Identification of Novel Drug-Drug Interactions as Out-of-Distribution Samples

Dimitrios Lezos
The American College of Greece—Deree
Athens, Greece
d.lezos@acg.edu

Dimitrios Vogiatzis
The American College of Greece—Deree &
National Center for Scientific Research, "Democritos"
Athens, Greece
dimitrv@acg.edu

***Abstract*—This work proposes a novel approach to predicting side effects in drug-drug interactions (DDIs) by identifying them as Out of Distribution (OoD) samples. The rise of polypharmacy has increased both the number of drug combinations administered to patients and the risk of DDIs, many such interactions remain undocumented in clinical trials or medical records, posing a significant challenge for patient safety. While traditional DDI prediction is typically framed as a supervised machine learning problem, novel side effects lack labeled data. To address this, we introduce a hybrid approach that combines diverse drug representations with OoD detection to capture unseen interactions. Our method integrates multiple drug embeddings generated from diverse characteristics, including biomedical texts, molecular structures, knowledge graphs and similarity profiles to generate rich latent representations that enable us to predict the presence of novel DDIs as OoD samples.**

Keywords—Drug-Drug Interaction, Machine Learning, Zero Shot Learning, Out of Distribution, k-Nearest Neighbors, Composite Embeddings

I. Introduction

As more drugs become available for the treatment of various ailments, patients often require multiple medications to address different aspects of their health [1], [2] while the widespread availability of prescription medications, over-the-counter drugs, and even self-medication options further contributes to the practice of polypharmacy. When a patient is administered with multiple drugs there might be interactions between them, affecting, enhancing or weakening their intended effects or even causing side effects making the need for careful management and prediction of possible interaction between drugs crucial. These kind of interactions are called drug-drug interaction (DDIs) [3]. Identifying harmful DDIs early can help prevent serious medical errors, enhance drug safety, and improve patient outcomes.

While many known DDIs are documented in medical literature and regulatory databases, novel (unseen) DDIs, which involve drug combinations that have not yet been clinically tested or reported with possible new adverse reactions, remain a critical challenge in pharmaceutical research and clinical practice. Many adverse drug reactions occur due to previously unknown interactions between drugs. Traditional experimental methods for detecting DDIs, such as in vitro studies and clinical trials, are expensive, time-consuming, and may not cover all possible drug interactions so being able to predict novel DDIs allows for more personalized medication plans by identifying safe drug combinations tailored to individual patient needs, so the prediction of novel DDIs is a crucial aspect of modern drug research and healthcare.

There are a number of proposed methods for predicting known DDIs that incorporate drug data from diverse sources like literature, molecular composition, knowledge graphs, structural similarity profiles and others to generate embeddings that feed various machine learning systems achieving excellent accuracy and recall [4], [5], [6], [7]. But the prediction of novel adverse drug reactions for which no training data exists is still a challenging task that has been addressed in only a few cases [10], [11], [12].

In this work we address the task of predicting the presence of novel DDIs not present in the training data, by using diverse sources to create rich representations of the drugs, that play a significant role in the prediction [13], and then identifying the novel DDIs as OoD cases compared to the known DDIs.

II. Literature Review

DDI prediction has been extensively studied [1], [5], [6], [7]. DDI prediction methods can be broadly categorized to Deep Neural Network (DNN) based methods, Convolutional Neural Network (CNN) based methods, Graph Neural Network (GNN) based methods, Knowledge Graph (KG) based methods [8], [9] and Multimodal based methods. They use different sources of information and embedding generation methods to model the drugs and their interactions. Similarly the problem of ZSL has been addressed by a number of publications mainly in the domain of computer vision based on the way they define the semantic information of the classes they can be split to Word based methods, Attribute based methods and Textual description based methods [14], [15], [16], [17], [18]. But the implementation in the DDI domain is quite difficult as Zero Shot classification is done by using semantic information of unseen classes which is very difficult to acquire in DDI. A more related area is the Generalized OoD detection methods [19] which is closely related to Anomaly Detection, Novelty Detection, Open Set Recognition and Outlier Detection. The Generalized OoD detection, which as defined by previous work [19] includes all the above fields, can be broadly categorized to Classification based methods, Density-based methods, Distance based methods and Reconstruction based methods.

For the specific task of ZSL / OoD detection in the area of DDI there are only a few publications namely Astras [10] which uses word embeddings to represent drugs and interactions and predicts DDI based on the architecture described in ESZSL [20], ZeroDDI [11] that extracts class-level and attribute-level semantics and distributes the embeddings in a unit sphere to achieve better class isolation and "Learning to Describe" [12] that uses a language model-based DDI predictor and a reinforcement learning (RL)-based information selector, enabling the selection of concise and pertinent text for accurate DDI prediction on new drugs.

In this work we combine elements of the above areas into a three-step method where first we create a composite embedding by collecting diverse rich representations of the drugs with which we train a DNN to create an Interactions Probabilities Vector (IPV) that we use as a much lower dimension space representation of each drug pair. In the second step we identify and remove the pairs with no interactions and in the third and final step we predict the drug pairs that will produce novel DDIs when administered together by identifying them as OoD samples compared to the known DDI pairs.

III. Research Design & Methodology

A. Architecture

Given a dataset of DDIs we split it into Seen and Unseen sets, where the Unseen set contains some interaction categories not present in the Seen one, we want by using only the Seen set for training, to predict if a pair of drugs in the Unseen set, when administered together, will produce novel DDIs not yet seen by our system. To achieve this, we use a three-step method:

Step 1: We use four separate embeddings for each drug that encapsulate different characteristics of the pharmaceutical substances, that after we reduce in size by Principal Component Analysis (PCA) we concatenate to create the representation of each drug and again concatenate to create the representation of each drug pair. We represent the interactions observed when the drugs in the pair were administered together with One-Hot Encoding and use the Seen set as training input to a DNN that learns to output the interactions as an Interactions Probabilities Vector (IPV) that we use a lower dimension space representation of the drug pairs of both the Seen and Unseen drug pairs on subsequent steps.

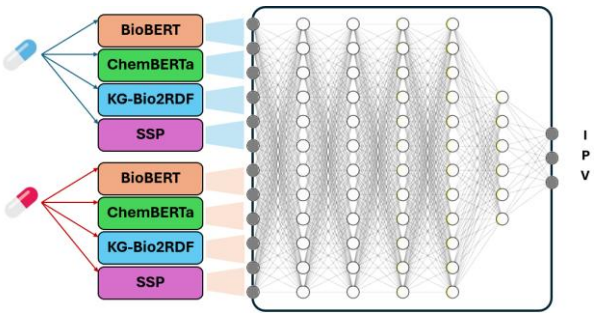

Fig. 1. Interactions Probabilities Vector Generation for a pair of drugs

Step 2: We identify the drug pairs that have no interactions and eliminate them from the next step.

Step 3: We predict the presence of novel DDIs by calculating the distance of each Unseen drug pair's IPV over the IPVs of the Seen set and comparing it with the distance distribution of the Seen drug pairs' IPVs over the same Seen set so that we can identify the OoD samples as novel DDIs by calculating the k-Nearest-Neighbours (kNN) distribution values as shown in kNN-OoD [33].

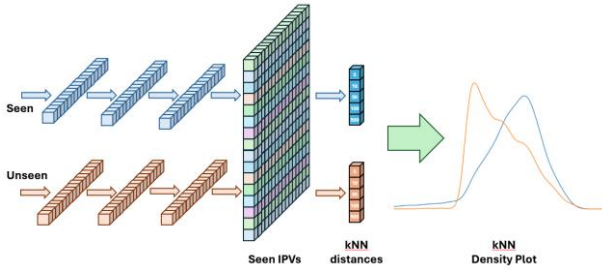

Fig. 2. kNN Distances Generation

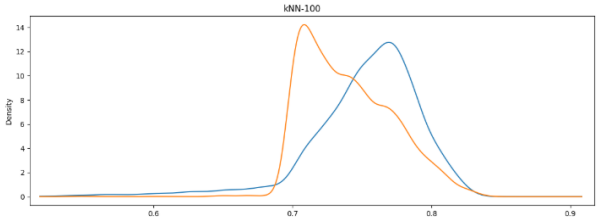

Fig. 3. In Distribution (blue) and Out of Distribution (orange) kNN Density Plot

B. Dataset Used

We used the latest TWOSIDES dataset[1] [34], that consists of drug-drug-interaction triplets that represent interaction side effects observed when administering two drugs together. When there are multiple interactions recorded then the drugs pair has multiple triplets, each having a different interaction. In our dataset preparation we combine these triplets so that each drug-drug pair exists in a single sample and is associated with all the different interactions that are recorded for that pair. For the drugs used in the DDI triplets we had their RxNORM ids used in TWOSIDES, mapped to the corresponding DrugBank ones to use embeddings produced by existing publications over that dataset.

C. Drug Embeddings

In our study we used four different embeddings for each drug that were concatenated. These are listed next:

1. Drug Name from BioBERT [21] transformer. Bio-BERT is a domain-specific language representation model pre-trained on large-scale biomedical corpora. We extracted the embeddings by providing Bio-BERT with the Drug name and received a 768 element vector.

2. Drug SMILES [22] molecular structure from the ChemBERTa transformer [23] similar to the one used in DeepARV [24]. ChemBERTa is a BERT transformer pretrained by a dataset of 77M unique SMILES from PubChem [38]. We extracted the embeddings by providing ChemBERTa with the Drug SMILES and received a 768 element vector.

3. Drug position and relations in the Bio2RDF Knowledge Graph [25] as generated by Graphembedding4DDI [26]. Bio2RDF creates a large RDF graph that interlinks data from major biological databases related to biological entities such as drug, protein, pathway and disease. In Graphembedding4DDI, DrugBank, KEGG and PharmGKB [27] datasets within Bio2RDF project release

[1] https://nsides.io/

4.0 were used as the background knowledge graph. The DDI links from these knowledge graphs were removed to eliminate bias on the prediction task. RDF2Vec [28] with Skip-Gram was used to traverse the graph and produce embeddings with 200 elements.

4. Structural Similarity Profile (SSP) between the known Drugs, generated by their SMILES molecular structure representation as defined and calculated in DeepDDI [29] with the use of Mol2Vec [30] and Modred [31]. SSP is a feature vector capturing structural features of individual drugs. SSP contains pairwise structural similarity scores obtained from the comparison between the input drug and all the 2,159 approved drugs of DrugBank as a fixed comparison target. SSP was generated for each drug in the input drug pair. Structural similarity between two drugs was measured by Tanimoto coefficient, which is defined as the number of common chemical fingerprints divided by the number of all the chemical fingerprints of the two drugs being compared. This provides an embedding with 2,159 elements.

D. *Input Dimentionality Reduction*

Each separate embedding before concatenation is reduced to a vector of 128 elements through PCA reduction, keeping 0.91 covariance from BioBERT, 0.93 covariance from ChemBERTa, 0.95 covariance from Bio2RDF KG and 0.99 covariance from SSP. This results in an input vector of 1024 elements for each drugs' pair (4x128=512 elements for each drug and 512x2=1024 elements for each pair).

E. *Out of Distribution Identification*

In order to identify the DDI's IPVs that indicate OoD items we used a method based on kNN, similar to the one described in kNN-OoD [33]. To accomplish this:

- We calculate the Cosine distances of each drug pair's IPV in the Seen set over a base population, usually consisting of the whole dataset of Seen pairs.
- For these distances we calculate kNN values for various numbers of k
- We define cutoff points that will separate the samples (e.g. using cutoff value of 0.23 for k=10 means that we separate those samples that have the 10 nearest neighbours closer than the 0.23 distance from those that have the 10 nearest neighbours further that the 0.23 distance).
- We then calculate the kNNs for the Unseen pairs and use the cutoff points to identify the OoD samples. We select the cutoff point value by trying to include as many of the Seen pairs as possible while leaving as many of the Unseen pairs out of the classification.

IV. Experiments

A. *Dataset*

We processed the TWOSIDES dataset of DDI triplets, keeping from a total of 42,919,390 triplets the 26,037,737 containing drugs for which we had available all the four embeddings. By using One-Hot Encoding for the interactions we grouped the data to 106,917 drug pairs each having one or more interactions. These were split to 55,137 samples consisting the Seen set and 51,780 samples consisting the Unseen set.

1) Use of Interaction Categories

In order to reduce the dataset size with the use of OpenAI LLM we categorized each on the 12,227 discrete interactions to one or more of 55 interaction categories that we used instead. We also added a 56th category denoting "No-Interaction" that is used for the negative samples.

2) Negative Samples

Given the Closed World assumption [32] drug pairs not present in the dataset are considered as negative samples. For the 1,038 drugs found in the positive samples there are 970,545 possible negative samples. We make sure that the negative samples added in the Seen set contain only drugs already present in that set. For the negative samples added in the Unseen set we make sure that one drug comes from the Unseen and the other from the Seen set.

We have been sceptical for the use of negative samples, since they are not created from medical reports and are somewhat artificial in their nature. They also occupy very distinct centers in the kNN distribution of the OoD step thus skewing the results. We have decided to identify them in Step 2 and remove them from the dataset used in Step 3.

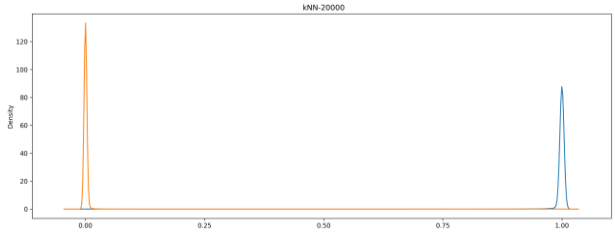

Fig. 4. kNN Density Plot of Negative (blue) vs Positive (orange) samples of the Seen set over a base population of the Negative samples of the Seen set.

3) Split Between Seen and Unseen Data

Each pair of drugs in the dataset is linked with one or more types of interactions which we categorized to one or more of the 56 categories of interactions. The dataset is split into Seen and Unseen pairs and our goal is to train over the Seen set and be able to predict novel interaction categories in the Unseen set. So the pairs associated with the categories that have the fewest occurrences, 18 and 45, were used for the Unseen set and the rest the Seen set. Note that the other categories are common for the Seen and Unseen sets and are present in both but the Seen set does not contain any pair that generates any of the two Unseen interaction categories, so the model is not trained for them.

B. *Embeddings*

1) BioBERT: We created the embeddings with the 'dmis-lab/biobert-v1.1' model and the corresponding tokenizer. where we passed the common name of each drug into the model and used the produced 'last_hidden_state' state of the model for a vector size of 768 elements, reduced to 128 elements through PCA.

2) ChemBERTa: We created the embeddings by using 'seyonec/PubChem10M_SMILES_BPE_450k' model and the corresponding tokenizer where we passed the SMILES representation of each drug into the model and used the produced 'last_hidden_state' state of the model for a vector size of 768 elements, reduced to 128 elements through PCA.

3) Bio2RDF Knowledge Graph: We used the embeddings provided by Graphembedding4DDI [26] in file 'Entity2Vec_sg_200_5_5_15_2_500_d5_uniform.txt' for each drug for a vector size of 200 elements, reduced to 128 elements through PCA.

4) Structural Similarity Profile SSP: We used the drug similarity vectors provided by DeepDDI [29] in file 'drug_similarity.csv' for each drug for a vector size of 2,159 elements reduced to 128 elements through PCA.

C. Main Experiment[2]

1) Interactions Probabilities Vector Generation

We created the Interactions Probabilities Vector (IPV) by using the seen pairs to train a fully connected DNN with 5 hidden layers with the "ReLU" activation function [39]. The 4 first hidden layers have 1024 neurons each while the last hidden layer has 256 neurons, each layer has a 0.25 dropout rate. The final output layer consists of 52 output neurons with the "sigmoid" activation function. The loss function used was BinaryCrossentropy and for the optimizer we used the Adam optimizer with a learning rate of 0.0001. We trained this DNN for 100 epochs with a batch size of 100 and a validation_split of 0.2. The Seen pairs were used as training data and the Unseen pairs as Test data.

The output metrics of this DNN for the Test data are presented in the following table:

TABLE I. TOTAL TRAINING METRICS OF THE IPV GENERATING DNN

Binary Accuracy	0.8747
Mean Square Error	0.0934
R2 score	0.6263
Precision	0.8985
Recall	0.9028

After the training we pass both the Seen and the Unseen samples through the DNN to generate their IPVs.

2) Negative Samples Detection

To identify the negative samples we used the OoD method. We used as base population only the negative samples in the Seen set and a k value of 20000. We then used the density plot of the kNN distribution of positive vs negative Seen samples to define the cutoff point of 0.05. Using this over the Unseen set we get a classification with the following confusion matrix, classification report and the ROC curve:

TABLE II. NEGATIVE SAMPLES DETECTION CONFUSION MATRIX

	Classified Positive	Classified Negative
Positive Samples	48188	3592
Negative Samples	5851	45929

TABLE III. NEGATIVE SAMPLES DETECTION CLASSIFICATION REPORT

	Precision	Recall	F1-Score	Support
Positive Samples	0.89	0.93	0.91	51780
Negative Samples	0.93	0.89	0.91	51780
Accuracy			0.91	103560
Macro Avg.	0.91	0.91	0.91	103560
Weighted Avg.	0.91	0.91	0.91	103560

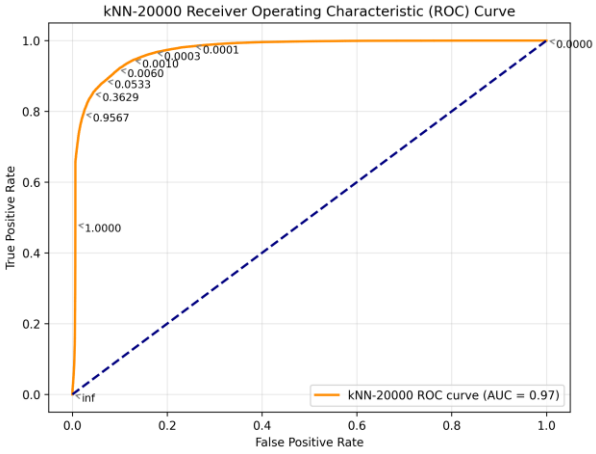

Fig. 5. ROC Curve of Negative vs Positive samples in the Unseen set.

These results can become even more accurate by using a more realistic mix of negative samples between seen and unseen datasets and by using a dedicated classification step (e.g. Neural Network, Logistic Regression etc.). Since these negative samples considerably skew the results of the following step, they have been somewhat artificially introduced in the samples and can be identified with high accuracy, we have decided to execute the next steps only with the positive samples.

3) OoD Samples Detection

To evaluate the OoD detection method more realistically we used as a Test set a dataset containing the Unseen set and an equal number of samples from the Seen set, excluding any negative samples from both, this resulted in a dataset of 103,560 samples. We used as base population only the positive samples of the Seen set and k=10. We then used the density plot of the kNN distribution to define the cutoff point of 0.745. Using this over the Test set we get a classification with the following confusion results:

TABLE IV. TEST SET KNN CONFUSION MATRIX

	Predicted OoD	Predicted InD
OoD	33558	18222
InD	13697	38083

TABLE V. TEST SET KNN CLASSIFICATION REPORT

	Precision	Recall	F1-score	Support
OoD	0.71	0.65	0.68	51780
InD	0.68	0.74	0.70	51780
Accuracy			0.69	103560
Macro Avg.	0.69	0.69	0.69	103560
Weighted Avg.	0.69	0.69	0.69	103560

[2] Code available in https://github.com/Dimitris-Lezos/NovelDDI-OoD

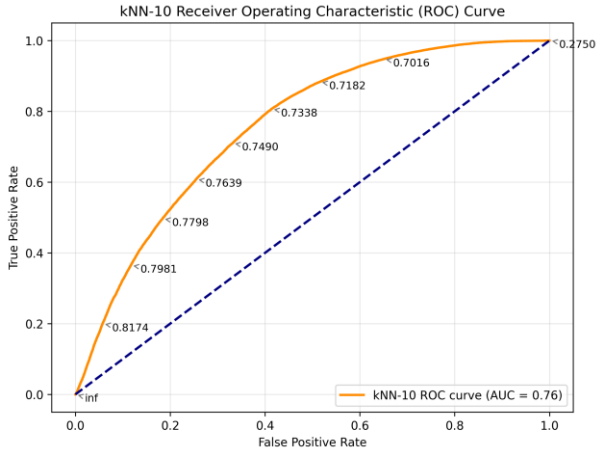

Fig. 6. ROC Curve of kNN distribution of the Test set.

D. Ablation

In order to investigate the importance of each embedding we conducted an ablation study, over a smaller setup and we used the following six test cases to compare:

- All four embeddings used (ALL-4)
- All but the SSP embeddings used (without SSP)
- All but the Bio2RDF Knowledge Graph embeddings used (without Bio2RDF)
- All but the BioBERT embeddings (without BioBERT)
- All but the ChemBERTa embeddings (without ChemBERTa)
- Only SSP and BioBERT embeddings (SSP-BioBERT)

The table below shows the results of each case we use Precision as our main indicator since we selected a InD/OoD cutoff value that maximized the Precision:

TABLE VI. OOD PRECISION AND CONFUSION MATRICES FOR EACH EMBEDDING CASE

Case	Precision	Confusion Matrix		
			Predicted OoD	***Predicted InD***
ALL-4	**0.701**	***OoD***	700	291
		InD	302	689
Without SSP	0.695	***OoD***	688	303
		InD	301	690
Without Bio2RDF	0.698	***OoD***	613	378
		InD	221	770
Without BioBERT	0.697	***OoD***	645	346
		InD	260	737
Without ChemBERTa	0.700	***OoD***	636	355
		InD	240	751
SSP-BioBERT	0.695	***OoD***	606	385
		InD	219	772

The results indicate that the SSP and BioBERT embeddings are the most significant ones but all four embeddings contribute to the results.

E. Conclusion / Discussion of Findings

This study successfully developed and validated a machine learning methodology for predicting novel drug-drug interactions not present in the training dataset. The promising results obtained highlight the model's ability to uncover previously unknown risks associated with drug combinations. A thorough ablation analysis was conducted to assess the individual contributions of the four drug embeddings utilized. The findings show that each embedding contributes unique and non-redundant information to the model's predictive performance. The observed decrease in accuracy when using only the two most influential embeddings underscores the importance of capturing diverse drug characteristics.

In the future we can focus on the 12,227 individual interactions as reported in TWOSIDES rather than on the 54 broad categories. We could also improve the distinction between In and Out of Distribution samples by following a methodology similar to DICE [40] that uses only the most significant weights and/or CIDER [41] that normalizes the embeddings over a hypersphere to maximize the distances between class centers. Also, we could investigate the case where novel drugs are completely missing from the training data, not only as pairs that generate interaction categories as in the current study. Furthermore, there are different ways to combine embeddings like Splicing, Matrix Convolution, Arithmetic mean, Self-attention. Finally, we can improve the embeddings in the SSP by working only on 585 drugs.

REFERENCES

[1] Lu, D. Y., Lu, T. R., Yarla, N. S., Wu, H. Y., Xu, B., Ding, J., & Zhu, H. (2017). Drug combination in clinical cancer treatments. Reviews on recent clinical trials, 12(3), 202-211.

[2] Fisusi, F. A., & Akala, E. O. (2019). Drug combinations in breast cancer therapy. Pharmaceutical nanotechnology, 7(1), 3-23.

[3] Asada, M., Miwa, M., & Sasaki, Y. (2018). Enhancing drug-drug interaction extraction from texts by molecular structure information. arXiv preprint arXiv:1805.05593.

[4] Luo, H., Yin, W., Wang, J., Zhang, G., Liang, W., Luo, J., & Yan, C. (2024). Drug-drug interactions prediction based on deep learning and knowledge graph: A review. Iscience, 27(3).

[5] Han, K., Cao, P., Wang, Y., Xie, F., Ma, J., Yu, M., ... & Wan, J. (2022). A review of approaches for predicting drug–drug interactions based on machine learning. Frontiers in pharmacology, 12, 814858.

[6] Li, X., Xiong, Z., Zhang, W., & Liu, S. (2024). Deep learning for drug‐drug interaction prediction: A comprehensive review. Quantitative Biology, 12(1), 30-52.

[7] Zhang, Y., Deng, Z., Xu, X., Feng, Y., & Junliang, S. (2023). Application of artificial intelligence in drug–drug interactions prediction: a review. Journal of chemical information and modeling, 64(7), 2158-2173.

[8] Nováček, V., & Mohamed, S. K. (2020). Predicting polypharmacy side-effects using knowledge graph embeddings. AMIA Summits on Translational Science Proceedings, 2020, 449.

[9] Zitnik, M., Agrawal, M., & Leskovec, J. (2018). Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics, 34(13), i457-i466

[10] Astras, N., & Vogiatzis, D. (2024, September). Prediction of Drug-Drug Interactions with Zero-Shot Learning. In Proceedings of the 13th Hellenic Conference on Artificial Intelligence (pp. 1-8).

[11] Wang, Z., Xiong, Z., Huang, F., Liu, X., & Zhang, W. (2024). ZeroDDI: A Zero-Shot Drug-Drug Interaction Event Prediction Method with Semantic Enhanced Learning and Dual-Modal Uniform Alignment. arXiv preprint arXiv:2407.00891.

[12] Zhu, F., Zhang, Y., Chen, L., Qin, B., & Xu, R. (2024). Learning to describe for predicting zero-shot drug-drug interactions. arXiv preprint arXiv:2403.08377.

[13] Tian, Y., Wang, Y., Krishnan, D., Tenenbaum, J. B., & Isola, P. (2020). Rethinking few-shot image classification: a good embedding is all you need?. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16 (pp. 266-282). Springer International Publishing.

[14] Wang, W., Zheng, V. W., Yu, H., & Miao, C. (2019). A survey of zero-shot learning: Settings, methods, and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2), 1-37.

[15] Xian, Y., Schiele, B., & Akata, Z. (2017). Zero-shot learning-the good, the bad and the ugly. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4582-4591).

[16] Pourpanah, F., Abdar, M., Luo, Y., Zhou, X., Wang, R., Lim, C. P., ... & Wu, Q. J. (2022). A review of generalized zero-shot learning methods. IEEE transactions on pattern analysis and machine intelligence, 45(4), 4051-4070.

[17] Tan, C., Xu, X., & Shen, F. (2021). A survey of zero shot detection: Methods and applications. Cognitive Robotics, 1, 159-167.

[18] Ubaid, M., & Hasan, M. T. (2023). Zero Shot Learning.

[19] Yang, J., Zhou, K., Li, Y., & Liu, Z. (2024). Generalized out-of-distribution detection: A survey. International Journal of Computer Vision, 132(12), 5635-5662.

[20] Romera-Paredes, B., & Torr, P. (2015, June). An embarrassingly simple approach to zero-shot learning. In International conference on machine learning (pp. 2152-2161). PMLR.

[21] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, 36(4), 1234-1240.

[22] Weininger, D. (1988). 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.

[23] Chithrananda, S., Grand, G., & Ramsundar, B. (2020, October 19). CHEMBERTA: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction. arXiv.org. https://arxiv.org/abs/2010.09885

[24] Pham, T., Ghafoor, M., Grañana-Castillo, S., Marzolini, C., Gibbons, S., Khoo, S., ... & Siccardi, M. (2024). DeepARV: ensemble deep learning to predict drug-drug interaction of clinical relevance with antiretroviral therapy. NPJ systems biology and applications, 10(1), 48.

[25] Dumontier, M., Callahan, A., Cruz-Toledo, J., Ansell, P., Emonet, V., Belleau, F., & Droit, A. (2014, October). Bio2RDF release 3: a larger connected network of linked data for the life sciences. In Proceedings of the 2014 international conference on posters & demonstrations track (Vol. 1272, pp. 401-404). Citeseer.

[26] Celebi, R., Uyar, H., Yasar, E., Gumus, O., Dikenelli, O., & Dumontier, M. (2019). Evaluation of knowledge graph embedding approaches for drug-drug interaction prediction in realistic settings. BMC bioinformatics, 20, 1-14.

[27] Hewett, M., Oliver, D. E., Rubin, D. L., Easton, K. L., Stuart, J. M., Altman, R. B., & Klein, T. E. (2002). PharmGKB: the pharmacogenetics knowledge base. Nucleic acids research, 30(1), 163-165.

[28] Ristoski, P., & Paulheim, H. (2016). Rdf2vec: Rdf graph embeddings for data mining. In The Semantic Web–ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part I 15 (pp. 498-514). Springer International Publishing.

[29] Ryu, J. Y., Kim, H. U., & Lee, S. Y. (2018). Deep learning improves prediction of drug–drug and drug–food interactions. Proceedings of the national academy of sciences, 115(18), E4304-E4311.

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

[31] Moriwaki, H., Tian, Y. S., Kawashita, N., & Takagi, T. (2018). Mordred: a molecular descriptor calculator. Journal of cheminformatics, 10(1), 4.

[32] Reiter, R. (1981). On closed world data bases. In Readings in artificial intelligence (pp. 119-140). Morgan Kaufmann.

[33] Sun, Y., Ming, Y., Zhu, X., & Li, Y. (2022, June). Out-of-distribution detection with deep nearest neighbors. In International Conference on Machine Learning (pp. 20827-20840). PMLR.

[34] Tatonetti, N. P., Ye, P. P., Daneshjou, R., & Altman, R. B. (2012). Data-driven prediction of drug effects and interactions. Science translational medicine, 4(125), 125ra31-125ra31.

[35] Knox, C., Wilson, M., Klinger, C. M., Franklin, M., Oler, E., Wilson, A., ... & Wishart, D. S. (2024). DrugBank 6.0: the DrugBank knowledgebase for 2024. Nucleic acids research, 52(D1), D1265-D1275.

[36] Kuhn, M., Letunic, I., Jensen, L. J., & Bork, P. (2016). The SIDER database of drugs and side effects. Nucleic acids research, 44(D1), D1075-D1079.

[37] Kanehisa, M. (2002, November). The KEGG database. In 'In silico'simulation of biological processes: Novartis Foundation Symposium 247 (Vol. 247, pp. 91-103). Chichester, UK: John Wiley & Sons, Ltd.

[38] Kim, S., Thiessen, P. A., Bolton, E. E., Chen, J., Fu, G., Gindulyte, A., ... & Bryant, S. H. (2016). PubChem substance and compound databases. Nucleic acids research, 44(D1), D1202-D1213.

[39] Agarap, A. F. (2018). Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375.

[40] Sun, Y., & Li, Y. (2022, October). Dice: Leveraging sparsification for out-of-distribution detection. In European conference on computer vision (pp. 691-708). Cham: Springer Nature Switzerland.

[41] Ming, Y., Sun, Y., Dia, O., & Li, Y. (2022). How to exploit hyperspherical embeddings for out-of-distribution detection?. arXiv preprint arXiv:2203.04450.