Leveraging Machine Learning for PFAS Toxicity Estimation

Chukwudumebi Ubogu ubogu@student.chalmers.se

Gritta Joshy gusjoshgr@student.gu.se

Natalia Alvarado gusalvarsi@student.gu.se

Yunyi Xu yunyix@student.chalmers.se

DIT892 - Project in Data Science

Supervised by: Rocío Mercado Oropeza and Richard Beckmann

GitHub Repository for the replication code

1 Overview

Perfluoroalkyl and polyfluoroalkyl substances (PFAS) are a group of manufactured chemicals that due to their properties have enjoyed widespread application in almost all facets of manufacturing and consequently consumer consumption (European Chemical Agency (ECHA), 2024). However, with this widespread use and a growing body of studies, concerns about their environmental and health effects have grown concurrently. To date pathways to establishing the toxicity of PFAS compounds are still dominated by traditional methods, which are expensive and time-consuming. As such, it will leave global regulatory bodies always playing catchup regarding an updated list of PFAS compounds. Fortunately, in the age of AI, and machine learning, numerous strides have been made in employing computational methods in healthcare, that can be applied to toxicity prediction of chemical compounds. The process of drug discovery has benefited from machine learning, and some of the techniques that accelerate the drug discovery process can now be easily applied to the determination of toxicity, particularly for the PFAS chemical group. A few studies have experimented with computational methods to leverage knowledge from established toxicity knowledge from traditional methods, to predict toxicity of an ever-growing list of PFAS compounds. Our study aims to contribute to this emerging field by replicating and building upon existing methodologies in PFAS toxicity prediction. In addition to a thorough literature review, we conducted a replication study where we applied machine learning models to predict PFAS toxicity. This involved implementing and testing several ML models, allowing us to compare their predictive capabilities on a large PFAS dataset. To enhance our analysis, we visualized key aspects of the data, including toxicity rank correlations and model performance metrics, using density plots, confusion matrices, and other visualization tools. These visualizations helped identify patterns in the data and assess the strengths and weaknesses of each model. Overall, this project not only assesses the current state of computational PFAS toxicity prediction but also contributes by validating, replicating, and extending existing methodologies. Through our replication study, visualization analysis, and model comparisons, we provide a comprehensive evaluation of computational methods in toxicity prediction, ultimately supporting the ongoing development of faster, cost-effective, and scalable solutions for PFAS toxicity assessment.

2 Introduction

This chapter outlines previous studies on the topic.

2.1 History and applications of Perfluoroalkyl and polyfluoroalkyl substances (PFAS)

Since the discovery of Perfluoroalkyl and polyfluoroalkyl substances (PFAS) in the 1940s their use in the manufacture of things in everyday life is pervasive. Also known as forever chemicals, this manufactured compound comprises a larger group of synthetic chemicals with alluring properties that make them highly versatile in a wide range of manufacturing.

In 1938, a DuPont chemist accidentally discovered polytetrafluoroethylene, which became widely known by its marketed name, Teflon , this was the first of what later became the (PFAS) compound. Through the 1940s and 1950s, DuPont ac-

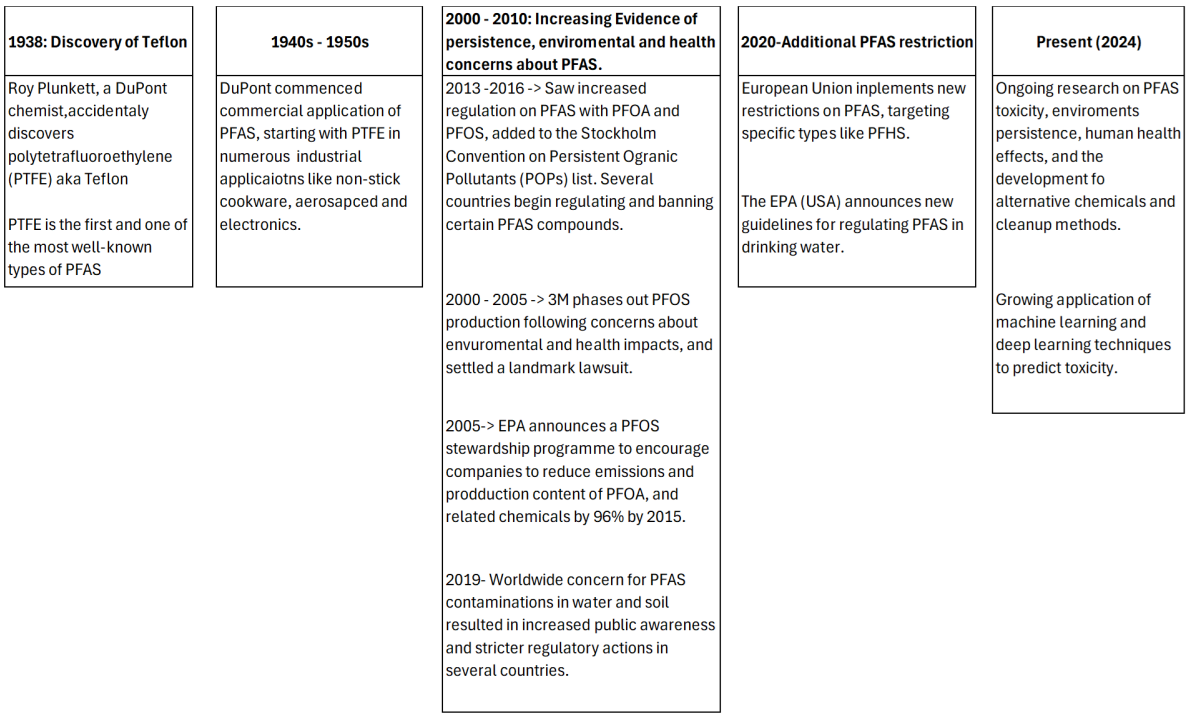


Figure 1: Timeline of PFAS compound events

celerated the commercial use of Teflon in cookware, electronics and electronics. Shortly after, the use of Teflon spread in manufacturing across the globe and additional PFAS compounds were found, growing the list of what is now the problematic PFAS, "forever" chemicals. Some of the properties behind the popularity of these compounds are briefly explained. PFAS are highly thermally stable and so resist degradation at higher temperatures making them ideal candidates for use in firefighting foams. In addition to this property, they are hydrophobic and oleophobic (oil and water resistant) which has led to their widespread use in non-stick cookware able to resist higher temperatures. This property is also coveted in textile manufacturing and is the primary characteristic that defines Teflon. PFAS are chemically stable and resist reaction with other compounds. They are resistant to acids, chemicals and bases. This has led PFAS to find widespread use in industrial manufacturing under harsh conditions. Lastly, they are durable and persist in the environment for many decades (Joudan & Lundgren, 2022). Racz & Kempisty (2021) succinctly illustrate the PFAS pathway that led to their global proliferation.

Ironically, these properties are the reason why, for more than seven decades after their discovery, there is growing agreement that the PFAS group is a problematic class of compounds. They may have disruptive effects on the environment as well as on biological pathways in humans and other organisms.

PFAS bioaccumulate in the environment, and organisms, including humans, long time exposure can quickly lead to higher levels of concentration in human tissue and the environment. PFAS as a group has been found to disrupt bodily functions, particularly the endocrine system, responsible for the regulation of hormones that have implications for reproduction, the immune system and in children has been linked to developmental delays. While not all PFAS compounds are considered problematic, many are considered toxic to varying levels particularly due to their proximity in characteristics and chemical traits (Barry et al., 2013; Mayr et al., 2016; Joudan and Lundgren, 2022)

2.2 Efforts to understand chemical toxicity

Understanding the link between a compound and its biological effects, toxicity or bioactivity, for example, is done through several processes. *In-vitro* assessments see experiments conducted outside of a living organism typically in a controlled environment, like culture flasks and petri dishes. *In vivo*,

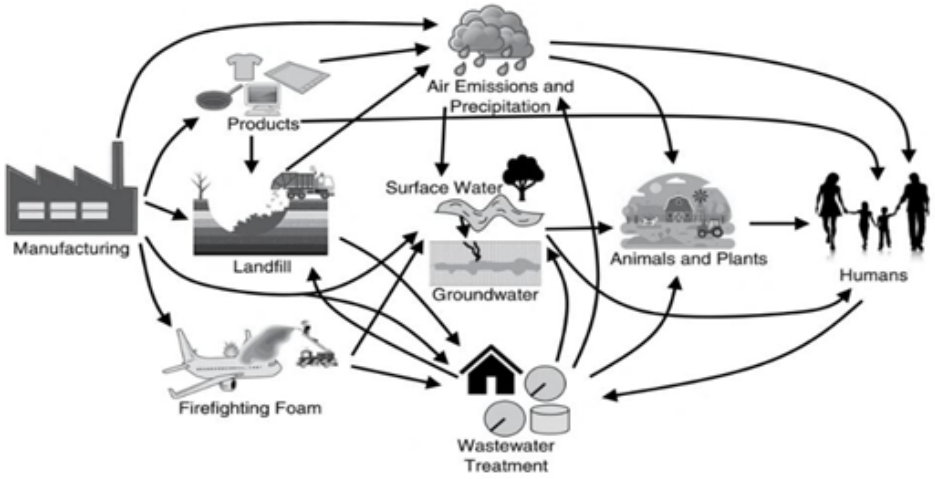


Figure 2: Flow of PFAS from manufacturing to the environment and eco-systems. *Source*: Racz & Kempisty, 2021, page 7

assessments are conducted within the living organism, where the effects under investigation are directly observed in the intended (or similar) organisms. These studies are preferred for their insight into realistic biological responses. However, they are often time-consuming, the most expensive, and quickly run into ethical dilemmas and problems. In silico studies refer to computational simulations to map biological processes and predict outcomes. Computational methods can be applied to molecular dynamics, pharmacokinetics, and now with machine learning, toxicology. Studies under the latter classification, have proven the fastest and the most cost-effective, but remain hampered by the availability and quality of input data. However, combining these types of studies works well to shorten the knowledge accumulation process, whereby in silico studies flag what *in-vitro* testing should be prioritised, and if successful, which should be progressed for validation through *in-vivo* studies Barry et al., 2013; Mayr et al., 2016; Raies & Bajic, 2016). Consistency in the predictions from machine learning models has thus become a handy tool to aid regulatory bodies on which chemicals to focus efforts and investigations.

2.3 Hazardous nature of PFAS compounds

Several studies have investigated the mechanisms through which PFAS group are toxic. Barry et al., (2013) found, within a population of 69,00 participants, a link between PFAS concentration levels and kidney and testicular cancers. Similarly, disruptive pathways to the endocrine system were linked to the prevalence of PFAS compounds. They have been linked to metabolic disorders like type 2 diabetes, liver toxicity, developmental delays, and thyroid malfunction. PFAS have also been linked to immunotoxicity in humans, causing higher incidences of autoimmune (Liu et al., 2007; Gustavsson et al., n.d.; Hayman et al., 2021; Jane L Espartero et al., 2022; Rudzanova et al., 2023).

Environmentally, PFAS compounds have also been found to bioaccumulated along the aquatic good food chain, for example in the Great Lakes basin of the United states of America and wastewater plants (Gustavsson et al., n.d.; Kannan et al., 2005; Hayman et al., 2021).
Despite their widespread proliferation, the estimated costs of potential PFAS cleanup are astronomical and with unproven efficacy. Clean-up cost estimates run into trillions, currently between 20 and 7000 trillion USD (Ling, 2024).) While the efficacy of PFAS removal is under investigation, predicting toxicity levels for these compounds offers the authorities some insight into which chemicals to focus, on and arrest their unabated accumulation.

2.4 Computational Methods in Toxicity prediction

While industry works to investigate how PFAS can be removed from the environment and regulatory bodies, around the world, work to discontinue, ban or limit the industrial usage of a growing list of PFAS, the need to predict the toxicity of PFAS and PFAS like compounds remains. The progress that has been made in computation toxicology is due to a growing database of chemical toxicities, an increase in the computational of critical hardware and advancement in machine learning techniques that can take advantage of these advancements (Feinstein et al., 2021).

Computational methods have proliferated drug discovery, and with the right dataset, some methods can also be applied in chemical toxicology. During the former process, with existing data on active compounds, including their structure, their biological efficacy, and interactions with intended targets, similarity searches can be done to find other potential compounds with the intended effects. So too can databases with established toxicity levels for compounds, along with their structures, and biotoxicities be used to assess the toxicity of similar compounds (Mayr et al., 2016; Feinstein et al., 2021; Kim et al., 2021).

The concept of structural-activity relationship (SAR) rests on the premise that the biological activities of a chemical are related to its molecular structure. Quantitative Structural-Activity Relationship (QSAR) is a set of analysis models that use chemical properties and the molecular structure to describe the compounds (Benfenati et al., 2013). The bioactivity, of interest in the case of the PFAS compound is death in the living organism, and the quantified part will be the concentration the PFAS compound that causes the death.

Raies & Bajic (2016) explain the five steps involved in developing a prediction model. Biological data gathering for associations between chemicals and toxicity endpoints, calculations of molecular descriptors of the chemicals in question; generation of a prediction model, accuracy evaluation of the prediction model, and its interpretation. QSAR is what permits the application of computational methods, in various forms, for the determination of toxicity of chemicals. The reasons for the paucity of studies that specifically use computational methods to establish the toxicity of PFAS compounds are these:

- The definition of toxicity is broad, and for studies to be comparable, they must have used the same measure of toxicity to limit the potential for misclassification.
- Sufficient studies reporting the comparable measures of toxicity, on a sufficiently large subset of PFAS compounds must exist to train and validate models. This improves the statistical confidence in the toxicity prediction of such models on PFAS compounds that have yet to be studied (Benfenati et al., 2013).
- The complexity of the structures is difficult to reduce to encompassing generalizations from which toxicity can be derived.

While computational studies on PFAS toxicity remain lean, there have been a few noteworthy ones. Three studies merit mention, Bhhatarai & Gramatica (2011), Feinstein *et al.,* (2021), and Mayr *et al.*, (2016). The first two studies investigated the toxicity of PFAS compounds in lab rats using a lethal dose that killed 50 per cent of the observed mice population shortly after the chemical was administered (LD_{50}).
The first study exploited a QSAR relationship, using experimental data, to predict toxicity of PFAS, including the most toxic that should be prioritised for risk assessment. To start, experimental datasets were necessary to extract structure and responses. The study employed multiple linear regression based on theoretical molecular descriptors. Training and prediction sets are then built using experimental datasets. Subsequently, the two sets were used to statistically derive predictive models, which verified 376 per and polyfluorinated chemicals. Essentially, the study built and verified a "similarity measure" of known PFAS toxins and used that to toxic properties of "similar" chemicals for which no toxic data is available.

Feinstein et al. (2021) used publicly available datasets on oral rats LD_{50} to create a dataset with more than 30,000 compounds, from which 519 had carbon-fluorine bonds corresponding to PFAS-like molecules. The authors bench-marked several machine learning models; additionally, they used transfer learning; a particularly useful method when experimental datasets are small. Additionally, extra architecture was used so that models identified regions of prediction with greater

confidence while avoiding those with high uncertainty. Several ML models were used, a deep neural network, a random forest regressor, a Gaussian process regression and a graph neural network with different expressions of the molecules such as the Extended-Connectivity Fingerprints (ECFP), Mordred descriptors and graphs. Much like in Mayr et al., (2016), the strongest model prediction came from the DNN Mordred model. However, when validated against a database of PFAS-like compounds, the model was found to be overconfident.

2.5 Purpose of the study

The objective of this study is to reproduce the work and findings of Feinstein et al. (2021). To assess whether when given the same data set on existing toxicity levels for a select group of PFAS compounds, if using the same class of machine learning algorithms results in the same predictions. In addition, if possible, within the allotted time frame, alternative algorithms will be tested to improve the predictive power of the models in (Feinstein et al., 2021). Alternative hyper parameters and different rejection options as presented in Geifman & El-Yaniv (2019)) will also be investigated.

3 Methodology

For the purpose of reproducing the results of the predictive models of Feinstein et al. (Feinstein et al., 2021), this study first attempts to replicate the authors' work, followed by evaluating potential improvements including revised data processing, adjusted training and test splits, and optimized hyperparameter tuning for the models considered: Random Forest regression (RF), Gaussian Process regression (GP), Deep Neural Network (DNN), and Graph Convolutional Neural Network (GCN).

3.1 Tree-Based Models and Ensemble Methods

Tree-based algorithms, particularly those grouped under ensemble methods such as Random Forest (RF) and Gradient Boosted Trees, are commonly applied in toxicity prediction due to their interpretability, robustness, and ability to handle smaller datasets, which is particularly valuable for PFAS toxicity data where observations are limited.

In particular, the RF model constructs an ensemble of trees, in this case with ECFPs and Mordred descriptors which minimizes overfitting and is particularly useful in predicting toxicity levels. Feature importance metrics derived from RF allows for identification of key variables that correlate with PFAS toxicity. Other studies, such as Fernandez et al. (Fernandez et al., 2023), confirm that RF models achieve robust toxicity predictions without extensive preprocessing or normalization.

Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ with N samples, where $\mathbf{x}_i \in R^d$ represents the input features (e.g., ECFPs and Mordred descriptors) and $y_i \in R$ denotes the target toxicity level, RF operates as follows:

- **Tree Construction**: For each of the M trees in the forest:
  - A subset $\mathcal{D}_m \subset \mathcal{D}$ is created by sampling with replacement.
  - A decision tree T_m is trained on $\mathcal{D}_m$, with splits selected to minimize the impurity (e.g., Gini impurity or mean squared error for regression tasks).
- **Prediction**: For a new instance $\mathbf{x}^*$, the RF prediction $\hat{y}$ is the average of the predictions from all individual trees (Breiman, 2001):

$$\hat{y} = \frac{1}{M} \sum_{m=1}^{M} T_m(\mathbf{x}^*).$$

- **Feature Importance**: Let I_j represent the importance score of feature j. Feature importance is estimated by evaluating the reduction in impurity attributed to each feature across all nodes and trees.

The RF approach reduces overfitting through ensemble averaging and is well-suited for toxicity prediction, where interpretability is enhanced by feature importance metrics, highlighting variables correlated with PFAS toxicity.

3.2 Gaussian Process Regression (GP)

This method provides a non-parametric approach to model the relationship between inputs and outputs by assuming a Gaussian process prior over the space of functions. Let $f(\mathbf{x})$ represent the latent function governing the toxicity prediction. According to Jie Wang (Wang, 2023), the Gaussian Process can be defined as:

$$P(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}|\boldsymbol{\mu}, \mathbf{K}),$$

where:

- $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ represents the observed data points,
- $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]$ the function values,
- $\boldsymbol{\mu} = [m(\mathbf{x}_1), \ldots, m(\mathbf{x}_n)]$ the mean function,
- $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ the kernel function, which is positive definite.

With no observation, we default the mean function to $m(\mathbf{X}) = 0$, assuming the data is normalized to zero mean. The Gaussian process model is thus a distribution over functions whose shapes (smoothness) are defined by $\mathbf{K}$.

Limitations of tree-based methods include their inability to leverage gradient-based loss functions or nonlinear relationships in the same manner as neural networks. Additionally, model generalization may suffer if not carefully tuned, requiring a controlled train-test split to avoid overfitting.

3.3 Deep Neural Network (DNN)

The deep neural network (DNN) model is a class of machine learning algorithms premised on *"the information processing flow* of the brain. It is a network bases architecture that learns the molecular representation of compounds, and enhances these process to predict molecular properties, the property of interest in this study is the toxicity prediction (Wiercioch and Kirchmair, 2023).

DNNs consist of multiple hidden layers, each containing nodes (neurons) that capture complex relationships that are not efficiently modeled by linear methods. For toxicity prediction, DNNs are structured with fully connected layers and used backpropagation with gradient descent to minimize loss functions between predicted and true LD50 values. To work with the DNN model, Feinstein et al. (Feinstein et al., 2021) used ECPFs and Mordred descriptors for independently training the models.

A DNN can be defined as a layer composition such as (Zhou et al., 2022):

$$\psi_\Theta(x) = \mathbf{W}\phi_\theta(x) + \mathbf{b},$$

where $\phi_\theta(x)$ represents the feature mapping, and $\mathbf{W}$ and $\mathbf{b}$ are parameters of the linear classifier.

where the MSE Loss with Regularization is expressed as:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{2N}\sum_{k=1}^{K}\sum_{i=1}^{n}\|\psi_\Theta(x_{k,i}) - \mathbf{y}_k\|_2^2 + \frac{\lambda}{2}\|\Theta\|_F^2,$$

where:

- $\|\psi_\Theta(x_{k,i}) - \mathbf{y}_k\|_2^2$: Squared error between the predicted and true output.
- $\|\Theta\|_F^2$: Frobenius norm regularization term.
- $\lambda > 0$: Regularization parameter.

While DNNs provide superior predictive capacity over tree-based methods, they require significant data and computational resources. Interpretability also poses a challenge, as identifying which features contribute to model predictions is more complex compared to simpler models like Random Forests.

3.4 Graph Convolutional Neural Network (GCN)

Graph Convolutional Neural Networks (GCNs) extend traditional neural networks by processing graph-structured data, enabling the model to learn directly from the structural relationships within molecular graphs. Each molecule is represented as a graph, where nodes represent atoms, including different information about its features and edges represent bonds. This could be particularly relevant for toxicity prediction tasks, as molecular structure can offer important information on properties and toxicity.

GCNs are suited for modeling molecular structures in toxicity prediction, where compounds can be represented as graphs $G = (V, E)$ with nodes V corresponding to atoms and edges E to bonds.

- Following the work of (Mehta et al., 2023), a GCN can be represented as:

$$\mathbf{F}^{(i+1)} = \sigma\left(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{F}^{(i)}\mathbf{W}^{(i)}\right)$$

  - $\mathbf{F}^{(i+1)}$: Feature matrix of all nodes at layer $i+1$.
  - σ: Activation function (non-linear).
  - $\hat{\mathbf{A}}$: Graph adjacency matrix with added self-loops.
  - $\hat{\mathbf{D}}$: Diagonal matrix of node degrees.
  - $\mathbf{F}^{(i)}$: Feature matrix of all nodes at layer i.
  - $\mathbf{W}^{(i)}$: Learnable weight matrix for layer i.

This formulation allows the GCN to exploit graph-based structural information for toxicity

prediction, capturing both atomic and bond-level features inherent in chemical compounds.

Despite the structural advantages of GCNs, their performance is limited by the size of available toxicity data, which is currently insufficient for extensive model training. Like DNNs, GCNs face interpretability challenges, as determining the specific molecular features driving toxicity predictions is difficult. Overfitting is also a risk due to the high capacity of these models, necessitating careful regularization and validation techniques.

4 Data and Data processing

Information on toxicity levels for PFAS is very scarce, given this problematic, Feinstein et al. propose their dataset LDToxDB, an aggregation of different sources on toxicity levels of 13,329 compounds.

4.1 Molecule Descriptors

In chemical applications, machine learning often represents molecules as vectorized numbers known as molecular descriptors or representations. These vectors and structural characteristics and other properties of molecules, allowing for comparisons across various datasets. They can also be utilized to design new molecules that share similarities but possess more desirable attributes(Lai et al., 2022). Several representations are commonly used in toxicity prediction models:

1. **SMILES (Simplified Molecular Input Line Entry System)**: SMILES is a text-based format that encodes a molecule's structure using atomic symbols and bond connectivity. This is the input for generating various molecular descriptors, including 2D and 3D features (Lai et al.,2022).

2. **ECFP (Extended Connectivity Fingerprints) and Mordred descriptors**: ECFPs are a widely used method for encoding molecular structures. These fingerprints explore the topological space around atoms in a molecule by iterating through substructures, then convert them into a fixed-length bit string. For example, a 2048-bit ECFP4 represents sub-structural connectivity up to a radius of four atoms around each center. Mordred descriptors is a different way to capture the molecular characteristics of interest of a molecule, using the Mordred software proposed by Moriwaki et al, 2018.

3. **Molecular Graphs**: In graph-based representations, molecules are modeled as graphs, where atoms are nodes and bonds are edges. This allows more flexible and detailed encoding of molecular structures. Graph convolutional networks (GCNs) process these graphs by assigning specific attributes to atoms and bonds, such as atomic identity or bond order, finding more intricate relationships within the molecular structure (Feinstein et al., 2021).

4.2 Data processing

In addition to different techniques to capture the chemical properties of molecules, the datasets used in this paper varies. As *in-vivo* acute oral toxicity measurement of PFAS is limited, and in order to address the lack of PFAS toxicity, the authors constructed an expanded dataset. **LDToxDB**, which has 13,329 unique compounds (toxic substances) of any type with oral rat LD_{50} measurements aggregated from the EPA toxicity estimation. It included all PFAS and PFAS-Like compounds.

LDToxDB-PFAS-like database looks at the LDToxDB, highlights molecules with two or more C−F bonds, but are not confirmed as actual PFAS. As these compounds with two or more C−F bonds could be polyfluorinated, they may not yet be designated as actual PFAS in various databases. This database has only 519 compounds, referred to as LDToxDB-PFAS-like.

LDToxDB-PFAS (Pfas8k): contains 8,163 PFAS compounds that were extracted from the EPA DSSTox database, most of which do not have a LD_{50} number. Only 58 of these, have known toxicity levels. This 58 are present in the LDToxDB database, and in the LDToxDB-PFAS-like database.

5 Results of the original paper and replication attempts

The dataset is divided into training and validation sets using a five-fold cross-validation approach, testing both random and stratified splits.
The authors report results as the mean accuracy across all folds. For classification into EPA categories, the model first predicts the negative log

Model	Input	Reported Acc.	Reported R2	Reported MAE	Reported RMSE	Repl. Acc.	Repl. R2	Repl. MAE	Repl. RMSE
DNN	Mordred	0.680	0.658	0.342	0.516	0.517	0.185	0.591	0.787
DNN	2048-bit ECFP	0.644	0.611	0.385	0.549	0.6402	0.611	0.390	0.549
GCN	Graphs	0.641	0.623	0.380	0.541	0.575	0.453	0.468	0.651
GP	10 Mordred, 200-bit ECFP	0.650	0.627	0.376	0.538	0.662	0.644	0.360	0.524
RF	Mordred	0.660	0.647	0.372	0.523	0.646	0.633	0.379	0.533
RF	4096-bit ECFP	0.623	0.584	0.410	0.569	0.596	0.524	0.448	0.607

Table 1: Reported performance vs replication

of the LD50. These predictions are then assigned to bins according to EPA classification standards. The results are presented in table **??**, where the random splitting was used for cross validation.

5.1 Graph Convolutional Neural Networks

The GCN proposed by the original authors consists of five convolutional layers with a convolutional base of 64, two multilayer perceptrons, a dropout rate of 0.153, a learning rate of 0.008, and a batch size of 64.

Before being input into the model, each molecule is converted into a graph:

- Edges represent bond type, conjugation, and whether the bond is part of a cyclic ring.
- Nodes represent atoms and include features such as atom type (S, Si, F, O, C, I, P, Cl, Br, N), degree, number of hydrogens, implicit valence, formal charge, number of radical electrons, hybridization, and aromaticity.

The results of the replication attempts are summarized in table 1.

5.2 Random Forests

The ensemble model used in the original paper is a random forest for regression problems. The model was fine-tuned to get the most optimal hyperparameters, with number of estimators of 4096, maximum tree depth 32, maximum sample splitting of 2, and minimum samples leaf 1.

In the replication attempt, adjustments to the model were made to explore potential improvements in performance. In particular, the number of estimators(the trees) were reduced from 4096 to 1000, as well as the max tree depth was reduced from 32 to 20. The authors were then able to assess the model's behavior under different configurations and to investigate whether these modifications could change the model behavior significantly. The results of running the model with different input features were highly similar to the ones validated by the original study as shown in table 1.

5.3 Gaussian Process

The Gaussian Process (GP) model demonstrated its effectiveness by using a specialized setup that utilized pre-trained Random Forest (RF) models as feature selectors. These RF models were trained on Mordred and ECFP features and served to identify the most relevant descriptors for toxicity prediction. By selecting the 10 most important Mordred descriptors and 200 ECFP bits based on RF Gini importance, the GP model was able to focus on a reduced feature space, simplifying the learning process and reducing computational complexity.

Why Dimensionality Reduction?
Reducing dimensionality in the GP setup was critical to avoid overfitting and ensure computational efficiency. The GP model's kernel-based structure is highly flexible but computationally expensive, with complexity increasing significantly as the number of features grows. Feature reduction using RF selectors ensured that only the most informative features were used, allowing the GP model to maintain its probabilistic strength while handling the dataset effectively.

In the replication attempt, an alternative setup was explored for the GP model by bypassing dimensionality reduction and utilizing the entire feature set. This approach provided the GP model with access to all Mordred descriptors and ECFP bits, preserving potentially subtle but important relationships that may have been lost

during feature selection. While this adjustment increased computational cost, it resulted in a slight improvement in accuracy compared to the original setup (as shown in Table **??**). This suggests that for certain datasets or models, preserving the full feature set may be advantageous, particularly when the relationships between features and outcomes are complex or not well-understood.

5.4 Deep Neural Networks

While DNNs, like other ML architectures are data intensive processes, the identification process of the molecular properties, that relies on nonlinear interactions makes this architecture on of interest to investigated potential chemical features to toxicity levels. As DNNs are layered, and can enable hierarchical learning, this class of ML architecture was excel at modeling these nonlinear relationships (Wiercioch and Kirchmair, 2023). DNNs are also adaptable to heterogeneous formats, such as SMILES, strings, molecular finger prints and graph based representations.

5.5 Differences in results

After several attempts, we were unable to fully replicate Feinstein et al. (2021) study. With no change to the code provided by the authors through their GitHub repository AI4PFAS, alongside necessary libraries upgrades to prevent deprecation errors (*full library set in Annex H*), The replication results varied from the original study. Small changes, such as replacing the function `mean_squared_error` to `root_mean_squared_error` were performed because the current **sklearn** library has deprecated some of the functions used in the original study.

Additionally, RDKit, the library used to convert SMILES strings to molecular representations, failed to handle some molecules during the replication process. These failures were inconsistent between local machines and the student cluster server, introducing further challenges. To address this, molecules that could not be successfully converted to RDKit objects were dropped from the dataset to ensure consistent processing across all environments.

Almost all the ML computations performed lower than was reported by (Feinstein et al., 2021). In the original study, the best-performing model was the DNN with Mordred descriptors. However, during the replication studies, the *GP model emerged as the best performer with an accuracy of 0.662*. This is likely due to its probabilistic nature and its ability to handle noisy and scarce data. Additionally, its kernel-based structure enables excels at modeling complex, non-linear relationships; an advantage over simpler methods like RF and data-hungry models like DNN and GCN. The RF model with mordred descriptors performed similarly to the original study. These models are inherently robust and perform well with smaller datasets, which may explain why they outperformed neural network-based models under the replication conditions.

The data-intensive ML architectures, DNN and GCN, require extensive data for training. The big surprise of the replication attempt was the underwhelming performance of the DNNs, in particular the DNN with mordred descriptors. While it is hard to be certain about why this the case the following are plausible explanations.

The DNN_ECFP had only one hidden layer but 2048 neurons, with DNN_Mordred had four hidden layers with 256 neurons. It also used a multilayer perceptron which treats all input features independently. In general, neural networks with limited depth, even with many neurons, can struggle to learn hierarchical (complex features) effectively.

In addition, the data intensity of these ML architectures, a few authors ((Tahıl et al., 2024), (Clark, 2011)) have suggested that 'SMILES may do poorly with stereochemistry, as it does not inherently validate stereochemistry'. As such, it becomes possible to encode invalid or chemically impossible configurations. This makes the downstream process error-prone. This is demonstrated with a coincidental discovery a molecule the authors found in the dataset, but one that chemists confirmed is an impossible configuration. This notwithstanding, this molecule was in the database, and had an actual NegLog50 which raised the question of how it was that an impossible molecule has an actual recorded toxicity. As these method require a lot of data to learn, and given the already limited dataset, even a small presence of these types of molecules, reduce the ability for the neural networks from learn.

Overall, the replication attempt yielding worse results compared to the original study was unexpected. These results highlight the challenges of replicating machine learning experiments, where

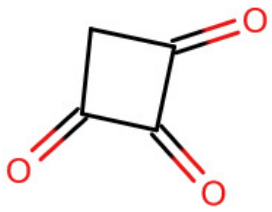


Figure 3: Impossible molecule

differences in data handling, model training, and computational resources can significantly impact outcomes.

However, investigation of these combination of things is no indication of remedy for the problem, and by reference a solution. These findings motivated a deeper dive into the dataset's structure and properties, which is discussed in the next section.

6 Exploring reasons behind a failed replication attempts

From the onset of the project, it was assumed that the publicly available documentation of the data analysis carried out by Feinstein et al. (2021) was sufficient to arrive at their conclusions. The failed attempt led the project to pivot to investigation the reasons behind the outcome. This led to further investigation of the data.

The first step was to assess the similarity between the training and validation datasets, based on the hypothesis that a random split might have caused significant differences between them.

Tanimoto similarity scores (a measure of structural similarity for molecules) were calculated, revealing that 2440 molecules have a similarity of 1 with another, approximately 18% of the samples of the full dataset (pre-split) had a similarity score of one within the same dataset. This prompted a closer examination of these seemingly identical molecules to understand their characteristics better.

6.1 Investigation of Molecules in LD5Tox database

A closer examination of the data revealed that some molecules were recorded with stereochemistry while others were not, suggesting the possibility that they represented the same molecule. Comparing the negative log of the LD50 values for molecules with a Tanimoto similarity of 1 showed that most of these values were close to zero, as shown in figure 4, indicating minimal differences, which can be attributed to slightly different calculation results.

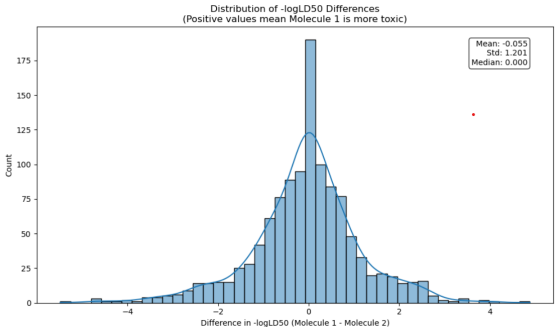


Figure 4: Distribution of -logLD50 Differences

In addition, different toxicity values among structurally identical molecules suggest distinct biological activities, and including these duplicates could inappropriately bias model training by overweighting certain structural patterns. This is supported by the scatterplot in Figure 5, which shows the correlation between toxicity values of pairs of molecules with Tanimoto similarity of **1**. The points are dispersed around the red dashed diagonal line (representing perfect correlation), indicating a weak correlation between the toxicity values of these identical molecules.

The calculated Pearson correlation coefficient was 0.195, with a p-value of 0.005, suggesting a weak but statistically significant positive correlation. This implies that while there is a slight tendency for structurally identical molecules to exhibit similar toxicity, other factors such as biological activity may contribute to variability, further supporting the treatment of these molecules as functionally distinct entities.

Since a detailed analysis of molecules recorded with and without stereochemistry was beyond the scope of this work, it was decided to simplify the preprocessing step by removing stereochemistry from the SMILES representations. This resulted in duplicate molecules, of which only one instance was retained. This led to a drop of 15.53% of the dataset, for which the overall distribution of the EPA classes and the PFAS did not seem thoroughly affected as show in fig. 6.

After processing the removed data, the number of compounds was reduced from 13,329 to 12,237. The dataset was thereby split into a training set of size 9838, and a hold-out test set of

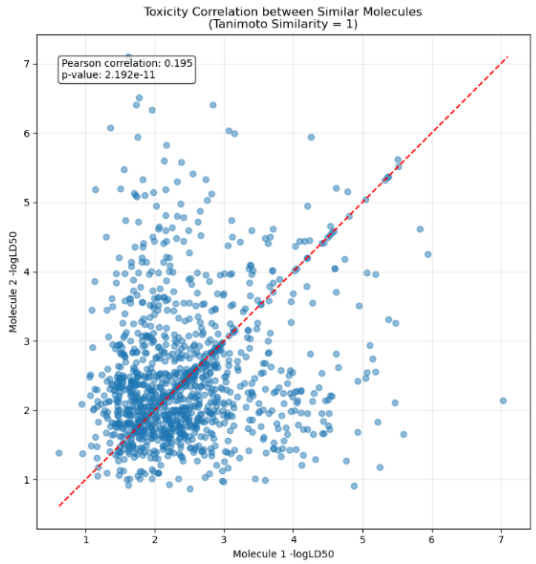


Figure 5: Toxicity Correlation between Similar Molecules

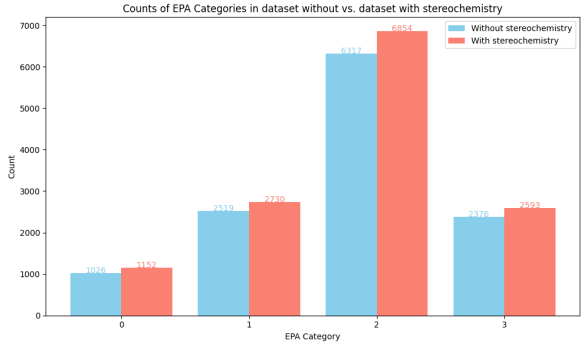


Figure 6: Count of EPA categories before and after stereochemistry removal

2399, with 443 and 50 pfas-like compounds respectively. While the training set was further divided into train and validation sets during cross validation, the hold-out test set was used to assess the performances(i.e. generalization) of the models.

6.2 Evaluation on modified dataset

The performance of the models were evaluated using a 5-fold cross-validation strategy, testing both random and stratified splits. The results are thereby used to compare across different models and with the results of the original study. After evaluating the models, the models were retrained on the train set with the selected hyperparameters, which were subsequently evaluated on the hold out test set.

6.2.1 RF

The Random Forest(RF) regression model using both mordred descriptors and fingerprints of 4096 bits as input features. Given that during replication attempts, using 1000 estimators and maximum tree depth of 20 in the replication, similar results were achieved, for this scenario on the modified dataset, a setup with 1000 estimators, a maximum tree depth of 32, a minimum samples split of 2, and a minimum samples leaf of 1 was used in training the model. As a similar performance was achieved by the replication effort, where the number of estimators was reduced from 4096 to 1000, the hyperparameter was therefore kept for further exploration on the revised dataset. This was done to increase the computational efficiency for handling high-dimensional data like ECFP and Mordred descriptors during the training phase, without significantly sacrificing predictive accuracy.

6.2.2 GP

The Gaussian Process Regression model has a similar setup as in the replication attempt. For each fold in the cross-validation process the pretrained RF models for Mordred and ECFP features were loaded, and their learned weights were used as feature selectors for the GP model. The model utilized feature selection by taking all available features (mordered and ecfp) from the RF model without performing any dimensionality reduction. The predictions were later generated on the test set and saved for evaluation.

6.2.3 GCN

The graph convolutional network(GCN) was assessed on two types of tasks: regression, using the mean squared error (MSE) loss following the original authors' design, and classification, using cross-entropy loss to assess whether a different loss function would help the neural network get better results. A combination of hyperparameters was tested, including learning rates of 0.01, 0.008, and 0.001, and feature dimensions of 50 and 100. The combination of a learning rate of 0.008 and 100 dimensions was the reported hyperparameters for the study of (Feinstein et al., 2021). The other specifications follow those of the original study.

Regression Task: For this task, the results demonstrated that the choice of learning rate and feature dimensions had a noticeable impact on performance:

- Learning Rate: A learning rate of 0.001 (smaller than the original study) outperformed both 0.01 and 0.008 in average, achieving less variance in the accuracy prediction and lower MSE values. Both the learning rates of 0.01 and 0.008 occasionally led to instability in training. This points to either a noisy dataset or a complex relation to be modeled.

- Feature Dimensions: Using 100 dimensions yielded better results compared to 50 dimensions whenever the learning rate was also smaller, this is likely to be due to complex molecular structures being better explained with 100 dimensions and losing important information when compressed into 50; this could also point at the gradients varying across layers.

Classification task: In the classification task, the model's accuracy was evaluated using cross-entropy loss. The results highlighted clear trends in hyperparameter effects:

- Learning Rate: The learning rate of 0.001 again performed better with less variance around the accuracy mean. Both 0.01 and 0.008 performed slightly worse, with both showing occasional overfitting. This could be explained as evidence of the model needed smaller steps for the loss function find the lowest point.

- Feature Dimensions: Similar to the regression task, 100 dimensions slightly outperformed 50 dimensions in overall accuracy.

Both the regression and classification task performed similarly, with s smaller advantage of the regression GCN predicting the EPA class through the toxicity value prediction. This could be due to the classification results having small differences in the probabilities for each result; the confusion matrix offers insight into both models predicting often to the adjacent class. The regression model could be capturing more details for the predictions.

Contrary to expectations, the random split had slightly better overall results than the stratified split; this improvement underscores the importance of stratification to ensure class distribution parity between folds. It could be an effect of the random split models being better at predicting the majority class than the minority ones and so achieving more accuracy. However, stratified splits for the regression and classification tasks achieved more consistent accuracy across folds. The performance comparison is available in Annex 3.

The loss plots for the classification GCN indicate that the model begins to overfit fairly quick, around 25 epochs. In contrast, the regression GCN displays spikes in its loss that do not appear to follow a clear pattern. The relative stability of the classification GCN could be attributed to the nature of the cross-entropy loss, which is more forgiving to certain errors. On the other hand, the regression GCN might be more sensitive to issues such as a non-normal distribution of the target data, making it more prone to instability (See section B in the Annex).

Given the consistent performance across folds, the tasks for regression and classification were measured against the test data with a learning rate of 0.001 and 100 dimensions. The model was initialized with random weights for three different iterations each to check for performance. The model shows overall stability and a better performance of the classification task on the unseen data which is expected do to easier boundary decisions imposed in classification tasks. For both tasks, the results show a large standard deviation as shown in Table 4 and Table 5 in Annex C.

Given the tendency of the models to classify to the nearest class instead of the actual class, a voting system was put in place for the regression and classification tasks by computing the mode of the iterations per task. This increased the performance by two percentage points for regression and one percentage point in classification as show by Fig. 16 in Annex D.

6.2.4 DNN

For the study, the DNNs used linear units and a nonlinear activation functions. The neurons are stacked into sequential layers, which formed a multilayer perceptron (MLP).

The DNN using mordred descriptors: *Four hidden layers with 256 neurons*, *a batch size of 256*, using the *Adam optimiser*, with a 0.01 learning rate, and batch normalisation between every layer.

For the DNN using ECFP descriptors: *one single hidden layer with 2048 neurons*, with a *batch size of 512*,also with the *Adam optimiser*, and a *learn-*

ing rate of 0.001.

While Annex G below details the full results of the approach taken under DNNs for this study, a highlight of the approach now follows. For the modified data, similar architecture was used for the different finger print expressions. One used a cross validation process during the training phase, before using the pre-trained model for prediction, while the other simply trained and made predictions. Both approaches made prediction on the blind test set,for an unbiased view on the generalisability of both approaches. Training losses guided the epoch choice, settled from a point where stability persisted, so while the initial study used 1000 epochs, this attempt used 500. While no approach was glaringly better than the original paper's attempts, the simple predict, and fit fared even worse.

6.2.5 Results of cross validation

Looking at these EPA Classification Accuracy box plots 7, it is seen that simpler models (Gaussian Process and Random Forest) consistently outperform the more complex deep learning models across both random and stratified sampling methods. GP models using combined features and RF models using Mordred descriptors achieve the highest accuracy, while SMILES-based features consistently show the lowest performance. While random sampling shows wider performance variance, stratified sampling produces more consistent results.

6.3 Results on test dataset

The results of the chosen models on the test dataset go in line with the results from the cross-validation efforts as shown in table 2 and confusion matrices in Fig. 8. The best performing models are GP using both mordred descriptors and ECFPs, followed by the RF with mordred descriptors.The DNN using ECFP descriptors also performed well. Amongst the lowest performing model is the GCN regression and DNN Mordred models; the former at 0.585 accuracy and DNN Mordred at 0.498 accuracy.

This enforces the need to find an adequate model for the data that is available; for example this particular dataset might not be nearly large enough to derive meaningful results using neural networks. Additionally, the data might be noisy. The comparison of random and stratified splits revealed additional insights. While stratified splits ensured class distribution parity and consistent performance across folds, random splits slightly outperformed them overall. This discrepancy indicates that the random split may have increased the dominance of majority classes, affecting the model's ability to generalize across all categories.

6.4 Can PFAS-Like Toxicity be predicted?

The performance of the models on PFAs toxicity estimation is worse than the general toxicity prediction. This might be the case because there are less proteins with PFAS structures to train the models with shown by the result on the PFAS resulting in the test dataset. The densities of similarities of the PFAS and non-PFAS molecules in the test dataset to the molecules on the training dataset show, somehow similar distributions but with one higher peak of lower similarities for the PFAS molecules as seen in figure 9 which could be driving the differences in the results.

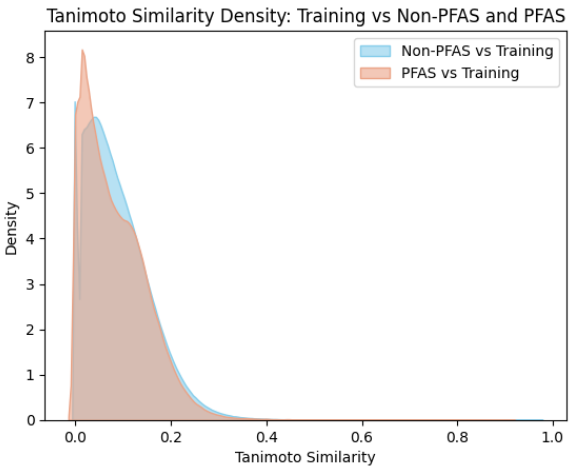


Figure 9: Density of similarities between the PFAS and Non-PFAS in the test data to the training data

Some models such as GCN have as little as 28% accuracy and up to 44%, where the classification model varies the most, having the highest and lowest accuracy (as shown in Table 6) while regression has less variation but equally unimpressive performance (Table 7). These tables are available in Annex E.

To further investigate the potential reasons for the lack of success of the models at predicting toxicity at a general level, the similarity of outliers (defined as predictions that are 4 z-scores away from the actual values for all the models) and of

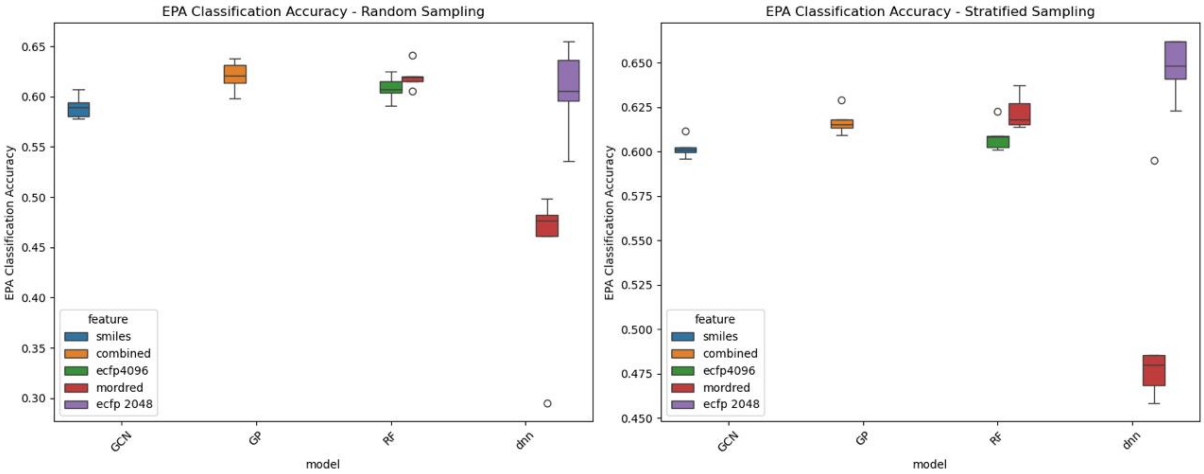


Figure 7: Cross validation comparisons

Model	Input	Accuracy	St. deviation
DNN	Mordred descriptors	0.498	0.056
DNN	ECFP descriptors	0.647	0.016
GCN Regression	Graph	0.585	0.492
GCN classification	Graph	0.605	0.488
GP	Mordred & ECFP	0.628	0.639
RF	Mordred	0.624	0.591
RF	ECFP 4096	0.602	0.598

Table 2: Performance metrics for different models and inputs.

those molecules that were always misclassified for all the models to the training data was compared.

As available in Figure 17 in Annex F, the densities for both groups show the non-outliers having a peak of 0 similarities but overall very similar densities. A similar work was performed in the molecules that were always misclassified (across all models) where the molecules hard to classify have less occurrences of zero similarity compared to the other molecules (Fig. 10).

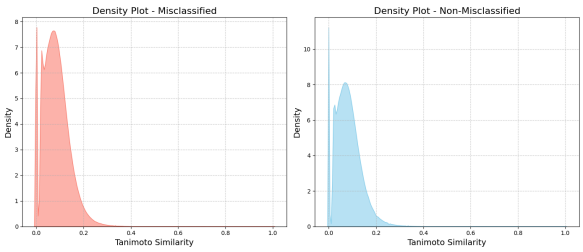


Figure 10: Always misclassified similarities to the training dataset vs the other molecules

In specific, PFAS molecules that were not correctly classified in any model have more zero similarity scores and lower scores overall than the PFAS that were correctly classified at least once, as seen in Fig. 11. This provides evidence of a need for more PFAS in training data to be able to train a better classifiers.

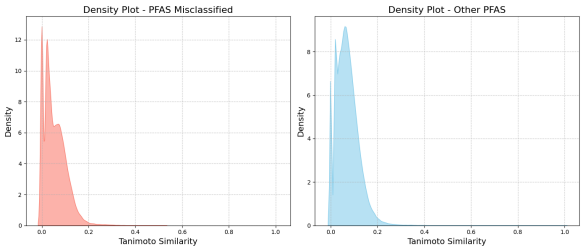


Figure 11: Always misclassified PFAS similarities to the training dataset vs the other PFAS

7 Discussion

The replication study revealed several important insights about machine learning approaches for PFAS toxicity prediction. While the results reported by Feinstein et al. (2021) was not not fully able to be replicated, the investigation uncovered key factors that influence model performance and highlighted important considerations for fu-

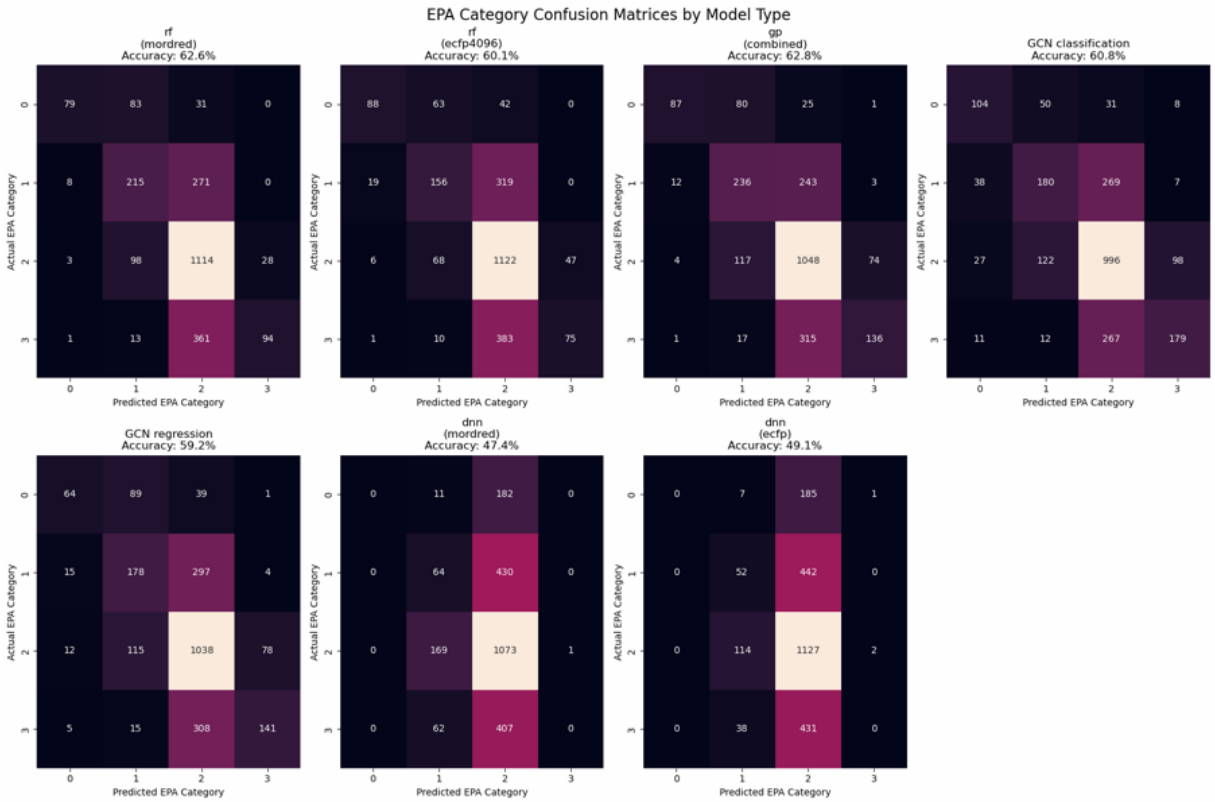


Figure 8: Confusion matrices for all models

ture work in this area. Some of the notable discussions are summarized below:

- The presence of stereochemically identical molecules with different SMILES representations and toxicity values introduced noise into the dataset. The overlap between structurally similar molecules with different toxicity values suggests that additional factors beyond chemical structure influence toxicity.
- The models studied performed notably worse on PFAS compounds compared to general toxicity prediction, as discussed earlier, this could be owing to the fact that there was insufficient amount of data with PFAS structures to be trained on the models. In addition, the analysis of similarity distributions revealed that PFAS molecules in the test set had lower similarity to training data. Furthermore, the presence of chemically impossible configurations in the dataset raises questions about data quality.
- Simpler models (GP and RF) demonstrated more robust performance compared to deep learning approaches when working with limited data.

8 Conclusion

This study makes several key contributions to PFAS toxicity prediction research while highlighting important challenges in the field. Our attempts to reproduce previous machine learning results revealed the critical need for detailed methodology reporting and careful data preprocessing in chemical toxicity prediction. The finding that simpler models like Gaussian Process and Random Forest outperformed more complex approaches suggests that, given current data limitations, these may be more appropriate for PFAS toxicity prediction tasks. The study identified several critical areas for future work, including the need for larger, higher-quality PFAS toxicity datasets, better methods for handling molecular stereochemistry, and specialized architectures for PFAS compound analysis. This research provides valuable insights into both the current limitations and future opportunities in computational PFAS toxicity prediction.

A Annex: GCN hyper-parameter exploration results

Table 3: Results of model across folds

Sampling	Model	Specifications	Accuracy
Random	**GCN classification**	Learning rate 0.001 — 100 dimensions	**0.586703**
		Learning rate 0.001 — 50 dimensions	0.576234
		Learning rate 0.01 — 100 dimensions	0.568914
		Learning rate 0.01 — 50 dimensions	0.584364
		Learning rate as study — 100 dimensions	0.578975
		Learning rate as study — 50 dimensions	0.569730
	GCN regression	Learning rate 0.001 — 100 dimensions	**0.589650**
		Learning rate 0.001 — 50 dimensions	0.581215
		Learning rate 0.01 — 100 dimensions	0.551126
		Learning rate 0.01 — 50 dimensions	0.576029
		Learning rate as study — 100 dimensions	0.583550
		Learning rate as study — 50 dimensions	0.583650
Stratified	**GCN classification**	Learning rate 0.001 — 100 dimensions	0.581725
		Learning rate 0.001 — 50 dimensions	0.581317
		Learning rate 0.01 — 100 dimensions	0.573998
		Learning rate 0.01 — 50 dimensions	0.580201
		Learning rate as study — 100 dimensions	**0.585486**
		Learning rate as study — 50 dimensions	0.580809
	GCN regression	Learning rate 0.001 — 100 dimensions	**0.602155**
		Learning rate 0.001 — 50 dimensions	0.590162
		Learning rate 0.01 — 100 dimensions	0.565361
		Learning rate 0.01 — 50 dimensions	0.578675
		Learning rate as study — 100 dimensions	0.575930
		Learning rate as study — 50 dimensions	0.588838

B Annex: GCN loss plots for best performing model

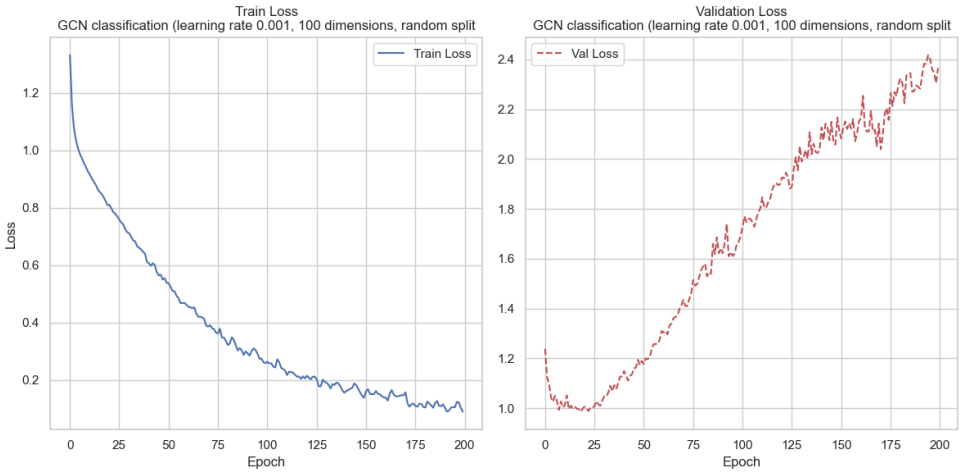


Figure 12: GCN classification loss plots on random split

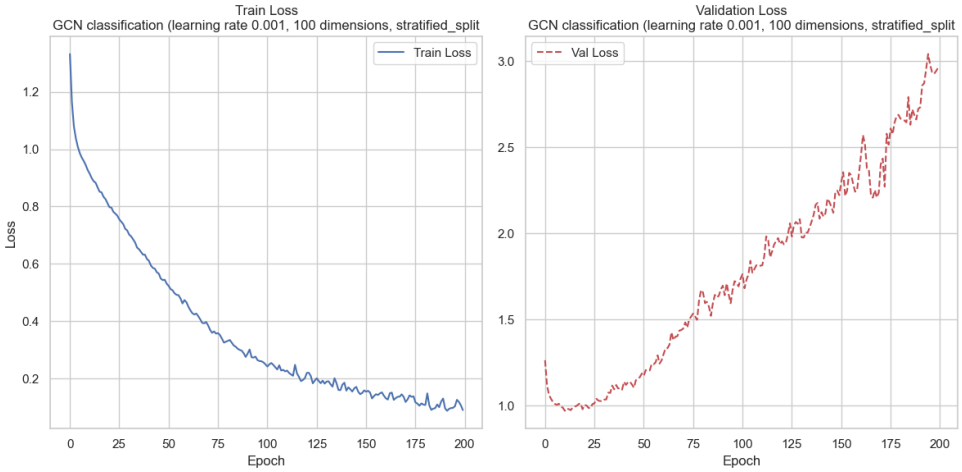


Figure 13: GCN classification loss plots on stratified split

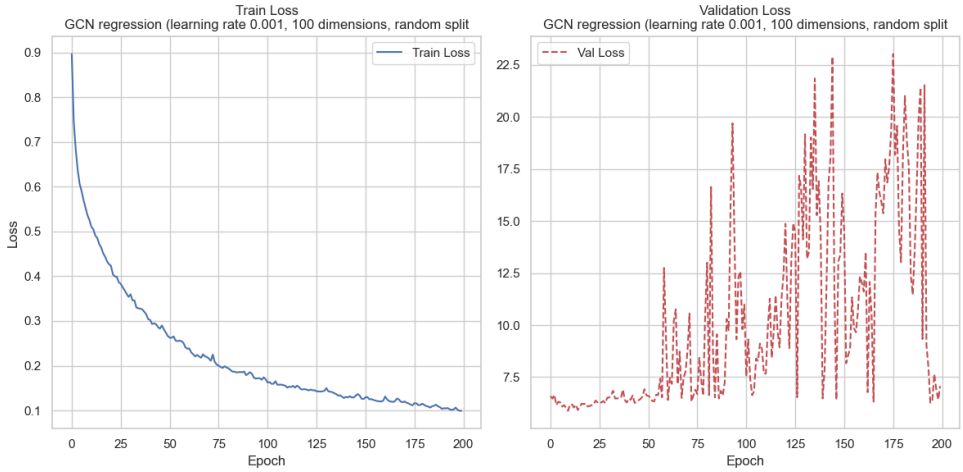


Figure 14: GCN regression loss plots on random split

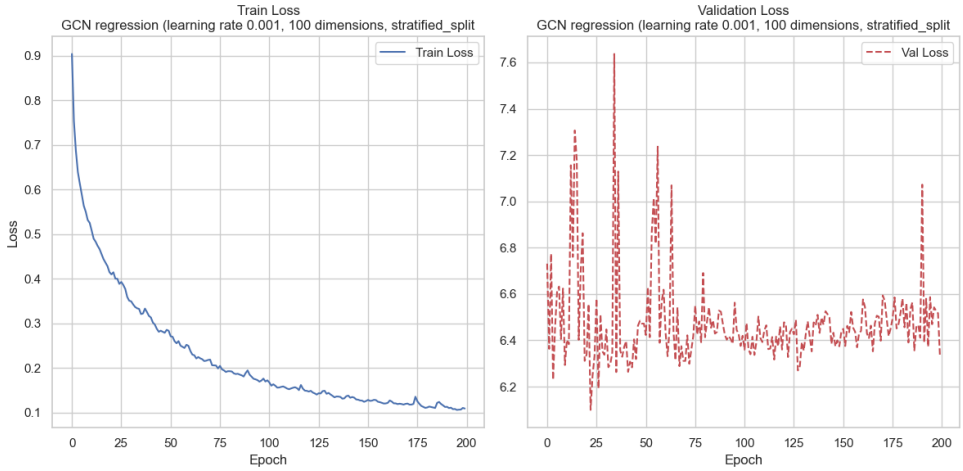


Figure 15: GCN regression loss plots on stratified split

C Annex: Details of the results of the GCN on test data

Table 4: Results of GCN Classification Model on Test Data

Iteration	Av. accuracy	St. deviation
1	0.608170	0.488261
2	0.600667	0.489863
3	0.609837	0.487888

Table 5: Results of GCN Regression Model on Test Data

Iteration	Av. accuracy	St. deviation
1	0.592330	0.491504
2	0.566486	0.495663
3	0.599833	0.490034

D Results of voting system of the GCNs

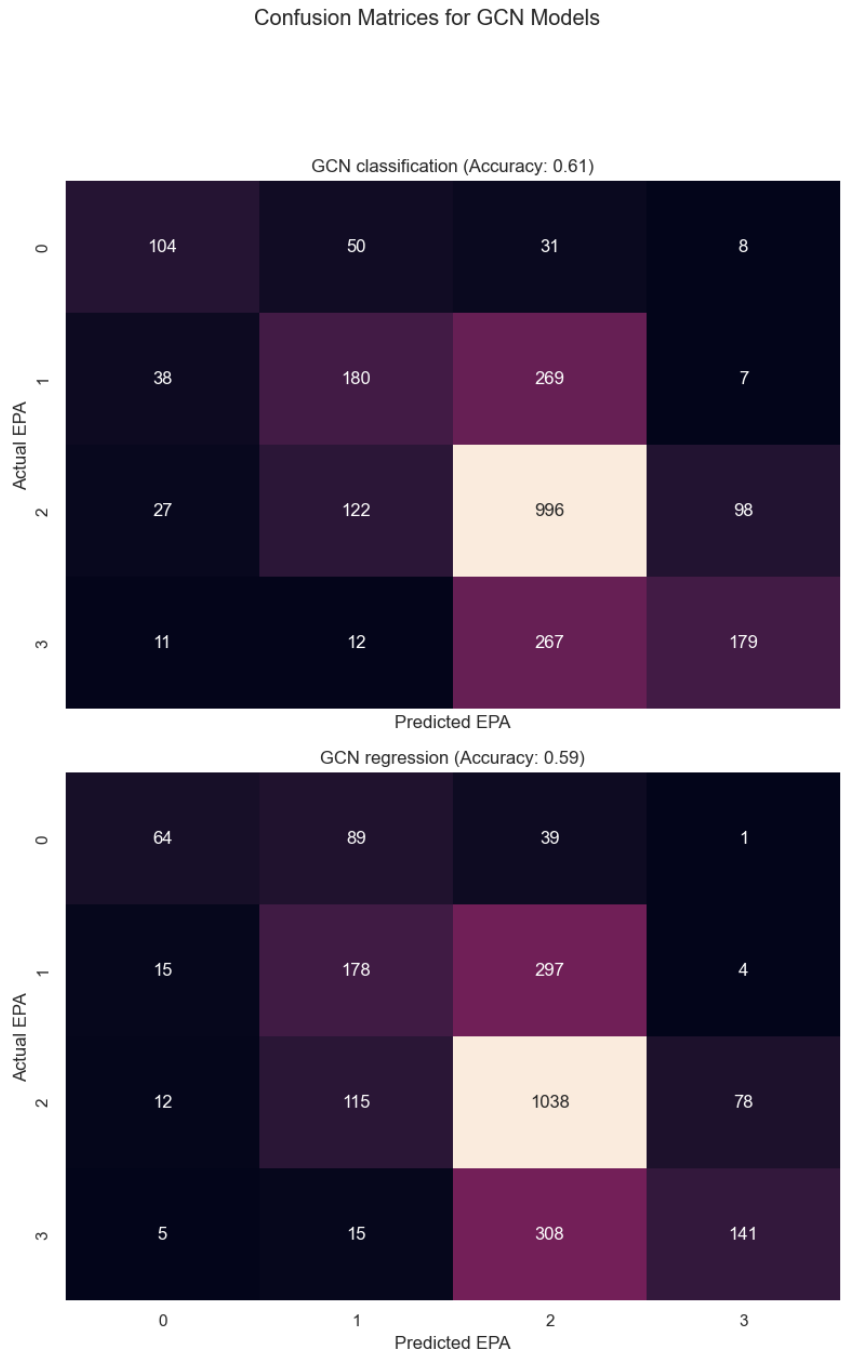


Figure 16: Confusion matrices for the GCN's voting system predictions

E Results of the GCN on the PFAS in test data

Table 6: Accuracy of GCN classification on PFAS

Iteration	Av. accuracy	St. Deviation
0	0.36	0.484873
1	0.28	0.453557
2	0.46	0.503457

Table 7: Accuracy of GCN regression on PFAS

Iteration	Av. accuracy	St. Deviation
0	0.30	0.462910
1	0.34	0.478518
2	0.44	0.501427

F Annex: Outliers similarities to training data

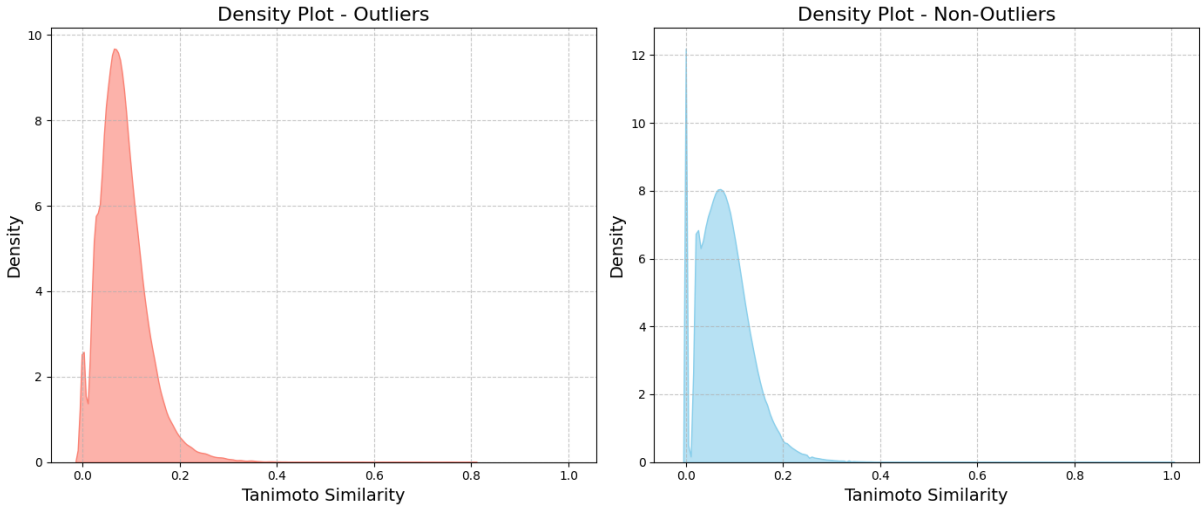
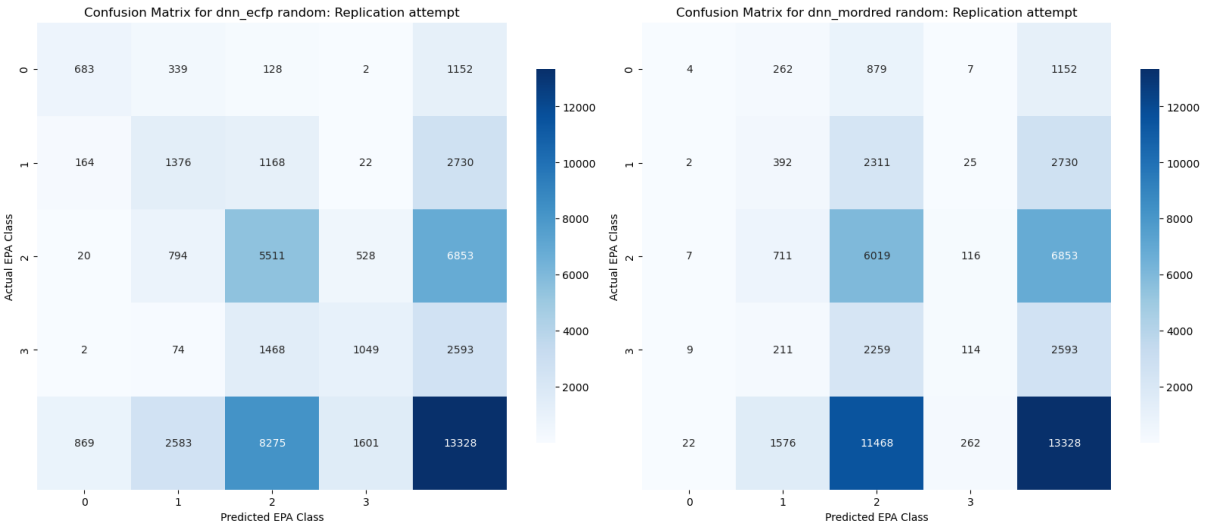


Figure 17: Outliers vs Non-outliers similarities to the training dataset

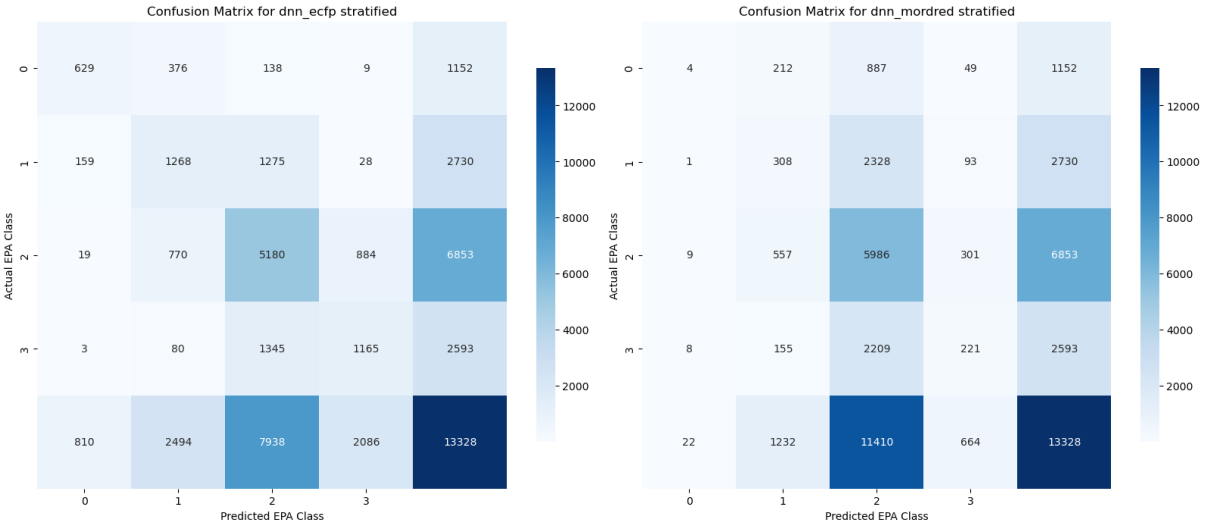
G Annexure: Comparison output for DNN Architecture

When the random approaches were considered. The confusion matrices for both random and stratified sampling approaches were:



((a)) CM: DNN_ECFP ((b)) CM: DNN_ECFP

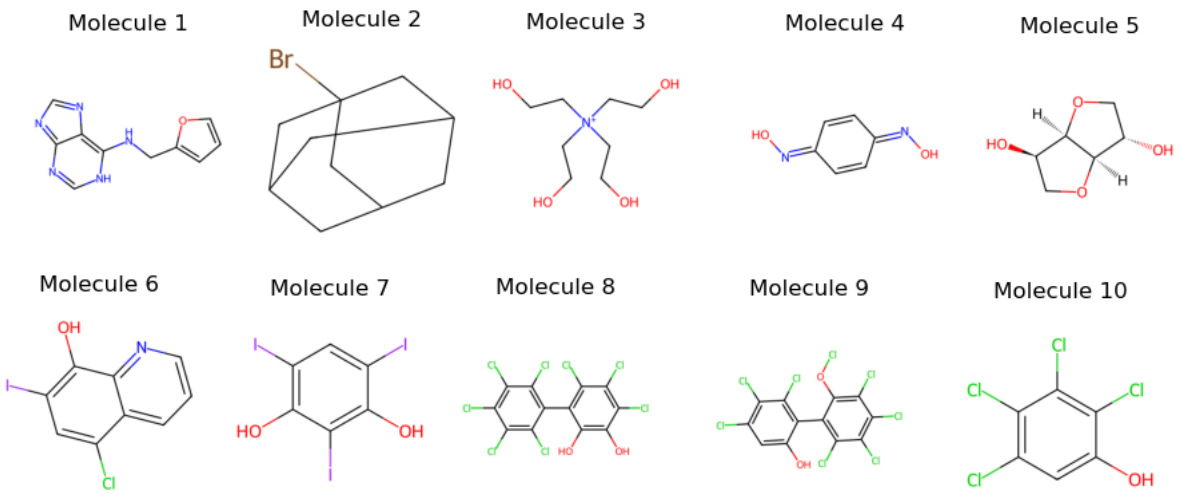
Figure 18: Confusion Matrices: DNNs for random sampling



((a)) CM: DNN_ECFP ((b)) CM: DNN_ECFP

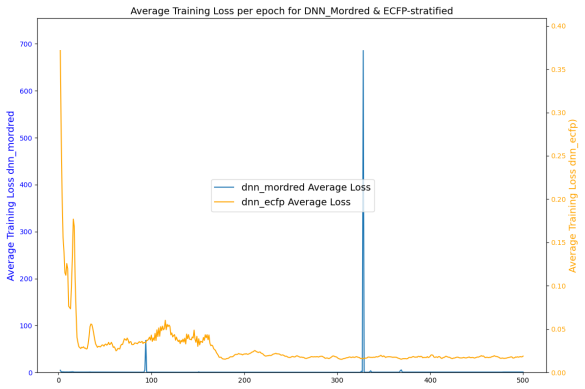
Figure 19: Confusion Matrices: DNNs for stratified sampling

Under the random and stratified sampling, with both ECFP and Mordred models, the top most misclassified molecules overlapped. It is unclear why these molecules were most the most misclassified, without some understanding about chemistry on how this differently these molecules can be put together as the SMILES can result in different smiles for the same molecules. This was however outside the scope of the project, but could be worth further investigate in future research.

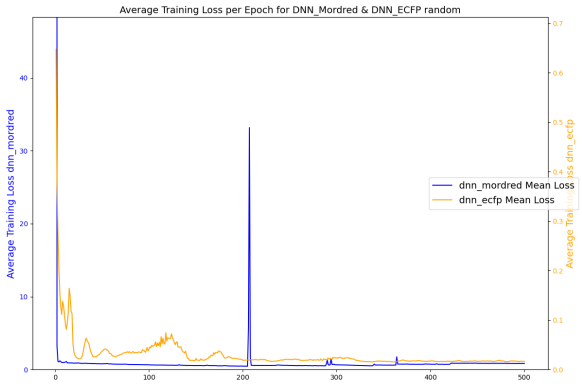


G.1 Training Losses and Epoch choice

Under this sampling approach the training losses for random and stratified approaches are detailed below



((a)) Avg Training loss per epoch: DNN_Mordred and ECFP - Stratified



((b)) Avg Training loss per epoch: DNN_Mordred and ECFP - Random

While the study was unable to explain the why after several hundred epochs, and particularly in the dnn_mordred models, the losses spike before returning to normal, it was clear that after 400 epochs, these spontaneous spikes ceased. As such for the rest of the attempts, *number of epoch 400* was chosen. Moreover, under the replication attempt, assessing the various performance indices, indicated that appeared to be a relatively dense with a relatively tight range of the accuracy across the different folds (under their cross-validation approach). This instability in losses, particularly for the Mordred descriptors, could explain why they are they are the worst performaning when of all the models, despite the higher performance of the ECFP descriptions under the same model architect.

G.2 Alternative Attempts to improve results

The first attempt by the original authors, applied the cross validation on the entire dataset. This meant that they failed to keep apart a holdout test set, for their predicted model. The prediction could not be a blind one as their model had seen every data point, once the training was complete.

Under efforts to improve the results, section 6 in the main text describes changes to the dataset to improve toxicity predictability of PFAS-like compounds, and to assess the generalisability on truly blind data. Two approaches used are below and their summary follows:

G.2.1 Training Losses for Simple-split model (using 80-20 split)

The Training losses under this simple approach using the modified data, while within a tighter range, still have the problem of instability, (spike), after several epochs, before then settling again. Similarly as while recreating the authors approach, this instability only occurred during for DNN using mordred descriptors. Subsequently the resulting model from the training was used to predicted the holdout test set and the result of which are illustrated below

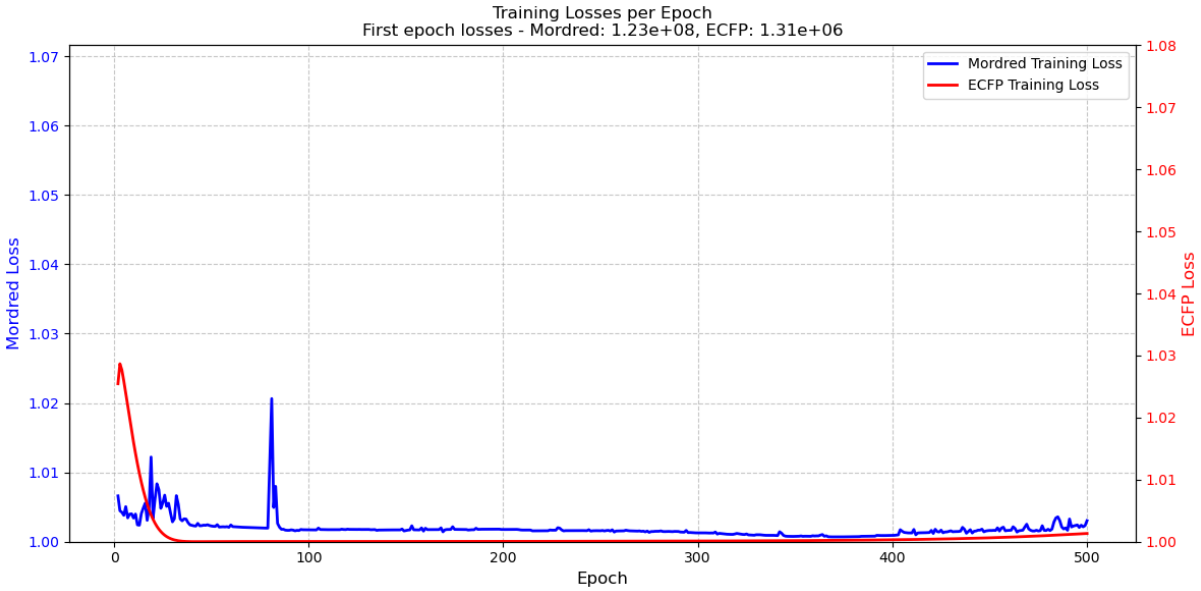


Figure 21: Simple-Split Training Losses

Looking at the prediction power of the model, the Mordred had an accuracy of 0.474 and the ECFP and accuracy of 0.491. When comparing their 10 most *misclassified* molecules, we find there is in no overlap there either. The figures below demonstrate this. ——————————

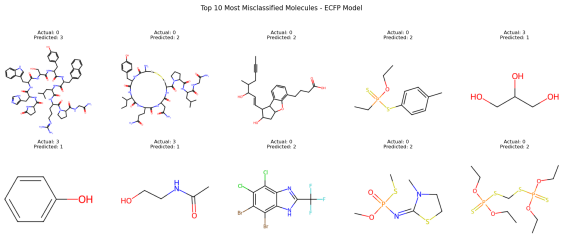


Figure 22: Misclassified on ECFP descriptors

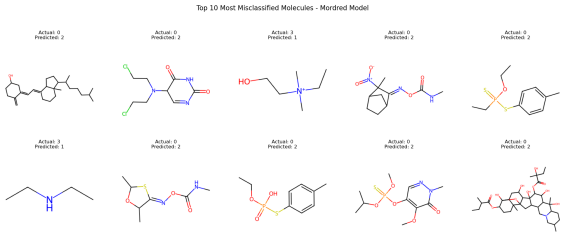


Figure 23: Misclassified on Mordred descriptors

G.2.2 5-Fold Cross-validation-predict with 80/20 split

The Training losses under the 5 fold cross-validation approach using the modified data, exhibits a similar pattern to the above-mentioned approached: the Mordred descriptors exhibit the most unstable training losses, albeit later stabilising after the spike.

When using 5-fold cross-validation to train a model, in actuality, one estimates 5 models. As such, there are several options for making predictions on a blind test set. However, the commonly and generally recommended approach is to train a final model, done after cross-validation is complete. The final model is trained using ALL of the training data with the best hyper-parameters found during cross validation. The losses during the initial 5 Fold cross-validation training, and the final model with the best of the hyper-parameters from the cross-validation is illustrated below.

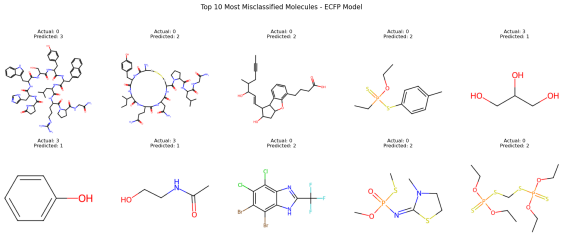


Figure 24: Cross-validation losses

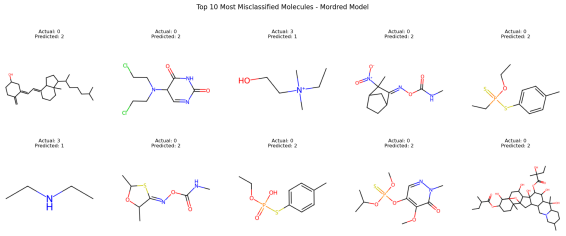


Figure 25: Final Model Losses-from Cross validation

Subsequently the resulting model from the training was used to predicted the holdout test set and the result of which are illustrated below

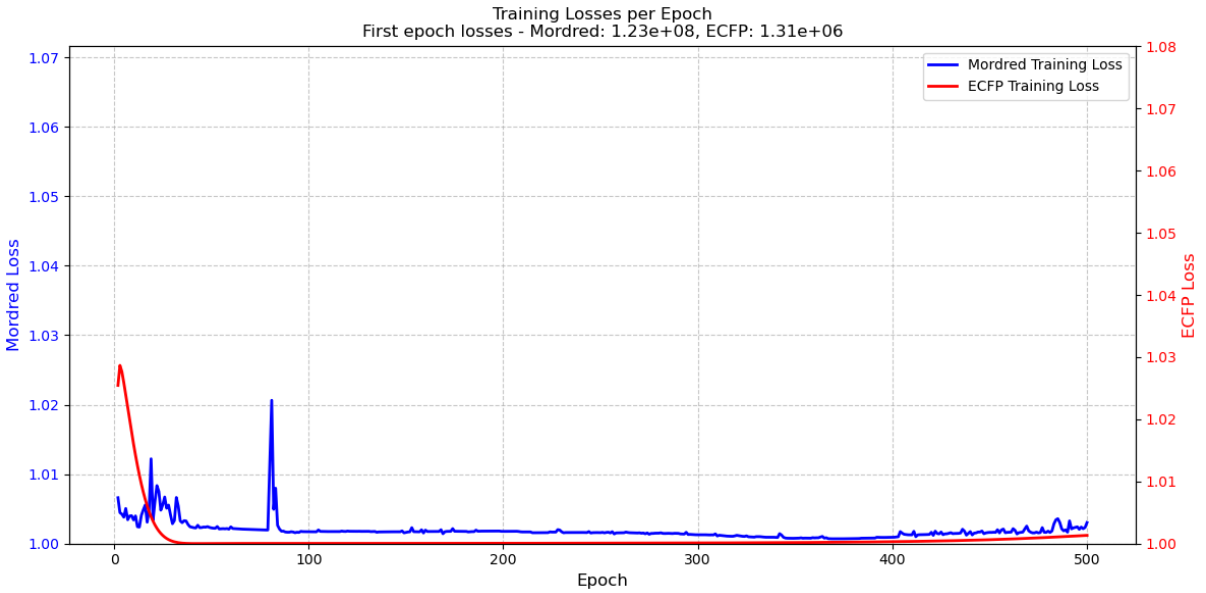


Figure 26: Simple-Split Training Losses

Looking at the prediction power of the model, the Mordred had an accuracy of 0.498 and the ECFP and accuracy of 0.647. When comparing their 10 most *misclassified* molecules, we find there is in no overlap there either. The figures below demonstrate this.

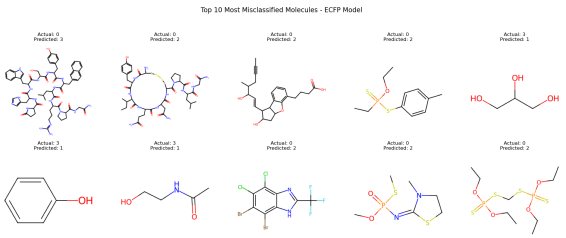


Figure 27: Misclassified on ECFP descriptors

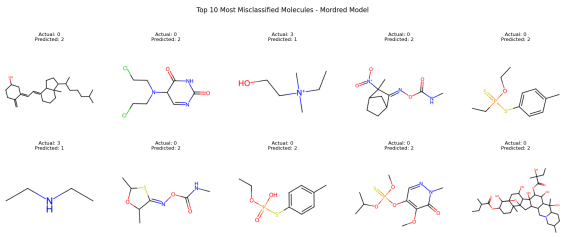


Figure 28: Misclassified on Mordred descriptors

H Annex: Libraries used for replication attempt

```
Pyhton version 3.11

Libraries:
absl-py==2.1.0
aiohappyeyeballs==2.4.3
aiohttp==3.10.8
aiosignal==1.3.1
anyio==4.6.0
argon2-cffi==23.1.0
argon2-cffi-bindings==21.2.0
arrow==1.3.0
asttokens==2.4.1
astunparse==1.6.3
async-lru==2.0.4
asyncio==3.4.3
attrs==24.2.0
autograd==1.7.0
babel==2.16.0
beautifulsoup4==4.12.3
bleach==6.1.0
certifi==2024.8.30
cffi==1.17.1
charset-normalizer==3.3.2
check-shapes==1.1.1
cloudpickle==3.0.0
colorama==0.4.6
comm==0.2.2
contourpy==1.3.0
cycler==0.12.1
debugpy==1.8.6
decorator==5.1.1
defusedxml==0.7.1
Deprecated==1.2.14
disutils==1.4.32.post2
dm-tree==0.1.8
dropstackframe==0.1.1
executing==2.1.0
fastjsonschema==2.20.0
filelock==3.16.1
flatbuffers==24.3.25
fonttools==4.54.1
fqdn==1.5.1
frozenlist==1.4.1
fsspec==2024.9.0
gast==0.6.0
google-pasta==0.2.0
gpflow==2.9.2
grpcio==1.66.2
h11==0.14.0
h5py==3.12.1
```

```
httpcore==1.0.6
httpx==0.27.2
idna==3.10
ipykernel==6.29.5
ipython==8.28.0
ipywidgets==8.1.5
isoduration==20.11.0
jedi==0.19.1
Jinja2==3.1.4
joblib==1.4.2
json5==0.9.25
jsonpointer==3.0.0
jsonschema==4.23.0
jsonschema-specifications==2023.12.1
jupyter==1.1.1
jupyter-console==6.6.3
jupyter-events==0.10.0
jupyter-lsp==2.2.5
jupyter_client==8.6.3
jupyter_core==5.7.2
jupyter_server==2.14.2
jupyter_server_terminals==0.5.3
jupyterlab==4.2.5
jupyterlab_pygments==0.3.0
jupyterlab_server==2.27.3
jupyterlab_widgets==3.0.13
keras==3.5.0
kiwisolver==1.4.7
lark==1.2.2
libclang==18.1.1
Markdown==3.7
markdown-it-py==3.0.0
MarkupSafe==2.1.5
matplotlib==3.9.2
matplotlib-inline==0.1.7
mdurl==0.1.2
mistune==3.0.2
ml-dtypes==0.4.1
mpmath==1.3.0
multidict==6.1.0
multipledispatch==1.0.0
namex==0.0.8
nbclient==0.10.0
nbconvert==7.16.4
nbformat==5.10.4
nest-asyncio==1.6.0
networkx==3.3
notebook==7.2.2
notebook_shim==0.2.4
numpy==1.26.4
opt_einsum==3.4.0
```

```
optree==0.12.1
overrides==7.7.0
packaging==24.1
pandas==2.2.3
pandocfilters==1.5.1
parso==0.8.4
pillow==10.4.0
platformdirs==4.3.6
prometheus_client==0.21.0
prompt_toolkit==3.0.48
protobuf==4.25.5
psutil==6.0.0
pure_eval==0.2.3
pycparser==2.22
Pygments==2.18.0
pyparsing==3.1.4
python-dateutil==2.9.0.post0
python-json-logger==2.0.7
pytz==2024.2
pywin32==306
pywinpty==2.0.13
PyYAML==6.0.2
pyzmq==26.2.0
rdkit==2024.3.5
referencing==0.35.1
requests==2.32.3
rfc3339-validator==0.1.4
rfc3986-validator==0.1.1
rich==13.9.1
rpds-py==0.20.0
scikit-learn==1.5.2
scipy==1.14.1
Send2Trash==1.8.3
setuptools==75.1.0
six==1.16.0
sniffio==1.3.1
soupsieve==2.6
stack-data==0.6.3
sympy==1.13.3
tabulate==0.9.0
tensorboard==2.17.1
tensorboard-data-server==0.7.2
tensorflow==2.17.0
tensorflow-intel==2.17.0
tensorflow-probability==0.24.0
termcolor==2.4.0
terminado==0.18.1
tf_keras==2.17.0
threadpoolctl==3.5.0
tinycss2==1.3.0
torch==2.4.1
```

```
torch-geometric==2.6.1
tornado==6.4.1
tqdm==4.66.5
traitlets==5.14.3
types-python-dateutil==2.9.0.20240906
typing_extensions==4.12.2
tzdata==2024.2
uri-template==1.3.0
urllib3==2.2.3
wcwidth==0.2.13
webcolors==24.8.0
webencodings==0.5.1
websocket-client==1.8.0
Werkzeug==3.0.4
wheel==0.44.0
widgetsnbextension==4.0.13
wrapt==1.16.0
yarl==1.13.1
```

References

V. Barry, A. Winquist, and K. Steenland. 2013. Perfluorooctanoic acid (pfoa) exposures and incident cancers among adults living near a chemical plant. *Environmental Health Perspectives*, 121(11–12):1313–1318.

E. Benfenati, A. Manganaro, and G. Gini. 2013. Vega-qsar: Ai inside a platform for predictive toxicology. In *CEUR Workshop Proceedings*, volume 1107, pages 21–28.

Leo Breiman. 2001. Random Forests. *Machine Learning*, 45(1):5–32.

Alex M. Clark. 2011. Accurate specification of molecular structures: The case for zero-order bonds and explicit hydrogen counting. *Journal of Chemical Information and Modeling*.

European Chemical Agency (ECHA). 2024. Per- and polyfluoroalkyl substances (PFAS). `https://echa.europa.eu/hot-topics/perfluoroalkyl-chemicals-pfas`. [Accessed on: 2024-09-19].

J. Feinstein, G. Sivaraman, K. Picel, B. Peters, V.-M. Alvero, A. Ramanathan, M. MacDonell, I. Foster, and E. Yang. 2021. Uncertainty-informed deep transfer learning of perfluoroalkyl and polyfluoroalkyl substance toxicity. *Journal of Chemical Information and Modeling*, 61:5793–5803.

Nicolas Fernandez, A. Pouyan Nejadhashemi, and Christian Loveall. 2023. Large-scale assessment of pfas compounds in drinking water sources using machine learning. *Water Research*, 243:120307.

S. Joudan and R. Lundgren. 2022. Taking the "f" out of forever chemicals. the right solvent mix breaks down perfluorinated organic acids. *Science*, 377(6608).

Andreas Mayr, Günter Klambauer, Thomas Unterthiner, and Sepp Hochreiter. 2016. Deeptox: Toxicity prediction using deep learning. *Frontiers in Environmental Science*, 3.

Bhavya Mehta, Kush Kothari, Reshmika Nambiar, and Seema Shrawne. 2023. Benchmarking toxic molecule classification using graph neural networks and few shot learning.

Gökhan Tahıl, Fabien Delorme, Daniel Le Berre, Éric Monflier, Adlane Sayede, and Sébastien Tilloy. 2024. Stereoisomers are not machine learning's best friends. *Journal of Chemical Information and Modeling*, 64:5451–5469.

Jie Wang. 2023. An intuitive tutorial to gaussian process regression. *Computing in Science amp; Engineering*, 25(4):4–11.

Magdalena Wiercioch and Johannes Kirchmair. 2023. Dnn-pp: A novel deep neural network approach and its applicability in drug-related property prediction. *Expert Systems with applications*, 213(X):1–14.

Jinxin Zhou, Xiao Li, Tianyu Ding, Chong You, Qing Qu, and Zhihui Zhu. 2022. On the optimization landscape of neural collapse under MSE loss: Global optimality with unconstrained features. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 27179–27202. PMLR.