Modular Multi–Source Prediction of Drug Side–Effects With DruGNN

Pietro Bongini [ID], Franco Scarselli [ID], Monica Bianchini [ID], Giovanna Maria Dimitri,
Niccoló Pancino, and Pietro Lió [ID]

Abstract—Drug Side–Effects (DSEs) have a high impact on public health, care system costs, and drug discovery processes. Predicting the probability of side–effects, before their occurrence, is fundamental to reduce this impact, in particular on drug discovery. Candidate molecules could be screened before undergoing clinical trials, reducing the costs in time, money, and health of the participants. Drug side–effects are triggered by complex biological processes involving many different entities, from drug structures to protein–protein interactions. To predict their occurrence, it is necessary to integrate data from heterogeneous sources. In this work, such heterogeneous data is integrated into a graph dataset, expressively representing the relational information between different entities, such as drug molecules and genes. The relational nature of the dataset represents an important novelty for drug side–effect predictors. Graph Neural Networks (GNNs) are exploited to predict DSEs on our dataset with very promising results. GNNs are deep learning models that can process graph–structured data, with minimal information loss, and have been applied on a wide variety of biological tasks. Our experimental results confirm the advantage of using relationships between data entities, suggesting interesting future developments in this scope. The experimentation also shows the importance of specific subsets of data in determining associations between drugs and side–effects.

Index Terms—Deep learning, drugs side–effect prediction, graph neural networks, node classification, transductive learning

✦

1 INTRODUCTION

Dᴿᵁᴳ Side–Effects (DSEs) represent a common health risk, with an estimated 3.5% of all hospital admissions, and approximately 197,000 annual deaths, in Europe alone, related to adverse drug reactions [1]. Such adverse outcomes turn out to be extremely expensive for public care systems. Drug–related morbidity and mortality are estimated to have cost nearly 177.4 billion in the United States alone in the year 2000 [2]. As prescription drug use is increasing [3], the numbers and costs related to DSEs are also expected to rise. DSEs are a huge problem for pharmaceutical companies, as their occurrence during clinical trials slows down drug discovery processes and prevents many candidate molecules from being selected as commercial drugs [4]. Therefore, predicting DSEs before submitting a molecule to clinical trials is extremely important to avoid health risks for participants and cut drug development costs [5].

• *Pietro Bongini and Niccoló Pancino are with the Department of Information Engineering, University of Florence, 50121 Florence, Italy. E-mail: {pietro.bongini, nikypancino}@gmail.com.*
• *Franco Scarselli, Monica Bianchini, and Giovanna Maria Dimitri are with the Department of Information Engineering and Mathematics, University of Siena, 53100 Siena, Italy. E-mail: {franco, monica}@diism.unisi.it, giovanna.maria.dimitri@gmail.com.*
• *Pietro Lió is with the Department of Computer Science and Technology, University of Cambridge, CB2 1TN Cambridge, U.K. E-mail: pl219@cam.ac.uk.*

Computational prediction methods, and in particular deep learning methods, are techniques of growing importance in this scope [6]. DSEs are, in fact, triggered by complex biological mechanisms, involving interactions between different entities, such as drug functional groups, proteins, genes, and metabolic processes. As a consequence, an efficient predictor should be capable of processing heterogeneous data, accounting for the relationships among different data types [7]. In the last decade, DSE computational prediction methods have evolved from simple predictors based on euclidean data [5] or drug similarity [8] to Machine Learning (ML) methods based on Support Vector Machines [9] or clustering [10], and to more complex predictors based on Random Forests [11] or deep learning [6]. Current machine learning methods for DSE prediction have increased the number and variety of features considered for computing the predictions, but they are still widely based on euclidean (vectorial) data, whereas the relevant information for DSE prediction is relational in nature. This is a limit, since the relational information must undergo a preprocessing to be transformed into vectors, with an inevitable loss of information. In addition, preprocessing methods usually require re–thinking when new features are added.

In recent years, Graph Neural Networks (GNNs) [12] have become a solid standard for predicting and generating graph–structured data, thanks to their capability of processing relational data directly in graph form, with minimal loss of information, and in a flexible way [13]. After their introduction in 2005 [14], [15], many different models have been added to the GNN family, e.g., Graph Convolution Networks (GCNs) [16], spectral GCNs [17] [18], GraphSAGE [19], GraphNets [20], Message–Passing Neural Networks [21],

and Graph Attention Networks [22], just to mention the most important ones.

Models of the GNN family have repeatedly proved to be more efficient and accurate than non–graph–based predictors on many node, edge, and graph property prediction tasks. Moreover, GNNs have been employed in a wide variety of biological and chemical tasks [23], also including several drug–discovery related problems [24]. In particular, the GNN model used in the present study has been employed for the prediction of protein–protein interactions [25], and for the generation of molecular graphs of potential new drug candidates [26].

In this work, we propose a new method for single–drug side–effect prediction based on GNNs, and we build a graph dataset for this task, accounting for drug–gene, drug–drug, and gene–gene relationships. To the best of our knowledge, this is the first machine learning approach to be able to exploit directly graph structured relational data, for the prediction of single–drug side–effects. GNNs were already used for a related but different task, namely the prediction of polypharmacy side–effects. Polypharmacy side–effects are triggered by the combined use of two or more drugs. Direct and indirect interactions between the drugs are the key mechanisms behind these adverse reactions, which can be foreseen based on the structures of the drugs and their interactions with human genes. Predicting the probability of such events before prescribing the drugs can save the patient's health. This problem was addressed with GNNs both by analyzing the network of drugs and protein targets [27], and by applying a graph co–attention model over the two graphs describing a pair of drugs' structural formulas [7]. Differently, the side–effects of a single drug are mainly triggered by its interactions with the human organism, and can therefore be determined based on these interactions, on the structural features of the drug, and on similarities with other drugs for which the side–effects are known. Nevertheless, as far as we know, no single–drug side–effect predictor based on GNNs has been proposed yet. Moreover, we believe that predicting a single drug side–effect could answer an immediate research question, namely, which are the expected side–effects of this new candidate drug?

The main contributions of the paper are as follows.

- The first contribution of this work consists in the construction of a relational dataset for the prediction of DSEs, made with data coming from well–known publicly accessible resources. The dataset is a single heterogeneous graph, in which two types of nodes (drugs and genes) share three types of edges (drug–gene, drug–drug, and gene–gene relationships). Both drug and gene nodes have features, accounting respectively for their chemical properties and for their characteristics and function.

- The second contribution of our work consists in a GNN–based method, called DruGNN, for the prediction of DSEs on the new dataset we constructed. The prediction is set up as a multi–class multi–label node classification problem (applied only to drug nodes, and not to gene nodes), in which each DSE corresponds to a class. We adopt a mixed inductive–transductive learning

scheme [28], that exploits both the features of drugs and genes (induction path) and the information on the side–effects of known drugs (transduction path), in order to predict the side–effects of new drugs. The whole method is flexible, since the graph dataset can be easily extended to include other node features and further relationships without changing the machine learning framework [29].

- The approach has been assessed in an *in silico* experimentation setting, with very promising results, showing a good classification accuracy. The performance of DruGNN are compared to those of similar graph–based models (using the same inductive–transductive scheme) and to those of a deep Multi–Layer Perceptron (MLP) that cannot exploit relational information. Finally, two ablation studies, one over the set of side–effects, and the other over the set of features, show the model robustness and the contribution to learning brought by each single data source.

- The usability of DruGNN is discussed, as it can be exploited for the prediction of DSEs of new drugs without retraining. It is sufficient to add a new compound, or a batch of new compounds, each represented by a new graph node, and to predict their classes exploiting the same inductive–transductive learning scheme exploited for training and testing. Interesting future developments in this direction are also analyzed: the same approach could be replicated on a tissue–specific basis, by exploiting tissue–specific transcriptomics and DSE information.

The rest of this paper is organized as follows: Section 2 describes the dataset, its construction process, and the data sources; Section 3 sketches the GNN–based prediction method; Section 4 presents the results we obtained, and a discussion on their relevance and meaning; Section 5 discusses the expected use of our method; Section 6 draws conclusions on this work and summarizes the main results obtained. A more detailed description of the GNN model is provided in the Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2022.3175362.

2 DATASET

Computational methods for the prediction of DSEs have mainly relied on euclidean derived features so far. Even methods, like [30], that do use topological information (i.e., about the metabolic network), compress it into a euclidean space before processing. Since DSEs are triggered by complex biological phenomena, data for predicting DSEs are heterogeneous and come from multiple sources. Drug protein targets are of key importance, as highlighted by the good results of Sparse Canonical Correlation Analysis between drug targets and DSEs [5]. Chemical drug features play an important role too [31], as well as metabolic data [10]. Combining all these pieces of information, even in euclidean form, yields the best results when using deep learning predictors [6]. As a consequence, to build our dataset, we integrated information from all of these sources. The main novelty of our approach consists in building a graph
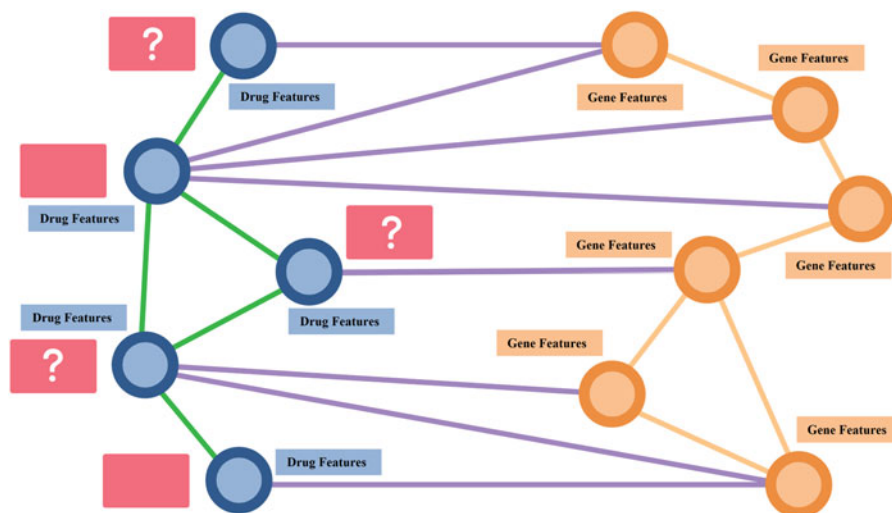
Fig. 1. Illustration of the graph composition. Drug nodes are represented as blue coloured circles, while gene nodes are represented as orange coloured circles. Red rectangles represent classes.

with these data, and processing the graph as it is, without forcing data objects into euclidean vectors of features.

Our dataset consists of a single graph, in which each drug, as well as each gene, is mapped to a node. Both drug nodes and gene nodes are described by feature vectors. Edges represent drug–drug relationships, drug–gene interactions, and gene–gene interactions. Side–effect labels are associated to each drug node. These label will be used, according to the inductive–transductive scheme, as either transductive features for known drugs, or class supervisions for new drugs. A sketch of the graph is provided in Fig. 1.

We chose to use gene nodes instead of protein nodes, because genes are more informative than proteins, as a gene node allows us to summarize all the information concerning the gene itself and its various protein products. Using proteins, each of our gene nodes would correspond to a subgraph of some protein nodes and it would be difficult to track and manage all the relationships between products of the same gene and the rest of the graph.

The associations between drugs and side–effects were downloaded from the SIDER database [32], which collects DSE information by aggregating multiple public information sources, summing up to 5,868 side–effects occurring on 1,430 drugs, with a total of 139,756 entries, each accounting for the association of a single drug to a specific side–effect. In our graph, a node was created for each drug. Each side–effect corresponds to a class. Our set of gene nodes, as well as the gene–gene edges, representing the interactions between two genes or their products, were constructed by downloading protein–protein interactions (PPI) information from the Human Reference Interactome (HuRI) [33], and mapping each protein to the gene it is a product of. The product–gene associations were instead obtained from Biomart [34]. Drug–protein interactions (DPI) were downloaded from the STITCH database [35], one of the most complete and up–to–date DPI databases available. Once again, using Biomart, each protein was mapped to the gene it is a product of, obtaining the links between drug nodes and gene nodes.

Drug features were retrieved from PubChem [36], which provides seven chemical descriptors for each molecule in our dataset, as well as the SMILES string describing its structure. The seven chemical descriptors consist in: molecular weight (MW), polar surface area (PA), xlogp coefficient (LP), heavy atom count (AC), number of hydrogen bond donors (HD), number of hydrogen bond acceptors (HA), and number of rotatable bonds (RB). In order to better describe each drug molecule, we also translated its SMILES representation to the corresponding structural formula, and extracted its substructure fingerprint, using RDKit software.[1] In order to keep the feature vector size of drug nodes similar to that of gene nodes (gene feature vectors are 140–dimensional and will be described in the following), we opted for drug substructure fingerprints of size 128, bringing the total size of the drug feature vector to 135.

Drug substructure fingerprints were also exploited to build the drug–drug set of edges, accounting for similarity relationships between molecules. In particular, we measured the Tanimoto similarity [37] of the fingerprints of each pair of drugs, adding an edge only to those pairs which were above a similarity threshold (which was set as a hyperparameter at graph construction). Fingerprints were extracted with RDKit again, but with size 2048, in order to better estimate the Tanimoto similarity. This similarity coefficient takes into account the substructure groups two molecules have in common, by calculating the distance between their fingerprints. The Tanimoto coefficient is inversely proportional to this distance and represents the best and more common measure of similarity between molecules [38]. Given the fingerprints Fp_a and Fp_b of two drugs a and b, the Tanimoto similarity between the two is defined as in Eq. (1)

$$T(a, b) = \frac{Fp_a \times Fp_b}{\|Fp_a\|^2 + \|Fp_b\|^2 - Fp_b \times Fp_a} \tag{1}$$

Gene features were obtained from two sources. Biomart [34] provided 3 pieces of information: the chromosome, which was one–hot encoded for a total of 25 features
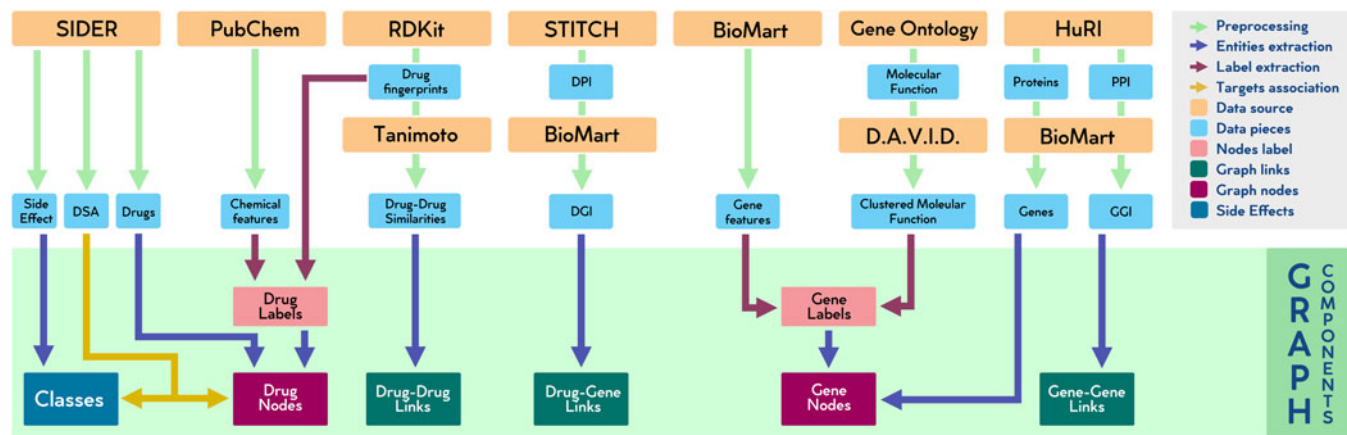
Fig. 2. Sketch of the dataset construction. Each data source is represented by an orange rectangle. Cyan rectangles represent data pieces. Preprocessing steps are represented by green arrows, which can include feeding data in input to other sources to obtain refined data. Graph node subsets are represented by purple rectangles, with their labels sketched as pink rectangles. Green rectangles are subsets of graph edges, while the blue rectangle represents the classes (side–effects). Red arrows represent the composition of feature labels from data pieces, while blue arrows show the composition of graph entities (nodes, edges, classes). The yellow arrow represents the association of drug nodes to side–effect classes.

(22 regular chromosomes, plus X, Y, and mitochondrial DNA); the strand the gene is codified on ($+1$ or -1); the percentage of GC content. Chromosomal location suggests drug impact on epistatic mechanisms or chromatin environmental changes. Gene ontology [39] provided the molecular function ontology terms to which each gene is mapped. Gene Ontology is composed of three different ontologies. While the cellular component ontology is not interesting in our case, the molecular function and biological process ontologies are highly interconnected, with each one being highly informative over the other. The molecular function ontology has a precise semantics while the biological process ontology lacks this feature. As a consequence we decided to use the molecular function ontology. The ontology terms describe the molecular function of each gene, which, combined to the gene–gene interaction links, allow to reconstruct the metabolic network. Since Gene Ontology mapped our genes to a total of 3,422 terms, we clustered the terms to those appearing at the higher levels of the molecular function ontology, using DAVID [40] [41]. This produced 113 unique terms, which were one–hot encoded and concatenated to the gene features obtained from Biomart (for a total of 140 features on each gene node).

We subsequently selected only side–effects with a sufficient number of occurrences in SIDER: In order for the network to be able to learn the associations of each side–effect, we applied a minimum threshold of 100 occurrences, reducing the number of side–effects in our dataset to 360. After this first filtering step, drugs without side–effects were also removed, reducing the number of drug nodes in the graph to 1,341. Genes with incomplete features were also discarded, along with their gene–gene interactions, bringing the number of gene nodes to 7,881. All the drugs have complete feature vectors, and at least one DPI. DPIs and DSEs of removed drugs were also removed. Drugs are mapped to 360 classes (one for each side–effect), with 96,477 total positive occurrences, and 515,160 negative ones. In this sense, belonging to the positive class for a drug means that it produces the particular side–effect. The target for each drug must be evaluated in relation to every possible side effect (360) since the addressed problem is both multi–class and multi–label. A total of 331,623 edges is instead present in the final version of the graph: 12,002 gene–gene interaction links, 314,369 DPI, and 5,252 drug–drug similarity links (with a minimum Tanimoto threshold of 0.7). The dataset construction, with all the source databases and preprocessing steps, is sketched in Fig. 2.

3 METHOD

Graph Neural Networks (GNNs) have seen an important and steady development in the last decade. Their main feature is the capability of processing graph–structured data with minimal information loss [15]. In our work, we exploit the original GNN model [12], and in particular its most recent implementation [25], to build a DSE predictor called DruGNN. An overview of the GNN model is provided in the Appendix, available in the online supplemental material, of this paper. The final task is to predict the label of drug nodes only, solving a node–based classification problem, with multiple classes (360 side–effects) and in a multi–label setting (each drug can cause multiple side–effects). In order to provide repeatable and comparable results, we set a dataset split and always use that split throughout the experimentation. The test set contains 10% of the drug nodes and is fed to the network only at test time. In our experiments, we also retain a 10% of the drug nodes as a validation set, in order to check overfitting and stop the training procedure when this occurs. The rest of the nodes (80%) is exploited as a training set.

In our work, we make use of a mixed inductive–transductive learning scheme [29], in which the network learns the node–class associations exploiting a double mechanism. In standard inductive learning, the GNN model would predict the side–effects of drugs based on the node features of drugs and genes, and the graph connectivity. In a transductive learning setup, the GNN model would make the predictions based on the known side–effects of other drugs. In our mixed inductive–transductive learning scheme, the GNN model exploits both mechanisms at the same time.

The learning scheme applied in this work consists in splitting the training set into ten batches. The network

TABLE 1
Hyperparameter Values Analyzed During the Grid
Search Procedure, and Best Configuration Obtained

Hyperparameter	Values	Best Config.
Activation	relu, selu, tanh, sigmoid	relu
Initial Learning Rate	$10^{-2}, 10^{-3}, 10^{-4}$	10^{-3}
State Dimension	10,50,100,200	50
Hidden Units	100,200,500	200
Neighborhood Aggregation	average, sum	average
Dropout Rate	0.5, 0.3, 0.1, 0.0	0.0

learns the input–supervision association on one training batch at a time, while the other nine batches are exploited as a transduction set. The features of each drug node in the transduction set are augmented with the transductive features, corresponding to the occurrence of the 360 side effects on that node. When analyzing the validation set, the full training set is exploited, in the same way as described before, as the transduction set. When analyzing the test set, the transduction set is composed of both the validation set and the training set.

This scheme is particularly appropriate for the expected use of our dataset and tool: the idea is to exploit the known DSE associations to predict the DSEs of newly inserted drugs, and the mixed inductive–transductive scheme simulates this behaviour at training, validation, and test times.

The network hyperparameters were tuned with an extensive grid–search over the validation set. In particular, we analyzed all the hyperparameter values described in Table 1 and their combinations. Each element in the grid was analyzed by measuring the average model accuracy in a training/validation experiment with five repetitions.

After tuning the hyperparameters, in order to check the learning capabilities of the DruGNN on our dataset, and in particular the effect on the learning process of the reduction or expansion of the set of side–effects, we set up a dedicated series of experiments. In this part of the experimentation, which consists in an ablation study over the set of side–effects, our model was trained and tested on versions of our dataset with progressively reduced numbers of side–effects: only the most k common side–effects were retained, with k assuming values $\{360, 240, 120, 80, 40, 20, 10, 5\}$.

To evaluate the importance of the contributions of the different data sources, we carried out another ablation study. We grouped the features and the edges by source and eliminated one feature/edge group at a time from the dataset, evaluating the performance of the model in absence of that group. The performance gap obtained gives an estimate of the importance of the features that were kept out. There are seven feature/edge groups in our dataset, each of which was analyzed in an experiment repeated five times. Once again, we always used the same dataset split and the same transductive learning scheme described for the previous experimentation.

Eventually, the DruGNN was compared to other competitive GNN models with different characteristics, in order to assess its performance with respect to the alternative

solutions. In particular we focused on two powerful models: GCNs [16], which exploit convolutions to aggregate information coming from different locations across the graph — and have shown competitive performance on many different tasks; GraphSAGE [19], which are versatile networks that can be configured with various aggregation and state updating functions—being potentially competitive on every graph dataset. Additionally, we also compared to a simple Multi–Layer Perceptron (MLP), in order to assess the difference between a graph–based model and a euclidean predictor. It was not possible to include previously published DSE predictors in the comparison, as our dataset is completely novel, and graph–structured, making it impossible to adapt to the feature sets of the predictors available in the literature (we remind that no graph–based predictor was published for this task). In particular, after a small optimization over the validation set, we used a three–layered MLP.

4 RESULTS AND DISCUSSION

The hyperparameter search described in Section 3 produced a model with an accuracy over the validation set of 87.22%. The same model, evaluated on the held–out test set obtained an accuracy of 86.30%. Since the DSEs are unbalanced classes in function of their frequency of occurrence in the dataset, we investigated their frequency distribution and its effect on model accuracy. Fig. 3 shows the histogram of class frequencies, and the average accuracy and standard deviation on each bin of the histogram. While our filtering steps allowed to exclude the large bulk of side–effects with just few occurrences, most of the classes still have low frequencies. This means that the accuracy of the model is expected to be high in these cases, and low for more frequent side–effects, as the model is globally encouraged to predict non–occurrences. Interestingly, the accuracy instead shows only a mild tendency to deteriorate for DSEs of medium frequency to then rise again for very frequent ones. Globally, the model shows to be able to manage the unbalanced classes.Given the best model configuration obtained in this first set of experiments, we investigated the contribution of the side–effects to the learning capability of the network. We ranked the side–effects by occurrences, and then we progressively reduced the size of the set of side–effects, by selecting only the most common ones. The average accuracy over five repetitions was measured over the held–out test set. Results are reported in Table 2.

Since we are dealing with a multi–class multi–label classification task, each class membership can be seen as a problem to be learned independently and in parallel with respect to all the other classes. As a consequence, the first expectation would be that increasing the number of classes, the network would have to learn a more complex algorithm, needing to solve more problems in parallel. On the contrary, the results reported in Table 2 show a clear tendency of improvement of the performance for larger sets of side–effects. This counter–intuitive behaviour is due to the network ability of learning intermediate solutions, which are useful for all or large subsets of the classes, with an effect very similar to transfer learning. This is particularly evident in our system, in which transfer learning between classes is fundamental because of the relatively small dimension of
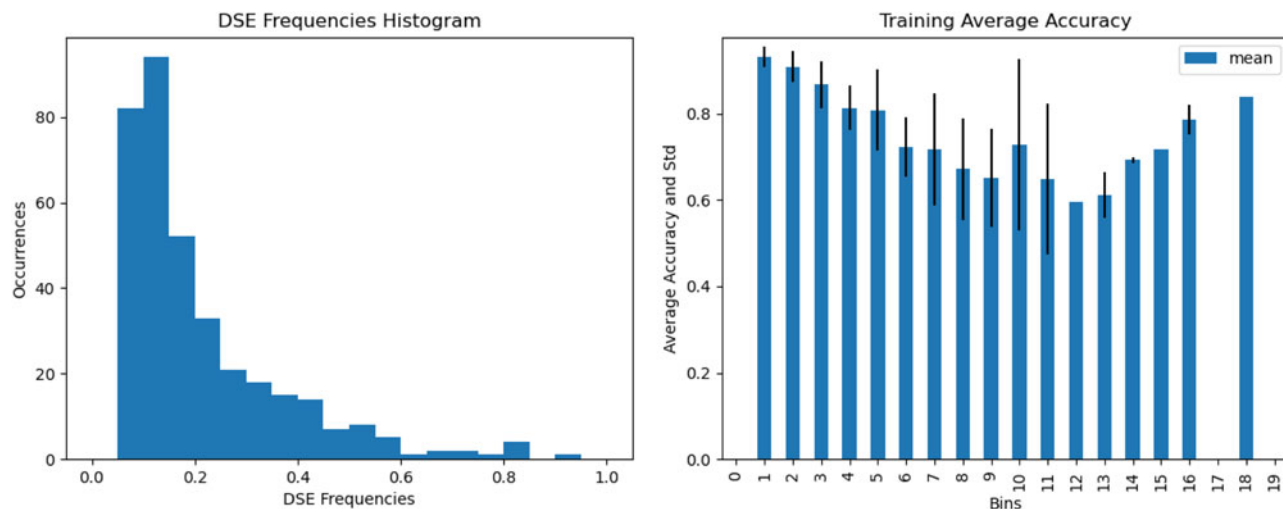
Fig. 3. Histogram of the frequency distribution of DSEs in our dataset (Left) and average classification accuracy of DruGNN by DSE frequency (Right). The histogram is composed of 20 bins of equal width in the frequency range (0-1). The same bins are used to observe the model class–specific accuracy in function of the class frequency: each column corresponds to the average accuracy over the corresponding histogram bin. Standard deviation is also shown (black line). Please notice that no accuracy is provided for bins 0, 17, and 19, because they are empty: no DSE falls in the corresponding frequency sub–range, as shown in the histogram. In the bin 0 case, this is caused by our filtering options, described in Section 2.

the set of drugs, with the additional bonus of avoiding over-fitting. An inversion of this behaviour can be observed at lower set dimensions (up to 20), where transfer learning becomes less easy and convenient and the network learns to treat each class independently. The unbalanced nature of the problem also plays a role, though. The side–effects with less occurrences are highly unbalanced in favour of the negative class, while the DSEs with more occurrences are unbalanced in favour of the positive class. The balance shift as less common DSEs are removed likely plays an important role in this scope.

A second ablation study was carried out on the feature/ edge groups coming from different data sources. The accuracy of the model, trained and tested in absence of the data group, was evaluated and averaged over five repetitions of the same experiment. Since the groups of features are of different sizes, to better weigh the importance of each, we also measured the DPF (Difference Per Feature) score: this is the performance difference with respect to the complete model, divided by the number of features in the group. The description, and the corresponding performance loss observed in the ablation study, of each data group, are reported in Table 3.

Table 3 shows that each data source has a positive contribution on the GNN learning process. In particular, deleting the drug fingerprints brings the largest performance drop. Substructure fingerprints are efficient embeddings for the molecular structure [42], which was previously demonstrated

to be fundamental to determine the side–effects of drugs [6]. Which can be explained by the importance of the drug substructures in determining the side–effects, but also by the large number of features (128) assigned to this data group.

Proportionally, looking at the DPF score, the seven Pub-Chem descriptors have the highest contribution, as it could be expected given their chemical relevance. The gene features also have a relevant impact on performance, with the Biomart derived features having a DPF equal to that of drug fingerprints. Edges also showed to be important, as deleting each edge set leads to a performance drop. Instead results suggest that drug similarity relations are the less important, likely because drug similarity can be inferred by the network on the basis of the fingerprints and of the drug–gene interactions.

Although each group of features and edges has a positive contribution to model performance, the small performance drop obtained by switching them off tells us that the model is robust. In fact, it works almost as well as the complete

TABLE 3
Average Accuracy Percentage (Acc.%), and Difference With Respect to the Model Trained on the Complete Feature Set (Diff.), Obtained on the Test Set by Training and Testing the Model on Our Dataset in Absence of the Corresponding feature/ edge Group (Group)

Group	DPI	PPI	DDS	FP	PC	GO	BM
Type	E	E	E	DF	DF	GF	GF
Acc.%	86.14	86.18	86.23	85.81	86.20	86.17	86.20
Diff.	0.16	0.12	0.07	0.49	0.10	0.13	0.10
Count	-	-	-	128	7	113	27
DPF	-	-	-	0.004	0.014	0.001	0.004

Each group is associated to a type: E (Edges), DF (Drug Features), GF (Gene Features). For DF and GF data groups, the number of features (Count) and the difference per feature (DPF) are reported: the latter is obtained by dividing the difference (Diff.) by the number of features in the group (Count). Our data groups are: DPI (Drug–Protein Interactions), PPI (Protein–Protein Interactions), DDS (Drug–Drug Similarity), FP (FingerPrints), PC (PubChem), GO (Gene Ontology), and BM (BioMart).

TABLE 2
Average Accuracy Percentage (Acc.%), and Average Balanced Accuracy Percentage (Bal.%) Obtained on the Test Set by Training and Testing the Model on Progressively Smaller Sets of Side–Effects (DSEs)

DSE	360	240	120	80	40	20	10	5
Acc.%	86.3	81.5	73.2	68.5	63.0	61.8	67.1	74.7
Bal.%	58.1	58.6	60.0	60.5	62.1	59.9	57.7	56.2

TABLE 4
Comparison Between Different Models of the GNN Family

Model	Configuration	Avg. Acc. %
DruGNN	$K = 6, SD = 50, DL = 1 \times 200$	86.30%
GCN	$CL = 2 \times 36, DL = 116$	82.94%
GraphSAGE	$CL = 2 \times 72, DL = 1 \times 168$	83.11%
MLP	$DL = 3 \times 25$	77.98%

Model configuration is reported, all of the models were optimized with a small hyperparameter search. K: maximum number of state update iterations for DruGNN; SD: state dimension for DruGNN; DL: number of dense layers and units in each dense layer; CL: number of convolutional layers and units in each convolutional layer. For the GCN and the GraphSage, the dense layer is the last one before the output layer.

version even when entire sets of edges or features are deleted. We can therefore hypothesize the following. On the one hand, GNNs are expected to be robust, on the basis of previous systematic ablation studies that demonstrated their capabilities on many types of graph datasets [43]. On the other hand, the large quantity of features and edges, and the heterogeneous nature of our data sources, likely boost the model's robustness.

Moreover, to assess the capabilities of DruGNN with respect to other GNN variants, and with respect to non–graph–based euclidean models, we carried out a comparison with GraphSage [19], GCNs [16], and with a simple Multi–Layer Perceptron (MLP) model trained on a vectorized version of our drug data. The MLP gives a measure of the results that can be achieved by applying a traditional euclidean predictor on our dataset. The GCN and the GraphSage are trained with the same transductive scheme as DruGNN. All the models were trained with the binary cross–entropy loss function, Adam optimizer [44], and an initial learning rate equal to 10^{-4}. A maximum of 500 epochs was allowed for each model, with early stopping on the validation loss, and recovery of the best weights. As expected, all the graph–based models outperformed the standard MLP, showing the advantage given by representing the dataset as relationships on a graph and by learning directly on the graph structure. Moreover, using a GraphSAGE or a GCN approach on this task did not allow to reach the same results we obtained with DruGNN, as shown in Table 4. This can be explained by the fact that our model is particularly efficient on node property prediction tasks, as the one presented in the current paper, while the other GNN models tend to aggregate nodes on a larger scale, getting an advantage on graph property prediction tasks. This is also in line with theoretical studies on GNNs that demonstrate the processing capabilities by simulating the Weisfeiler–Lehman test [43]. Model evaluation is based on the average accuracy percentage obtained over 10 runs of training and testing on the same dataset split.

5 USABILITY OF DRUGNN IN REAL PRACTICE

DruGNN is meant as a tool of real usage, that can help healthcare and pharmacology professionals to predict side–effects of newly discovered drugs or other compounds not yet classified as commercial drugs. The dataset and the software are publicly available on GitHub,[2] so that both assets can be exploited in further scientific research and by the

whole community. The datasets we used for training DruGNN can be considered as de facto standards in terms of experimental results included. SIDER [32] represents one of the most comprehensive databases available for drug–side–effect associations. The same could be stated for the Human Reference Interactome (HURI) [33], Biomart [34], Stitch [35], and Pubchem [36]. Being these datasets comprehensive of the most up to date relevant biological information, and considering the fact that no unique assessment of the best dataset to be used in these cases exists [45], we believe that with DruGNN we provide an efficient tool, robust to possible variations that might appear considering different biological data sources. Furthermore, both the dataset and the algorithm are scalable: adding new compounds to predict their side–effects does not compromise the network usability (i.e., the network does not need to be retrained from scratch). In fact, GNNs do not learn the data configuration itself, they rather generalize the processes of message passing and state updating, which maintain their validity regardless of the modifications to the graph structure [12].

An example of such usage is represented by the prediction of the side–effects of Amoxicillin (PubChem CID: 2171), which is part of the held–out test set (and therefore never seen during the training or validation phases). Amoxicillin has been determined to be similar to the following drugs, listed by PubChem CID: 2173, 2349, 2559, 4607, 4730, 4834, 8982, 15232, 22502, 6437075. It also interacts with 76 genes. No other information but the fingerprint and PubChem features of Amoxicillin are available to the model. The network correctly predicts the following side–effects, listed by the SIDER id: C0000737 (Abdominal pain), C0001824 (Agranulocytosis), C0002792 (Anaphylactic shock), C0002871 (Anaemia), C0002878 (Haemolytic Anaemia), C0003467 (Anxiety), C0006840 (Candida infection), C0011991 (Diarrhoea), C0012833 (Dizziness), C0013378 (Dysgeusia), C0015230 (Rash), C0017178 (Gastrointestinal disorder), C0018681 (Headache), C0019080 (Haemorrhage), C0027497 (Nausea), C0033774 (Pruritus), C0038362 (Stomatitis), C0042075 (Urinary tract disorder), C0042109 (Urticaria), C0042963 (Vomiting), C0267792 (Hepatobiliary disease), C0917801 (Insomnia). It fails to predict these side–effects: C0002994 (Angioedema), C0008370 (Cholestasis), C0009319 (Colitis), C0011606 (Dermatitis exfoliative), C0014457 (Eosinophilia), C0036572 (Convulsion). Please notice that Angioedema, Colitis, Dermatitis exfoliative, and Convulsion are indicated as very rare for Amoxicillin. Cholestasis has relatively few occurrences in the dataset, and is therefore difficult to predict. Moreover, the network shows good predictive capabilities on side–effects which are common in the whole drug class Amoxicillin belongs to (represented by the similar compounds in the dataset). In addition, the network predicts only one side–effect which is not associated to Amoxicillin in the supervision: C0035078 (Renal failure).

As shown in the example, to predict the side–effects of a new compound, it is sufficient to retrieve information (coming from wet–lab studies and from the literature) on its interactions with genes, and to know its structural formula. A possible limitation of this approach is represented by the difficulty in determining the drug–gene interaction of a

2. https://github.com/PietroMSB/DrugSideEffects

newly discovered drug, though we expect some wet–lab studies could provide at least some basic information before submitting the compound to DSE prediction and then, eventually, to clinical trials. RdKit can be used to calculate the fingerprint, and consequently the similarity to other drugs in the dataset. The PubChem features can either be obtained from a database, or calculated with RdKit. It is then sufficient to insert the compound in the dataset and to predict its side–effects with DruGNN. Visualisation of the DruGNN results and the excepts from the database could then be directly used by doctors and pharmacists.

6   CONCLUSION

Combining data from multiple sources is crucial for a deep neural network to learn complex mechanisms regulating the occurrence of drug side–effects. In particular, the relational information on the interactions of drugs and genes is well described by a graph structure. Integrating these entities and their relations, we built a graph dataset thought for training and testing graph–based DSE predictors. Graph Neural Networks (GNNs) showed very good learning capabilities on this dataset, suggesting that a predictor based on GNNs could help anticipate the occurrence of side–effects. Furthermore, its application on new candidate drugs would help saving time and money in drug discovery studies, also preventing health issues for the participants to the clinical tests.

DruGNN is a modular approach to DSE prediction and is robust to ablation. Moreover, it is easily usable on new drug compounds: it is sufficient to add the new drug, with its features and gene interactions, as a node in the graph, and to run the prediction of its classes. The model does not need retraining, and the same inductive–transductive learning scheme can be used for future additions of compounds and predictions of their side–effects. The prediction relies on a modular multi–omics robust approach, based on information retrieved from publicly available sources. In principle, the same graph could be exploited also to predict the drug–gene interactions of new compounds, by applying link prediction over the gene set.

Since drug structure fingerprints are a very important piece of information in our dataset, a limitation of our approach is the loss of information implied by using fingerprints instead of the full drug structure. Another limitation of DruGNN is the lack of tissue specific information which could be exploited to predict side–effects on a tissue–specific basis. Indeed, the level of interaction between a drug and a gene depends on the tissue: taking this variability into account is important to build a more accurate predictor.

Consequently, an interesting future direction is represented by the development of a GNN–based predictor that could analyse the structural formulas of the molecules, represented as graphs. These molecular graphs could be augmented with features coming from the gene side and drug–gene relations. In this scope, the algorithm could even be combined with generative models, like MG^2N^2 [26], that generate molecular graphs of possible drug candidates in large quantities. The task of the DSE predictor would be to screen out all the candidate compounds with high probabilities of occurrence of particular side–effects. A suitable variant of this approach would be to exploit a hierarchical GNN model to analyze a graph of graphs: the higher level graph would be analogous to our dataset, while the lower level would be represented by the structural graphs of the compounds. The structural graphs might also include spatial features in the labels of drug nodes and bond edges. This improvement is expected to have a cost in terms of computational complexity, though.

Another very interesting direction is that of specializing the predictor presented in this work, in order to take into account tissue–specific data (i.e., gene expression) and fine–tune a dedicated version of the model for each tissue. This could be made possible by exploiting tissue specific side–effect targets, leading to a more detailed prediction which could also be personalized, given the gene expression values of each individual, as expected in the context of precision medicine. The architecture of DruGNN allows the use of tissue transcriptomics (from Gtex data) and tissue methylomics, although implementing this will require further experimental tests which we plan as future work.

Also including a layer of protein nodes between drugs and genes would be possible, yet we plan to keep the dataset as simple as possible, and using genes looks as the best tradeoff. In fact, there is a lack and sparsity of information on protein abundances. Instead, the information on gene expression is particularly abundant. Information on alternative splicing will be also relevant. Further development on tissue specific drug response will be based on tissue gene–related resources such as GTEX.

REFERENCES

[1]   H. Khalil and C. Huang, "Adverse drug reactions in primary care: A scoping review," *BMC Health Serv. Res.*, vol. 20, no. 1, pp. 1–13, 2020.
[2]   F. R. Ernst and A. J. Grizzle, "Drug–related morbidity and mortality: Updating the cost–of–illness model," *J. Amer. Pharmaceut. Assoc.* (1996), vol. 41, no. 2, pp. 192–199, 2001.
[3]   E. D. Kantor, C. D. Rehm, J. S. Haas, A. T. Chan, and E. L. Giovannucci, "Trends in prescription drug use among adults in the United States from 1999–2012," *Jama*, vol. 314, no. 17, pp. 1818–1830, 2015.
[4]   M. L. Billingsley, "Druggable targets and targeted drugs: Enhancing the development of new therapeutics," *Pharmacology*, vol. 82, no. 4, pp. 239–244, 2008.
[5]   S. Mizutani, E. Pauwels, V. Stoven, S. Goto, and Y. Yamanishi, "Relating drug–protein interaction network with drug side effects," *Bioinformatics*, vol. 28, no. 18, pp. i522–i528, 2012.
[6]   O. C. Uner, R. G. Cinbis, O. Tastan, and A. E. Cicek, "DeepSide: A deep learning framework for drug side effect prediction," 2019, *bioRxiv:843029*.
[7]   A. Deac, Y. H. Huang, P. Veličković, P. Liò, and J. Tang, "Drug–drug adverse effect prediction with graph co–attention," 2019, *arXiv:1905.00534*.
[8]   W. Zhang, Y. Chen, S. Tu, F. Liu, and Q. Qu, "Drug side effect prediction through linear neighborhoods and multiple data source integration," in *Proc. IEEE Int. Conf. Bioinf. Biomed.*, 2016, pp. 427–434.
[9]   I. Shaked, M. A. Oberhardt, N. Atias, R. Sharan, and E. Ruppin, "Metabolic network prediction of drug side effects," *Cell Syst.*, vol. 2, no. 3, pp. 209–213, 2016.
[10] G. M. Dimitri and P. Liò, "DrugClust: A machine learning approach for drugs side effects prediction," *Comput. Biol. Chem.*, vol. 68, pp. 204–210, 2017.
[11] A. Cakir, M. Tuncer, H. Taymaz-Nikerel, and O. Ulucan, "Side effect prediction based on drug–induced gene expression profiles and random forest with iterative feature selection," *Pharmacogenomics J.*, vol. 21, pp. 673–681, 2021.

[12] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[13] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[14] F. Scarselli, S. L. Yong, M. Gori, M. Hagenbuchner, A. C. Tsoi, and M. Maggini, "Graph neural networks for ranking web pages," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell.*, 2005, pp. 666–672.

[15] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "Computational capabilities of graph neural networks," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 81–102, Jan. 2009.

[16] T. N. Kipf and M. Welling, "Semi–supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Representations*, 2017.

[17] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and deep locally connected networks on graphs," in *Proc. 2nd Int. Conf. Learn. Representations*, 2014.

[18] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[19] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.

[20] P. W. Battaglia et al., "Relational inductive biases, deep learning, and graph networks," 2018, *arXiv:1806.01261*.

[21] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1263–1272.

[22] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[23] J. Zhou et al., "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[24] J. Kim, S. Park, D. Min, and W. Kim, "Comprehensive survey of recent drug discovery using deep learning," *Int. J. Mol. Sci.*, vol. 22, no. 18, 2021, Art. no. 9983.

[25] N. Pancino et al., "Graph neural networks for the prediction of protein–protein interfaces," in *Proc. Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn.*, 2020, pp. 127–132.

[26] P. Bongini, M. Bianchini, and F. Scarselli, "Molecular generative graph neural networks for drug discovery," *Neurocomputing*, vol. 450, pp. 242–252, 2021.

[27] M. Zitnik, M. Agrawal, and J. Leskovec, "Modeling polypharmacy side effects with graph convolutional networks," *Bioinformatics*, vol. 34, no. 13, pp. i457–i466, 2018.

[28] A. Rossi, M. Tiezzi, G. M. Dimitri, M. Bianchini, M. Maggini, and F. Scarselli, "Inductive–transductive learning with graph neural networks," in *Proc. IAPR Workshop Artif. Neural Netw. Pattern Recognit.*, 2018, pp. 201–212.

[29] G. Ciano, A. Rossi, M. Bianchini, and F. Scarselli, "On inductive-transductive learning with graph neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 758–769, Feb. 2022.

[30] Y. Yamanishi, M. Kotera, M. Kanehisa, and S. Goto, "Drug–target interaction prediction from chemical, genomic and pharmacological data in an integrated framework," *Bioinformatics*, vol. 26, no. 12, pp. i246–i254, 2010.

[31] E. Pauwels, V. Stoven, and Y. Yamanishi, "Predicting drug side–effect profiles: A chemical fragment–based approach," *BMC Bioinf.*, vol. 12, no. 1, pp. 1–13, 2011.

[32] M. Kuhn, I. Letunic, L. J. Jensen, and P. Bork, "The SIDER database of drugs and side effects," *Nucleic Acids Res.*, vol. 44, no. D1, pp. D1075–D1079, 2016.

[33] K. Luck et al., "A reference map of the human binary protein interactome," *Nature*, vol. 580, no. 7803, pp. 402–408, 2020.

[34] D. Smedley et al., "The BioMart community portal: An innovative alternative to large, centralized data repositories," *Nucleic Acids Res.*, vol. 43, no. W1, pp. W589–W598, 2015.

[35] D. Szklarczyk, A. Santos, C. Von Mering, L. J. Jensen, P. Bork, and M. Kuhn, "STITCH 5: Augmenting protein–chemical interaction networks with tissue and affinity data," *Nucleic Acids Res.*, vol. 44, no. D1, pp. D380–D384, 2016.

[36] S. Kim et al., "PubChem in 2021: New data content and improved web interfaces," *Nucleic Acids Res.*, vol. 49, no. D1, pp. D1388–D1395, 2021.

[37] T. T. Tanimoto, "IBM internal report 17th," IBM, Tech. Rep., 111957.

[38] D. Bajusz, A. Rácz, and K. Héberger, "Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations?," *J. Cheminformat.*, vol. 7, no. 1, pp. 1–13, 2015.

[39] M. Ashburner et al., "Gene ontology: Tool for the unification of biology," *Nat. Genet.*, vol. 25, no. 1, pp. 25–29, 2000.

[40] D. W. Huang, B. T. Sherman, and R. A. Lempicki, "Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources," *Nat. Protoc.*, vol. 4, no. 1, pp. 44–57, 2009.

[41] D. W. Huang, B. T. Sherman, and R. A. Lempicki, "Bioinformatics enrichment tools: Paths toward the comprehensive functional analysis of large gene lists," *Nucleic Acids Res.*, vol. 37, no. 1, pp. 1–13, 2009.

[42] S. Vilar and G. Hripcsak, "The role of drug profiles as similarity metrics: Applications to repurposing, adverse effects detection and drug–drug interactions," *Brief. Bioinf.*, vol. 18, no. 4, pp. 670–681, 2017.

[43] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," in *Proc. Int. Conf. Learn. Representations*, 2018.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[45] A. K. Bajpai et al., "Systematic comparison of the protein-protein interaction databases from a user's perspective," *J. Biomed. Informat.*, vol. 103, 2020, Art. no. 103380.

Pietro Bongini received the bachelor's degree in information engineering and the master's degree with honors in computer and automation engineering from the University of Siena, in July 2015 and 2018, respectively. He is currently working toward the PhD degree in smart computing with the University of Florence. Currently, he is affiliated with the Department of Information Engineering and Mathematics at the University of Siena. His main research interests are bioinformatics and machine learning on structured data.

Franco Scarselli rreceived the PhD degree in computer science and automation engineering from the University of Florence. Currently, he is an associate professor with the University of Siena. In 2013, he received the Italian scientific habilitation as full professor. The research activity of Franco Scarselli is in the field of machine learning, with a particular focus on neural networks for graphs, deep learning and approximation theory. From an applicative point of view, his focus is on computer vision and information retrieval. He has been involved in numeros research projects related to the mentioned topics founded by companies, national and international entities.

Monica Bianchini received the Laurea degree cum laude in applied mathematics in 1989 and the PhD degree in computer science and control systems in 1995 from the University of Florence, Italy. She is currently an associate professor with the Department of Information Engineering and Mathematics of the University of Siena. Her main research interests are in the field of machine learning, with emphasis on neural networks for structured data and deep learning, approximation theory, bioinformatics, and image processing. She served/serves as an associate editor for *IEEE Transactions on Neural Networks, International Journal of Knowledge-Based and Intelligent Engineering Systems, International Journal of Computers in Healthcare*, and has been the editor of numerous books and special issue in international journals on neural networks/structural pattern recognition. She is a permanent member of the editorial board of IJCNN, ICANN, ICPR, ESANN, ANNPR, and KES.

Giovanna Maria Dimitri received the MPhil degree (with distinction) in advanced computer science from the University of Cambridge, Cambridge, U.K., in July 2015, under the supervision of Prof Pietro Lió, with the dissertation on predicting drugs side effects from a combination of chemical and biological profiles, and the PhD degree in computer science from the University of Cambridge, under the supervision of Prof Pietro Lió, with the dissertation on multilayer network methodologies for brain data analysis and modeling. She is a life member of Clare Hall College, Cambridge. Previously she received the bachelor's and master's thesis (both 110/110 cum laude) in computer and automation engineering from the University of Siena. She is currently a postdoc researcher, working on deep learning for biomedical image processing. She is the author of several papers in international peer reviewed journals. Her research interests include developing deep learning and machine learning models for biomedical data analysis.

Niccoló Pancino received the bachelors degree in computer and information engineering from the University of Siena, in 2016 and the masters degree Cum Laude in computer and automation engineering from the University of Siena, in 2019. He is currently working toward the PhD degree in smart computing with the University of Florence. Under the supervision of Prof. Bianchini and Prof. Scarselli, he defended his master's thesis Graph Neural Network for the prediction of Protein-Protein Interactions. His research interests focus on developing deep learning models for graph-structured data processing.

Pietro Lió received the PhD degree in complex systems and non-linear dynamics and second PhD degree in genomic sciences PL develops new key methodologies in deep learning. He is an internationally leading researcher in both Bioinformatics and machine learning, member of the Cambridge centre of AI and Medicine, The Cambridge Center for Computational Biology and The European Laboratory for Machine Learning (Ellis). These methodologies provide grounds for effective and actionable interfacing computer science and machine learning with biomedical fields and also set the vision for future medicine care.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.