

Using Machine Learning Models to Predict the Impact of Chemicals on p53

Group 9: Utkarsh Nattamai Subramanian Rajkumar, Isabelle Du Plessis, Hannah Snyder

May 1, 2023

Abstract

Cancer is one of the most common causes of death worldwide, second only to heart disease [RSR18]. One way cancer has been studied is through the impact that p53’s function has on tumor growth. P53 is a protein family that is known to be a tumor suppressor gene, so studying the function of p53 is one way to gain insight into the likelihood that tumors will develop. Recently, scientists have been using machine learning as a way to learn more about biological problems and questions. This provides a cheaper and quicker alternative to other methods such as animal testing and clinical trials. To learn more about cancer and p53, we have developed two different approaches to classify chemicals that have been shown to have some impact on the function of p53. These approaches include non graph approach (logistic regression, support vector machines, random forest) and graph approach (graphical neural network). Chemical features were derived from Simplified Molecular Input Line-Entry System (SMILES) sequences corresponding to the atoms and structure of each compound. The fraction of CSp3 hybridization, molecular weight, and number of hydrogen acceptors were the features selected to be used with the non graph approach models. Graph representation of SMILES strings were used for the graph approach model. Each non-graphical approach models have a decent AUROC score in the range from 0.75 to 0.79. The graphical approach has an AUROC score in the range from 0.75 to 0.81. Each of these scores is adequate for a machine learning model, however, ideally it would be higher.

1 Introduction and Related Works

1.1 Cancer & p53 Protein

In 2020, there were almost 10.0 million deaths from various types of cancer [SFS⁺21]. Along with this, there were around 19.3 million new cancer cases, increasing from 2018 when there were approximately 18.1 million new cases of cancer [BFS⁺18]. While there are many mutations that cause cancerous tumors to form, mutations in tumor suppressor p53 are found in more than 50% of human cancers [Lev20]. There are no other proteins that have been studied to have such a strong correlation with cancers. Carcinogenic chemicals can cause DNA damage such as strand breaks, chromosome aberrations, and altered DNA repair mechanisms, which can lead to gene mutations and cancer [SGG⁺16]. In order to help decrease the number of cases of cancer around the world, studying the ways that chemicals impact p53 proteins and their functionality is an interesting new approach.

1.2 Lack of Data on Properties of Toxic Chemicals

The U.S. Environmental Protection Agency (EPA) has developed an Integrated Risk Information System (IRIS), compiled of toxicity information including cancer descriptors of only 487 chemicals [iri]. However, the EPA has also designated over 86,000 chemicals as potentially toxic as part of the Toxic Substances Control Act (TSCA) [TSC]. Humans are exposed to potentially hazardous chemicals in virtually all industrialized activities, so it is crucial to address environmentally-induced cancer, as the environment is responsible for 80% of cancer tumors [TRLL09]. Only a fraction of the TSCA chemicals have undergone full characterization and assessment for risk. Expanding the current data on potentially carcinogenic chemicals is essential to progress in cancer prevention and understanding life-threatening mutations. A list of key characteristics for toxic carcinogens has been developed to address the EPA and the National Toxicology Program’s (NTP) concerns [SGG⁺16]. These characteristics include electrophilicity, genotoxicity, and immunosuppressive, however, it is not feasible to experimentally screen for these characteristics on a large scale. Additionally, these factors can interact with each other to affect the level of risk.

It is difficult to experimentally determine these properties, and there is a high risk associated with handling many of these chemicals. Federal programs such as Tox21 have been implemented to push for the development of new methods for assessing chemical toxicity [Pro]. Due to a major lack of information on potentially hazardous substances, computational models are promising in predicting toxicity based on physiochemical properties that affect how substances interact with humans and the environment [ZMW⁺17].

1.3 Machine Learning Approach

Scientists have been using machine learning as a way to predict chemical properties such as toxicity, and one study even discovered that models can be more reliable than previously done animal studies [Har19]. Some common machine learning models that have been studied for predicting toxicology include linear regression, k-nearest neighbors, support vector machines, decision trees, neural networks, and random forest algorithms [Bas00]. Logistic regression models have become very popular for predictive medical data science [Jak06]. Using information about a certain event, a logistic regression model predicts the probability the event will occur or not. Binary logistic regression model is used when there are only two outcomes of the event, which is common during classification problems as well [log]. Support vector machines (SVMs) use algorithms to implement statistical learning. They utilize machine learning to avoid over-fitting and increase accuracy while classifying data [Jak06]. SVMs are known to work well with high-dimensional, and noisy data, which is a common problem when using biological datasets [N+04]. Random forest classifiers are based on a combination of decision trees, created using training data and internally validated. They are very popular for large datasets because it also works well with high-dimensional data and if there are highly correlated predictors [BJKK12].

Another model that is used is a graphical neural network (GNN). A GNN works by creating nodes to represent data and using algorithms to evaluate the relationship between these nodes. By doing so, it uses the topological relationships to create a graph-like structure to give insight into the data [SGT+08].

Quantitative structure-activity relationship (QSAR) models have been used for toxicity prediction and risk assessment. Essentially, these models aim to find relationships between chemical structure and biological activity, and they can be used to classify chemicals that have not yet been tested. QSAR models generally consist of 1) the design of a training set of chemicals, 2) selection of descriptors that relate structure to biological function, and 3) use of statistical methods to find differences in structure that correlate to differences in function [PFTW03]. Machine learning approaches similar to QSAR models can be used to predict the biologically impactful features of structural variation. Different chemical functional groups and their 3D orientations affect whether compounds are harmful or not [KSM+19].

Machine learning approaches have been successfully applied to toxin carcinogenicity prediction of untested chemicals. One study used SVM and ANN models to accurately classify chemicals as genotoxic (directly damaging to DNA) and epigenetic carcinogens (do not directly damage DNA) and noted that more informative descriptors as features may improve accuracy [TRLL09]. Another study aims to recognize structural alerts for genotoxicity using QSAR-like models [Ben05]. They describe how properties like molecular weight, size, solubility, reactivity, and metabolism can be used to predict absorption and carcinogenicity. They conclude that these models allow for the identification of groups of chemicals with significantly different mechanisms of action to be further experimentally validated. To address, the challenge of limited data on known human carcinogens, data-driven, machine-learning, QSAR modeling approaches have been developed to achieve more diverse training sets for model development and prediction of chemical compound properties [CRC+23]. This previous work confirms that applying machine-learning models to classify untested chemicals for carcinogenic properties, such as the impact on the p53 gene, is promising for expanding limited knowledge on toxic chemicals.

1.4 Simplified Molecular Input Line-Entry System Data

When using machine learning models in prediction of chemical properties, there are different ways chemicals can be represented as data. Chemical compounds each have a unique arrangement of atoms connected by bonds, and these structures can be represented using several methods, one being a string sequence format. Simplified Molecular Input Line-Entry System (SMILES) is a chemical notation language developed to create in-line descriptions of chemicals. They are designed to be user-friendly, machine-friendly, and have unique graph structures [Wei88]. Canonical SMILES were developed to account for different notations representing the same chemical compound due to stereochemistry and to standardize chemical sequences for computational use [O2]. SMILES can be used to various chemoinformatics tasks, as hundreds of chemical features can be derived from the sequences, such as the number of atoms, number of bonds, formal charge, hydrogen donors, etc. These 2D numerical features as well as 3D structural features can be obtained from SMILES using libraries such as RDKit (G. Landrum, RDKit: Open-source cheminformatics, 2006.). Different chemical functional groups and their 3D orientations affect whether compounds are actively harmful or not [KSM+19].

SMILES are commonly used in predictive chemoinformatics due to their versatility and applicability to many chemical and biological problems. Features from SMILES have been used to predict many properties, such as druglikeness, which is related to the absorption of chemicals and their activity in humans [KLAL21]. One study utilized SMILES with several methods of feature selection (CFS, Pcorr, Scorr, RF, and LASSO) with different models, aiming to improve understanding of what features are the most successful in predicting quantum chemistry properties without using data with atomic coordinates [PMS+20]. They found that using numerical features from SMILES was nine times more accurate

than 3D representations of molecular structure (Coulomb matrix). Another study classified chemical compounds from the Tox21 dataset based on whether they would bind to proteins or not using SMILES notation with a CNN model [HSK⁺18]. They found that the richer feature space with SMILES-derived features led to higher accuracy than traditional fingerprint methods. Despite no explicit substructure data, the model could learn chemical motifs from the SMILES sequences.

Different representations of SMILES have advantages and disadvantages. One study extracts numerical features from the SMILES strings as well as a molecular image [KSM⁺19]. Comparison of FCNN on the numerical features, CNN on the molecular images, and RNN on vector representation of SMILES strings revealed that the FCNN with numerical features performed best, but the highest accuracy was achieved when combining all three models. However, they note that encoding molecular images for large molecules is more difficult, so this approach likely works best with relatively small compounds.

Additionally, there is value in the interpretability of simple models for toxicity prediction. How the number of SMILES numerical features affects accuracy in toxicity prediction has been evaluated, and they found that their top 3 features played the most critical role in toxicity prediction, significantly more than the top 29 features [KMNS19]. They additionally note that a major benefit of simpler models is that they are less computationally intensive.

Studying the features derived from the SMILES sequence of chemicals that have been shown to impact p53 could lead to preventative measures to reduce human exposure to these toxicants.

1.5 Our Models

To study p53 and its role in tumor growth, we selected a dataset from MoleculeNet which include classification information for 7,831 which have been shown to impact the function of p53 [ZW17]. Of the models we have read about, we used the following models on our chosen dataset: logistic regression, support vector machines, random forest, and GNN. In creating these models, we hope to be able to predict how chemicals impact the function of p53 based on their structure. With this information, there could be limitations placed on the use of these chemicals to reduce human exposure. This would help to decrease the number of mutations in p53, and hopefully help decrease future cancer rates.

After creating each of our models, we evaluated the significance using a variety of quantitative measures. Based on these, each of our models had scores indicating significance and that accurate models were created. Each non-graphical approach model had a decent AUROC score in range from 0.75 to 0.79. The graphical approach had an AUROC score in range from 0.75 to 0.81. Based on these scores, our model’s AUROC is slightly lower than many other biological prediction models using MoleculeNet.

2 Data

We used the Tox21 dataset from the MoleculeNet website. This dataset was curated from Tox21 chemical information and includes quantitative toxicity data for 12 different mechanistic targets and 7,831 chemicals for each of these targets. One of these targets includes p53, which is directly correlated to mutations that lead to cancerous tumors. The below subsections will describe the general as well as the specific approach-based preprocessing steps that were performed for this project.

2.1 General Preprocessing

Since we were interested in knowing the chemical compounds effect on the p53 mechanistic target, we extracted both the smiles and SR-p53 columns from the dataset into a new Pandas dataframe (called srP53Data). Once we created this new dataframe, we noticed that there were several missing values in the SR-p53 column. To overcome this problem, we dropped all the rows with missing values. We thought it was a good idea to drop the missing rows because we felt we cannot make assumptions about missing values especially for this particular scenario. For instance, the chemical compound might have a target label value of 1 instead of a 0 or vice versa. Therefore, we felt it was best to drop the rows with missing values. This action did not affect the overall quality and distribution of the dataset (the number of observations per class was still imbalanced). Figure 1 shows the number of observations for the two target classes (0 and 1).

2.2 Non Graph Approach Preprocessing Steps

The non graph approach mentioned in section four required some preprocessing steps before we could develop our machine learning models. The first preprocessing step was to acquire simple molecular descriptor feature values for each of the

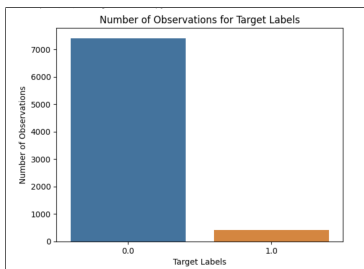


Figure 1: view of the imbalanced number of observations between the two target labels

chemical compounds in our dataframe. To get the feature values, we first retrieved all the SMILES strings from the smiles column of our dataframe. We then referred to [a brief tutorial by Oxford Protein Informatics Group](#) to turn each SMILES string into a vector of molecular descriptors values using functions from the RDKit library. From the different descriptor values, we choose the values for the simple descriptor features for all the chemical compounds. In this process, we referred to the paper [FEG⁺20] to get a list of all the simple descriptor features. According to the paper, these descriptor features were the most commonly used descriptors. This was our justification for using these specific feature values. Once the simple descriptor feature values were retrieved for all the SMILES strings, we performed two steps before feature selection. We first removed the smiles column from the dataframe as we no longer needed it for this approach. We also changed the type of the SR-p53 column (containing the target labels) from float to int. After the two previous steps, we performed feature selection. We thought performing feature selection was important to prevent us from experiencing overfitting issues later during the model development phase. SkLearn’s SelectKBest() function was utilized for selecting the top three features. After applying this function to our data, we saw that the model chose FractionCSP3, MolWt, NumHAcceptors as the top three features. CSP3 is the fraction of carbon bond saturation in a chemical [CMLBF17]. This is a very good indicator of chemical structure because the backbone of most chemicals is pure carbon, so information about the backbone of the chemical can be predictive of related structures. Molecular Weight (MolWt) is one factor that impacts how quickly a chemical is absorbed into the body [DZW⁺11]. This can increase the toxicity of a chemical because how a chemical is absorbed into the body plays a large role in the toxicity of the chemical, and thus how much it impacts the body. The number of hydrogen acceptors (NumHAcceptors) and hydrogen bonds, in general, have been studied to be a very important mechanism for molecule separation in a compound [AJ17]. This process can also be directly related to the rate of absorption of a chemical into the body. After getting the top three features from the feature selection process, we normalized our data using the StandardScaler() method from Sklearn. We thought this was a good idea because we noticed MolWt column values were much larger than the values in the FractionCSP3 and NumHAcceptors columns. Therefore normalizing the dataset would allow the features to be on a similar scale.

2.3 Graph Approach Preprocessing Steps

The graph approach mentioned in section four also required some preprocessing steps. The main goal of preprocessing for this approach was to convert the SMILES string for each of the chemical compounds into a graph representation. Before performing any steps to convert the SMILES string into a graph representation, we shuffled our dataset to ensure the samples were not in any specific order. To successfully convert the SMILES strings into graph representations, we first retrieved all the SMILES strings from the smiles column of our data frame. We then utilized Pytorch Geometric library’s from_smiles() function (part of torch_geometric.utils.smiles module) to convert each SMILES string into an initial graph. We then constructed an updated graph from the initial graph. This updated graph included the specific target label and SMILES string along with all the information in the initial graph for a chemical compound. We also developed our custom PyTorch dataset. This was done because we thought creating our custom dataset would help simplify efforts related to creating dataloaders for later stages of model development process.

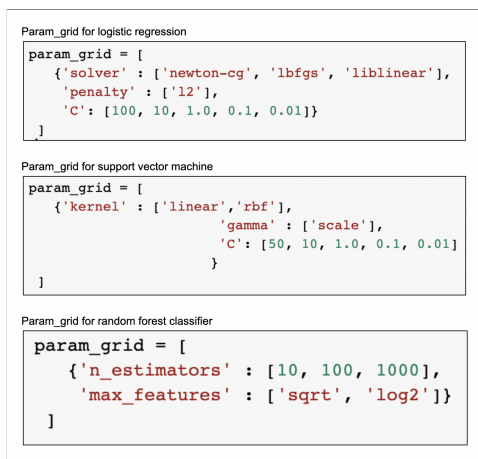
3 Methods

We tried two different approaches to solve this problem. The first one was a non-graph approach and the second one was a graph approach. The below subsections will provide detailed information regarding both approaches.

3.1 Non Graph Approach

After the necessary preprocessing steps were applied to this approach as mentioned in subsection 2.2 Non-Graph Approach Preprocessing Steps, we had to train three machine learning models using our preprocessed data. These machine learning

models include logistic regression, support vector machine (support vector classifier), and random forest classifier. We decided to try these specific machine learning models as they were popular choices for such classification tasks. Before we could start developing the individual models, we first split our preprocessed data into train, validation, and test sets. The split ratio was 80%, 10%, and 10% for train, validation, and test sets respectively. This was a random split. The reason we used a random split is because MoleculeNet website recommended random splitting for the Tox21 dataset. We then started the model development process. For each model, we tried to see the performance of the model using two different resampling techniques (undersampling and oversampling). This was done because we thought the resampling techniques would help the classifier better predict on the originally less represented class. It is also important to note that the resampling techniques were only applied to the training set. We first started out by trying the undersampling technique for each model. We used `RandomUnderSampler()` function from the `imbalanced-learn` library to successfully undersample our training data. We undersampled from only the majority class so that the number of majority class samples was equal to the number of minority class samples after undersampling. Additionally, the results of this function were assigned to new variables (`X_train_rus` and `y_train_rus`) to ensure we did not modify the original training set. We then used the `hypopt` library for hyperparameter tuning for our model. We used this particular library for this process because it allowed us to conduct grid search hyperparameter optimization using our specific validation set (which was not affected by resampling). We felt this was the right thing to do because we wanted to evaluate our model using the validation set that represented the true class distribution. Another intuition is that evaluating the model on the undersampled validation set would lead to the model performing well on the validation set but performing poorly on the test set which is not ideal. `Param_grids` were used during grid search hyperparameter optimization process for all the three models. `Param_grid` contains all the hyperparameters as well as the possible values we are interested in exploring. The `param_grid` for each of the models is shown in Figure 2.



```

Param_grid for logistic regression
param_grid = [
  {'solver': ['newton-cg', 'lbfgs', 'liblinear'],
   'penalty': ['l2'],
   'C': [100, 10, 1.0, 0.1, 0.01]}
]

Param_grid for support vector machine
param_grid = [
  {'kernel': ['linear', 'rbf'],
   'gamma': ['scale'],
   'C': [50, 10, 1.0, 0.1, 0.01]}
]

Param_grid for random forest classifier
param_grid = [
  {'n_estimators': [10, 100, 1000],
   'max_features': ['sqrt', 'log2']}
]

```

Figure 2: view of the `param_grid` for each of the three models

As seen in Figure 2, these specific `param_grids` were used during the grid search hyperparameter optimization process for each of the models. **Note:** we did not have more values for some of the hyperparameters in specific `param_grids` to reduce the time needed to determine the hyperparameter values. Once the desired hyperparameter values were found, we combined the undersampled training set with the validation set. We then trained each of the models using the right hyperparameter values obtained from the hyperparameter optimization process on the combined set. The test set was then used to evaluate the final model for each of the three models. In general, we used the three sets similar to how it was taught during the machine learning toolbox lecture.

We also tried the oversampling technique for each model. For this technique, we used SMOTE to generate synthetic samples from the minority class. SMOTE (Synthetic Minority Oversampling Technique) creates new instances of minority class examples from the existing examples. It utilizes the k nearest neighbor algorithm to create the new minority class samples. Similar to the undersampling technique, we applied the oversampling technique only on the training data. Specifically, we used the `SMOTE()` function from the `imbalanced-learn` library to successfully oversample our training data. We oversampled from only the minority class so that the number of majority class samples was equal to the number of minority class samples after oversampling. Additionally, the results of this function was assigned to new variables (`X_train_sm` and `y_train_sm`) to ensure we did not modify the original training set. The rest of the model development process, hyperparameter tuning and model development after the tuning process, was the same as previously mentioned for undersampling except we used `X_train_sm` and `y_train_sm` as our training set instead of `X_train_rus` and `y_train_rus`. The results of three models using both undersampling and oversampling techniques is described in the evaluation section of the paper.

3.2 Graph Approach

After the necessary preprocessing steps were applied to this approach as mentioned in subsection 2.3 Graph Approach Preprocessing Steps, we had to train a graph neural network (GNN) for classification using our preprocessed data. After we were able to have a graph representation for each chemical compound and create our custom PyTorch dataset, we split our preprocessed data into train, validation, and test sets. We used 80% of the data for training, 10% of the data for validation, and the remaining 10% for test set. We also created separate dataloaders for the three sets to simplify efforts related to batching the data and other related things. We then performed a series of experiments to determine the appropriate model architecture and hyperparameter values such as learning rate. For every single trial, we training our model on the training set and evaluated on the validation set. We continued experimenting with different architectures and hyperparameter values and selected the architecture and hyperparameter values that produced the highest AUROC value on the validation set. We then used this particular architecture and hyperparameter values and evaluated on the test set to get an evaluation of our final model. Our final model had two GCNConv layers and one linear layer. The first layer was the GCNConv layer which had an in_channel value of 9 and out_channel value of 64. The second layer was the GCNConv layer which had an in_channel value of 64 and out_channel value of 32. The last layer was the linear layer which had an in_channel value of 32 and out_channel value of 2. Additionally, we used a learning rate value of 0.001, Adam optimizer, and CrossEntropyLoss() as our loss function. The details of the model architecture is also shown in Figure 3.

```
GraphGCN(  
  (conv1): GCNConv(9, 64)  
  (conv2): GCNConv(64, 32)  
  (lin1): Linear(in_features=32, out_features=2, bias=True)  
)
```

Figure 3: view of GNN architecture used for the graph approach

As described in homework three, to train this GNN for graph classification, we followed the following three steps:

1. Embed each node by performing multiple rounds of message passing.
2. Aggregate the node embeddings into a single graph embedding (readout layer). We used PyTorch Geometric’s `global_mean_pool()` function for our readout layer similar to the homework. **Note:** we also added dropout after this step.
3. Train classifier on the graph embedding.

4 Evaluation

For both approaches, we used the AUROC score to evaluate the performance of the various models. We used this metric because it was the recommended evaluation metric for the Tox21 dataset by the [MoleculeNet](#) website. The below subsections will cover the evaluation results for each of the two approaches.

4.1 Evaluation Results of Non-Graph Approach and Comparison to Other Model Results

As previously described in the methods section of the paper, we used three machine-learning models for this approach. These models include logistic regression, support vector machine (support vector classifier), and random forest classifier. We also tested two resampling techniques to handle the imbalanced data for all three models. The reason we tested on two resampling techniques is because we wanted to see if either one had drastic effects on our model performance. Figure 4 below shows the AUROC score for the three models for both the resampling techniques.

	Logistic Regression	Support Vector Classifier	Random Forest Classifier
Using Undersampling Technique	0.7817	0.7522	0.7627
Using Oversampling Technique	0.7844	0.7843	0.7706

Figure 4: view of the AUROC scores for the three models for both the resampling techniques

As we can see in Figure 4, the AUROC scores for the different models are in the range from 0.75 to 0.79. These results are slightly lower than the [results](#) mentioned on the MoleculeNet website. Figure 6 shows the results from the website for

the Tox21 dataset.

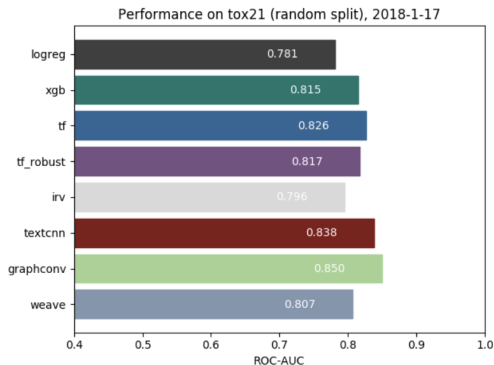


Figure 5: AUROC scores for different models from the MoleculeNet website for Tox21 dataset

It is important to note that we cannot directly compare our results with the results mentioned on the MoleculeNet website. This is because we only focused on one mechanistic target (p53) while they focused on multiple mechanistic targets. So the AUROC scores shown in Figure 5 might be the average over multiple mechanistic targets. Additionally, for the most part, they used different models for this classification task. However, we can use the results in Figure 5 to get an approximation of whether our models produced results that are similar to the latest results found on the website. Figure 6 shows the ROC curve for the various models with different resampling techniques. Our three models are robust because they are able to achieve the same range of AUROC scores even with variations in the validation and test sets.

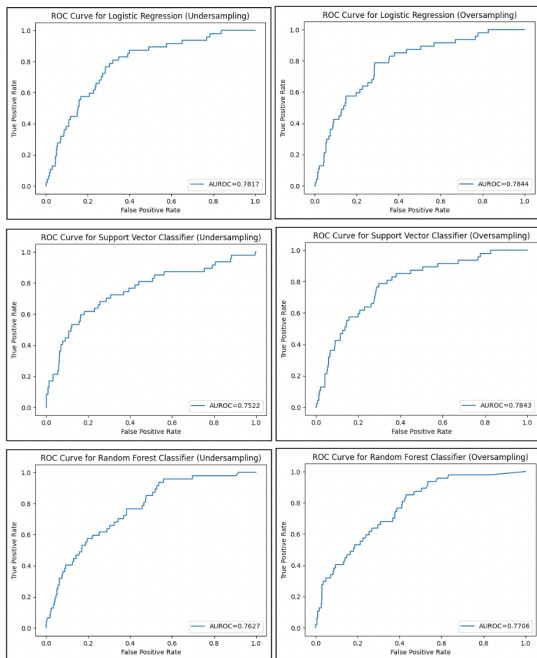


Figure 6: ROC curves for the models with different resampling techniques

4.2 Evaluation Results of Graph Approach

Similar to the non graph approach, we used AUROC score to evaluate our GNN. We used weighted average to calculate AUROC since this accounts for class imbalance. The AUROC score on the test set was in range from 0.75 to 0.81. This value is less than a related model’s (GraphConv) AUROC score shown in Figure 5. However, as mentioned in the previous subsection, we cannot directly compare our results with the results mentioned in Figure 5. This is because we focused on only one mechanistic target while the results from Figure 5 might have been the average AUROC score over all the different mechanistic targets in the Tox21 dataset. But we can use the results in Figure 5 to get an approximation of

our model’s performance. Additionally, we did not include the ROC curve for the GNN to acknowledge the report page limit. Our graph approach is robust because even with variations in three sets due to shuffling, our GNN is able to achieve similar AUROC scores on the test set. Additionally there are no overfitting or underfitting issues since the AUROC scores for the validation and test set are in same range.

5 Conclusion and Discussion

5.1 Updates to our Project After the Proposal

Our final project needed to be modified in order to create functioning models with a good dataset. Because of this, it is very different from the proposal. However, all adjustments were made with guidance from Dr. Luo. First, the dataset we used was different than the one we had proposed to use. The previous dataset focused on chemicals that had been studied to impact the thyroid and the assays they were tested with. We switched from this biological data to the SMILES data based on chemical structure. Because of the switch, we needed a much larger dataset. Also, we originally stated we would be using F-value to analyze the accuracy of the models, however switching the dataset meant that we wanted to use what MoleculeNet recommended, as described previously. Because of this, we used AUROC score and ROC curves to evaluate our models.

5.2 Limitations to our Models

While there are many studies that have gone into detail about the role that p53 plays in tumor suppression, there are some that claim that p53 is too complex, and previous studies have only touched the surface of the biological mechanisms behind tumor suppression [LG22]. The role of p53 may be very complex, as it can play a role in several different processes including cell cycle arrest, DNA repair, and apoptosis [BMA14]. Because of this, it is important to understand that these models do not give direct insight into if chemicals will cause tumors to grow or make a person get cancer when exposed. However, they can give some insight into a way that they can impact the function of p53 as a protein family. Another limitation we experienced was regarding the hyperparameter tuning process while developing our GNN. We did not use any library functions for hyperparameter tuning. Rather we manually tried out different architectures and hyperparameter values and selected the architecture and values that resulted in the highest AUROC score on the validation set. While this helps to some degree, we are not really covering a large space in terms of exploring the hyperparameter values. This is a limitation because it prevents us from developing an even better-performing model. Another limitation is that we used random splitting method to split our data into train, validation, and test sets. This is a limitation because during random splitting we are not ensuring equal representation of our data on all the different sets. This means that there is a chance of encountering a scenario where the training set contains all the 0 target labels and the validation set and test set contain all the 1 target labels. This would result in poor model performance. We can address this limitation by performing stratified splitting to ensure we have the same percentage of classes in each split.

There are also limitations to studying the features of these chemicals alone due to the mechanisms of action that take place in the body. For example, when molecules bind to each other, the toxicity of certain chemicals can depend on the proteins involved in the biological pathway [LL21]. Additional proteins involved in p53 tumor suppression may do so directly or indirectly, and it is possible there may be confounding interactions with other chemicals. These results may narrow down and suggest what chemicals may be worth further studying for carcinogenic properties related to p53, however, experimental validation would be necessary to make public health recommendations.

5.3 Future Directions of the Study

Given the time limit for this project, there are many ways we would expand on the project. The first thing we would do is work towards improving the accuracy and AUROC scores by exploring more chemical features, as well as testing other metrics (such as precise-recall curve and F1 score) to see how well our models perform for both the classes. It would be interesting to compare the use of 2D numerical features and 3D structural features as other studies have for this particular case. Next, we would use other oncogenes to improve the model’s connection to predicting cancer. While p53 is known to be one of the most important protein families when thinking about tumor growth and suppression, there are others that we could look into and add on to our original dataset.

6 Contributions and Source Code

Individuals worked as the lead on certain sections: Utkarsh as Code Lead, Isabelle as Biological Background, and Hannah as Data Collection Lead. Here is our source code: [Google Colab Link](#). This code is well documented for easy readability.

References

- [AJ17] Inteaz Ahmed and Sung Hwa Jung. Applications of metal-organic frameworks in adsorption/separation processes via hydrogen bonding interactions. *Chemical Engineering Journal*, 310:197–215, 2017.
- [Bas00] II Baskin. Ii (2018). *Machine Learning Methods in Computational Toxicology. Methods Mol. Biol.*, pages 119–139, 1800.
- [Ben05] Romualdo Benigni. Structureactivity relationship studies of chemical mutagens and carcinogens: mechanistic investigations and prediction approaches. *Chemical Reviews*, 105(5):1767–1800, 2005. PMID: 15884789.
- [BFS⁺18] Freddie Bray, Jacques Ferlay, Isabelle Soerjomataram, Rebecca L Siegel, Lindsey A Torre, and Ahmedin Jemal. Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*, 68(6):394–424, 2018.
- [BJKK12] Anne-Laure Boulesteix, Silke Janitza, Jochen Kruppa, and Inke R König. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6):493–507, 2012.
- [BMA14] Kathryn T Bieging, Stephano Spano Mello, and Laura D Attardi. Unravelling mechanisms of p53-mediated tumour suppression. *Nature Reviews Cancer*, 14(5):359–370, 2014.
- [CMLBF17] Lisa Chedik, Dominique Mias-Lucquin, Arnaud Bruyere, and Olivier Fardel. In silico prediction for intestinal absorption and brain penetration of chemical pesticides in humans. *International Journal of Environmental Research and Public Health*, 14(7):708, 2017.
- [CRC⁺23] Elena Chung, Daniel P. Russo, Heather L. Ciallella, Yu-Tang Wang, Min Wu, Lauren M. Aleksunes, and Hao Zhu. Data-driven quantitative structure–activity relationship modeling for human carcinogenicity by chronic oral exposure. *Environmental Science & Technology*, 57(16):6573–6588, 2023. PMID: 37040559.
- [DZW⁺11] Yonghong Deng, Weijian Zhang, Yuan Wu, Haifeng Yu, and Xueqing Qiu. Effect of molecular weight on the adsorption characteristics of lignosulfonates. *The Journal of Physical Chemistry B*, 115(49):14866–14873, 2011.
- [FEG⁺20] Benedek Fabian, Thomas Edlich, Hélène Gaspar, Marwin Segler, Joshua Meyers, Marco Fiscato, and Mohamed Ahmed. Molecular representation learning with language models and domain-relevant auxiliary tasks. *arXiv preprint arXiv:2011.13230*, 2020.
- [Har19] Thomas Hartung. Predicting toxicity of chemicals: software beats animal testing. *Efsa Journal*, 17:e170710, 2019.
- [HSK⁺18] Maya Hirohara, Yutaka Saito, Yuki Koda, Kengo Sato, and Yasubumi Sakakibara. Convolutional neural network based on smiles representation of compounds for detecting chemical motif. *BMC bioinformatics*, 19:83–94, 2018.
- [iri] Integrated risk information system. <https://www.epa.gov/iris>.
- [Jak06] Vikramaditya Jakkula. Tutorial on support vector machine (svm). *School of EECS, Washington State University*, 37(2.5):3, 2006.
- [KLAL21] Hyunseob Kim, Jeongcheol Lee, Sunil Ahn, and Jongsuk Ruth Lee. A merged molecular representation learning for molecular properties prediction with a web-based service. *Scientific Reports*, 11(1):11028, 2021.
- [KMNS19] Abdul Karim, Avinash Mishra, M. A. Hakim Newton, and Abdul Sattar. Efficient toxicity prediction via simple features using shallow neural networks and decision trees. *ACS Omega*, 4(1):1874–1888, 2019.
- [KSM⁺19] Abdul Karim, Jaspreet Singh, Avinash Mishra, Abdollah Dehzangi, M. A. Hakim Newton, and Abdul Sattar. Toxicity prediction by multimodal deep learning. In Kouzou Ohara and Quan Bai, editors, *Knowledge Management and Acquisition for Intelligent Systems*, pages 142–152, Cham, 2019. Springer International Publishing.
- [Lev20] Arnold J Levine. p53: 800 million years of evolution and 40 years of discovery. *Nature Reviews Cancer*, 20(8):471–480, 2020.
- [LG22] Yanqing Liu and Wei Gu. The complexity of p53-mediated metabolic regulation in tumor suppression. *Seminars in Cancer Biology*, 85:4–32, 2022. Targeting Cellular Signaling Pathways.

- [LL21] Sangrak Lim and Yong Oh Lee. Predicting chemical properties using self-attention multi-task learning based on smiles representation. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 3146–3153, 2021.
- [log] What is logistic regression? <https://www.ibm.com/topics/logistic-regression#:~:text=Resources-,What%20is%20logistic%20regression%3F,given%20dataset%20of%20independent%20variables.>
- [N⁺04] William Stafford Noble et al. Support vector machine applications in computational biology. *Kernel methods in computational biology*, 71:92, 2004.
- [O2] Noel M O’Boyle. Towards a universal smiles representation-a standard method to generate canonical smiles based on the inchi. *Journal of cheminformatics*, 4:1–14, 2012.
- [PFTW03] Roger Perkins, Hong 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.
- [PMS⁺20] Gabriel A. Pinheiro, Johnatan Mucelini, Marinalva D. Soares, Ronaldo C. Prati, Juarez L. F. Da Silva, and Marcos G. Quiles. Machine learning prediction of nine molecular properties based on the smiles representation of the qm9 quantum-chemistry dataset. *The Journal of Physical Chemistry A*, 124(47):9854–9866, 2020. PMID: 33174750.
- [Pro] National Toxicology Program. Toxicology in the 21st century (tox21). <https://ntp.niehs.nih.gov/whatwestudy/tox21/index.html#:~:text=The%20Toxicology%20in%20the%2021st,substances%20%20adversely%20affect%20human%20health.>
- [RSR18] Hannah Ritchie, Fiona Spooner, and Max Roser. Causes of death. *Our World in Data*, 2018. <https://ourworldindata.org/causes-of-death>.
- [SFS⁺21] Hyuna Sung, Jacques Ferlay, Rebecca L Siegel, Mathieu Laversanne, Isabelle Soerjomataram, Ahmedin Jemal, and Freddie Bray. Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*, 71(3):209–249, 2021.
- [SGG⁺16] Martyn T. Smith, Kathryn Z. Guyton, Catherine F. Gibbons, Jason M. Fritz, Christopher J. Portier, Ivan Rusyn, David M. DeMarini, Jane C. Caldwell, Robert J. Kavlock, Paul F. Lambert, Stephen S. Hecht, John R. Bucher, Bernard W. Stewart, Robert A. Baan, Vincent J. Coglian, and Kurt Straif. Key characteristics of carcinogens as a basis for organizing data on mechanisms of carcinogenesis. *Environmental Health Perspectives*, 124(6):713–721, 2016.
- [SGT⁺08] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.
- [TRLL09] N.X. Tan, H.B. Rao, Z.R. Li, and X.Y. Li. Prediction of chemical carcinogenicity by machine learning approaches. *SAR and QSAR in Environmental Research*, 20(1-2):27–75, 2009. PMID: 19343583.
- [TSC] About the tsca chemical substance inventory. <https://www.epa.gov/tsca-inventory/about-tsca-chemical-substance-inventory>.
- [Wei88] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988.
- [ZMW⁺17] Qingda Zang, Kamel Mansouri, Antony J. Williams, Richard S. Judson, David G. Allen, Warren M. Casey, and Nicole C. Kleinstreuer. In silico prediction of physicochemical properties of environmental chemicals using molecular fingerprints and machine learning. *Journal of Chemical Information and Modeling*, 57(1):36–49, 2017. PMID: 28006899.
- [ZW17] Evan N. Feinberg Joseph Gomes Caleb Geniesse Aneesh S. Pappu Karl Leswing Vijay Pande Zhenqin Wu, Bharath Ramsundar. Moleculenet: A benchmark for molecular machine learning, 2017.