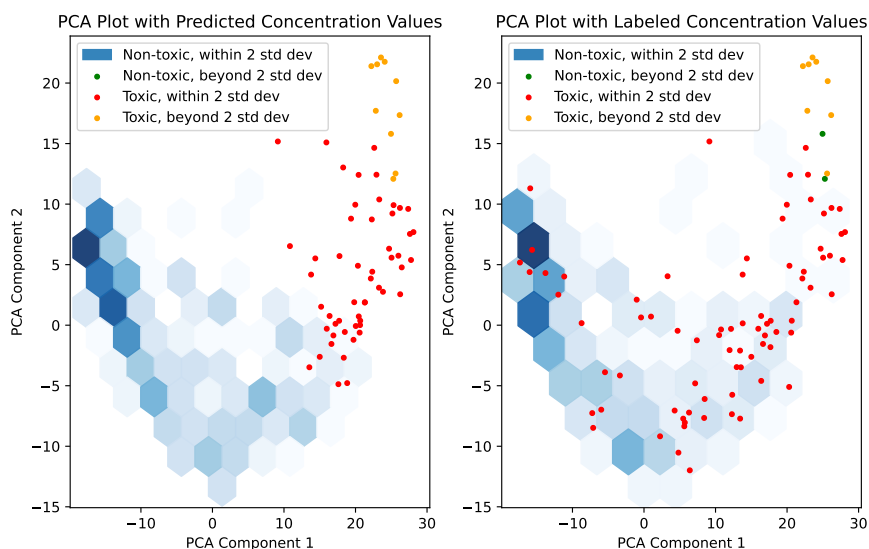


Figure A.1: PCA plots for Fold 1, 2, and 4 of Dataset 1 illustrating the placement of toxic and uncertain chemicals in the TRIDENT chemical space. Chemicals classified as non-toxic and within two standard deviations of the mean cosine similarity of the training data are shown as blue hexagons. The shade of blue indicates the density of chemicals in each area, with deeper shades representing higher densities and lighter shades indicating lower densities. Chemicals classified as non-toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in green. Chemicals classified as toxic within two standard deviations of the mean cosine similarity of the training data are shown in red. Chemicals classified as toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in orange. The PCA plots to the left shows the classifications of points using the predicted values in the dataset. The PCA plots to the right shows classifications using the labeled values in the dataset.

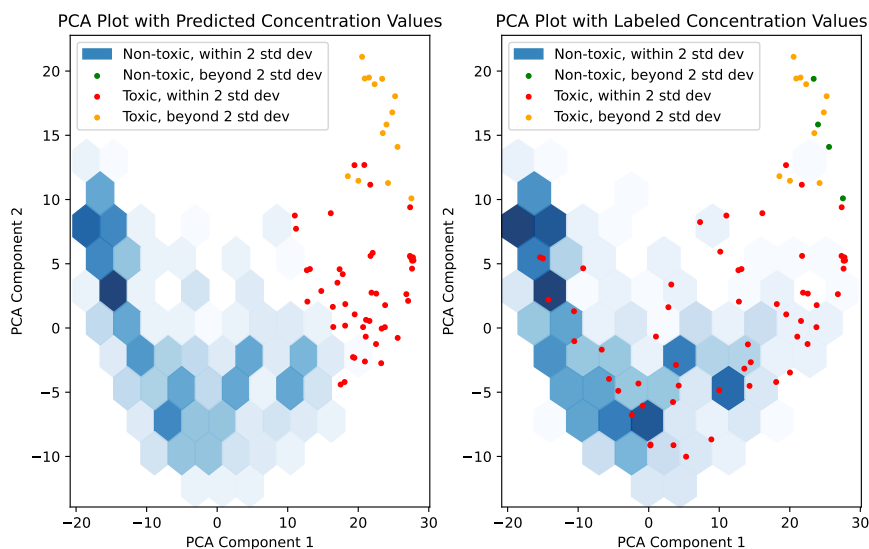


(a) PCA Fold 5

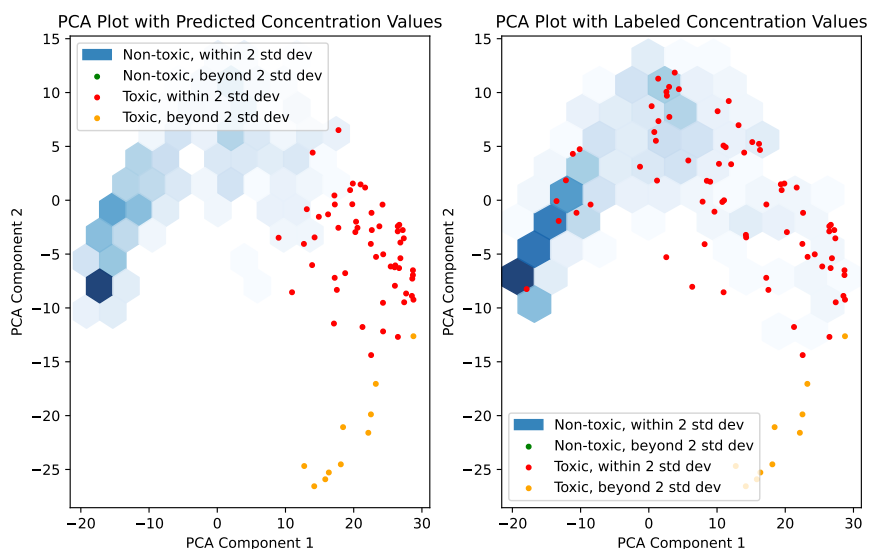


(b) PCA Fold 6

Figure A.2: PCA plots for Folds 5 and 6 of Dataset 1 illustrating the placement of toxic and uncertain chemicals in the TRIDENT chemical space. Chemicals classified as non-toxic and within two standard deviations of the mean cosine similarity of the training data are shown as blue hexagons. The shade of blue indicates the density of chemicals in each area, with deeper shades representing higher densities and lighter shades indicating lower densities. Chemicals classified as non-toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in green. Chemicals classified as toxic within two standard deviations of the mean cosine similarity of the training data are shown in red. Chemicals classified as toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in orange. The PCA plots to the left shows the classifications of points using the predicted values in the dataset. The PCA plots to the right shows classifications using the labeled values in the dataset.

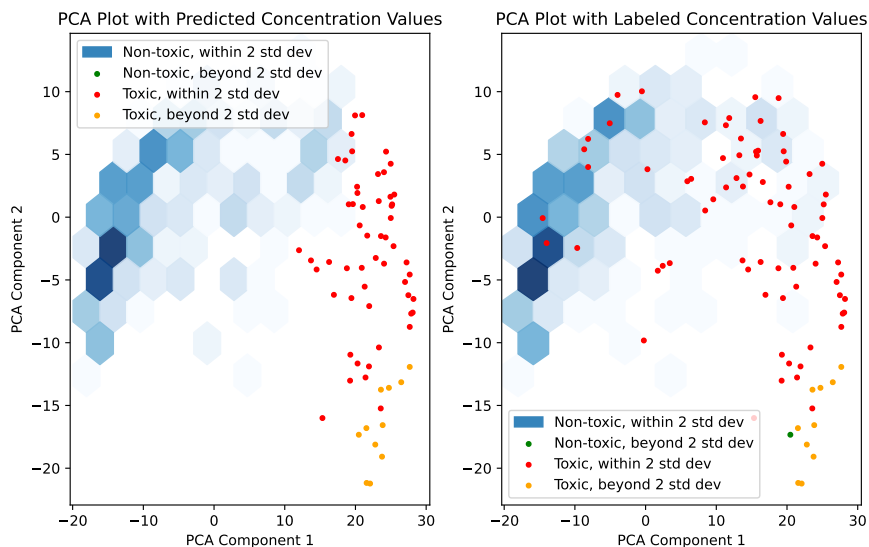


(a) PCA Fold 7

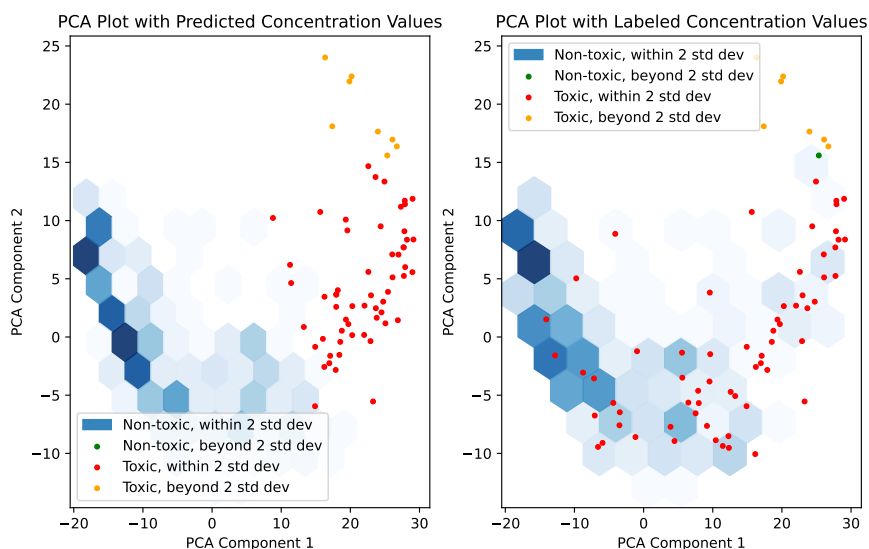


(b) PCA Fold 8

Figure A.3: PCA plots for Folds 7 and 8 of Dataset 1 illustrating the placement of toxic and uncertain chemicals in the TRIDENT chemical space. Chemicals classified as non-toxic and within two standard deviations of the mean cosine similarity of the training data are shown as blue hexagons. The shade of blue indicates the density of chemicals in each area, with deeper shades representing higher densities and lighter shades indicating lower densities. Chemicals classified as non-toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in green. Chemicals classified as toxic within two standard deviations of the mean cosine similarity of the training data are shown in red. Chemicals classified as toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in orange. The PCA plots to the left shows the classifications of points using the predicted values in the dataset. The PCA plots to the right shows classifications using the labeled values in the dataset.



(a) PCA Fold 9



(b) PCA Fold 10

Figure A.4: PCA plots for Fold 9 and 10 of Dataset 1 illustrating the placement of toxic and uncertain chemicals in the TRIDENT chemical space. Chemicals classified as non-toxic and within two standard deviations of the mean cosine similarity of the training data are shown as blue hexagons. The shade of blue indicates the density of chemicals in each area, with deeper shades representing higher densities and lighter shades indicating lower densities. Chemicals classified as non-toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in green. Chemicals classified as toxic within two standard deviations of the mean cosine similarity of the training data are shown in red. Chemicals classified as toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in orange. The PCA plots to the left shows the classifications of points using the predicted values in the dataset. The PCA plots to the right shows classifications using the labeled values in the dataset.

Table A.1: Table presenting chemicals located in the region of the PCA plot that is typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classifies them as toxic. Their CAS number and name are displayed for each chemical. (Fold 1.)

CAS	Name
75-86-5	Acetone cyanohydrin
741-58-2	Bensulide
63449-39-8	Chlorowax 40
593-08-8	2-Tridecanone
111-85-3	1-Chlorooctane
2437-29-8	Malachite Green Oxalate
300-76-5	Naled
106-50-3	p-Phenylenediamine
643-79-8	o-Phthalaldehyde
10108-73-3	Cerium nitrate
100-25-4	1,4-Dinitrobenzene
1306-25-8	Cadmium telluride

Table A.2: Table presenting chemicals located in the region of the PCA plot that is typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classifies them as toxic. Their CAS number and name are displayed for each chemical. (Fold 2.)

CAS	Name
281-23-2	Adamantane
78-95-5	Chloroacetone
2778-42-9	1,3-Bis(1-isocyanato-1-methylethyl)benzene
6257-64-3	Benzenamine, 4,4'-azobis[N,N-dimethyl-
21564-17-0	2-(Thiocyanomethylthio)benzothiazole
591-89-9	Mercuric potassium cyanide
10476-85-4	Strontium dichloride hexahydrate
64628-44-0	Triflumuron
106-51-4	1,4-Benzoquinone
10102-90-6	Pyrophosphoric acid, copper salt
13453-32-2	Thallium(III) chloride
882-33-7	Diphenyl disulfide

Table A.3: Table presenting chemicals located in the region of the PCA plot that is typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classifies them as toxic. Their CAS number and name are displayed for each chemical. (Fold 3.)

CAS	Name
15667-63-7	1-Cyanoallyl acetate
107-02-8	Acrolein
2896-70-0	4-Oxo-2,2,6,6-tetramethylpiperidinoxy
751-83-7	(3-Methyl-5-(2,6,6-trimethyl-1-cyclohexen-1-yl)-2,4-pentadienyl) triphenylphosphonium sulphate
2224-44-4	4-(2-Nitrobutyl)morpholine
14008-58-3	N-Butyl-N'-nicotinoylurea
556-61-6	Methyl isothiocyanate
79617-96-2	Sertraline
68092-46-6	Zinc m-toluate
1193-21-1	4,6-Dichloropyrimidine
2682-20-4	2-Methyl-4-isothiazolin-3-one hydrochloride
13473-90-0	Aluminum nitrate nonahydrate
143-50-0	Chlordecone
115-86-6	Triphenyl phosphate
10124-36-4	Cadmium sulfate
7166-19-0	beta-Bromo-beta-nitrostyrene
528-29-0	1,2-Dinitrobenzene
12135-22-7	Palladium dioxide
13815-17-3	Tetraamminepalladium(II) dichloride
72459-58-6	Triazoxide

Table A.4: Table presenting chemicals located in the region of the PCA plot that is typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classifies them as toxic. Their CAS number and name are displayed for each chemical. (Fold 4.)

CAS	Name
96792-67-5	Ethanone, 1-(hexahydrodimethyl-1H-benzindenyl)-
2654-57-1	4-Methyl-1-phenylpyrazolidin-3-one
27668-52-6	Dimethyloctadecyl[3-(trimethoxysilyl)propyl]ammonium chloride
1912-26-1	Trietazine
685-88-1	Diethyl fluoromalonate
54-11-5	Nicotine
18268-76-3	2-Chloro-4-hydroxy-5-methoxybenzaldehyde
6623-41-2	2-Amino-4,5-dimethylphenol
603-85-0	2-Amino-3-nitrophenol
80324-43-2	Basic Brown 1
62229-08-7	Sulfurous acid, lead salt, dibasic
69094-18-4	2,2-Dibromo-2-nitroethanol
76-06-2	Chloropicrin
7761-88-8	Silver nitrate
12062-24-7	Silicate(2-), hexafluoro-, copper(2+) (1:1)
7723-14-0	Phosphorus

Table A.5: Table presenting chemicals located in the region of the PCA plot that is typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classifies them as toxic. Their CAS number and name are displayed for each chemical. (Fold 5.)

CAS	Name
4170-30-3	cis-Crotonaldehyde
2094-99-7	N-propan-2-ylidene-3-prop-1-en-2-ylbenzamide
2869-34-3	Tridecylamine
36362-09-1	2-(Decylthio)ethanamine hydrochloride
143-16-8	Dihexylamine
1397-94-0	Antimycin A
10605-21-7	Carbendazim
18181-70-9	Iodofenphos
581-64-6	Thionine
56840-61-0	Romucide
10294-26-5	Silver sulfate
554-84-7	3-Nitrophenol
108-80-5	Cyanuric acid
130-16-5	5-Chloro-8-hydroxyquinoline
615-67-8	Chlorohydroquinone
7439-91-0	Lanthanum
13463-41-7	Zinc pyrithione

Table A.6: Table presenting chemicals located in the region of the PCA plot that is typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classifies them as toxic. Their CAS number and name are displayed for each chemical. (Fold 6.)

CAS	Name
106-95-6	Allyl bromide
1647-16-1	1,9-Decadiene
28801-69-6	Tributyl(neodecanoyloxy)stannane
78-97-7	Lactonitrile
113158-40-0	Fenoxaprop
55792-61-5	2'-(Octyloxy)-acetanilide
818-08-6	Dibutyltin oxide
115-90-2	Fensulfothion
128-03-0	Potassium dimethyldithiocarbamate
137-42-8	Metam-sodium
68890-66-4	Piroctone olamine
95-81-8	2-Chloro-5-methylaniline
3400-09-07	Chlorimide
121-88-0	2-Amino-5-nitrophenol
108-98-5	Benzenethiol

Table A.7: Table presenting chemicals located in the region of the PCA plot that is typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classifies them as toxic. Their CAS number and name are displayed for each chemical. (Fold 7.)

CAS	Name
107-18-6	Allyl alcohol
13893-53-3	2-Amino-2,3-dimethylbutanenitrile
136-53-8	Zinc 2-ethylhexanoate
333-43-7	O-Ethyl S-p-tolyl ethylphosphonodithioate
2589-57-3	Dimethyl 2,2'-azobis(2-methylpropionate)
9012-76-4	Chitosan
12108-13-3	Methylcyclopentadienyl manganese tricarbonyl
162881-26-7	Phenylbis(2,4,6-trimethylbenzoyl)phosphine oxide
24279-39-8	4-Amino-3,5-dichlorobenzotrifluoride
54982-83-1	1,4-Dioxacyclohexadecane-5,16-dione
10028-15-6	Ozone
1314-87-0	Lead(II) sulfide

Table A.8: Table presenting chemicals located in the region of the PCA plot that is typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classifies them as toxic. Their CAS number and name are displayed for each chemical. (Fold 8.)

CAS	Name
74-90-8	Hydrogen cyanide
5989-27-5	D-Limonene
67124-09-8	2-Propanol, 1-(tert-dodecylthio)-
123-03-5	Cetylpyridinium chloride
3252-43-5	Dibromoacetonitrile
1111-67-7	Cuprous thiocyanate
123-31-9	Hydroquinone
7783-06-04	Hydrogen sulfide
73398-89-7	3,6-Bis(diethylamino)-9-(2-(methoxycarbonyl)phenyl)xanthylium tetrachlorozincate

Table A.9: Table presenting chemicals located in the region of the PCA plot that is typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classifies them as toxic. Their CAS number and name are displayed for each chemical. (Fold 9.)

CAS	Name
111-83-1	1-Bromooctane
124088-59-1	Benzyltrimethyloctadecyl ammonium 3-nitrobenzenesulfonate
71873-51-3	C.I. Acid Yellow 218
41198-08-7	Profenofos
64-00-6	m-Cumenyl methylcarbamate
7786-34-7	Mevinphos
66215-27-8	Cyromazine
1306-23-6	Cadmium sulfide
3570-55-6	Bis(2-mercaptoethyl) sulfide
19168-23-1	Ammonium hexachloropalladate(IV)

Table A.10: Table presenting chemicals located in the region of the PCA plot that is typically occupied by non-toxic chemicals, $PC1 < 0$, but whose labeled value classifies them as toxic. Their CAS number and name are displayed for each chemical. (Fold 10.)

CAS	Name
83-26-1	Pindone
27458-92-0	Isotridecanol
119313-12-1	2-Benzyl-2-dimethylamino-1-(4-morpholinophenyl)-1-butanone
1338-02-9	Naphthenic acids, copper salts
16529-65-0	Docosanoic acid
78-48-8	Tribufos
19398-06-2	(2-Ethylphenyl)hydrazine
147-24-0	Diphenhydramine hydrochloride
150-75-4	p-(Methylamino)phenol sulphate
1464-42-2	L-selenomethionine
87-63-8	2-Chloro-6-methylaniline
937-14-4	3-Chloroperoxybenzoic acid
49663-84-5	pentazinc chromate octahydroxide
1420-06-0	Trifenmorph

Table A.11: Table presenting chemicals whose labeled values classify them as non-toxic that have a cosine similarity z-score beyond two standard deviations from the training data’s mean cosine similarity for all 10 folds. Their CAS number and name are displayed for each chemical.

CAS	Name
17109-49-8	Edifenphos
23149-52-2	Thiosulfuric acid (H ₂ S ₂ O ₃), silver(1+) salt (1:2)
506-64-9	Silver cyanide
24934-91-6	Chlormephos
1192-89-8	Mercury, bromophenyl-
7533-79-1	Phosphorothioic acid, O-(2,5-dichloro-4-iodophenyl) O,O-diethyl ester
90-03-9	o-(Chloromercuri)phenol
2275-14-1	Phenkapton
57808-65-8	Closantel
75-15-0	Carbon disulfide
28343-61-5	4-Hydroxy-2,5,6-trichloroisophthalonitrile
683-18-1	Dibutyltin dichloride
2554-06-05	2,4,6,8-Tetramethyl-2,4,6,8-tetravinylcyclotetrasiloxane
121-75-5	Malathion
115-27-5	Chlorendic anhydride
82-68-8	Pentachloronitrobenzene
260-94-6	Acridine
1067-33-0	Acetic acid;dibutyltin

Table A.12: Table displaying the validation metrics for big ($L1 \geq 2$) and small $L1 < 2$ errors for the SVM model.

Validation MSE small error: 0.21752099596716684
Validation MSE big error: 3.745423299745528
Correlation Pearson small error: 0.20317460033650697
Correlation Pearson big error: -0.08598489044674255
Correlation Spearman small error: 0.19693095458811166
Correlation Spearman big error: 0.004926108374384235

Table A.13: Table displaying the validation metrics for big ($L1 \geq 2$) and small $L1 < 2$ errors for the kNN model.

Validation MSE small error: 0.21673568
Validation MSE big error: 3.8634408
Correlation Pearson small error: 0.17219976
Correlation Pearson big error: 0.32054102
Correlation Spearman small error: 0.16818801638462735
Correlation Spearman big error: 0.23596059113300488

Table A.14: Table displaying the validation metrics for big ($L1 \geq 2$) and small $L1 < 2$ errors for the naive kNN model.

Validation MSE small errors: 0.21854526
Validation MSE big errors: 3.7929041
Correlation Pearson small error: 0.1696001
Correlation Pearson big error: 0.30295858
Correlation Spearman small error: 0.1840172791947062
Correlation Spearman big error: 0.23546798029556643

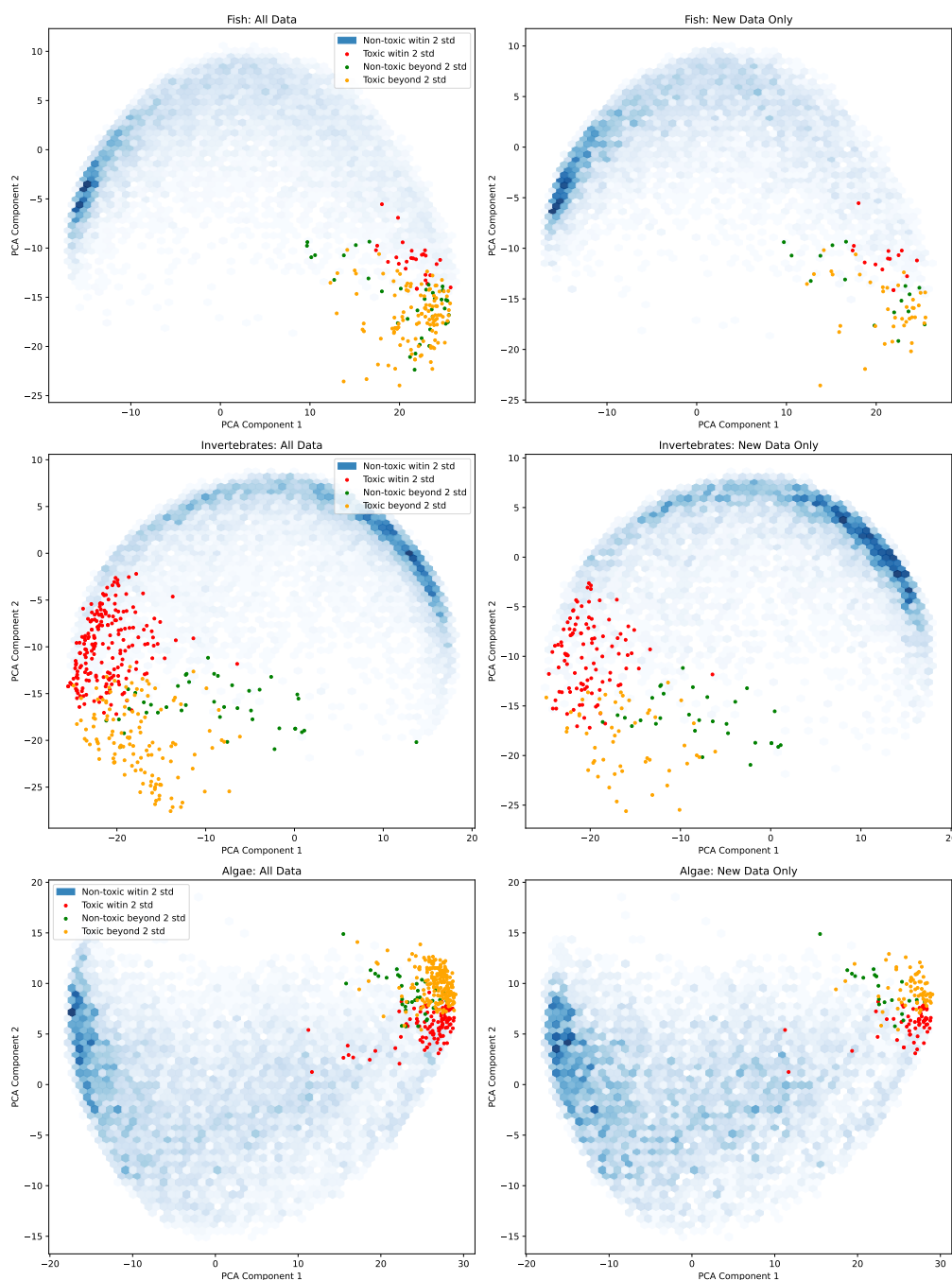


Figure A.5: Figure showing the PCA scatterplots for Dataset 2 with endpoint EC50. The plots to the left contain all chemicals in the dataset with predictions for each species group. The plots to the right contain only chemicals that were not used in the training of the model. Chemicals classified as non-toxic and within two standard deviations of the mean cosine similarity of the training data are shown as blue hexagons. The shade of blue indicates the density of chemicals in each area, with deeper shades representing higher densities and lighter shades indicating lower densities. Chemicals classified as non-toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in green. Chemicals classified as toxic within two standard deviations of the mean cosine similarity of the training data are shown in red. Chemicals classified as toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in orange.

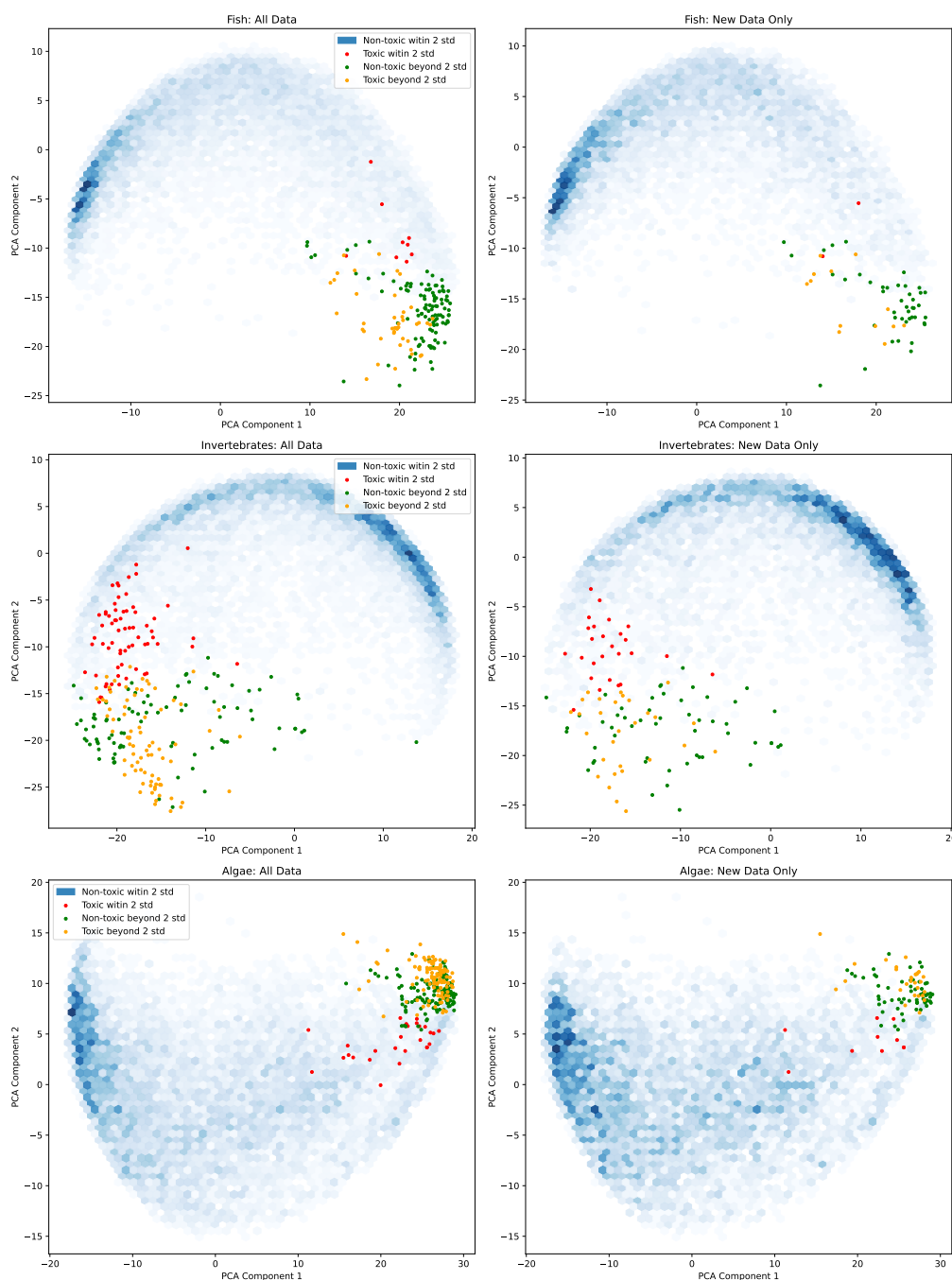


Figure A.6: Figure showing the PCA scatterplots for Dataset 2 with endpoint EC10. The plots to the left contain all chemicals in the dataset with predictions for each species group. The plots to the right contain only chemicals that were not used in the training of the model. Chemicals classified as non-toxic and within two standard deviations of the mean cosine similarity of the training data are shown as blue hexagons. The shade of blue indicates the density of chemicals in each area, with deeper shades representing higher densities and lighter shades indicating lower densities. Chemicals classified as non-toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in green. Chemicals classified as toxic within two standard deviations of the mean cosine similarity of the training data are shown in red. Chemicals classified as toxic beyond two standard deviations of the mean cosine similarity of the training data are shown in orange.

Table A.15: Table presenting the chemicals classified as toxic for the endpoint EC10, not present in the training data for the model. The chemical is classified as toxic for the intersection of invertebrates and algae. For each chemical, their CAS number and name are displayed.

CAS	Name
17955-88-3	1,1,1,3,5,5,5-heptamethyl-3-octyltrisiloxane
5356-84-3	1,1,1,5,5,5-hexamethyl-3-[(trimethylsilyl)oxy]-3-vinyltrisiloxane
84870-65-5	2-[[4-(diethylamino)-2-methylphenyl]azo]-5-nitrobenzene-1,3-dicarbonitrile
59130-69-7	Hexadecyl 2-ethylhexanoate
1116-70-7	Tributylaluminium

Table A.16: Table presenting the chemicals classified as toxic for the endpoint EC50, not present in the training data for the model. The chemical is classified as toxic for all species groups. For each chemical, their CAS number and name are displayed.

CAS	Name
3349-36-8	Dibutoxydibutylstannane

Table A.17: Table presenting the chemicals classified as toxic for the endpoint EC50, not present in the training data for the model. The chemical is classified as toxic for the intersection of invertebrates and fish. For each chemical, their CAS number and name are displayed.

CAS	Name
92-78-4	4'-chloro-3-hydroxy-2-naphthanilide
84-58-2	4,5-dichloro-3,6-dioxocyclohexa-1,4-diene-1,2-dicarbonitrile
85750-13-6	4-[(2-chloro-4-nitrophenyl)azo]-N-ethyl-N-[2-[1-(2-methylpropoxy)ethoxy]ethyl]aniline
3349-36-8	Dibutoxydibutylstannane
7803-51-2	Phosphine
119345-01-6	Phosphorous trichloride, reaction products with 1,1'-biphenyl and 2,4-bis(1,1-dimethylethyl)phenol
118712-89-3	transfluthrin (ISO); 2,3,5,6-tetrafluorobenzyl (1R,3S)-3-(2,2-dichlorovinyl)-2,2-dimethylcyclopropanecarboxylate

Table A.18: Table presenting the chemicals classified as toxic for the endpoint EC50, not present in the training data for the model. The chemical is classified as toxic for the intersection of algae and fish. For each chemical, their CAS number and name are displayed.

CAS	Name
3349-36-8	Dibutoxydibutylstannane
1067-55-6	Dibutyldimethoxystannane
7798-23-4	Tricopper bis(orthophosphate)

Table A.19: Table presenting the chemicals classified as toxic for the endpoint EC50, not present in the training data for the model. The chemical is classified as toxic for the intersection of invertebrates and algae. For each chemical, their CAS number and name are displayed.

CAS	Name
17955-88-3	1,1,1,3,5,5,5-heptamethyl-3-octyltrisiloxane
84870-65-5	2-[[4-(diethylamino)-2-methylphenyl]azo]-5-nitrobenzene-1,3-dicarbonitrile
125304-04-3	A mixture of: isomers of 2-(2H-benzotriazol-2-yl)-4-methyl-(n)-dodecylphenol; isomers of 2-(2H-benzotriazol-2-yl)-4-methyl-(n)-tetracosylphenol; isomers of 2-(2H-benzotriazol-2-yl)-4-methyl-5,6-didodecyl-phenol. n=5 or 6
24577-34-2	Bis[(2-ethyl-1-oxohexyl)oxy]dioctylstannane
2587-76-0	Chlorotrioctylstannane
3349-36-8	Dibutoxydibutylstannane
12158-74-6	Dicopper hydroxide phosphate
1120-49-6	Didecylamine
59130-69-7	Hexadecyl 2-ethylhexanoate
629-70-9	Hexadecyl acetate
26272-90-2	Hexadecyl chloroformate
1943-84-6	Hexadecyl isocyanate
5538-95-4	N-dodecylpropane-1,3-diamine
13566-03-5	Palladium sulphate
3375-31-3	Palladium(II) acetate
68002-58-4	Quaternary ammonium compounds, di-C14-18-alkyldimethyl, Me sulfates
68413-68-3	Tetraamminepalladium(2+) dihydroxide
3590-84-9	Tetraoctyltin
1116-70-7	Tributylaluminium
1120-02-1	Trimethyloctadecylammonium bromide
869-59-0	Trioctylstannane

DEPARTMENT OF MATHEMATICAL SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden
www.chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY