Final Report: An Interactive Adverse Drug Effect Network

Team 23: Lauren Fogel, Meghna Godbole, Asma Khimani, Harshini Vummadi

1 Introduction

In 2021, 64.8% of adults in the USA had taken prescription medication in the last year [1]. Adverse drug reactions (ADRs), harmful or unintended effects resulting from medication use [2], and drug-drug interactions (DDIs), which occur when one drug affects the efficacy of another [3], pose significant public health risks. In 2022, more than 1.25 million severe adverse events were reported, along with over 175,000 deaths [4]. Current tools to assess ADRs require users to input specific drugs and focus on prescription drugs, leaving a gap for over-the-counter (OTC) drugs that may interact unexpectedly with prescriptions. **Formal Problem Definition** Unlike existing tools, the primary objective of our project is to create a network visualization tool that maps relationships between prescription drugs and OTC medications based on ADRs. This tool will allow users to identify potential harmful interactions between drugs and predict medications with possible adverse drug effects using machine learning.

2 Literature Survey

2.1 Drug Interaction Databases

The FDA Adverse Event Reporting System (FAERS) [5] is a database that gathers reports on adverse drug events from various individuals to support the FDA's safety surveillance of drug and biologic products. Min et al. [6] used FAERS to analyze painkiller-related ADRs only, but we aim to extend this type of analysis to include both OTC and prescription drugs.

DrugBank [7] serves as a resource for detailed information on drug mechanisms, targets, and interactions to facilitate drug discovery and understanding. This will aid us in identifying interactions based on chemical structure and severity. We aim to enhance DrugBank's search tool, which requires users to input drugs and checks only for interactions between those queried.

DDInter [8] is a drug interaction database designed to enhance DrugBank [7] by offering detailed information on interaction mechanisms, severity levels, and drug alternatives. It features an interaction checker for user input and drug interaction visualizations. We aim to improve this by incorporating predictive features that will alert users to other potential drug interactions.

DDInter 2.0 [9] features an improved user interface, expanded data coverage, and more interaction types, plus a visualization tool with advanced filtering options. We aim to incorporate these features into our network visualization by constructing a drug network and allowing users to view smaller networks for specific drugs.

PubChem [10] stores chemical, structural, and regulatory data. For accurate predictions with our machine learning (ML) model, we will use this chemical compound data for related drugs to identify interactions. We aim to create a simpler visualization compared to PubChem's detailed, chemistry-focused data.

2.2 Network Visualization

The Diet-Drug Interactions (DDID) [11] database helps users understand how diet affects medications by providing a network visualization of drug and food interactions based on correlation studies. It is broad and non-interactive in nature, so our aim is to create a more detailed, interactive visualization using tools like D3 [12], a data visualization library.

ReactomeFIViz [13] visualizes drug-target interactions for FDA-approved drugs within a genome-wide functional interaction network. This interactive tool lets users explore drug interactions, preventing potential ADRs. While similar to our design, we aim to enhance its cluttered visualization for better clarity and usability.

Udrescu et al. [14] uses topological community detection to cluster drugs in a DDI network, forming functional categories and analyzing relationships with a zoom in feature. We plan to build on their approach, incorporating similar clustering techniques while enhancing the design by reorganizing and simplifying the visualization to improve clarity and reduce clutter.

2.3 Utilizing Chemical Data, Structural Data, and ML Algorithms

Vilar et al. [15] proposes a methodology to predict DDIs based on molecular structure similar-

ity using DrugBank data and the Tanimoto coefficent for drug pair comparison. It inspired our approach to use Tanimoto similarity scores and interaction data in a ML algorithm to improve predictions. We plan to further improve this method by combining similarity scores and real-world drug interaction data for better predictive power.

To predict drug-target interactions, Kim et al. [16] uses drug-target protein similarity measures derived from datasets such as chemical structure, side effects, and DDIs, applying KL1LR and SVM classification methods, measuring protein similarity, and predicting interactions. This is crucial for our project to predict previously unreported ADRs. We plan to explore more ML methods while consolidating relevant data, as this paper focuses only on two classification methods and uses a variety of data.

A 2023 paper [17] enhanced drug-target interaction prediction with a three-phase model using advanced feature extraction, selection, and classification for drugs and proteins. While it incorporates diverse chemical and structural data, it overlooks broader interaction data like ADRs and real-world evidence from OTC medications. We plan to integrate network visualization with prediction features based on chemical data.

Farhana et al. [18] utilized classifiers, random forest and SVM, to predict pharmacodynamic DDIs based on features like side effect and chemical similarity, and target protein accuracy. Their work demonstrates ML approaches for DDI prediction using Tanimoto similarity. However, this paper focuses solely on pharmacodynamic interactions and could be enhanced by including additional interaction types as features.

3 Methods

3.1 Intuition

Our project has two key innovations. First, we present drug-drug interactions differently. Unlike existing tools that require users to input two drugs, our tool allows users to search for a single drug and see other drugs in the database that interact with it, helping users identify potentially harmful interactions without needing to know which drugs to compare. The results are shown in an interactive network, sorted by commonality based on historical interaction data.

The second innovation is a prediction function using machine learning to predict DDIs between common OTC and prescription drugs. Our model improves upon existing tools by incorporating features like chemical structural similarity, protein target similarity, and real-world adverse event data from FAERS.

3.2 Network and Visualization

3.2.1 Data Preprocessing

The primary data source for our network visualization is DDInter [8]. We downloaded the data for all drugs on the site and consolidated the data into a file containing all of the nodes (drugs), and all the edges (drug-drug pairs) with their associated level of severity of the interaction (Minor, Moderate, or Major). We then added an additional property to each edge, using supplemental data from FAERS. From FAERS, we pulled counts of adverse events for each drug-drug pair, then added this as a property to each edge. We used a batching process to retrieve the counts similar to what was used in the ML model. The data from DDInter was combined with the FAERS count and prediction scores to form the json files that create our network visualization.

3.2.2 Visualization Development

We created our visualization tool with D3 [12]. Our visualization tool allows the user to search for and select a drug of interest from the drop down menu, then visualize the ten drugs with the highest counts of interactions with the selected drug. These interactions are connected by colored edges which represent the level of severity of that interaction. The user is able to drag and pin nodes, as well as mouse over the edges to view more information about the level of severity and the classes of drugs involved in the interaction. In addition to the search button, we created a predict button. When pressed, a new network visualization is shown, with ten OTC drugs we predicted would have an interaction with the drug of interest. The network visualization looks similar to the one generated through DDInter, but instead, the nodes are ordered by prediction

score ensuring that the user only sees the most confident predictions. The mouse over provides the specific prediction score so the user can see how confident the model is about the prediction.

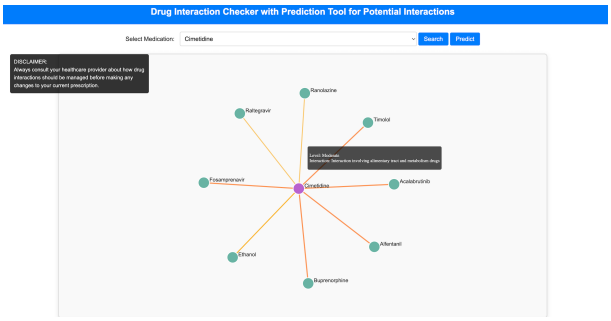

Figure 1: Search Function of the Interactive Network Visualization, depicting the ten most common DDIs for a user-selected drug

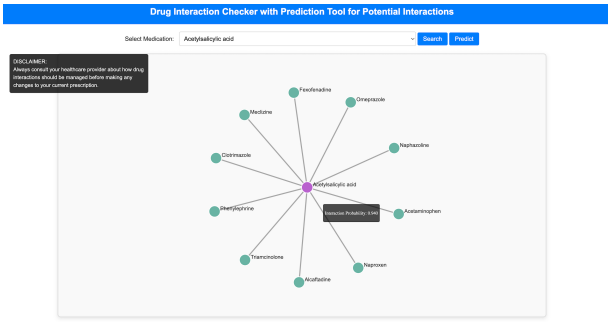

Figure 2: Predict Function of the Interactive Network Visualization, depicting the ten most likely OTC drugs to have an interaction with a user-selected drug

3.3 ML Model Development

3.3.1 Data Preprocessing

The prediction function uses data from DrugBank FAERS. To extract DrugBank data from the large XML file, we found a script [19] on GitHub that we modified to obtain extensive data. We then developed Python scripts that got data into a simpler format, with just the matched drug-drug interactions, as well as the documented interactions between these drugs. We also obtained general codes for matching all drugs in the DrugBank dataset and drug ids to resolve issues with unknown pairs. To compute chemical similarity scores for each drug pair, we needed to extract SMILES codes, text based representations of drug chemical structures, for each drug from DrugBank. We obtained this data in a separate data extraction process with a script using python's RDkit package[20]. To compute protein target similarity scores for each drug pair, we extracted target lists for each individual drug from the DrugBank database. This data was obtained using the script [19] for extracting DrugBank data, and then processed further to get a list of targets for each drug. From there, vectors comparing the targets in each drug pair were created, denoting each protein target as 0 or 1 based on which drugs targeted which proteins.

3.3.2 Feature Engineering and Selection

To predict DDIs accurately, we created a robust feature set that captured a variety of characteristics that contribute to DDIs. **Computing Chemical Similarity.** We computed chemical similarity to assess the structural similarity between the drug molecules. We utilized one of the most common ways to calculate chemical similarity, the Tanimoto Coefficent (TC), which measures the overlap between the molecular fingerprints of two compounds using a bitwise comparison to determine the presence or absence of substructures. Given two fingerprints A and B, the TC is calculated where $|A \cap B|$ is the number of common features and $|A \cup B|$ is the total number of unique features.

$$TC(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

In order to compute TC, we used RDkit to convert the SMILE strings into molecular objects for further processing [20]. We then generated Morgan fingerprints, which are binary vector representations of the absence or presence of certain substructures. Lastly, using the Morgan fingerprints of the drugs we calculated TC similarity scores for each drug pair, where a high TC similarity score indicates the drugs share common substructures and serves as an indicator of potential interactions. Initially, we explored the summary statistics of the dataset by visualizing the distribution of TC similarity scores for drug pairs with interactions (positive) and drug pairs with no interactions (negative). Figure 3 shows the density of TC scores for both classes. Both classes have significant overlap, in the lower similarity score ranges, which suggests that the there

could be a challenge distinguishing interactions based solely on TC scores and additional features may be necessary for better classification.

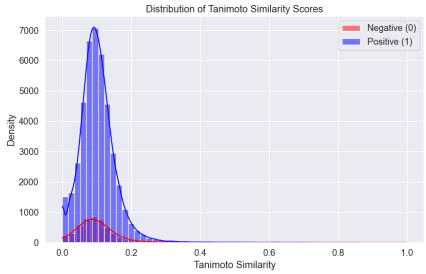

Figure 3: This histogram shows the density of Tanimoto similarity scores for drug pairs with interactions (Positive) and those without interactions (Negative).

Computing Adverse Event Frequency and Severity Scores To capture real-world interaction patterns, we retrieved data from FAERS APIs to create two features, adverse event frequency and adverse event severity scores. Adverse event frequency represents the number of reports when a drug pair. Severity scores represent a weighted score based on the number of serious outcomes using reports of hospitalization, life-threatening, disability, and death. However, FAERS reports often include all drugs a patient is on which could introduce confounding effects. To mitigate this, we filtered FAERS reports to those with three drugs or less to reduce the complexity of cumulative interactions. When it came to calling the FAERS API to retrieve this data, we ran into many issues. To speed up the lengthy execution, we used parallel execution by calling ThreadPoolExecutor from the concurrent.futures Python package [21]. Then, we used a batching method to run successive calls in batches of 1000 drug pairs, which resulted in the need for a total of 110 batches to cover our dataset of almost 110,000 drug pairs. This was a python script which used a subprocess to run the main API calls, and calculated the number of batches it would need to run based on the desired batch size, then iterated through these batches. We used this same method for calling our training data as well. **Computing Protein Target Similarity Scores** Drug targets are the location in a cell, often a protein, where a drug binds, causing a change in function and therapeutic effect [7]. Similar targets may mean drugs are trying to bind in the same spots, which can cause adverse effects or impact drug effectiveness [22]. We created a feature to capture protein target similarity by using Jaccard similarity, which requires vectors comparing each drug's targets, and then calculates the similarity between the two [22]. Although the calculation is similar to TC similarity scores, it instead uses protein target similarity vectors. A similarity score of 1 means the drugs share the exact same protein targets, and a score of 0 means the drugs have no common targets.

$$Jaccard\_similarity(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

The data came from DrugBank, so did not require any additional data collection.

3.3.3 Model Training

We utilized a random forest classifier algorithm for DDI prediction due to its capability to handle complex, non-linear data relationships. The dataset was split into three subsets: train, validation, and test. The dataset was first split into a training and validation set (80%) and a test set (20%) using train_test_split with a random seed for reproducibility. The training set was further split into a training set and a validation set. The training set was used for model training and hyperparameter tuning, the validation set was reserved for model calibration, and the test set was used in the final evaluation of the model's performance. **Hyperparameter Tuning** The hyperparameters of the model were optimized using random search and five fold cross validation over predefined parameters. This approach efficiently explored possible parameter combinations on the training set, allowing us to identify parameters that provided strong initial performance. In addition to model specific hyperparameters, the model was analyzed using scoring metrics from the scikit-learn library [23] - precision, f1, recall, roc auc, and brier score. Our criteria focused on parameters that could prevent overfitting, an issue that the class imbalance could cause towards the large majority class. **Model Calibration** After the best hyperparameters were identified, the model was

calibrated using the sigmoid method in the CalibratedClassifierCV function from scikit-learn using the validation set to improve the reliability of predicted probability scores. Calibration was essential for this project because it ensures that the probabilities generated by the model are meaningful and interpretable when integrating it with the network visualization. The final calibrated model was evaluated on the test set to assess its performance across the scoring metrics. **Model Evaluation Metrics** Given the large class imbalance between the majority (drug pairs with known interactions, Class 1) and the minority (drug pairs with unknown or no interactions, Class 0) classes, the model is naturally biased toward performing well on the majority class. This can cause disproportionately high recall for Class 1 at the cost of performance on Class 0. However, our primary focus is minimizing false negatives for Class 1 because missing an interaction could lead to harmful consequences for users. In contrast, a false positive would cause unnecessary concern or more effort to confirm the interaction, but would not result in the same harm caused by a false negative. Thus, we prioritized recall for Class 1 to ensure the model correctly identifies interacting drug pairs. However, while recall for the majority class is critical, focusing only on this metric can lead to trade-offs, such as a higher number of false positives or reduced overall performance across other metrics, such as precision and recall for Class 0. Thus, we choose the best model based on minimizing false negatives, striking a balance between both classes while keeping recall high for Class 1, and achieving reliable probability distributions for real-world applicability.

3.4 Experiments and Evaluation

3.4.1 Network Visualization Evaluation

In order to evaluate our network we asked people outside our project for feedback when using our visualization. We asked several questions about ease of use and design choices and recorded feedback for future use. Originally, we had known and predicted nodes appearing on the same graph, resulting in a total of 20 nodes for each target node. We received feedback that this made the network much harder to read and users preferred our visualization where the networks were separated. Another piece of feedback we received was about our mouse-overs. We originally had mouse-overs over every node relay information, but got responses that there were too many mouse-overs and it made moving around the network too chaotic, so we decided to reduce the mouse-overs and only keep the ones on the edges, since those have the most pertinent information. We also received various feedback on the color schemes for both the nodes and edges, and incorporated those changes based on our opinions.

3.4.2 ML Model Evaluation

The overarching goal of this project is to answer the question: can we predict DDIs using machine learning? To answer this, a random forest model was trained and evaluated on a dataset of 38,738 drug pairs, where 34,798 drug pairs had an interaction (Class 1) and 3,940 drug pairs did not have an interaction (Class 0). The evaluation focused on handling the observed class imbalance, maintaining high recall for the majority class while minimizing false negatives, and ensuring reliable probability distributions that could be used to predict the probability of adverse interactions between OTC-prescription drug pairs.

We optimized the hyperparameters of the model using random search, where we prioritized recall as the scoring metric, with five-fold cross validation and obtained the following best parameters: n_estimators= 100, min_samples_split= 2, min_samples_leaf= 5, max_depth= None, class_weight= {0: 1, 1: 10}. Using the optimized hyperparameters, the model was effective in identifying class 1 instances, achieving a recall of 1.00, which means all drug pairs with interactions were correctly identified. However, the model struggled to correctly classify class 0 instances, with a recall of only 0.12 and an f1 score of 0.22. Additionally, the Brier score was 0.0718, indicating okay but improvable probability calibration. These results suggested that the model did excellent in capturing drug pairs with interactions, but needed further refinement to improve its handling of non-

interacting drug pairs and the reliability of its probability estimates. To address this, we calibrated the model using the sigmoid method. The calibration resulted in significant improvements for class 0, with recall increasing from 0.12 to 0.35 and f1 score increasing from 0.22 to 0.50. Although there was a slight decrease in recall for class 1 (from 1.00 to 0.99), this trade-off was acceptable because the calibrated model maintained strong performance for interactions while improving its handling of non-interacting drug pairs. Furthermore, the Brier score decreased to 0.0634, which shows improved probability calibration and reliability in predictions.

To understand the model's decision making process, feature importance scores were extracted from the best random forest model. Figure 5 shows that adverse event frequency and severity score contributed the most to the model's predictions, while chemical similarity and protein target similarity contributed the least. This is an interesting result because it suggests that patterns in real-world reported outcomes may be highly beneficial in predicting potential drug interactions rather than structural or biological similarities.

Table 1: Selected Random Forest Model - Classification Report

Class	Precision	Recall	F1	Support
0	0.84	0.35	0.50	765
1	0.93	0.99	0.96	6983

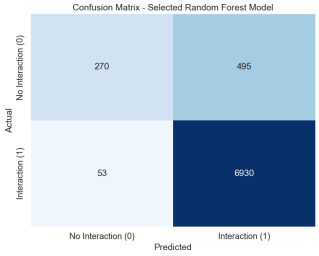

Figure 4: This confusion matrix shows the performance of the random forest model in predicting adverse drug interactions and minimizing the number of false negatives, 53.

3.5 Conclusions and Discussion

We developed an interactive network visualization and ML algorithm to map OTC drugs to prescription drugs and predict the probability of

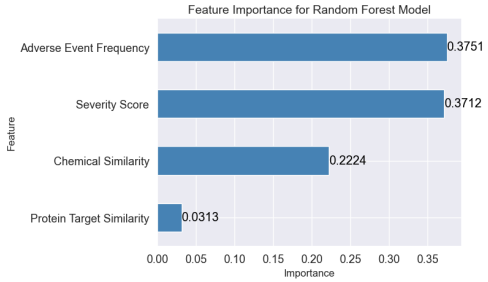

Figure 5: Feature importance plot from the selected random forest model. The longer bars indicate greater importance in predicting DDIs.

interactions between them. It fills a gap of current interaction checkers, which mainly focus on DDIs between prescription drugs.

The interface features a search bar for easy drug selection and displays interaction severity and type on hover. Users can also interact with the ML model to predict potential drug interactions and their probabilities. The model uses real-world adverse event reports, chemical, and biological data, helping users identify and prevent harmful interactions for safer medication use. **Limitations and Future Directions.** We observed a class imbalance in the dataset used to train the ML model due to the lack of data available on drug pairs that do not have interactions. Additionally, we relied on FAERS reports to create features, which may be may be incomplete or biased since it is a voluntary reporting system. However, focusing on reports with fewer drugs helps reduce confounding effects but also may leave out significant real-world scenarios. Visualization constraints include only displaying the top ten interacting drugs for each drug, which may inadvertently overlook other relevant drug interactions. In contrast, displaying too many medications may cause the visualization to become too cluttered. In the future, we hope to expand our visualization to display more drug interactions in a user friendly format. Additionally, we plan on introducing customizable filters for specific populations, such as pediatric or geriatric users, to offer more personalized assessments.

Contributions. All team members have contributed equally.

References

[1] Centers for Disease Control and Prevention. Quickstats: Percentage of adults aged 18 years who took prescription medication during the past 12 months, by sex and age group — national health interview survey, united states, 2021. *MMWR Morbidity and Mortality Weekly Report*, 72:450, 2023.

[2] J. J. Coleman and S. K. Pontefract. Adverse drug reactions. *Clinical Medicine (London, England)*, 16(5):481–485, 2016.

[3] Nicholas T. Lappas and Courtney M. Lappas. Chapter 3 - drug interactions. In Nicholas T. Lappas and Courtney M. Lappas, editors, *Forensic Toxicology (Second Edition)*, pages 45–71. Academic Press, San Diego, second edition edition, 2022.

[4] Sharath Kommu and Philip Whitfield. Adverse drug reactions. In *StatPearls [Internet]*. StatPearls Publishing, 2024.

[5] U.S. Food and Drug Administration. FDA Drug Event API. https://open.fda.gov/apis/drug/event/, n.d. Accessed: 2024-10-11.

[6] Jae Min, Vicki Osborne, Allison Kowalski, and Mattia Prosperi. Reported adverse events with painkillers: data mining of the us food and drug administration adverse events reporting system. *Drug safety*, 41:313–320, 2018.

[7] Craig Knox, Mike Wilson, Christen M Klinger, Mark Franklin, Eponine Oler, Alex Wilson, Allison Pon, Jordan Cox, Na Eun Chin, Seth A Strawbridge, et al. Drugbank 6.0: the drugbank knowledgebase for 2024. *Nucleic acids research*, 52(D1):D1265–D1275, 2024.

[8] Guoli Xiong, Zhijiang Yang, Jiacai Yi, Ningning Wang, Lei Wang, Huimin Zhu, Chengkun Wu, Aiping Lu, Xiang Chen, Shao Liu, et al. Ddinter: an online drug–drug interaction database towards improving clinical decision-making and patient safety. *Nucleic acids research*, 50(D1):D1200–D1207, 2022.

[9] Yao Tian, Jiacai Yi, Ningning Wang, Chengkun Wu, Jinfu Peng, Shao Liu, Guoping Yang, and Dongsheng Cao. Ddinter 2.0: an enhanced drug interaction resource with expanded data coverage, new interaction types, and improved user interface. *Nucleic Acids Research*, page gkae726, 2024.

[10] Sunghwan Kim, Paul A Thiessen, Evan E Bolton, Jie Chen, Gang Fu, Asta Gindulyte, Lianyi Han, Jane He, Siqian He, Benjamin A Shoemaker, et al. Pubchem substance and compound databases. *Nucleic acids research*, 44(D1):D1202–D1213, 2016.

[11] Yanfeng Hong, Hongquan Xu, Yuhong Liu, Sisi Zhu, Chao Tian, Gongxing Chen, Feng Zhu, and Lin Tao. Ddid: a comprehensive resource for visualization and analysis of diet–drug interactions. *Briefings in Bioinformatics*, 25(3):bbae212, 2024.

[12] Observable. D3.js.

[13] Aurora S Blucher, Shannon K McWeeney, Lincoln Stein, and Guanming Wu. Visualization of drug target interactions in the contexts of pathways and networks with reactomefiviz. *F1000Research*, 8, 2019.

[14] Lucreţia Udrescu, Laura Sbârcea, Alexandru Topîrceanu, Alexandru Iovanovici, Ludovic Kurunczi, Paul Bogdan, and Mihai Udrescu. Clustering drug-drug interaction networks with energy model layouts: community analysis and drug repurposing. *Scientific reports*, 6(1):32745, 2016.

[15] Santiago Vilar, Rave Harpaz, Eugenio Uriarte, Lourdes Santana, Raul Rabadan, and Carol Friedman. Drug—drug interaction through molecular structure similarity analysis. *Journal of the American Medical Informatics Association*, 19(6):1066–1074, 2012.

[16] Shinhyuk Kim, Daeyong Jin, and Hyunju Lee. Predicting drug-target interactions

using drug-drug interactions. *PloS one*, 8(11):e80129, 2013.

[17] Hengame Abbasi Mesrabadi, Karim Faez, and Jamshid Pirgazi. Drug–target interaction prediction based on protein features, using wrapper feature selection. *Scientific Reports*, 13(1):3594, 2023.

[18] NA Farhana, FM Afendi, A Fitrianto, and SH Wijaya. Classification modeling of support vector machine (svm) and random forest in predicting pharmacodynamics interactions. In *Journal of Physics: Conference Series*, volume 1863, page 012067. IOP Publishing, 2021.

[19] Rosa Hernansaiz Ballesteros. Parsing_drugbank. https://gist.github.com/rosherbal/56461421c69a8a7da775336c95fa62e0, 2023. GitHub Gist. Accessed: 2024-10-31.

[20] G Landrum. Rdkit: open-source cheminformatics http://www. rdkit. org. *Google Scholar There is no corresponding record for this reference*, 3(8), 2016.

[21] Python Software Foundation. *concurrent.futures — Launching parallel tasks*, 2023. Added in version 3.2.

[22] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466, 2018.

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